JURNAL ELEKTRO DAN INFORMATIKA
SWADHARMA
TERAKREDITASI SINTA 4
P-ISSN : 2774 - 5775 | E-ISSN : 2774 - 5767
Volume 6 Nomor 2 – Juli 2026

JEIS
Penerbit
Lembaga Penelitian dan Pengabdian Kepada Masyarakat (LPPM)
Institut Teknologi dan Bisnis Swadharma (ITBS)

ISSN 2774 – 5775 | eISSN 2774-5767

JEIS : JURNAL ELEKTRO DAN INFORMATIKA
SWADHARMA

TERAKREDITASI SINTA 4
SK No. 355/DST/D.D1/HM.01.01/2026

PENGANTAR REDAKSI

Dengan ucapan puji dan syukur kami panjatkan ke hadirat Tuhan Yang Maha Esa. Karena berkat rahmat dan hidayahnya Jurnal Elektro dan Informatika Swadharma (JEIS) Institut Teknologi dan Bisnis (ITB) Swadharma Volume 6 Nomor 2 Edisi Juli 2026 dapat diterbitkan. Pada tanggal 24 Juli 2026 menjelang eidisi ini diterbitkan, tim pengelola JEIS mendapatkan kabar gembira dengan keluarnya hasil Reakreditasi Jurnal, JEIS memperoleh **Akreditasi SINTA 4** berdasarkan Keputusan Direktur Jenderal Sains dan Teknologi Kemendikti Saintek RI No. 355/DST/D.D1/HM.01.01/2026 mulai Vol 5 No 2 Juli 2025 sampai dengan Vol 10 No 1 Januari 2030.

Jurnal ilmiah ini memuat makalah hasil penelitian, studi literature, pemodelan, simulasi, studi pustaka, dan hasil pemikiran lainnya di bidang elektro dan Informatika. Pada edisi ini memuat 17 (tujuh belas) paper. Dari 17 paper tersebut, 2 paper berasal dari internal ITB Swadharma dan 15 paper lainnya berasal dari luar ITB Swadharma, yaitu Universitas Bina Sarana Informatika Jakarta (8 Paper), UIN KH. Abdurrahman Wahid Pekalongan (4 paper), dan sisanya masing-masing 1 paper dari Sekolah Tinggi Teknologi Informasi NIIT I-Tech Jakarta, Institut Bakti Nusantara, Lampung, Universitas KH Abdul Wahab Hasbullah Jombang.

Redaksi mengucapkan terima kasih kepada para penulis yang telah mengirimkan papernya untuk diterbitkan pada edisi ini. Sementara beberapa paper lainnya yang sudah ada di redaksi namun belum dapat diterbitkan akan kami muat pada edisi berikutnya.

Redaksi mengharapkan saran dan kritik yang membangun dari seluruh pembaca, utamanya Sivitas Akademika ITB Swadharma demi meningkatkan mutu jurnal ilmiah pada edisi yang akan datang.

Jakarta, 30 Juli 2026

Managing Editor

ISSN 2774 – 5775 | eISSN 2774-5767

JEIS : JURNAL ELEKTRO DAN INFORMATIKA

SWADHARMA

Volume 06 Nomor 02, Juli 2026

DAFTAR ISI

JURNAL ELEKTRO & INFORMATIKA
SWADHARMA (JEIS)
VOLUME 06 NOMOR 02 JULI 2026
ISSN 2774 – 5775 | eISSN 2774-5767

OPTIMASI *INTER-VLAN ROUTING* JARINGAN *WIRELESS* MENGGUNAKAN METODE *NETWORK DEVELOPMENT LIFE CYCLE* PADA CROWN HORECA

Andy Dharmalau[1)], Lela Nurlaela[2)], Khusnul Khoiriyah[3)], Tuhfatul Habibah Hasibuan[4)], Artha Murin[5)]
[1,2,3,4,5]Prodi Teknik Informatika, Fakultas Teknologi, ITB Swadharma

Correspondence author: A Dharmalau, andy.d@swadharma.ac.id, Jakarta, Indonesia

Abstract

Information technology, particularly computer networks, has become a fundamental aspect in various fields, including economics, education, communications, and entertainment. Crown Horeca already has several network devices and uses the internet. Instability often occurs in the wireless network used due to high traffic exceeding capacity. Wireless network optimisation is needed to improve efficiency and the quality of network service. The purpose of this research is to design a Virtual Local Area Network (VLAN) using Inter-VLAN Routing techniques to improve performance and network segmentation management. The study used the Network Development Life Cycle (NDLC) method with data collection techniques of direct observation, interviews, and literature review. Ping test results showed no Request Time Out (RTO), indicating stable inter-floor connections. Message delivery between devices on the network, as simulated in Packet Tracer, ran smoothly, proving that inter-VLAN communication was functioning as expected.

Keywords: *network optimisation, Inter-VLAN Routing, wireless*

Abstrak

Teknologi informasi, khususnya dalam jaringan komputer, telah menjadi aspek mendasar dalam berbagai bidang, termasuk perekonomian, pendidikan, komunikasi, dan hiburan. Crown Horeca telah memiliki beberapa perangkat jaringan dan menggunakan jaringan internet. Ketidakstabilan sering kali terjadi pada jaringan *wireless* yang digunakan karena tingginya trafik yang melebihi kapasitas. Diperlukan optimasi jaringan *wireless* guna meningkatkan efisiensi dan kualitas layanan jaringan. Tujuan penelitian untuk merancang jaringan *Virtual Local Area Network* (VLAN) menggunakan teknik *Inter-VLAN Routing* untuk meningkatkan performa dan manajemen segmentasi jaringan. Penelitian menggunakan metode *Network Development Life Cycle* (NDLC) dengan teknik pengumpulan data observasi langsung, wawancara dan studi pustaka. Hasil pengujian *ping* menunjukkan tidak adanya *Request Time Out* (RTO), menunjukkan koneksi antar lantai berjalan stabil. Pengiriman pesan antar perangkat di jaringan menggunakan simulasi *Packet Tracer* berjalan lancar, membuktikan bahwa komunikasi antar VLAN berjalan dengan baik dan sesuai harapan.

Kata Kunci: optimasi jaringan, *Inter-VLAN Routing, wireless*

A. PENDAHULUAN

Teknologi informasi, khususnya jaringan komputer, telah menjadi aspek mendasar dalam berbagai bidang, termasuk perekonomian, pendidikan, komunikasi, dan hiburan (Alukadinata et al., 2021; Dharmalau et al., 2022). Hal ini dapat dilihat dari luasnya penggunaan jaringan komputer, baik dalam skala umum maupun pribadi. Semakin meningkatnya kebutuhan akan akses dan komunikasi, kinerja jaringan harus tetap optimal. Oleh karena itu, operator jaringan dan penyedia layanan internet (ISP) perlu memiliki strategi yang efektif dalam mengatasi permasalahan yang dapat mengganggu stabilitas dan keandalan jaringan (Irawati & Fitriansyah, 2025).

Crown Horeca adalah sebuah nama perusahaan yang didirikan pada tahun 2011, merupakan perusahaan penyedia bisnis penjualan alat dapur, mesin restoran, dan berbagai macam peralatan, serta *Kitchen Equipment.*

Fasilitas internet pada perusahaan Crown Horeca terdiri dari beberapa perangkat jaringan seperti 1 *Server, 1 Switch, 1 Router*, dan 4 *Modem*. Menggunakan jaringan internet melalui provider dengan jumlah *bandwidth* 100 MB (*Mega Byte*). Fasilitas internet tersebut terbagi menjadi 3 bagian, yaitu 40 MB untuk lantai 4 yang terdiri dari ruang *Server*, ruang Manager, *Staff IT (Information Technology) & Digital*, serta Finance. Sementara itu, 40 MB lainnya dialokasikan untuk lantai 3 yang mencakup ruang *Staff Sales*, *Finance*, dan Admin. Sisanya 20 MB digunakan untuk jaringan wireless yang tersebar di seluruh lantai melalui tambahan *Router*.

Pada jaringan *wireless* ini sering terjadi ketidakstabilan karena tingginya *traffik* yang melebihi kapasitas, Hal ini disebabkan banyaknya staff mengakses aplikasi media sosial tanpa adanya pemblokiran. Akibatnya lalulintas data pada jaringan menjadi padat, dan node bekerja melebihi kapasitas *buffer* karena keterbatasan memori, yang berkontribusi pada rendahnya *throughput*, ketidakstabilan jaringan, *delay*, *packet loss*, dan *jitter*. Selain itu, manajemen bandiwidth yang belum teratur menjadi masalah tersebut.

Untuk itu diperlukan manajemen *bandwidth* untuk optimasi jaringan wireless guna meningkatkan efisiensi dan kualitas layanan jaringan (Sitanggang et al., 2025). Untuk mengatasi permasalahan yang ada, diperlukan pendekatan yang terstruktur dalam perancangan dan penglolaan jaringan agar performanya lebih optimal(Kurniawan & Santosa, 2022).

Mengacu pada penelitian yang pernah dilakukan membahas implementasi *VLAN, Inter-VLAN Routing, dan Access Control List (ACL)* untuk meningkatkan manajemen jaringan serta mengurangi *broadcast domain* (Sari et al., 2022). Penelitian lain membahas segmentasi jaringan menggunakan *VLAN* dan *Inter-VLAN Routing* untuk mengatasi *broadcast domain* yang besar serta ketidakstabilan koneksi hotspot (Fardiwiyono, 2022). Penelitian ini menerapkan *Hotspot Bridge*, yang memungkinkan pengguna berpindah antar *access point* tanpa harus login ulang, serta *fail over,* yang menjaga koneksi tetap stabil saat satu jalur internet mengalami gangguan. Penelitian selanjutnya oleh (Nirmalsari et al., 2023), penelitian ini menggunakan metode *Network Development Life Cycle (NDLC)* yang memiliki beberapa tahapan dalam penelitian tersebut yakni Analisis, Desain, Implementasi, Monitoring, Management.

Berdasarkan hasil penelitian yang pernah dilakukan terhadap pengembangan jaringan ini, dilakukan dengan menerapkan metode *Network Development Life Cycle (NDLC)* sebagai pendekatan sistematis dalam perancangan dan evaluasi jaringan (Gunawan et al., 2025). Metode ini digunakan dalam mengembangkan atau rancang infrastruktur dengan memantau kinerja jaringan. Sebagai pendekatan sistematis dalam perancangan dan evaluasi jaringan. Rancangan ini akan disimulasikan menggunakan *packet tracer* untuk menguji efektivitas *Inter-VLAN*

Routing dalam mengelola *trafik* jaringan (Syaifudin et al., 2020). *Packet tracer* dikembangkan oleh cisco, sebuah perusahaan yang intens dalam masalah jaringan (Awan, 2024). *Cisco packet tracer* berfungsi untuk memudahkan penggunanya untuk belajar, berlatih ataupun mendesain jaringan komputer (Leki et al., 2022). Sedangkan *Inter-VLAN Routing* merupakan proses untuk melakukan forwarding traffic dari *Vlan* yang satu ke Vlan lainnya dengan menggunakan *Router* (Sastra & Musyaffa, 2020). Pada jaringan ini sistem routing dapat terpusat hanya membutuhkan 1 *Router* dan 1 port interface untuk pembagian ip address yang akan dibuat dalam bentuk virtual yang kemudian akan di *trunk* menuju vlan-vlan lainnya yang terdapat di *switch* pada gedung gedung yang berbeda.

Tujuan dari penelitian ini untuk merancang sebuah jaringan *Virtual Local Area Network (VLAN)* menggunakan metode *Inter-VLAN Routing* untuk meningkatkan performa dan manajemen segmentasi jaringan.

B. METODE PENELITIAN

Penelitian ini menggunakan metode kualitatif, yang fokus pada pengamatan yang mendalam, dengan pendekatan studi lapangan untuk menganalisis perancangan jaringan menggunakan *Inter-VLAN Routing*. Fokus pada pemahaman konsep, analisis konfigurasi, serta evakuasi jaringan berdasarkan hasil simulasi di *packet tracer*.

Teknik pengumpulan data dalam penelitian ini meliputi observasi, wawancara, dan studi pustaka. Penjelasan lebih lanjut mengenai masing-masing teknik dijabarkan sebagai berikut:

Observasi adalah metode pengumpulan data dengan mengamati dan monitoring secara langsung struktur jaringan, internet dan spesifikasi perangkat keras yang tersedia pada Crown Horeca.

Interview merupakan salah satu metode yang untuk mendapatkan informasi mengenai permasalahan yang ada., dengan melakukan tanya jawab dengan beberapa Narasumber. Adapun pertanyaan yang diajukan terkait permasalahan saat ini.

Studi literatur juga dilakukan dengan tujuan untuk memperluas wawasan dan menjelaskan kajian pustaka berdasarkan teori-teori penunjang yang digunakan sebagai bahan penelitian seperti pengembangan jaringan komputer, perancangan jaringan komputer dan sebagainya. Adapun studi literatur ini didapatkan dari membaca buku, jurnal, artikel maupun dari internet.

C. HASIL DAN PEMBAHASAN

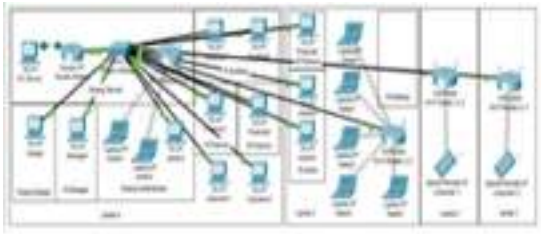
Sistem Jaringan yang ada di Crown Horeca sudah menjadi suatu hal kebutuhan pokok, karena menjadi sarana komunikasi dan pertukaran data yang digunakan oleh Manajer dan karyawan dalam berbagai kegiatan operasional perusahaan. Jaringan komputer ini sangat penting dalam mendukung berbagai aspek, mulai dari proses komunikasi internal hingga pengelolaan data dan administrasi. Akses internet digunakan sebagai media berbagi informasi dan media pembelajaran bagi para karyawannya.

Adanya Internet memungkinkan staff untuk mengakses sumber daya terkini, mengikuti tren industri, dan memperoleh pelatihan yang diperlukan untuk meningkatkan keterampilan karyawan.

Bagi bagian administrasi, jaringan komputer memainkan peran krusial dalam pengarsipan dokumen, manajemen data, serta pelaporan yang akurat dan tepat waktu. Dokumen-dokumen penting dapat disimpan dengan aman dan mudah diakses ketika diperlukan, mengurangi risiko kehilangan data dan mempercepat proses pengambilan keputusan.

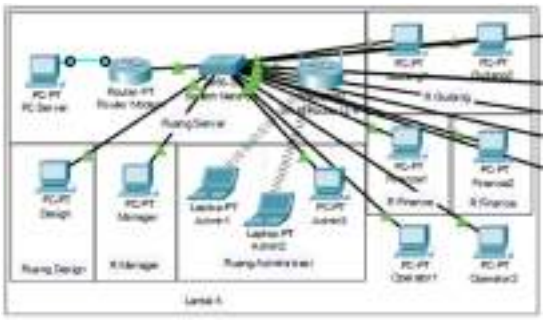
Untuk dapat memahami kondisi jaringan yang ada, berikut adalah denah ruangan dan infrastruktur jaringan yang berjalan pada Crown Horeca. Topologi jaringan yang digunakan di perusahaan yaitu *Topologi Tree*. Topologi ini dipilih karena mudah

dikembangkan, memiliki struktur hierarki yang efisien, serta gabungan dari Topologi *Bus* dan *Star*. Pada gambar 1 merupakan topologi seluruh ruangan yang ada.



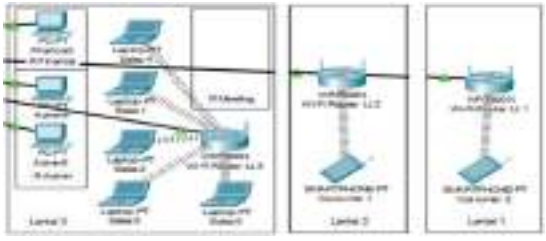
Gambar 1. Topologi Jaringan yang ada.

Pada gambar 2 merupakan detail dari topologi jaringan pada lantai 4.



Gambar 2. Topologi Jaringan Lantai 4

Pada gambar 3 merupakan detail dari topologi jaringan pada lantai 3, lantai 2 dan lantai 1.



Gambar 3. Topologi Jaringan Lantai 3,2,1

Permasalahan Sistem Secara Umum

Analisa yang dilakukan untuk permasalahan di Crown Horeca menggunakan metode *SWOT (Strength, Weakness, Opportunity, Threat).*

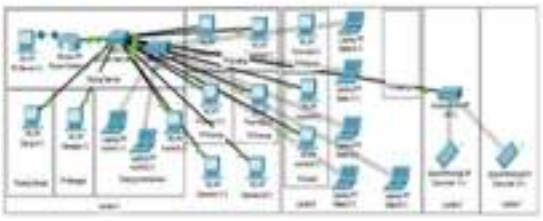
Infrastruktur jaringan pada Crown Horeca sudah cukup memadai, namun masih terdapat kendala dalam manajemen bandwidth dan pengelolaan *IP Address*, yang menyebabkan hambatan trafik jaringan, terutama saat jam sibuk. Selain itu, belum diterapkannya segmentasi jaringan (VLAN) menjadikan trafik data tidak efisien dan sulit dikelola secara terpisah antar divisi.

Keamanan Jaringan, kurangnya pengamanan jaringan dan tidak adanya segmentasi menyebabkan jaringan rentan terhadap akses tidak sah dari pihak luar maupun pengguna internal yang tidak berwenang. Maka diperlukan penerapan metode seperti pemblokiran situs non-produktif dan *Inter-VLAN Routing* untuk membatasi hak akses, memperkuat keamanan, dan meningkatkan efisiensi jaringan.

Analisis Kebutuhan Sistem Jaringan

Penggunaan metode *Inter-VLAN Routing* bertujuan untuk melakukan segmentasi jaringan secara *virtual (VLAN)*, agar dapat mengurangi jumlah perangkat jaringan yang sering kali menyebabkan terjadinya traffic pada Infrastruktur yang saat ini berjalan dan melakukan pembatasan pada distribusi IP Address agar mengurangi jumlah kelonggaran pada *IP Address* yang tidak terpakai. Melakukan pengurangan perangkat *Wireless Router* dan menggantinya dengan *Access Point*, melakukan segmentasi jaringan dengan mengkonfigurasi *Virtual Local Area Network (VLAN)* pada *Switch* dan konfigurasi *Inter-VLAN Routing* pada *Router* agar masing-masing segmen dapat terhubung.

Pada Gambar 4. adalah topologi jaringan usulan.



Gambar 4. Topologi Usulan

Tahapan atau proses yang dilakukan dalam melakukan konfigurasi pada perangkat *Router* secara menyeluruh dan sistematis,

JURNAL ELEKTRO & INFORMATIKA
SWADHARMA (JEIS)
VOLUME 06 NOMOR 02 JULI 2026
ISSN 2774 – 5775 | eISSN 2774-5767

Berikut adalah gambar tampilan untuk konfigurasi *Inter-VLAN Routing* pada *Router* Utama, yang digunakan sebagai penghubung antara masing-masing *VLAN* yang sudah dibagi di *Switch* Utama.

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#int fa0/0.100
Router(config-subif)#encap dot1q 100
Router(config-subif)#ip add 192.168.100.1 255.255.255.0
Router(config-subif)#exit
Router(config)#int fa0/0.103
Router(config-subif)#encap dot1q 103
Router(config-subif)#ip add 192.168.103.1 255.255.255.240
Router(config-subif)#exit
Router(config)#int fa0/0.104
Router(config-subif)#encap dot1q 104
Router(config-subif)#ip add 192.168.104.1 255.255.255.240
Router(config-subif)#exit
```

Gambar 5. Konfigurasi *Inter-VLAN Routing* pada *Router*

Konfigurasi *Switch* Utama mencakup penamaan perangkat, pengaturan *VLAN*, *port*, dan keamanan. Tampilan proses konfigurasi pembuatan *VLAN* pada *Switch* utama, masing-masing ruangan telah terbagi ke dalam *VLAN* yang sudah diberi nama sesuai nama ruangan tersebut. Pada Gambar 6.

```
Switch>enable
Switch#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#vlan 100
Switch(config-vlan)#name wireless
Switch(config-vlan)#exit
Switch(config)#vlan 103
Switch(config-vlan)#name lantai3
Switch(config-vlan)#exit
Switch(config)#vlan 104
Switch(config-vlan)#name lantai4
Switch(config-vlan)#exit
```

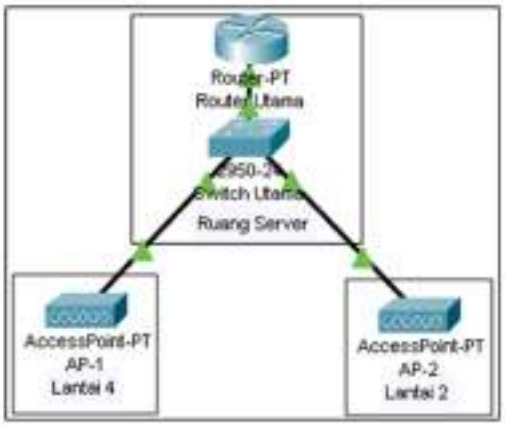
Gambar 6. Konfigurasi *VLAN*

Berikut adalah proses konfigurasi *VLAN Trunk* pada *Switch* Utama. Tampilan proses konfigurasi untuk mendaftarkan port yang ada pada *Switch* Utama ke *VLAN* yang telah dibuat sebelumnya, sekaligus mengubah mode pada *port Switch* Utama yang terhubung ke *Router* Utama dari mode *access* menjadi mode *trunk.* Pada Gambar 7.

```
Switch(config)#interface range fa0/3-11
Switch(config-if-range)#switchport mode access
Switch(config-if-range)#switchport access vlan 104
Switch(config-if-range)#exit
Switch(config)#interface range fa0/13-14
Switch(config-if-range)#switchport mode access
Switch(config-if-range)#switchport access vlan 103
Switch(config-if-range)#exit
Switch(config)#interface fa0/2
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 100
Switch(config-if)#exit
Switch(config)#interface fa0/12
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 100
Switch(config-if)#exit
Switch(config)#interface fa0/1
Switch(config-if)#switchport mode trunk
Switch(config-if)#exit
```

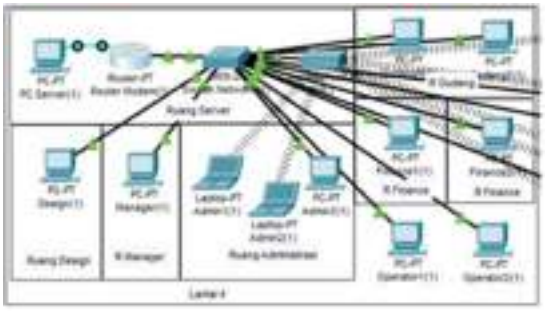
Gambar 7. Konfigurasi pendaftaran port *VLAN* dan pengaturan *port Trunk* ke *Router* Utama

Gambar pemetaan topologi jaringan yang terhubung dari *Router* ke *switch* utama lalu ke access point yang ada di lantai 4 dan 2. Pada Gambar 8.

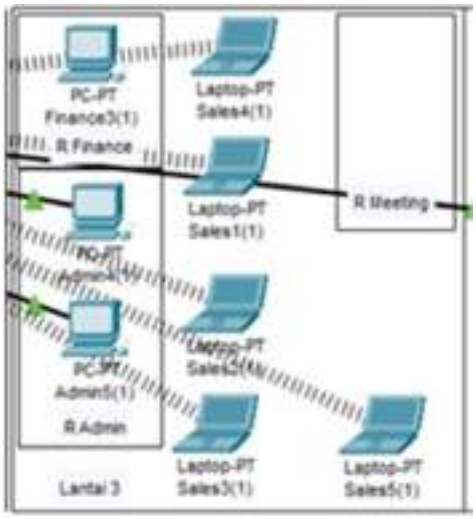


Gambar 8. Denah Jaringan *Crown Horeca*

Pada Infrastruktur Jaringan di Lantai 4, masing-masing *End-device* terhubung langsung dengan *Switch* Utama yang ada di Ruang *Server.*

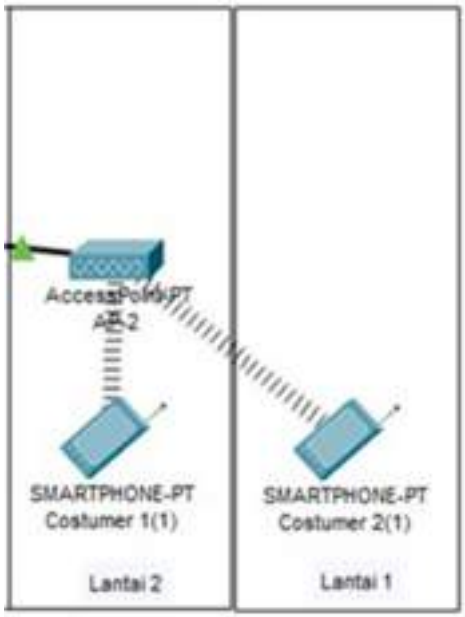


Gambar 9. Topologi Jaringan Lantai 4



Gambar 10. Topologi Jaringan Lantai 3

Gambar topologi jaringan lantai 3, dimana pada perangkat komputer itu terhubung langsung ke *Switch* Utama dan perangkat laptop terhubung langsung dengan Access Point.



Gambar 11. Denah Topologi Jaringan Lantai 2 dan 1

Gambar topologi jaringan lantai 2 dan 1, pada kedua lantai itu tidak ada perangkat yang terhubung menggunakan kabel hanya menggunakan *wireless* yang terhubung langsung dengan Access Point pada lantai 2.

Hasil Konfigurasi *Inter-VLAN Routing* pada *Router*. Dapat dilihat bahwa *Fast ethernet* 0/0 yang terhubung ke *Switch* Utama sudah terdaftar dalam list bridging pada *Router*, artinya setiap *VLAN* yang terdaftar dapat terhubung meskipun di segmentasi *VLAN*.

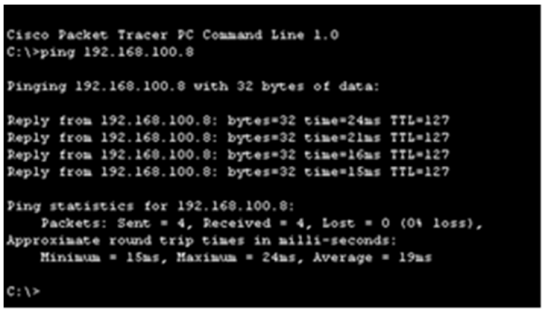
Gambar 12. *List Port* yang terdaftar *Inter-VLAN Routing*

Hasil konfigurasi *VLAN* pada perangkat *Switch* Utama yang ada di Ruang *Server*. Dapat dilihat bahwa *Fast ethernet* 0/2 sampai dengan *Fast ethernet* 0/14 terhubung ke *Switch* pada masing-masing ruangan sudah terdaftar dalam list *VLAN* di *Switch* Utama, artinya setiap *Switch* yang terdaftar pada *VLAN* sudah tersegmentasi.

Gambar 13. List Port yang terdaftar *VLAN Switch*

Hasil Pengujian Konfigurasi *Inter-VLAN Routing*, menggunakan tes ping dari perangkat yang terhubung pada *VLAN* lantai 4 ke *VLAN wireless*.

JURNAL ELEKTRO & INFORMATIKA
SWADHARMA (JEIS)
VOLUME 06 NOMOR 02 JULI 2026
ISSN 2774 – 5775 | eISSN 2774-5767



Gambar 14. *Test ping VLAN* lantai 4 ke *VLAN wireless.*

Rancangan *Access List* ada pada *Tabel Access List* dari Ruangan yang tersegmentasi dengan *VLAN*, akses dari luar *VLAN* yang tidak terkonfigurasi *Inter-VLAN Routing* tidak akan bisa mengakses ke dalam jaringan. Acces list dapat dilihat pada tabel 2 di bawah ini.

Tabel 1. Tabel Access List

Access List				
No	Ruang	IP *Address*	Nama VLAN	ID VLAN
1	Lantai 2	192.168.100.1	wireless	100
2	Lantai 3	192.168.103.1	Lantai 3	103
3	Lantai 4	192.168.104.1	Lantai 4	104

Berdasarkan perancangan *Inter-VLAN Routing*, dilakukan pengujian terhadap performa jaringan untuk membandingkan kondisi sebelum dan sesudah optimasi. Pengujian ini meliputi parameter delay (latency), *packet loss*, dan *bandwidth* sebagaimana ditunjukkan pada Tabel 2.

Tabel 2. Tabel Perbandingan Jaringan

Aspek	Sebelum Optimasi	Sesudah Optimasi
Segmentasi Jaringan	Tidak ada *VLAN*	Menggunakan *VLAN & Inter-VLAN Routing*
Delay (ms)	± 65 ms	± 15 ms
Packet Loss (%)	± 10%	0%
Bandwidth (Mbps)	± 55 Mbps	± 95 Mbps
Keamanan Jaringan	Tidak ada *ACL dan Port Security*	Menggunakan *ACL, Port Security, DHCP Snooping*
Akses Wireless	*Wireless Router* biasa	*Access Point IEEE 802.11ac (dual-band, VLAN tagging)*

Visualisasi hasil pengujian *bandwidth* dan jaringan dapat dilihat pada Gambar 15 dan gambar 16 dibawah ini.

Gambar 15. Hasil Perbandingan *bandwidth* sebelum dan sesudah Optimasi

Gambar 16. Hasil Perbandingan jaringan sebelum dan sesudah Optimasi

Hasil dari perancangan yang dilakukan menunjukkan bahwa jaringan yang telah diterapkan *Inter-VLAN Routing* berhasil mengatasi masalah yang ada. Pengujian ping dari masing-masing lantai ke lantai lainnya menunjukkan tidak adanya *Request Time Out (RTO)*, yang menunjukkan bahwa koneksi antar lantai berjalan stabil. Selain itu, pengiriman pesan antar perangkat di jaringan menggunakan simulasi *Packet Tracer* juga

berhasil dilakukan dengan lancar, membuktikan bahwa komunikasi antar *VLAN* berjalan dengan baik dan sesuai harapan

D. PENUTUP

Pada infrastruktur jaringan yang berjalan saat ini di Crown Horeca, terdapat beberapa permasalahan yang menghambat performa dan efisiensi jaringan.

Rancangan infrastruktur jaringan *Inter-VLAN Routing* pada Crown Horeca menggunakan dua perangkat utama, yaitu perangkat *Router* dan *Switch*. Perangkat *Switch* berfungsi sebagai wadah untuk *VLAN* yang melakukan segmentasi jaringan, sementara perangkat *Router* berfungsi sebagai penghubung antar segmen jaringan yang telah dikonfigurasi pada *Switch*, sehingga memungkinkan komunikasi yang efisien antara berbagai *VLAN*.

Hasil dari perancangan yang dilakukan menunjukkan bahwa jaringan yang telah diterapkan *Inter-VLAN Routing* berhasil mengatasi masalah yang ada. Pengujian ping dari masing-masing lantai ke lantai lainnya menunjukkan tidak adanya *Request Time Out* (RTO), yang menunjukkan bahwa koneksi antar lantai berjalan stabil. Selain itu, pengiriman pesan antar perangkat di jaringan menggunakan simulasi *Packet Tracer* juga berhasil dilakukan dengan lancar, membuktikan bahwa komunikasi antar *VLAN* berjalan dengan baik dan sesuai harapan.

Disarankan penggantian perangkat *wireless Router* dengan *Access Point* yang mendukung standar IEEE 802.11ac, beroperasi pada frekuensi 2,4 GHz (hingga 450 Mbps) dan 5 GHz (hingga 1300 Mbps), serta mendukung *dual-band concurrent* dan fitur *VLAN tagging*. *Access Point* ini juga memiliki dukungan *PoE (Power over Ethernet)*, sehingga pemasangannya lebih fleksibel. Dengan spesifikasi tersebut, perangkat mampu melayani hingga 200 pengguna secara bersamaan tanpa mengurangi kestabilan koneksi. Pemeliharaan jaringan secara berkala perlu dilakukan sebagai bagian dari prosedur standar perusahaan guna memastikan seluruh perangkat jaringan berjalan dengan baik. Kegiatan ini mencakup pembaruan perangkat lunak *(firmware)*, pemeriksaan catatan aktivitas jaringan, pemantauan lalu lintas data, serta pengujian koneksi antar divisi. Dengan melakukan perawatan rutin, potensi gangguan atau ancaman keamanan dapat dideteksi lebih awal dan dicegah sebelum mengganggu operasional perusahaan.

E. DAFTAR PUSTAKA

Alukadinata, A., Firdaus, M., & Ristiawan, R. (2021). Perancangan Sistem Informasi Penjualan Handphone di Forza Mufid Perdana Selular. *JIST: Jurnal Indonesia Sosial Teknologi*, *2*(7), 1100–1111. https://doi.org/10.36418/jist.v2i7.192

Awan. (2024). Simulasi Penggunaan Router sebagai penghubung antar Network ID yang berbeda dengan Cisco Packet Tracer. *Skena Teknologi : Jurnal Ilmiah Sains Dan Teknologi*, *1*(2), 10–18. https://ejournal.ibbi.ac.id/index.php/ST/article/view/30

Dharmalau, A., Ar-Rasyid, H., & Iskandarsyah, M. A. (2022). Implementasi Metode SWOT Pada Analisis Jaringan Area Lokal Sekolah. *JEIS: Jurnal Elektro Dan Informatika Swadharma*, *02*(1), 1–8. https://doi.org/10.56486/jeis.vol2no1.110

Fardiwiyono. (2022). *Analisis dan Perancangan Jaringan Berbasis Inter-VLAN di STMIK Widya Cipta Dharma* [Teknik Informnatika STMIK Widya Cipta Dharma]. https://repository.wicida.ac.id/4339/

Gunawan, A., Sundari, S. S., & Anwar, D. S. (2025). Implementasi Manajemen Bandwidth Metode Simple Queue Pada Jaringan Internet SMP Negeri 1

Jamanis. *JEIS: Jurnal Elektro Dan Informatika Swadharma*, *5*(1), 92–98. https://doi.org/10.56486/jeis.vol5no1.639

Irawati, & Fitriansyah, A. (2025). *Komunikasi Data*. Yogyakarta : Deepublish.

Kurniawan, P., & Santosa, K. (2022). Pengembangan VLAN dengan V Trunking Protokol (VTP) Mode Menggunakan Switch Cisco di Bank Syariah Indonesia. *Oktal : Jurnal Ilmu Komputer Dan Sains*, *1*(08), 1226–1233. https://journal.mediapublikasi.id/index.php/oktal/article/view/487

Leki, N., Djamen, A. C., & Mintjelungan, M. M. (2022). Penerapan Cisco Packet Tracer Sebagai media Pembelajaran Jaringan untuk Meningkatkan Hasil Belajar Siswa SMK. *Edutik : Jurnal Pendidikan Teknologi Informasi Dan Komunikasi*, *2*(1), 14 26. https://doi.org/10.53682/edutik.v2i1.3319

Nirmalsari, P., Fahmi, H., & Fadli, S. (2023). Implementasi Metode Network Development Life Cycle Pada Rancang Bangun Jaringan Wireless Berbasis Mikrotik. *Jurnal Teknik Mesin, Industri, Elektro Dan Informatika*, *2*(3), 72–87. https://doi.org/10.55606/jtmei.v2i3.2107

Sari, A. M., Desriyani, D., & Suryani, D. (2022). Pengembangan Dan Perancangan Local Area Network Pada Yayasan Joko Tinurut Jakarta. *Akrab Juara : Jurnal Ilmu-Ilmu Sosial*, *7*(3), 1–11. https://doi.org/10.58487/akrabjuara.v7i3.1861

Sastra, R., & Musyaffa, N. (2020). Implementasi Access Inter-VLAN Menggunakan Router. *INSANTEK : Jurnal Inovasi Dan Sains Teknik Elektro*, *1*(2), 77–81. https://ejournal.bsi.ac.id/ejurnal/index.php/khatulistiwa/issue/archive/index.php/insantek/article/view/9250

Sitanggang, Y. F. P., Meilano, R., & Saputra, M. H. (2025). Optimalisasi Manajemen Bandwidth Pada Jaringan Internet Menggunakan Metode PPDIOO di Politeknik Jambi. *ELTI : Jurnal Elektronika, Listrik Dan Teknologi Informasi Terapan*, *7*(1), 43–52. https://doi.org/10.37338/elti.v7i1.451

Syaifudin, A., Wahyuddin, M. I., & Ningsih, S. (2020). Redundancy Link dan Load Balancing Menggunakan Metode EtherChannel LACP dengan InterVLAN Routing. *JOINTECS (Journal of Information Technology and Computer Science)*, *5*(2), 137. https://doi.org/10.31328/jointecs.v5i2.1251

ANALISA EFISIENSI WAKTU KOMPUTASI DAN PENGGUNAAN MEMORI PADA ENAM ALGORITMA PENGURUTAN

Nafisha Putri Arsita[1)], Rahma Syarifa[2)], Tsaqib Fahmi Ahmad[3)], Imam Prayogo Pujiono[4)]
[1,2,3]Bisnis Digital, Fakultas Ekonomi dan Bisnis Islam, UIN K.H Abdurrahman Wahid Pekalongan
[4]Informatika, Fakultas Ekonomi dan Bisnis Islam, UIN K.H Abdurrahman Wahid Pekalongan

Correspondence author: NP Arsita, nafisha.putri.arsita.25012@mhs.uingusdur.ac.id, Pekalongan, Indonesia

Abstract

This study aims to analyze and compare the performance of six sorting algorithms (Tim Sort, Bubble Sort, Selection Sort, Insertion Sort, Quick Sort, and Merge Sort) in terms of computational time efficiency and memory usage. Implementation was carried out in Java using Visual Studio Code as the development environment, and testing was conducted on a device with an Intel Core i7-8665U (1.02 GHz) processor, 8 GB of RAM, a 64-bit Operating System, and a 256 GB SSD. All algorithms were tested using three different dataset sizes, namely 100, 1,000, and 5,000 data points, each with three repetitions to obtain more accurate results. Based on the test, Tim Sort showed the most efficient performance, maintaining stable computational time and memory usage across all dataset sizes. Quick Sort and Merge Sort have also been shown to perform well, especially for large data, in contrast to Bubble Sort, Selection Sort, and Insertion Sort, which become less efficient as data grows due to significantly increased processing time. Overall, this study concludes that Tim Sort is the most optimal algorithm in terms of speed and memory efficiency. Quick Sort and Merge Sort are recommended for sorting large-scale data. At the same time, basic algorithms like Bubble Sort, Selection Sort, and Insertion Sort are better suited to small datasets or to learning sorting concepts.

Keyword *: comparison, sorting algorithms, computation time, memory usage*

Abstrak

Penelitian ini bertujuan untuk manganalisis dan membandingkan performa enam algortima *sorting (Tim Sort, Bubble Sort, Selection Sort, Insertion Sort, Quick Sort, dan Merge Sort*) dari aspek efisiensi waktu komputasi dan penggunaan memori. Implementasi dilakukan menggunakan Bahasa pemrograman Java melalui *Visual Studio Code* sebagai lingkungan pengembangan, dan pengujian dijalankan pada perangkat dengan spesifikasi *Intel Core* i7-8665U (1,02 GHz), RAM 8 GB, Sistem Operasi 64-bit, serta SSD 256 GB. Semua algoritma diuji menggunakan tiga perbedaan ukuran dataset, yaitu 100 data, 1.000 data, dan 5.000 data, masing-masing dengan tiga kali pengulangan untuk mendapatkan hasil yang lebih akurat. Berdasarkan pengujian ditemukan bahwa *Tim Sort* menunjukkan kinerja yang paling efisien, mempertahankan waktu komputasi, dan pemakaian memori yang stabil pada semua ukuran dataset. *Quick Sort* dan *Merge Sort* juga terbukti memiliki performa kuat, khususnya ketika menangani data berukuran

JURNAL ELEKTRO & INFORMATIKA
SWADHARMA (JEIS)
VOLUME 06 NOMOR 02 JULI 2026
ISSN 2774 – 5775 | eISSN 2774-5767

besar, berbeda dengan *Bubble Sort, Selection Sort,* dan *Insertion Sort* yang menjadi kurang efisien karena peningkatan waktu proses yang signifikan seiring pertumbuhan data. Secara keseluruhan, penelitian ini menyimpulkan bahwa *Tim Sort* merupakan algoritma yang paling optimal dalam aspek kecepatan dan efisiensi memori. Adapun *Quick Sort* dan *Merge Sort* direkomendasikan untuk pengurutan data dalam skala besar, sedangkan algoritma dasar seperti *Bubble Sort, Selection Sort,* dan *Insertion Sort* lebih sesuai digunakan untuk dataset kecil atau keperluan pembelajaran konsep pengurutan.

Kata Kunci : perbandingan, algoritma pengurutan, waktu komputasi, penggunaan memori

A. PENDAHULUAN

Pengurutan data (*Sorting*) merupakan operasi fundamental dalam menentukan seberapa efisiens sistem dapat mengelola dan menyajikan data (Ghofur et al., 2025). Jumlah data yang diproses memiliki dampak langsung terhadap kinerja sistem Selama proses pengurutan (Sari et al., 2022). Dalam Bahasa pemrograman Java, terdapat berbagai algoritma pengurutan seperti *Bubble Sort, Selection Sort, Insertion Sort, Quick Sort, dan Merge Sort.* Setiap algoritma tersebut memiliki karakteristik berbeda dalam hal efisiensi memori dan waktu eksekusi (Sena et al., 2024).

Efisiensi memori dan waktu komputasi menjadi faktor utama dalam menentukan kinerja suatu algoritma pengurutan data (Basir, 2020). Pemilihan algoritma yang tepat dapat memimalkan penggunaan sumber daya memori dan waktu pemrosesan yang berlebihan (Pujiono et al., 2025). Seiring dengan perkembangan teknologi komputasi, berbagai algoritma pengurutan baru telah dikembangkan dengan tujuan meningkatkan efisiensi dan kecepatan dlam pengolahan dataset berukuran besar. Salah satu agoritma pengurutan modern yang banyak digunakan saat ini adalah *Tim Sort*.

Tim Sort merupakan algoritma hibdrida yang menggabungkan keunggulan dari *Insertion Sort* dan *Merge Sort,* sehingga mampu menghasilkan kinerja yang efisien, cepat, dan stabil pada berbagai kondisi data (Al-aydi & Abu-naser, 2025). Penelitian (Hanafi et al., 2022) menjelaskan bahwa langkah-langkah umum dalam algoritma *Tim Sort* meliputi : membagi array menjadi beberapa blok yang disebut *run*, menentukan ukuran *run*, umumnya 32 atau 64 elemen, kemudian mengurutkan elemen pada setiap *run* menggunakan *Insertion Sort*, dan terakhir menggabungkan setiap *run* yang telah diurutkan dengan metode *Merge Sort.*

Berbagi penelitian sebelumnya telah mengkaji efisiensi memori dan waktu komputasi pada algoritma pengurutan data yang diimplementasikan dalam Bahasa pemrograman Java sebagai upaya untuk meningkatkan kinerja komputasi. Penelitian (M. I. Ali et al., 2025) menganalisis efisiensi mempori dan waktu komputasi pada delapan algoritma *sorting* menggunakan Bahasa C++. Penelitian lainnya (H. Ali et al., 2021) membandingkan performa antara algoritma *Heap Sort* dan *Insertion Sort.* Selanjutnya penelitian (Mahrozi & Faisal, 2023) yang dilakukan pada tahun 2023 menganalisis perbandingan kecepatan dari segi waktu antara algoritma pengurutan data *Selection Sort* dan *Bubble Sort.* Penelitian lain (Iskandar et al., 2020) membahas analisis efisiensi memori pada algoritma *Bubble Sort* dan *Insertion Sort.* Sementara itu, penelitian yang dilakukan pada bulan Juli tahun 2025 (Musyaffa et al., 2025) mengkaji efisiensi memori dan waktu pada proses pengurutan data menggunakan algoritma ASA dan algoritma pengurutan tradisional berbasis

Python. Selain itu, penelitian (Isyqi et al., 2024) melakukan perbandingan kinerja antara algoritma *Bubble Sort*, *Insertion Sort*, dan *Selection Sort* menggunakan data numerik dalam implementasi bahasa Python. Riset tahun 2025 (Ilham et al., 2025) melakukan analisis komparatif terhadap lima algoritma pengurutan untuk menilai kecepatan pemrosesan serta efisiensi penggunaan memori ketika diimplementasikan menggunakan Bahasa pemrograman C++. Pada kajian lain (Gudiato et al., 2024), dilakukan analisis Strategi Algoritma Sorting Menggunakan Metode Komparatif pada Bahasa Pemrograman Java dengan Python. Sementara itu, penelitian sebelumnya (Latifah et al., 2021) mengkaji perbandingan kinerja antara algoritma Bubble Sort dan Selection Sort dalam konteks pemrograman paralel. Penelitian lain (Wijaya et al., 2024) menyoroti perbandingan algoritma pengurutan menggunakan Bahasa pemrograman JavaScript dengan fokus pada aspek waktu komputasi dan penggunaan memori. Selanjutnya, penelitian tahun 2025 (Sutanto et al., 2025) membahas efisiensi waktu pengurutan data antara *Bubble Sort, Insertion Sort, Merge Sort, Quick Sort* menggunakan bahasa Python. Pada penelitian berikutnya (Amanda et al., 2026), Evaluasi kinerja kombinasi algoritma sorting *Intro Sort*, *Bubble Sort* dan *Selection Sort.* Adapun penelitian tahun 2022 (Zulfa et al., 2022) membandingkan kinerja algoritma Bubble Sort, Shell Sort, dan Quick Sort dalam proses pengurutan bilangan acak menggunakan Bahasa pemrograman Java.

Pada penelitian ini dilakukan analisis terhadap efisiensi memori dan waktu komputasi pada algoritma pengurutan data, dengan membandingkan algoritma Bubble Sort, Selection Sort, Insertion Sort, Quick Sort, Merge Sort yang diimplementasikan menggunakan pemrograman Java. Data tersebut diuji dalam tiga variasi ukuran, yaitu 100, 1000, dan 5000 elemen. Setiap pengujian dilakukan sebanyak tiga kali dengan menggunakan aplikasi yang sama yaitu *Visual Studio Code* untuk memperoleh hasil yang lebih akurat. Pengukuran efisiensi memori dilakukan berdasarkan jumlah memori yang digunakan selama proses eksekusi, sedangkan waktu komputasi diukur dari durasi yang dibutuhkan komputer untuk menyelesaikan proses pengurutan data secara keseluruhan. Selanjutnya, kinerja algoritma *Tim Sort* dianalisis dan dikomparasikan dengan lima algoritma pengurutan lainya guna mengidentifikasi tingkat efisiensi *Tim Sort* dalam pemanfaatan memori serta waktu komputasi. Kontribusi utama penelitian ini adalah menyajikan analisis komparatif yang terukur antara *Tim Sort* dan algoritma pengurutan tradisional pada Bahasa pemrograman Java dengan dua perpsektif sekaligus yaitu efisiensi waktu dan efisiensi memori, serta merumuskan rekomendasi praktis pemilihan algoritma untuk berbagai skala data.

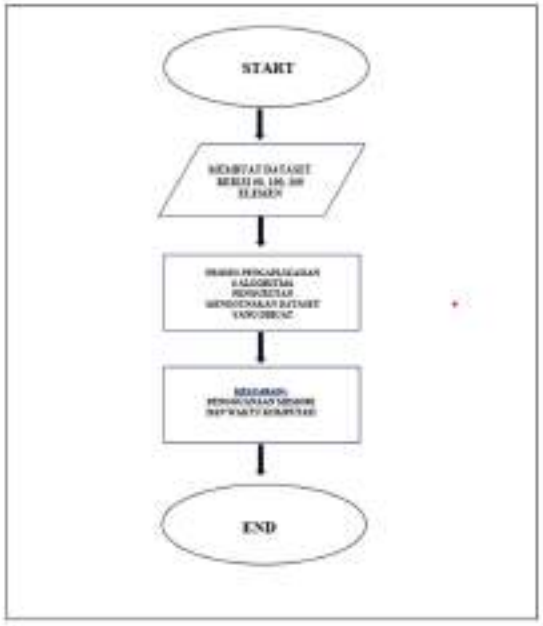
B. METODE PENELITIAN

Penelitian ini menggunakan metode komparatif dengan tujuan membandingkan tingkat efisiensi beberapa algoritma pengurutan data. Algoritma yang diuji meliputi *Tim Sort* sebagai algoritma modern serta beberapa algoritma pengurutan tradisional, yaitu *Bubble Sort, Selection Sort, Insertion Sort, Quick Sort, dan Merge Sort.* Melalui pendekatan komparatif ini, penelitian bertujuan mengetahui algoritma yang paling efisien berdasarkan penggunaan waktu komputasi dan memori. Langkah pertama dalam penelitian ini diawali dengan pembuatan dataset yang digunakan sebagai data uji. Dataset tersebut dibagi ke dalam tiga kategori ukuran, masing-masing memiliki rentang nilai antara 1 hingga 99. Dataset berukuran kecil berisi 100 elemen, dataset berukuran sedang berisi 1.000 elemen, dan dataset berukuran besar berisi 5.000 elemen. Seluruh elemen data dihasilkan menggunakan aplikasi *Random Number Generator Plus*. Pada aplikasi tersebut,

JURNAL ELEKTRO & INFORMATIKA
SWADHARMA (JEIS)
VOLUME 06 NOMOR 02 JULI 2026
ISSN 2774 – 5775 | eISSN 2774-5767

peneliti menentukan nilai minimum, nilai maksimum, serta jumlah elemen yang ingin dihasilkan untuk memperoleh kumpulan data acak secara otomatis.

Dataset yang telah diperoleh kemudian diimplementasikan ke dalam kode program menggunakan bahasa pemrograman Java. Setiap algoritma pengurutan diuji secara terpisah dengan dataset yang sama untuk memastikan hasil pengukuran bersifat objektif. Pengujian dilakukan dengan mengukur waktu komputasi dan penggunaan memori pada setiap algoritma. Proses pengukuran dilakukan dengan memanfaatkan kelas Runtime dalam Java untuk memperoleh data terkait memori sebelum dan sesudah proses pengurutan. Hasil pengujian dari setiap algoritma kemudian dibandingkan dan dianalisis untuk menentukan algoritma yang paling efisien berdasarkan kedua parameter tersebut. Gambar 1 menunjukkan tahapan dari penelitian ini.



Gambar 1. Diagram Alir Tahapan Penelitian

C. HASIL DAN PEMBAHASAN

Sistem Penelitian ini memanfaatkan bahasa pemrograman Java dan *Integrated Development Environment* (IDE) *Visual Studio Code Versi 1.105.1* sebagai sarana untuk pengembangan sekaligus eksekusi kode program. Proses pengujian dijalankan menggunakan laptop dengan spesifikasi Intel Core i7-8665U dengan kecepatan 1,02 GHz, memori 8 GB RAM, sistem Operasi 64 Bit, dan SSD berkapasitas 256 GB.

Algoritma yang diuji dalam penelitian ini meliputi *Tim Sort, Bubble Sort, Selection Sort, Insertion Sort, Quick Sort, dan Merge Sort*. Seluruh algoritma tersebut diuji menggunakan dataset yang sama, yaitu 100 data (array kecil), 1.000 data (array sedang), dan 5.000 data (array besar) menggunakan data numerik acak dengan rentang nilai 1 hingga 99.

Proses pengujian dilakukan sebanyak tiga kali untuk setiap ukuran dataset dengan rincian sebagai berikut:

1. Pengujian pertama hingga ketiga menggunakan dataset berisi 100 data (array kecil).
2. Pengujian keempat hingga keenam menggunakan dataset berisi 1.000 data (array sedang).
3. Pengujian ketujuh hingga kesembilan menggunakan dataset berisi 5.000 data (array besar).

Setiap algoritma diuji dan diulang selama tiga kali agar mendapat hasil yang tepat. Pengujian dilakukan menggunakan aplikasi *Visual Studio Code* sebagai lingkungan pengembangan untuk menjalankan program, sedangkan hasil pengujian dicatat menggunakan Microsoft Word.

Dataset lengkap pengujian dapat diakses melalui tautan bit.ly/4hVK923. Selama proses pengujian, setiap algoritma dievaluasi berdasarkan jumlah memori yang digunakan serta lama waktu yang diperlukan untuk menghasilkan urutan data. Berikut merupakan tautan hasil dokumentasi untuk masing-masing pengujian: link *Tim Sort*

(bit.ly/47EK9yI). Link *Bubble Sort* (bit.ly/3WF7rQ3). Link *Selection Sort* (bit.ly/4qKBNOD). Link *Insertion Sort* (bit.ly/3LjXhlw). Link *Quick Sort* (bit.ly/4p1QfAu). Dan link *Merge Sort* (bit.ly/4qNZqWy/).

Tim Sort

```
// === Fungsi utama TimSort ===
public static void timSort(int[] arr, int n) {
   int RUN = 32; // Ukuran blok kecil untuk Insertion Sort

   // 1. Urutkan setiap blok kecil menggunakan Insertion Sort
   for (int i = 0; i < n; i += RUN) {
      for (int j = i + 1; j < Math.min(i + RUN, n); j++) {
         int temp = arr[j];
         int k = j - 1;
         while (k >= i && arr[k] > temp) {
            arr[k + 1] = arr[k]; // Geser elemen lebih besar
            k--;
         }
         arr[k + 1] = temp; // Tempatkan elemen di posisi yang benar
      }
   }
   // 2. Gabungkan blok-blok yang sudah terurut secara bertahap
   for (int size = RUN; size < n; size *= 2) {
      for (int left = 0; left < n; left += 2 * size) {
         int mid = left + size - 1;
         int right = Math.min(left + 2 * size - 1, n - 1);
         if (mid < right) {
            // Merge dua blok terurut
            int[] leftArr = Arrays.copyOfRange(arr, left, mid + 1);
            int[] rightArr = Arrays.copyOfRange(arr, mid + 1, right + 1);
            int i = 0, j = 0, k = left;
            while (i < leftArr.length && j < rightArr.length) {
               arr[k++] = (leftArr[i] <= rightArr[j]) ? leftArr[i++] : rightArr[j++];
            }
            while (i < leftArr.length) arr[k++] = leftArr[i++];
            while (j < rightArr.length) arr[k++] = rightArr[j++];
         }
      }
   }
}
```

Kode program diatas merupakan implementasi dari algortima *Tim Sort*, Pola operasional algoritma *TimSort* merepresentasikan hibridisasi antara pendekatan *Insertion Sort* dan *Merge Sort*, yang dirancang untuk memaksimalkan efisiensi proses pengurutan. Algoritma ini diawali dengan pembagian struktur data array menjadi sejumlah segmen kecil yang dikenal sebagai *run*. Setiap *run* kemudian diurutkan secara individual menggunakan metode Insertion Sort, mengingat algoritma ini menunjukkan kinerja optimal untuk pengurutan data dalam skala terbatas. Setelah seluruh *run* berhasil tersusun secara terurut, tahap berikutnya melibatkan penggabungan (*merge*) *run*-run tersebut secara bertahap, mengikuti mekanisme yang serupa dengan algoritma *Merge Sort*. Proses penggabungan dilakukan secara sistematis hingga seluruh segmen array terintegrasi menjadi satu kesatuan yang tersusun secara konsisten. Pendekatan hibrida ini memungkinkan *TimSort* beroperasi secara optimal terutama ketika data awal sudah memiliki sebagian elemen yang tersusun, sekaligus mempertahankan stabilitas pengurutan, yakni urutan relatif elemen dengan nilai identik tetap terjaga. Pada akhirnya, algoritma ini menghasilkan array yang tersusun secara menaik (*ascending order*), mulai dari nilai terkecil hingga terbesar, dengan efisiensi signifikan baik dari sisi waktu proses maupun penggunaan memori.

Tabel 1. Hasil Pengujian Algoritma Tim Sort

Uji Ke	Hasil Pengujian Algoritma Tim Sort		
	Jumlah Data	Pemakaian Memori (KB)	Waktu Komputasi (milidetik)
1	100	40	1.2786
2	100	40	3456
3	100	40	1.5763
4	1.000	40	4.4448
5	1.000	40	3.8101
6	1.000	40	5.7755
7	5.000	251	22.8013
8	5.000	251	15.1905
9	5.000	251	25.1534

JURNAL ELEKTRO & INFORMATIKA
SWADHARMA (JEIS)
VOLUME 06 NOMOR 02 JULI 2026
ISSN 2774 – 5775 | eISSN 2774-5767

Berdasarkan hasil pengujian tabel 1, dari perspektif memori, rata-rata konsumsi memori pada pengujian dengan 100 data adalah 40 KB, dengan 1.000 data adalah 40 KB, dan dengan 5.000 data adalah 251 KB. Namun, dari perspektif waktu komputasi, waktu komputasi rata-rata untuk pengujian dengan 100 data adalah 1.40016 milidetik, dengan 1.000 data adalah 4.6768 milidetik dan dengan 5.000 data adalah 21.0484 milidetik.

Bubble Sort

```
int n = arr.length;
for (int i = 0; i < n - 1; i++) {          // Loop luar: setiap pass
   boolean swapped = false;                // Flag untuk cek pertukaran
   for (int j = 0; j < n - i - 1; j++) {   // Loop dalam: bandingkan elemen berurutan
      if (arr[j] > arr[j + 1]) {           // Jika tidak urut
         int temp = arr[j];                // Tukar posisi
         arr[j] = arr[j + 1];
         arr[j + 1] = temp;
         swapped = true;                   // Tandai ada pertukaran
      }
   }
   if (!swapped) break;                    // Hentikan jika array sudah terurut
}
```

Kode program diatas merupakan implementasi dari algortima *Bubble Sort*, Pola operasional algoritma *Bubble Sort* dilaksanakan melalui mekanisme perbandingan antar dua elemen yang berposisi bersebelahan dalam struktur data. Apabila elemen pada posisi kanan memiliki nilai numerik lebih kecil dibandingkan elemen pada posisi kiri, maka kedua elemen tersebut mengalami proses pertukaran (*swap*). Proses pertukaran ini dilakukan secara iteratif sepanjang satu siklus evaluasi, sehingga setiap pasangan elemen diperiksa secara menyeluruh. Setelah satu siklus selesai, algoritma melanjutkan ke siklus berikutnya dengan pola operasi yang identik, hingga tercapai kondisi di mana tidak terjadi pertukaran nilai sama sekali. Kondisi ini menandakan bahwa seluruh elemen telah tersusun secara benar sesuai urutan yang diinginkan. Dengan demikian, algoritma Bubble Sort mampu menghasilkan susunan data secara menaik (ascending order), dimulai dari nilai terkecil hingga nilai terbesar, melalui proses iteratif yang sistematis dan deterministik.

Tabel 2. Hasil Pengujian Algoritma Bubble Sort

Uji Ke	Hasil Pengujian Algoritma Bubble Sort		
	Jumlah Data	Pemakaian Memori (KB)	Waktu Komputasi (milidetik)
1	100	40	4.5252
2	100	40	1.3303
3	100	40	2.0375
4	1.000	40	8.6362
5	1.000	40	9.8187
6	1.000	40	9.0292
7	5.000	96	56.4221
8	5.000	96	30.6994
9	5.000	96	40.2875

Berdasarkan hasil pengujian tabel 2, jika dilihat dari segi memori, rata-rata pemakaian memori pada pengujian dengan 100 data adalah 40 KB, dengan 1.000 data adalah 40 KB, dan dengan 5.000 data adalah 96 KB. Namun, dari perspektif waktu komputasi, waktu komputasi rata-rata untuk pengujian dengan 100 data adalah 2.631 milidetik, dengan 1.000 data adalah 9.16136 milidetik, dan dengan 5.000 data adalah 188,097.3289. milidetik.

Selection Sort

```
int n = arr.length;

for (int i = 0; i < n - 1; i++) {
   int minIndex = i;

   // Cari elemen terkecil di sisa array
   for (int j = i + 1; j < n; j++) {
      if (arr[j] < arr[minIndex]) {
         minIndex = j;
      }
   }

   // Tukar elemen terkecil dengan elemen pertama dari bagian belum terurut
   if (minIndex != i) {
      int temp = arr[minIndex];
```

```
        arr[minIndex] = arr[i];
        arr[i] = temp;
    }
}
```

Kode program diatas merupakan implementasi dari algortima *Selection Sort*, Pola operasional algoritma *Selection Sort* dijalankan melalui mekanisme seleksi nilai terkecil dari keseluruhan elemen yang terdapat dalam array. Nilai terkecil yang teridentifikasi pada tahap awal kemudian ditukar (*swap*) dengan elemen yang berada pada posisi indeks pertama. Setelah pertukaran tersebut, proses pencarian nilai terkecil dilanjutkan pada subset elemen yang tersisa, dimulai dari indeks kedua, dan nilai terkecil yang ditemukan pada tahap ini ditukar dengan elemen pada posisi indeks kedua. Mekanisme pencarian dan pertukaran ini berlangsung secara iteratif hingga seluruh elemen menempati posisi yang sesuai sesuai urutan yang diinginkan. Dengan demikian, algoritma Selection Sort menghasilkan susunan data secara menaik (ascending order), dimulai dari nilai terkecil hingga nilai terbesar. Jumlah pertukaran yang terjadi bergantung secara langsung pada jumlah elemen dalam array, sedangkan proses pencarian nilai minimum memerlukan evaluasi berulang pada seluruh elemen yang tersisa pada setiap iterasi, yang memengaruhi kompleksitas waktu keseluruhan algoritma.

Tabel 3. Hasil Pengujian Algoritma Selection Sort

Uji Ke	Hasil Pengujian Algoritma Selection Sort		
	Jumlah Data	Pemakaian Memori (KB)	Waktu Komputasi (milidetik)
1	100	40	1.869
2	100	40	1.1346
3	100	40	1.9142
4	1.000	40	8.6803
5	1.000	40	8.7331
6	1.000	40	8.4616
7	5.000	96	31.0111
8	5.000	96	33.2489
9	5.000	96	41.1057

Berdasarkan hasil pengujian tabel 3, jika dilihat dari segi memori, rata-rata penggunaan memori pada pengujian dengan 100 data adalah 40 KB, dengan 1.000 data ialah 40 KB, dan dengan 5.000 data ialah 96 KB. Namun, dari perspektif waktu komputasi, waktu komputasi rata-rata untuk pengujian dengan 100 data adalah 1.63926 milidetik, dengan 1.000 data adalah 25.875 milidetik, dan dengan 5.000 data adalah 25.1219 milidetik.

Insertion Sort

```
int n = arr.length;

for (int i = 1; i < n; i++) {
    int key = arr[i];
    int j = i - 1;

    // Geser elemen yang lebih besar dari key ke kanan
    while (j >= 0 && arr[j] > key) {
        arr[j + 1] = arr[j];
        j--;
    }
    arr[j + 1] = key; // Tempatkan key di posisi yang
tepat
}
```

Kode program diatas merupakan implementasi dari algortima *Insertion Sort*, Pola operasional algoritma *Insertion Sort* dimulai dengan asumsi bahwa elemen pertama dalam array telah berada pada posisi yang benar, sementara elemen-elemen berikutnya masih dalam keadaan belum terurut. Proses pengurutan dimulai dari elemen kedua, yang dijadikan sebagai kunci (*key*). Elemen kunci ini kemudian dibandingkan secara berurutan dengan elemen-elemen di sebelah kirinya. Apabila elemen sebelah kiri memiliki nilai yang lebih besar daripada kunci, maka elemen tersebut digeser satu posisi ke arah kanan untuk memberikan ruang bagi penyisipan kunci. Pergeseran ini dilakukan secara iteratif hingga ditemukan posisi yang tepat, yakni posisi tempat sebelumnya memiliki nilai lebih kecil atau sama dengan kunci. Setelah posisi yang sesuai teridentifikasi, kunci disisipkan ke dalam posisi tersebut.

JURNAL ELEKTRO & INFORMATIKA
SWADHARMA (JEIS)
VOLUME 06 NOMOR 02 JULI 2026
ISSN 2774 – 5775 | eISSN 2774-5767

Prosedur ini berlanjut secara berurutan untuk elemen ketiga, keempat, dan seterusnya, sehingga pada setiap langkah, subarray di sebelah kiri elemen kunci selalu berada dalam keadaan terurut. Proses iteratif ini berakhir ketika seluruh elemen array telah melalui tahap perbandingan dan penyisipan, menghasilkan susunan data yang terurut secara menaik (*ascending order*), mulai dari nilai terkecil hingga nilai terbesar. Pendekatan ini memastikan stabilitas pengurutan, karena urutan relatif elemen dengan nilai identik tetap terjaga, sekaligus meminimalkan jumlah pergeseran yang diperlukan untuk data yang sebagian sudah terurut.

Tabel 4. Hasil Pengujian Algoritma *Insertion Sort*

Uji Ke	Hasil Pengujian Algoritma Insertion Sort		
	Jumlah Data	Pemakaian Memori (KB)	Waktu Komputasi (milidetik)
1	100	40	1.9303
2	100	40	1.9466
3	100	40	1.0181
4	1.000	40	6.1726
5	1.000	40	7.8368
6	1.000	40	7.6174
7	5.000	96	22.2042
8	5.000	96	22.9072
9	5.000	96	28.2614

Berdasarkan hasil pengujian tabel , jika dilihat dari segi memori, rata-rata pemakaian memori pada pengujian dengan 100 data adalah 40 KB, dengan 1.000 data adalah 40 KB, dan dengan 5.000 data adalah 96 KB. Namun, dari perspektif waktu komputasi, waktu komputasi rata-rata untuk pengujian dengan 100 data adalah 1.6316 milidetik, dengan 1.000 data adalah 7.20893 milidetik, dan dengan 5.000 data adalah 24.4576 milidetik.

Quick Sort

```
if (low < high) {
int pivotIndex = partition(arr, low, high); // Tentukan posisi pivot
   quickSort(arr, low, pivotIndex - 1);       // Rekursif ke kiri pivot
   quickSort(arr, pivotIndex + 1, high);      // Rekursif ke kanan pivot
}
```

Kode program diatas merupakan implementasi dari algortima *Quick Sort*, Pola operasional algoritma *Quick Sort* dimulai dengan pemilihan satu elemen sebagai pivot, yang berfungsi sebagai acuan utama dalam proses pengurutan. Seluruh elemen lain dalam array dibandingkan dengan nilai acuan, di mana elemen dengan nilai lebih kecil dari nilai acuan ditempatkan pada sisi kiri, sedangkan elemen dengan nilai lebih besar ditempatkan pada sisi kanan. Proses pembagian ini dikenal sebagai partitioning, yang menjamin bahwa pivot menempati posisi finalnya yang benar dalam urutan akhir setelah setiap partisi.

Setelah tahap partitioning selesai, algoritma diterapkan kembali secara rekursif pada subarray kiri dan kanan, yakni pada kelompok elemen dengan nilai lebih kecil maupun lebih besar daripada pivot. Proses rekursif ini terus berlangsung hingga setiap subarray hanya terdiri dari satu elemen atau tidak lagi memerlukan pengurutan tambahan. Pendekatan ini memungkinkan pembentukan struktur data yang terurut secara bertahap melalui pembagian dan penggabungan yang sistematis.

Dengan demikian, algoritma Quick Sort menghasilkan susunan data menaik (ascending order), dari nilai terkecil hingga terbesar, dengan pivot berperan sentral dalam menentukan posisi elemen dan memfasilitasi efisiensi pengurutan. Metode ini memiliki keunggulan dalam hal kompleksitas waktu rata-rata, meskipun kinerjanya dapat dipengaruhi oleh pemilihan pivot pada dataset tertentu.

Tabel 5. Hasil Pengujian Algoritma Quick Sort

Uji Ke	Hasil Pengujian Algoritma Quick Sort		
	Jumlah Data	Pemakaian Memori (KB)	Waktu Komputasi (milidetik)
1	100	40	2.4579
2	100	40	0.9396

Uji Ke	Hasil Pengujian Algoritma Quick Sort		
	Jumlah Data	Pemakaian Memori (KB)	Waktu Komputasi (milidetik)
3	100	40	0.8901
4	1.000	40	4.7131
5	1.000	40	3.6
6	1.000	40	5.2718
7	5.000	96	13.7851
8	5.000	96	14.6089
9	5.000	96	16.9416

Berdasarkan hasil pengujian tabel 5, ditinjaudari prespektif konsumsi memori, rata-rata penggunaan memori pada pengujian dengan 100 data adalah 40 KB, dengan 1.000 data adalah 40 KB, dan dengan 5.000 data adalah 96 KB. Namun, dari perspektif waktu komputasi, waktu komputasi rata-rata untuk pengujian dengan 100 data adalah 1.43183 milidetik, dengan 1.000 data adalah 13.5849 milidetik, dan dengan 5.000 data adalah 15.11186 milidetik.

Merge Sort

```
if (left < right) {
   int mid = (left + right) / 2;

   // Rekursi untuk bagian kiri dan kanan
   mergeSort(arr, left, mid);
   mergeSort(arr, mid + 1, right);

   // Gabungkan hasil pengurutan
   merge(arr, left, mid, right);
}
```

Kode program diatas merupakan implementasi dari algortima *Merge Sort*, Pola operasional algoritma *Merge Sort* dimulai dengan pembagian (divide) array secara rekursif menjadi dua subarray hingga setiap subarray hanya terdiri atas satu elemen, kondisi di mana setiap elemen secara implisit dianggap telah terurut. Tahap berikutnya adalah proses penggabungan (merge), di mana dua subarray yang lebih kecil digabung menjadi satu subarray yang lebih besar dengan mempertahankan keterurutan. Proses penggabungan dilakukan dengan membandingkan elemen-elemen dari kedua subarray dan menempatkannya secara berurutan sehingga membentuk urutan menaik (ascending order), mulai dari nilai terkecil hingga terbesar.

Prosedur pembagian dan penggabungan ini berlangsung secara iteratif dan rekursif hingga seluruh subarray kembali terintegrasi menjadi satu kesatuan array yang telah tersusun secara sempurna. Pendekatan ini merupakan implementasi dari strategi divide and conquer, yakni memecah masalah besar menjadi bagian-bagian yang lebih kecil, menyelesaikan setiap bagian secara independen, dan kemudian menggabungkannya kembali untuk memperoleh hasil pengurutan yang optimal. Metode ini menjamin stabilitas pengurutan, karena posisi relatif elemen dengan nilai identik tetap dipertahankan, serta efisiensi waktu yang konsisten terutama pada dataset dengan ukuran besar.

Tabel 6. Hasil Pengujian Algoritma Merge Sort

Uji Ke	Hasil Pengujian Algoritma Merge Sort		
	Jumlah Data	Pemakaian Memori (KB)	Waktu Komputasi (milidetik)
1	100	40	1.5774
2	100	40	0.9875
3	100	40	1.9832
4	1.000	40	33.0365
5	1.000	40	3.5722
6	1.000	40	5.0721
7	5.000	96	14.1419
8	5.000	96	18.155
9	5.000	96	14.4867

Berdasarkan hasil pengujian tabel 6, jika dilihat dari segi memori, rata-rata pemakaian memori pada pengujian dengan 100 data adalah 40 KB, dengan 1.000 data adalah 122 KB, dan dengan 5.000 data adalah 531 KB. Namun, dari perspektif waktu komputasi, waktu komputasi rata-rata untuk pengujian dengan 100 data adalah 1.49773 milidetik, dengan 1.000 data adalah 13.8936 milidetik, dan dengan 5.000 data adalah 15.59453 milidetik.

Dari hasil pengujian keenam algoritma yang sudah dilakukan, rangkuman rata-rata

JURNAL ELEKTRO & INFORMATIKA
SWADHARMA (JEIS)
VOLUME 06 NOMOR 02 JULI 2026
ISSN 2774 – 5775 | eISSN 2774-5767

hasil pengujian setiap algoritma dapat dilihat pada gambar 2 dan 3 dibawah ini.

Gambar 2. Diagram Rata-rata Pemakaian Memori

Gambar 3. Diagram Rata-rata Kecepatan memori

Selection Sort, Insertion Sort, Quick Sort, dan Merge Sort, diperoleh perbedaan kinerja yang signifikan dalam hal waktu komputasi dan pemakaian memori pada berbagai ukuran dataset. Secara umum, *Tim Sort* menunjukkan performa paling efisien dan stabil dibandingkan algoritma lainnya. Hal ini disebabkan oleh sifat hibridanya yang menggabungkan keunggulan Insertion Sort untuk data berukuran kecil dan *Merge Sort* untuk proses penggabungan berskala besar, namun pada sisi penggunaan memori jika semakin banyak data yang dimasukkan maka semakin banyak memori yang digunakan.

Algoritma *Quick Sort* juga menunjukkan waktu komputasi yang kompetitif pada berbagai ukuran data, terutama karena penerapan strategi *divide and conquer* yang efektif dalam membagi dan mengurutkan data. Namun, efisiensinya sangat bergantung pada pemilihan pivot, yang dapat memengaruhi performa secara signifikan.

Sementara itu, *Merge Sort* memberikan hasil yang konsisten dan stabil dengan kompleksitas waktu yang baik pada dataset besar, meskipun memerlukan penggunaan memori tambahan yang relatif lebih tinggi dibandingkan algoritma lain. Berbeda dengan ketiga algoritma tersebut, algoritma pengurutan sederhana seperti *Bubble Sort, Selection Sort, dan Insertion Sort* menunjukkan performa yang kurang efisien terutama pada dataset berukuran besar. Ketiga algoritma ini memerlukan waktu komputasi yang jauh lebih lama akibat banyaknya proses perbandingan dan pertukaran elemen.

D. PENUTUP

Kesimpulan dari penelitian ini, Tim Sort merupakan algoritma paling efisien dalam hal waktu dan penggunaan memori. Quick Sort dan Merge Sort juga cepat untuk data besar, sedangkan Bubble Sort, Selection Sort, dan Insertion Sort jauh lebih lambat dan hanya cocok untuk dataset kecil atau pembelajaran dasar. Berdasarkan hasil penelitian ini, pembaca disarankan untuk mempertimbangkan penggunaan Tim Sort atau Quick Sort ketika menangani proses pengurutan pada dataset berukuran sedang hingga besar karena keduanya menunjukkan kinerja komputasi yang lebih efisien. Selain itu, Merge Sort dapat menjadi alternatif ketika dibutuhkan hasil pengurutan yang stabil meskipun memerlukan memori tambahan. Sebaliknya, penggunaan algoritma Bubble Sort, Selection Sort, dan Insertion Sort sebaiknya dihindari untuk dataset besar karena waktu komputasi yang kurang optimal. Pemilihan algoritma pengurutan sebaiknya disesuaikan dengan kebutuhan skenario data, ukuran dataset, serta

keterbatasan sumber daya komputasi yang tersedia. Saran untuk penelitian selanjutnya disarankan menggunakan variasi ukuran data yag lebih besar, serta melakukan penukuran efisiensi memori dan waktu eksekusi secara langsung agar hasil analisis bersifat lebih komprehensif.

E. DAFTAR PUSTAKA

Al-aydi, B. M., & Abu-naser, S. S. (2025). *Comparative Study of Traditional and AI-Enhanced Sorting Algorithms : Comparative Study of Traditional and AI-Enhanced Sorting Algorithms : QuickSort , MergeSort , HeapSort , and TimSort*. (September). https://doi.org/10.13140/RG.2.2.13915.84005

Ali, H., Nawaz, H., Maitlo, A., & Soomro, I. (2021). Performance Analysis of Heap Sort and Insertion Sort Algorithm. *International Journal of Emerging Trends in Engineering Research*, *9*(5), 580–586. https://doi.org/10.30534/ijeter/2021/08952021

Ali, M. I., Fardiarsyah, R. D., Shodik, L., Kinanti, F. Z. D., & Pujiono, I. P. (2025). Analisis Komparatif Efisiensi Memori dan Waktu Komputasi pada 8 Algoritma Sorting menggunakan C++. *LogicLink*, *2*(1), 1–17. https://doi.org/10.28918/logiclink.v2i1.10868

Amanda, S., Anastasya, L. D., Sulistia, F., Purnamasari, N., & Pujiono, I. P. (2026). Analisis Perbandingan Efisiensi Algoritma Introsort Dengan Algoritma Tradisional Bubble Sort dan Selection Sort. *JEIS: Jurnal Elektro Dan Informatika Swadharma*, *6*(1), 155–165. https://doi.org/10.56486/jeis.vol6no1.1086

Basir, R. R. (2020). Analisis Kompleksitas Ruang dan Waktu Terhadap Laju Pertumbuhan Algoritma Heap Sort, Insertion Sort dan Merge dengan Pemrograman Java. *STRING (Satuan Tulisan Riset Dan Inovasi Teknologi)*, *5*(2), 109. https://doi.org/10.30998/string.v5i2.6250

Ghofur, A., Nazila, J., Holidiyah, I., Kholifah, S., Husaini Kulsum, F., Insiyah, N., Ibrahimy, U., Situbondo, K., & Timur, J. (2025). Penerapan Algoritma Insertion Sort Pada Aplikasi Pengolahan Data Mahasiswa Menggunakan Java Di Fakultas Sains Dan Teknologi Prodi Teknologi Informasi. *Eastasouth Journal of Positive Community Services*, *4*(01), 12–19. https://doi.org/10.58812/ejpcs.v4i01

Gudiato, C., Cahyaningtyas, C., & P., N. (2024). Decision Support System in Selecting Residential Houses using the S.A.W Method. and Fuzzy Logic. *G-Tech : Jurnal Teknologi Terapan*, *8*(1), 186–195. https://doi.org/10.33379/gtech.v8i1.3601

Hanafi, M. R., Faadhilah, M. A., Dwi Putra, M. T., & Pradeka, D. (2022). Comparison Analysis of Bubble Sort Algorithm with Tim Sort Algorithm Sorting Against the Amount of Data. *Journal of Computer Engineering, Electronics and Information Technology*, *1*(1), 29–38. https://doi.org/10.17509/coelite.v1i1.43794

Ilham, M. N., Setiawan, A. F., Kholifatun, I., Aldiansyah, M. H., & Pujiono, I. P. (2025). Comparative Analysis of Memory Performance and Processing Time of Five Sorting Algorithms Using C++ Programming Language. *Journal of Artificial Intelligence and Engineering Applications (JAIEA)*, *4*(3), 1950–1956. https://doi.org/10.59934/jaiea.v4i3.1051

Iskandar, I. D., Amirulloh, I., Pertiwi, M. W., Kusmira, M., Hikmah, A. B., & Supriadi,

D. (2020). Analysis of Bubble Sort and Insertion Sort Algorithm on Memory Efficiency Using Data Mining Approach. *Jurnal PILAR Nusa Mandiri*, *16*(1), 89–96. https://doi.org/10.33480/pilar.v16i1.1165

Isyqi, A., Astuti, G. D., Saputro, A. D., & Barryananda, M. A. (2024). Perbandingan Kinerja Algoritma Bubble Sort, Insertion Sort, Dan Selection Sort Menggunakan Data Bilangan Numerik Pada Program Python. *Seminar Nasional Sains Dan Teknologi "SainTek" Seri II*, *1*(2), 3047–6569. https://conference.ut.ac.id/index.php/saintek/article/view/2617

Latifah, R., Arriyanti, E., & Hakim, A. R. (2021). *Perbandingan Efisiensi Kinerja Algoritma Bubble Sort dan Algoritma Selection Sort Pada Parallel Programming* [STMIK Widya Cipta Dharma]. https://repository.wicida.ac.id/3687/

Mahrozi, N., & Faisal, M. (2023). Analisis Perbandingan Kecepatan Algoritma Selection Sort dan Bubble Sort. *Scientica: Jurnal Ilmiah Sains Dan Teknologi*, *1*(2), 89–98. https://doi.org/10.572349/scientica.v1i2.209

Musyaffa, M. Z., Raharjo, K., Faiz, M., Ammarulloh, S., & Pujiono, I. P. (2025). Efisiensi Memori dan Waktu: Array Sorting Algorithm vs Algoritma Pengurutan Tradisional Menggunakan Python. *JEIS: Jurnal Elektro Dan Informatika Swadharma*, *5*(2), 72–81. https://doi.org/10.56486/jeis.vol5no2.785

Pujiono, I. P., Kamal, M. R., Prayogi, A., Sari, C. A., & Ikhsanuddin, R. M. (2025). Algoritma Counting Sort Vs Algoritma Pengurutan Modern: Analisis Efisiensi Memori Dan Waktu Komputasi. *Jurnal Informatika Dan Teknik Elektro Terapan*, *13*(3). https://doi.org/10.23960/jitet.v13i3.6657

Sari, N., Gunawan, W. A., Sari, P. K., Zikri, I., & Syahputra, A. (2022). Analisis Algoritma Bubble Sort Secara Ascending Dan Descending Serta Implementasinya Dengan Menggunakan Bahasa Pemrograman Java. *ADI Bisnis Digital Interdisiplin Jurnal*, *3*(1), 16–23. https://doi.org/10.34306/abdi.v3i1.625

Sena, M. B., Hanun, R. M., Purnama, I. R., & Ardian, M. (2024). Perbandingan Kinerja Algoritma Sorting Dalam Pengurutan Data Menggunakan Bahasa Python. *Prosiding Seminar Nasional Sains Dan Teknologi Seri 02*, *1*(2), 310–314. https://conference.ut.ac.id/index.php/saintek/article/view/2633

Sutanto, D. S., Kirana, C., & Wahyuningsih, D. (2025). Analisis Efisiensi Waktu Bubble, Insertion, Merge, Dan Quick Sort Menggunakan Python. *Metik Jurnal*, *9*(1), 159–169. https://doi.org/10.47002/metik.v9i1.1053

Wijaya, S., Fauziah, & Harjanti, T. W. (2024). Perbandingan Algoritma Sorting dengan Menggunakan Bahasa Pemograman Javascript dalam Penggunaan Waktu Komputasi dan Penggunaan Memori. *STRING : Satuan Tulisan Riset Dan Inovasi Teknologi*, *8*(2), 294–302. https://doi.org/10.30998/string.v8i3.17972

Zulfa, M., Mikhael, M., & Sari, B. (2022). Analisis Perbandingan Algoritma Bubble Sort, Shell Sort, dan Quick Sort dalam Mengurutkan Baris Angka Acak menggunakan Bahasa Java. *Jurnal Ilmiah Wahana Pendidikan*, *8*(13), 237–246. https://doi.org/10.5281/zenodo.6962346

PERBANDINGAN EFISIENSI MEMORI DAN WAKTU KOMPUTASI ALGORITMA PENGURUTAN REKURSIF DAN ITERATIF DI PYTHON (*MERGE SORT* DAN *QUICK SORT*)

Regina Nur Aeni[1)], Alina Putri As Salwa[2)], Muhammad Farras Abiy[3)], Muhammad Noval Syafiq Sofi[4)], Imam Prayogo Pujiono[5)]
[1,2,3,4]Bisnis Digital, Fakultas Ekonomi dan Bisnis Islam, UIN K.H Abdurrahman Wahid Pekalongan
[4]Informatika, Fakultas Ekonomi dan Bisnis Islam, UIN K.H Abdurrahman Wahid Pekalongan

Correspondence author: RN Aeni, reginanuraeni23@gmail.com, Pekalongan, Indonesia

Abstract

This study aims to analyze and compare the computational time efficiency and memory usage of the recursive and iterative approaches in the Merge Sort and Quick Sort algorithms in Python. The method used is an experimental quantitative approach with performance testing on two data sizes, namely 10,000 and 1,000,000 random number elements. Execution time is measured using the time and timeit libraries, while memory usage is monitored using the psutil library. The results show that, in general, the iterative approach is more efficient in memory usage because it does not require a function call stack as in recursion. In terms of execution speed, recursive Quick Sort provides the fastest average time of 0.002 seconds on data of 10,000 elements, while on data of 1,000,000 elements, iterative Quick Sort shows better time performance than the recursive version. Iterative Merge Sort is consistently more memory efficient than recursive Merge Sort on both data sizes. The time difference between the recursive and iterative approaches is relatively small for medium-sized data, but memory consumption increases significantly with the recursive version as the data size increases. These findings indicate that the choice of implementation approach for sorting algorithms in Python requires considering the trade-off between speed and memory efficiency. The iterative approach is more suitable for applications with limited memory resources, while the recursive approach remains relevant for scenarios that prioritize simplicity of implementation and clarity of algorithm structure. The results of this study are expected to serve as a reference for software developers and algorithm researchers in determining optimal sorting implementation strategies in Python.

Keyword *: comparison, Quick Sort, Merge Sort, computational time, memory efficiency*

Abstrak

Penelitian ini bertujuan menganalisis dan membandingkan efisiensi waktu komputasi dan penggunaan memori antara pendekatan rekursif dan iteratif pada algoritma pengurutan *Merge Sort* dan *Quick Sort* menggunakan bahasa pemrograman Python. Metode yang digunakan adalah pendekatan kuantitatif eksperimental dengan pengujian kinerja pada dua ukuran data, yaitu 10.000 dan

JURNAL ELEKTRO & INFORMATIKA
SWADHARMA (JEIS)
VOLUME 06 NOMOR 02 JULI 2026
ISSN 2774 – 5775 | eISSN 2774-5767

1.000.000 elemen bilangan acak. Waktu eksekusi diukur menggunakan pustaka time dan timeit, sedangkan penggunaan memori dipantau dengan pustaka psutil. Hasil penelitian menunjukkan bahwa secara umum pendekatan iteratif lebih efisien dalam penggunaan memori karena tidak memerlukan tumpukan pemanggilan fungsi (*call stack*) sebagaimana pada rekursi. Dari sisi kecepatan eksekusi, *Quick Sort* rekursif memberikan waktu rata-rata tercepat sebesar 0,002 detik pada data 10.000 elemen, sedangkan pada data 1.000.000 elemen *Quick Sort* iteratif menunjukkan kinerja waktu yang lebih baik dibandingkan versi rekursif. *Merge Sort* iteratif konsisten lebih hemat memori dibandingkan *Merge Sort* rekursif pada kedua ukuran data. Perbedaan waktu antara pendekatan rekursif dan iteratif relatif kecil pada data berukuran menengah, namun konsumsi memori meningkat secara nyata pada versi rekursif seiring bertambahnya ukuran data. Temuan ini mengindikasikan bahwa pemilihan pendekatan implementasi algoritma pengurutan di Python perlu mempertimbangkan *trade-off* antara kecepatan dan efisiensi memori. Pendekatan iteratif lebih sesuai untuk aplikasi dengan keterbatasan sumber daya memori, sedangkan pendekatan rekursif tetap relevan untuk skenario yang mengutamakan kesederhanaan implementasi dan kejelasan struktur algoritma. Hasil penelitian ini diharapkan dapat menjadi rujukan bagi pengembang perangkat lunak dan peneliti algoritma dalam menentukan strategi implementasi pengurutan yang optimal pada Python.

Kata Kunci : perbandingan, *Quick Sort, Merge Sort*, waktu komputasi, efisiensi memori

A. PENDAHULUAN

Algoritma pengurutan adalah komponen penting dalam sistem komputasi modern, selain itu efisiensi algoritma juga hal yang sangat diperlukan dalam pengembangan suatu sistem (Ilham et al., 2025). Aspek utama terkait dengan efisiensi algoritma adalah efisiensi waktu dan memori. Efisiensi waktu adalah tahap kecepatan setiap algoritma untuk menyelesaikan masalah tergantung pada jumlah langkah yang dibutuhkan, Sementara itu, efisiensi memori adalah jumlah memori yang dibutuhkan selama eksekusi berlangsung (Ali et al., 2025). Efisiensi memori dalam pemrosesan data yang berskala besar menjadi faktor penting karena dapat mempengaruhi kecepatan sistem. Keseimbangan antara waktu dan efisiensi ruang merupakan faktor penentu keberhasilan dalam mengimplementasikan suatu algoritma (Musyaffa et al., 2025).

Pendekatan algoritma pengurutan yang digunakan dalam pemrograman adalah pendekatan rekursif dan pendekatan iteratif (Zahwa et al., 2025). Pendekatan rekursif bekerja dengan prinsip *divide and conquer*, yaitu membagi masalah menjadi submasalah yang lebih kecil hingga mencapai kondisi dasar, kemudian hasilnya digabungkan kembali untuk menghasilkan solusi akhir (Pratama et al., 2025). Sebaliknya, pendekatan iteratif menggunakan struktur pengulangan (*looping*) yang menghindari pemanggilan fungsi berulang sehingga dapat mengurangi penggunaan memori pada *stack frame* (Febriansyah et al., 2025).

Beberapa penelitian menunjukkan bahwa algoritma rekursif, seperti *Merge Sort* dan *Quick Sort*, cenderung lebih mudah dipahami dan diimplementasikan, namun memiliki kelemahan pada efisiensi memori karena setiap pemanggilan fungsi menambah beban pada tumpukan memori (Ala'Anzy et al., 2024). Sebaliknya, versi iteratif dari

algoritma yang sama mampu mengurangi penggunaan memori, tetapi terkadang mengalami penurunan kecepatan eksekusi akibat kompleksitas logika *loop* (Febriansyah et al., 2025).

Analisis perbandingan antara algoritma *Merge Sort* dan *Quick Sort* juga menunjukkan bahwa efisiensi waktu sangat bergantung pada karakteristik data yang diurutkan serta jumlah elemen yang diproses (Mishal et al., 2021). *Merge Sort* lebih stabil dan memiliki kompleksitas waktu O (n log n) dalam semua kasus, sedangkan *Quick Sort* memiliki kompleksitas rata-rata O (n log n) namun dapat memburuk menjadi O(n^2) pada kondisi data tertentu (Syahputra et al., 2026). Oleh karena itu, penelitian ini berfokus pada perbandingan efisiensi waktu dan memori antara implementasi rekursif dan iteratif pada algoritma *Merge Sort* dan *Quick Sort*, untuk mengetahui pendekatan mana yang lebih optimal dalam konteks pengolahan data besar.

B. METODE PENELITIAN

Penelitian ini akan menggunakan pendekatan kuantitatif eksperimental dengan metode komparatif, yang ditujukan untuk menganalisis perbandingan efisiensi algoritma pengurutan yang berbasis rekursif dan iteratif dalam bahasa pemrograman Python. Eksperimen diimplementasikan tentang dengan mengamati waktu komputasi dan penggunaan memori beberapa algoritma pengurutan yang diimplementasikan pada dua cara tersebut. Python dipilih sebagai bahasa kode karena ada fleksibilitas tinggi pertimbangan pustaka Python Waris yang hampir selengkap mungkin, dan bahasa itu sendiri, memungkinkan untuk memantau penggunaan sumber daya sistem secara online. Algoritma pengurutan yang digunakan adalah *Merge Sort* vs *Quick Sort*. Oleh karena itu, setiap algoritma diimplementasikan dalam dua versi, yaitu rekursif dan iteratif, untuk dibandingkan berdasarkan efisiensi waktu dan memori.

Populasi dan Sampel

Populasi dalam penelitian ini adalah seluruh algoritma pengurutan yang umum digunakan dalam analisis data dan pemrograman. Dari populasi tersebut, penelitian ini menggunakan metode *non-probability* sampling jenis *purposive sampling*, yaitu memilih algoritma berdasarkan pertimbangan karakteristik dan tujuan penelitian. Sampel yang digunakan terdiri dari dua algoritma utama, yaitu *Merge Sort* dan *Quick Sort*, karena keduanya mewakili algoritma *divide and conquer* dengan performa berbeda dan sering dipakai dalam pengolahan data berskala besar. Pengujian dilakukan pada dua ukuran dataset, yaitu 10.000 elemen dan 1.000.000 elemen, untuk melihat pengaruh skala data terhadap efisiensi algoritma.

Instrumen dan Pengumpulan Data

Instrumen utama dalam penelitian ini adalah program Python yang dirancang untuk mengukur dua variabel efisiensi waktu dan efisiensi memori.Waktu eksekusi diukur menggunakan pustaka *time* dan *timeit* dari Python untuk menghitung durasi proses pengurutan dalam satuan detik. Penggunaan memori diukur dengan memanfaatkan pustaka *psutil*, yang berfungsi untuk memperoleh informasi mengenai konsumsi memori selama proses berlangsung Setiap algoritma dijalankan secara terpisah dengan kondisi lingkungan yang sama guna memastikan keakuratan pengukuran.

Prosedur Penelitian

Tahapan penelitian dimulai dengan implementasi kedua algoritma pengurutan dalam dua versi: rekursif dan iteratif. Setiap versi diuji pada dua ukuran dataset, yaitu 10.000 dan 1.000.000 elemen, dengan tiga kali pengulangan untuk setiap skenario. Selama proses pengujian, data waktu eksekusi dan penggunaan memori dicatat secara otomatis menggunakan fungsi pemantauan yang telah terintegrasi dalam skrip Python. Setelah semua data terkumpul, dilakukan proses validasi untuk memastikan

JURNAL ELEKTRO & INFORMATIKA
SWADHARMA (JEIS)
VOLUME 06 NOMOR 02 JULI 2026
ISSN 2774 – 5775 | eISSN 2774-5767

tidak terdapat anomali pada hasil pengukuran.

Analisis Data

Data hasil pengujian dianalisis menggunakan kombinasi metode statistik deskriptif dan inferensial. Analisis deskriptif dilakukan untuk menghitung nilai rata-rata, minimum, maksimum, dan deviasi standar dari waktu eksekusi serta penggunaan memori masing-masing algoritma. Sedangkan analisis inferensial dilakukan dengan uji t dua sampel independen untuk menentukan apakah terdapat perbedaan signifikan antara pendekatan rekursif dan iteratif. Seluruh proses analisis dilakukan menggunakan Python (pustaka pandas dan scipy) serta SPSS untuk validasi hasil statistik.

Keterbatasan Penelitian

1. Ukuran Data: Pengujian hanya dilakukan pada dua ukuran dataset sehingga hasilnya mungkin belum mewakili seluruh rentang ukuran data dalam aplikasi nyata.
2. Jenis Algoritma: Penelitian ini hanya mencakup dua algoritma pengurutan, yaitu *Merge Sort* dan *Quick Sort* yang masing-masing diimplementasikan daman dua versi: rekursif dan iteratif.
3. Lingkungan Eksekusi: Kinerja algoritma dapat dipengaruhi oleh spesifikasi perangkat keras dan sistem operasi yang digunakan selama eksperimen.

C. HASIL DAN PEMBAHASAN

Sistem Penelitian dilakukan menggunakan bahasa Python (versi 3.12) dengan bantuan *Integrated Development Environment* (IDE) *Visual Studio Code*. Seluruh eksperimen dijalankan pada satu perangkat dengan spesifikasi berikut:

1. Sistem Operasi: Windows 11 64-bit
2. Memori: 12 GB RAM
3. Prosesor: Intel ® Core ™ i5-7300U CPU @ 2.60GHZ 2.71 GHz
4. Media Penyimpanan: SSD 512 GB

Hasil pengujian menunjukkan bahwa perbedaan implementasi antara pendekatan rekursif dan iteratif menghasilkan variasi kinerja yang signifikan pada dua algoritma yang diuji. Secara umum, pendekatan iteratif cenderung lebih efisien dalam penggunaan memori, sedangkan pendekatan rekursif unggul pada algoritma yang memanfaatkan konsep *divide and conquer* seperti *Merge Sort dan Quick Sort* terkait waktu eksekusi. Hal ini mengindikasikan bahwa untuk algoritma *divide and conquer*, *overhead* pemanggilan fungsi pada rekursi lebih dapat ditoleransi dibandingkan manfaat kecepatan yang ditawarkan.

***Merge Sort* Rekursif**

Gambar 1. Koding *Merge Sort* Rekursif di Python

Merge Sort menggunakan pendekatan *Divide and Conquer* secara rekursif. Fungsi *merge_sort* () bertanggung jawab untuk membagi data. Ini memotong array menjadi dua bagian secara berulang hingga setiap subarray hanya memiliki satu elemen (kasus dasar rekursi). Setelah pembagian selesai, fungsi merge () mengambil alih tugas menggabungkan dan mengurutkan. Fungsi ini membandingkan elemen dari dua subarray yang sudah terurut *(left dan right)*, lalu menggabungkannya menjadi satu array yang terurut sempurna. Proses penggabungan ini terus berulang dari bawah ke atas hingga seluruh data terurut.

Tabel 1. Statistik Deskriptif *Merge Sort* Rekursif (Ukuran Data: 10.0000)

Nilai	Waktu (Detik)	Memori (MB)
Nilai 1	0.045	2.4
Nilai 2	0.046	2.5
Nilai 3	0.048	2.6
Rata-rata	0.046	2.5
Deviasi Standar	0.00125	0.0817

Tabel 2. Statistik Deskriptif *Merge Sort* Rekursif (Ukuran Data: 1.000.000)

Nilai	Waktu (Detik)	Memori (MB)
Nilai 1	0.5	9
Nilai 2	0.5	9.5
Nilai 3	0.6	10
Rata-rata	0.533	9.5
Deviasi Standar	0.047	0.41

Tabel 3. Statistik Deskriptif *Merge Sort* Iteratif (Ukuran Data: 10.000)

Nilai	Waktu (Detik)	Memori (MB)
Nilai 1	0.016	2.1
Nilai 2	0.017	2.1
Nilai 3	0.16	2.2
Rata-rata	0.0163	2.13
Deviasi Standar	0.0005	0.058

Tabel 4. Statistik Deskriptif *Merge Sort* Iteratif (Ukuran Data: 1.000.000)

Nilai	Waktu (Detik)	Memori (MB)
Nilai 1	0.05	8.5
Nilai 2	0.05	8.8
Nilai 3	0.05	9.0
Rata-rata	0.05	8.8
Deviasi Standar	0.0	0.0

Merge Sort Iteratif

Gambar 2. Koding *Merge Sort* Iteratif di Python

Merge Sort Iteratif adalah pengurutan *bottom-up* yang menghindari rekursi. Fungsi kuncinya, merge (), menggabungkan dua subarray terurut menjadi satu. Fungsi utama *merge_sort_iterative*() mengelola prosesnya dengan menggunakan *variabel width* yang terus digandakan (1, 2, 4, \dots). Ini secara berulang memanggil merge () untuk menggabungkan pasangan subarray yang berdekatan dengan lebar saat ini, secara progresif mengurutkan seluruh data.

Quick Short Rekursif

Gambar 3. Koding *Quick Sort* Rekursif di Python

Pendekatan rekursif pada *Quick Sort* menerapkan konsep *Divide and Conquer* dengan membagi data menjadi dua bagian yang lebih kecil untuk diurutkan secara terpisah. Setiap bagian diproses melalui pemanggilan fungsi rekursif *quick_sort*(), kemudian hasilnya digabung kembali hingga seluruh data terurut sempurna. Proses ini berlanjut hingga setiap subarray berukuran satu elemen.

JURNAL ELEKTRO & INFORMATIKA
SWADHARMA (JEIS)
VOLUME 06 NOMOR 02 JULI 2026
ISSN 2774 – 5775 | eISSN 2774-5767

Tabel 5. Statistik Deskriptif *Quick Sort* Rekursif (Ukuran Data: 10.000)

Nilai	Waktu (Detik)	Memori (MB)
Nilai 1	0.002	0.6
Nilai 2	0.002	0.6
Nilai 3	0.002	0.6
Rata-rata	0.002	0.6
Deviasi Standar	0.0	0.0

Tabel 6. Statistik Deskriptif *Quick Sort* Rekurtif (Ukuran Data: 1.000.000)

Nilai	Waktu (Detik)	Memori (MB)
Nilai 1	0.55	6.5
Nilai 2	0.55	6.5
Nilai 3	0.55	6.5
Rata-rata	0.55	6.5
Deviasi Standar	0.0	0.0

***Quick Short* Iteratif**

Gambar 4. Koding *Quick Sort* Iteratif di Python

Algoritma ini tetap menerapkan konsep *divide and conquer* tanpa pemanggilan fungsi berulang, sehingga lebih efisien dalam penggunaan memori dan terhindar dari risiko *stack overflow*. Fungsi *partition* () memisahkan elemen berdasarkan *pivot*, lalu subarray kiri dan kanan diproses hingga seluruh data terurut. Pendekatan ini cocok untuk pengurutan data besar dengan keterbatasan memori.

Tabel 7. Statistik Deskriptif *Quick Sort* Iteratif (Ukuran Data: 10.000)

Nilai	Waktu (Detik)	Memori (MB)
Nilai 1	0.004	0.5
Nilai 2	0.004	0.5
Nilai 3	0.004	0.5
Rata-rata	0.004	0.5
Deviasi Standar	0.0	0.0

Tabel 8. Statistik Deskriptif *Quick Sort* Iteratif (Ukuran Data: 1.000.000)

Nilai	Waktu (Detik)	Memori (MB)
Nilai 1	0.045	6.15
Nilai 2	0.045	6.15
Nilai 3	0.045	6.15
Rata-rata	0.045	6.15
Deviasi Standar	0.0	0.0

Berdasarkan hasil pengujian memberikan gambaran yang jelas mengenai peforma dari dua algoritma pengurutan yang populer, yaitu *Merge Sort* dan *Quick Sort*, dalam dua pendekatan implementasi yang berbeda, yakni rekursif dan iteratif.

Dari sisi waktu eksekusi, algoritma *Quick Sort* versi rekursif menunjukkan performa terbaik dengan waktu eksekusi tercepat, yakni sekitar 0,002 detik pada pengujian dengan 10.000 elemen Implementasi *Quick Sort* rekursif menunjukkan waktu eksekusi yang cepat pada data acak berukuran menengah, namun penggunaan memori meningkat akibat tumpukan rekursi yang dalam (Ali et al., 2025). Sebaliknya, *Merge Sort* rekursif memiliki stabilitas tinggi dalam berbagai kondisi data, tetapi memerlukan ruang tambahan untuk proses penggabungan (*merge*) (Taiwo et al., 2020). Versi iteratif dari kedua algoritma umumnya memperlihatkan efisiensi memori yang lebih baik karena menghindari *call stack* rekursif, terutama pada pengolahan data berskala besar (Zheng, 2025).

Dalam uji eksperimental Python yang dilakukan pada ukuran data dari 1.000 hingga 1.000.000 elemen, *Quick Sort* iteratif

memberikan waktu eksekusi rata-rata yang lebih cepat dibandingkan *Merge Sort* iteratif, namun perbedaan efisiensinya semakin kecil pada ukuran data besar (Ali et al., 2025). Penggunaan modul *tracemalloc* menunjukkan bahwa *Merge Sort* rekursif menggunakan memori hingga 25–30% lebih banyak dibanding versi iteratif, sementara *Quick Sort* iteratif hanya membutuhkan tambahan sekitar 5–10% memori dari batas minimum (Fitro, 2023). Hal ini sejalan dengan temuan (Lin et al., 2021) yang menyatakan bahwa implementasi iteratif dapat menekan konsumsi memori tanpa penurunan signifikan pada waktu proses. Beberapa studi juga menegaskan bahwa bahasa Python memiliki *overhead* memori yang lebih besar dibanding bahasa tingkat rendah seperti C++, namun tetap mencerminkan tren yang sama dalam perbandingan algoritma rekursif dan iteratif (Ali et al., 2025).

Secara umum, hasil ini menunjukkan bahwa pemilihan pendekatan rekursif atau iteratif tidak hanya mempertimbangkan kecepatan, tetapi juga efisiensi memori, terutama untuk aplikasi yang berjalan pada sistem dengan sumber daya terbatas.

D. PENUTUP

Berdasarkan hasil implementasi dan pengujian Berdasarkan hasil implementasi dan pengujian menggunakan Python, dapat disimpulkan bahwa efisiensi waktu dan penggunaan memori pada algoritma pengurutan sangat dipengaruhi oleh pendekatan yang digunakan, yaitu rekursif atau iteratif, serta jenis algoritmanya. *Quick Sort* rekursif mampu memberikan waktu eksekusi yang cepat pada data acak berukuran menengah, tetapi membutuhkan memori lebih besar karena banyaknya pemanggilan fungsi rekursif. Sebaliknya, *Merge Sort* rekursif lebih stabil untuk berbagai jenis data, namun memerlukan ruang tambahan untuk proses penggabungan.

Versi iteratif dari kedua algoritma cenderung lebih efisien dalam penggunaan memori karena tidak menggunakan *call stack* rekursif, sehingga lebih sesuai untuk pemrosesan data dalam skala besar. Hasil pengujian terhadap ukuran data antara 1.000 hingga 1.000.000 elemen menunjukkan bahwa *Quick Sort* iteratif memiliki waktu eksekusi rata-rata yang lebih cepat dibandingkan *Merge Sort* iteratif, meskipun selisihnya semakin kecil pada ukuran data yang sangat besar.

Secara umum, pendekatan iteratif terbukti lebih hemat memori, sedangkan *Quick Sort* tetap menjadi pilihan terbaik dari sisi kecepatan rata-rata. Dengan demikian, pemilihan algoritma harus mempertimbangkan kebutuhan sistem, *Quick Sort* iteratif lebih tepat untuk kondisi dengan batasan memori dan tuntutan kecepatan, sedangkan *Merge Sort* rekursif cocok untuk kebutuhan stabilitas hasil dan performa yang konsisten.

E. DAFTAR PUSTAKA

Ala'Anzy, M. A., Mazhit, Z., Ala'Anzy, A. F., Algarni, A., Akhmedov, R., & Bauyrzhan, A. (2024). Comparative Analysis of Sorting Algorithms: A Review. *11th International Conference on Soft Computing & Machine Intelligence (ISCMI)*, 88–100. https://doi.org/10.1109/ISCMI63661.2024.10851593

Ali, M. I., Fardiarsyah, R. D., Shodik, L., Kinanti, F. Z. D., & Pujiono, I. P. (2025). Analisis Komparatif Efisiensi Memori dan Waktu Komputasi pada 8 Algoritma Sorting menggunakan C++. *LogicLink: Journal of Artificial Intelligence and Multimedia in Informatics*, *2*(1), 1–17. https://doi.org/10.28918/logiclink.v2i1.10868

Febriansyah, M. F., Gunawan, Rhamadani, M., & Sutabri, T. (2025). Perbandingan Pemanfaatan Algoritma Rekursif dan

Iteratif dalam Penyelesaian Struktur Data Pohon. *MIFORTEKH : Jurnal Manajemen Informatika & Teknologi*, *5*(1), 46–56. https://doi.org/10.51903/2mxmmg85

Fitro, A. (2023). Comparison of Efficiency Data Sorting Algorithms Based on Execution Time. *IJSRCSEIT : International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, *9*(2), 15–21. https://doi.org/10.32628/CSEIT2390151

Ilham, M. N., Setiawan, A. F., Kholifatun, I., Aldiansyah, M. H., & Pujiono, I. P. (2025). Comparative Analysis of Memory Performance and Processing Time of Five Sorting Algorithms Using C++ Programming Language. *Journal of Artificial Intelligence and Engineering Applications (JAIEA)*, *4*(3), 1950–1956. https://doi.org/10.59934/jaiea.v4i3.1051

Lin, L., Xu, Z., Huan, H., Jian, Z., & Li-Xin, L. (2021). An Empirical Comparison of Implementation Efficiency of Iterative and Recursive Algorithms of Fast Fourier Transform. *Machine Learning and Intelligent Communications*, *342*, 73–81. https://doi.org/10.1007/978-3-030-66785-6_9

Mishal, I., Al-Khatib, R., & Hiasa, R. (2021). Comparative Study of Two Divide and Conquer Sorting Algorithms: Modified Quick Sort and Merge Sort. *International Journal of Computer Applications*, *183*, 28–33. https://doi.org/10.5120/ijca2021921702

Musyaffa, M. Z., Raharjo, K., Faiz, M., Ammarulloh, S., & Pujiono, I. P. (2025). Efisiensi Memori dan Waktu: Array Sorting Algorithm vs Algoritma Pengurutan Tradisional Menggunakan Python. *JEIS: Jurnal Elektro Dan Informatika Swadharma*, *5*(2), 72–81. https://doi.org/10.56486/jeis.vol5no2.78

Pratama, M. B., Setiawan, R., & Sutabri, T. (2025). Integrasi Algoritma Rekursif pada Pemrosesan Data Multilevel untuk Aplikasi Berbasis AI. *MIFORTEKH : Jurnal Manajemen Informatika & Teknologi*, *5*(1), 57–66. https://doi.org/10.51903/g04chv51

Syahputra, C., Nafiisah, S. S., Gultom, S. R., Affandi, R., & Perdana, A. (2026). Analisis Performa Algortima Quick Sort dan Merge Sort Pada Pengurutan Data Besar (Big Data) Menggunakan Notasi Big-O. *Jurnal Informatika Dan Multimedia*, *6*(1), 367–379. https://doi.org/10.51903/informatika.v6i1.1661

Taiwo, O. E., Christianah, A. O., Oluwatobi, A. N., Aderonke, K. A., & Kehinde, A. J. (2020). Comparative Study of Two Divide and Conquer Sorting Algorithms: Quicksort and Mergesort. *Procedia Computer Science*, *171*, 2532–2540. https://doi.org/10.1016/j.procs.2020.04.274

Zahwa, S., Amelia, N. D., Nafila, R., Putri, R. A., & Pujiono, I. P. (2025). Perbandingan Efisiensi Memori dan Waktu Komputasi pada Algoritma Rekursif dan Iteratif dalam Operasi Pengurutan di C++. *RESTIKOM : Riset Teknik Informatika Dan Komputer*, *7*(1), 123–136. https://doi.org/10.52005/restikom.v7i1.428

Zheng, Y. (2025). Optimization Implementation and Performance Analysis of Divide-and-Conquer Algorithm Based on Python in Big Data Sorting and Retrieval. *Applied and Computational Engineering*, *178*, 40–46. https://doi.org/10.54254/2755-2721/2025.PO25411

ANALISIS EFEKTIVITAS TEKNIK *INDEXING* TERHADAP WAKTU EKSEKUSI *QUERY* SQL PADA BASIS DATA TRANSAKSI *E-COMMERCE*

M.Aldy Zulfa Saputra[1)], Muchammad Soufwan Fauzi[2)], Rangga Dzikri Fardiansyah[3)], Zaki Kafila Pamungkas[4)], Dicky Anggriawan Nugroho[5)], Imam Prayogo Pujiono[6)]

Prodi Informatika, Fakultas Ekonomi dan Bisnis Islam, UIN K.H Abdurrahman Wahid Pekalongan

Correspondence author: MAZ Saputra, m.aldy.zulfa.saputra24055@mhs.uingusdur.ac.id, Pekalongan, Indonesia

Abstract

This study examines the effect of indexing techniques on SQL query execution speed in a MySQL-based e-commerce transaction database. Performance issues often arise as data volume increases because the system must scan the entire table, resulting in slow response times. To analyze this problem, this study employed a pure experimental method by testing three index conditions: no index, a single index, and a composite index. Testing was conducted using a synthetic dataset of 20,000 rows depicting e-commerce transaction activity. Each condition was evaluated using a series of SELECT queries. Measurements were made through execution time and execution plans displayed by the EXPLAIN command. The results showed that using indexes provided a significant performance improvement. A single index successfully reduced execution time from 0.2380 seconds to 0.0740 seconds, while a composite index resulted in an execution time of 0.0900 seconds. Both types of indexes reduced the number of rows scanned across the entire table to only tens of rows. A single index is more suitable for simple queries with high selectivity, while a composite index is more effective for queries with more than one condition. However, implementing an index requires additional storage space and causes the data write process to run slightly slower. Overall, this study confirms that an indexing strategy tailored to query patterns and data characteristics can significantly improve the performance of e-commerce transaction databases.

Keyword *: effectiveness analysis, indexing, SQL queries, databases, e-commerce*

Abstrak

Penelitian ini mengkaji pengaruh teknik *indexing* terhadap kecepatan eksekusi *query* SQL pada basis data transaksi *e-commerce* berbasis MySQL. Gangguan kinerja sering muncul ketika jumlah data meningkat karena sistem harus melakukan pemindaian seluruh tabel yang menyebabkan waktu respon menjadi lambat. Untuk menganalisis permasalahan tersebut, penelitian ini menerapkan metode eksperimen murni dengan menguji tiga kondisi indeks yaitu tanpa indeks, indeks tunggal, dan indeks komposit. Pengujian dilakukan menggunakan *dataset* sintetis berjumlah 20.000 baris yang menggambarkan aktivitas transaksi *e-commerce*. Setiap kondisi dievaluasi menggunakan serangkaian *query* SELECT.

JURNAL ELEKTRO & INFORMATIKA
SWADHARMA (JEIS)
VOLUME 06 NOMOR 02 JULI 2026
ISSN 2774 – 5775 | eISSN 2774-5767

Pengukuran dilakukan melalui waktu eksekusi dan rencana eksekusi yang ditampilkan oleh perintah EXPLAIN. Hasil penelitian menunjukkan bahwa penggunaan indeks memberikan peningkatan performa yang sangat signifikan. Indeks tunggal berhasil menurunkan waktu eksekusi dari 0,2380 detik menjadi 0,0740 detik, sedangkan indeks komposit menghasilkan waktu eksekusi 0,0900 detik. Kedua jenis indeks ini mengurangi jumlah baris yang dipindai dari seluruh tabel menjadi hanya puluhan baris. Indeks tunggal lebih sesuai untuk *query* sederhana dengan *selectivity* yang tinggi, sedangkan indeks komposit lebih efektif untuk *query* yang memiliki lebih dari satu kondisi. Meskipun demikian, penerapan indeks memerlukan ruang penyimpanan tambahan dan menyebabkan proses penulisan data berjalan sedikit lebih lambat. Secara keseluruhan, penelitian ini menegaskan bahwa strategi *indexing* yang disesuaikan dengan pola *query* dan karakteristik data mampu meningkatkan performa basis data transaksi *e-commerce* secara signifikan.

Kata Kunci : analisa efektivitas, *indexing*, *query* sql, basis data, *e-commerce*

A. PENDAHULUAN

Perkembangan teknologi informasi telah menyebabkan hampir seluruh aktivitas manusia terekam dalam bentuk data, mulai dari interaksi pada media sosial, layanan perbankan, hingga berbagai transaksi digital. Kebutuhan akan pengelolaan data dalam jumlah besar tersebut menuntut tersedianya sistem yang mampu menyimpan, mengelola, dan mengolah informasi dalam volume besar secara cepat, tepat, dan andal (Ibna, 2024). Tanpa pengelolaan basis data yang efisien, proses penyajian informasi, pengambilan keputusan, serta layanan berbasis aplikasi berpotensi menjadi lambat dan tidak responsif. Kondisi ini semakin krusial pada aplikasi yang bergantung pada pemrosesan data secara waktu nyata *(real-time)*. Salah satu di antaranya ialah aplikasi *e-commerce* yang tidak hanya berfungsi sebagai sarana fasilitasi transaksi jual beli secara daring, tetapi juga sebagai sistem informasi yang menganalisis data konsumen, riwayat transaksi, dan perilaku pengguna sebagai dasar perumusan kebijakan dan pengambilan keputusan bisnis (Rahayu et al., 2025). Dengan demikian, kebutuhan akan sistem basis data yang mampu menangani volume data besar serta mengelola data secara efisien menjadi fondasi penting dalam pengembangan dan pengelolaan aplikasi *e-commerce* modern.

Dalam praktiknya, salah satu teknologi yang banyak digunakan untuk mengelola data transaksi pada aplikasi *e-commerce* adalah *Relational Database Management System* (RDBMS). RDBMS menyimpan data dalam bentuk tabel-tabel berelasi dan dikelola menggunakan *Structured Query Language* (SQL) sebagai bahasa standar untuk mendefinisikan dan mengakses data (Adisa & Nasution, 2023). Contoh RDBMS yang umum digunakan antara lain *MySQL*, *PostgreSQL*, dan *Microsoft SQL Server* (Ahsana et al., 2023). Di antara berbagai pilihan tersebut, *MySQL* menjadi salah satu RDBMS yang populer karena bersifat sumber terbuka (*open source*), menggunakan SQL sebagai bahasa utama, serta banyak diimplementasikan pada sistem informasi dan aplikasi web, termasuk sistem penjualan dan *e-commerce* (Nurhayati & Nasution, 2023). Namun, seiring meningkatnya volume data dan kompleksitas kebutuhan informasi, kinerja eksekusi *query* pada MySQL dapat menurun apabila tidak disertai perancangan struktur basis data dan penerapan teknik optimasi yang memadai, seperti pemanfaatan

indexing untuk menjaga efisiensi waktu respon *query* (Thoib et al., 2024).

Dalam konteks tersebut, platform *e-commerce* menjadi salah satu contoh nyata aplikasi yang sangat bergantung pada kinerja sistem basis data. Peningkatan volume transaksi daring, personalisasi layanan, integrasi dengan sistem pembayaran digital, serta pemanfaatan analitik data berskala besar menghasilkan lonjakan data transaksi yang harus diproses secara cepat dan konsisten. Kondisi ini menuntut sistem basis data mampu mengeksekusi *query* dengan waktu tanggap yang rendah, terutama pada skenario transaksi yang berlangsung secara waktu nyata *(real-time)* (Anchlia, 2024). Keterlambatan beberapa detik saja dalam pemrosesan *query* dapat berdampak pada terhambatnya proses bisnis, penurunan kualitas layanan, hingga hilangnya peluang transaksi pada platform *e-commerce* (Poggi et al., 2014).

Menanggapi permasalahan tersebut, salah satu pendekatan yang banyak digunakan adalah penerapan teknik *indexing*. Secara konsep, indeks merupakan struktur data yang dibangun di atas satu atau beberapa kolom tabel untuk mempercepat proses pencarian dan penyaringan data, sehingga sistem tidak perlu melakukan memindai semua tabel (*full table scan)* pada setiap eksekusi *query* (Anchlia, 2024). Dengan adanya indeks, basis data dapat langsung mengarah ke lokasi data yang relevan sehingga jumlah blok data yang harus diakses menjadi lebih sedikit dan waktu respon *query* dapat ditekan secara signifikan. Berbagai penelitian terdahulu menunjukkan bahwa *indexing* merupakan salah satu teknik yang paling efektif dalam meningkatkan kinerja *query* pada sistem basis data relasional. (Anchlia, 2024) menegaskan bahwa pemanfaatan indeks yang tepat mampu menurunkan waktu pencarian data dan mengoptimalkan rencana eksekusi *query*, sedangkan (Panggabean & Azizah, 2025) menunjukkan bahwa penambahan indeks pada kolom-kolom transaksi tertentu dalam aplikasi *e-commerce* dapat mempercepat eksekusi *query* dan meningkatkan efisiensi pencarian data. Penelitian lain pada MySQL yang mengkaji optimasi *query* berbasis indeks juga melaporkan bahwa pemilihan kolom dan jenis indeks yang tepat berkontribusi langsung terhadap perbaikan waktu respon pada berbagai skenario pengujian (Maesaroh et al., 2022; Permana et al., 2023). Meskipun demikian, sebagian besar studi tersebut masih berfokus pada pengujian umum atau konteks basis data tertentu dan belum secara spesifik menganalisis perbandingan efektivitas desain indeks tanpa indeks, *single index*, dan *composite index* terhadap waktu eksekusi *query* SQL pada basis data transaksi *e-commerce* dengan volume data puluhan ribu baris menggunakan MySQL. Kesenjangan inilah yang menjadi dasar perlunya penelitian ini untuk menghadirkan kajian empiris yang lebih terarah pada skenario transaksi *e-commerce* yang mendekati kondisi operasional nyata.

Berdasarkan kesenjangan penelitian tersebut, fokus kajian dalam penelitian ini diarahkan pada analisis efektivitas penerapan teknik *indexing* terhadap kinerja eksekusi *query* pada basis data transaksi *e-commerce* berbasis MySQL. Pertanyaan penelitian yang diajukan adalah: (1) bagaimana pengaruh penerapan teknik *indexing* terhadap waktu eksekusi *query* SQL pada basis data transaksi *e-commerce*; (2) seberapa besar perbedaan performa *query* antara kondisi tanpa indeks, *single index*, dan *composite index*; dan (3) jenis indeks apa yang paling efektif digunakan untuk meningkatkan efisiensi waktu eksekusi *query* pada sistem *e-commerce* berskala besar. Sejalan dengan rumusan masalah tersebut, penelitian ini bertujuan untuk menganalisis dan membandingkan kinerja waktu eksekusi *query* pada ketiga skenario desain indeks tersebut menggunakan basis data transaksi *e-commerce* berukuran puluhan ribu baris di MySQL, serta merumuskan rekomendasi strategi *indexing* yang dapat dimanfaatkan sebagai acuan bagi pengembang dan

pengelola basis data dalam mengoptimalkan performa sistem *e-commerce*.

B. METODE PENELITIAN

Penelitian ini menggunakan pendekatan kuantitatif eksperimental dengan jenis eksperimen murni (*true experiment*) untuk mengukur secara empiris pengaruh penerapan teknik *indexing* terhadap waktu eksekusi *query* SQL pada basis data transaksi *e-commerce*. Variabel independen dalam penelitian ini adalah skenario penerapan indeks, yang terdiri atas tiga kondisi, yaitu tanpa indeks, *single index*, dan *composite index*, sedangkan variabel dependen adalah waktu eksekusi *query*. Rancangan penelitian mengikuti model *pre-test–post-test control design*, di mana kinerja *query* diukur terlebih dahulu pada kondisi tanpa indeks, kemudian dibandingkan dengan kondisi setelah penerapan *single index* dan *composite index* sebagaimana digunakan pada penelitian eksperimen basis data oleh (Anchlia, 2024) dan (Panggabean & Azizah, 2025).

Penelitian dilaksanakan menggunakan satu perangkat laptop dengan sistem operasi Windows 11, prosesor Intel Core i7 Gen 12, RAM 16 GB, dan SSD 512 GB. Sistem manajemen basis data yang digunakan adalah MySQL Server 8.0.36 yang diakses melalui phpMyAdmin 5.2.1 dengan dukungan XAMPP sebagai server lokal. Populasi penelitian adalah data transaksi *e-commerce* yang memuat informasi pelanggan, produk, dan aktivitas pembelian. Dari populasi tersebut diambil sampel berupa *synthetic dataset* berukuran 20.000 baris yang diimpor ke dalam tabel *sales_transactions* dengan kolom utama *Customer_Id*, *Product_Category*, *Order_Date*, dan *Order_Status*. Pemilihan ukuran sampel ini dimaksudkan untuk merepresentasikan pola transaksi *e-commerce* pada volume data menengah yang relevan untuk pengujian kinerja basis data.

Teknik sampling yang digunakan adalah *purposive sampling* dengan memilih kolom dan pola *query* yang relevan dengan aktivitas pencarian data pada sistem *e-commerce*, seperti pencarian transaksi berdasarkan pelanggan, kategori produk, dan rentang waktu pemesanan. Penelitian diawali dengan pembuatan skema basis data dan pengisian tabel *sales_transactions* menggunakan *dataset* tersebut. Selanjutnya disusun serangkaian *query* SELECT yang mewakili pola akses data umum. Pada tahap pertama, seluruh *query* dieksekusi pada kondisi tanpa indeks untuk memperoleh kinerja dasar. Tahap berikutnya adalah penerapan *single index* pada kolom *Customer_Id*, diikuti pengujian ulang *query* yang sama. Tahap terakhir adalah penerapan *composite index* pada kombinasi kolom *Customer_Id* dan *Order_Date*, kemudian dilakukan pengujian ulang. Pada setiap skenario, perintah EXPLAIN digunakan untuk menganalisis rencana eksekusi *query* dan pola akses data.

Data yang diperoleh berupa waktu eksekusi *query* dan informasi rencana eksekusi dianalisis secara kuantitatif deskriptif. Analisis dilakukan dengan menghitung rata-rata, nilai minimum, maksimum, dan simpangan baku waktu eksekusi untuk setiap jenis *query* pada masing-masing skenario indeks, serta membandingkan kinerja antar skenario melalui persentase peningkatan atau penurunan waktu eksekusi. Hasil analisis kemudian disajikan dalam bentuk tabel, grafik, dan uraian naratif yang menjelaskan efektivitas masing-masing skenario indeks terhadap efisiensi waktu eksekusi *query* pada basis data transaksi *e-commerce*. Keabsahan hasil dijaga melalui replikasi pengujian pada setiap skenario dan pengendalian lingkungan eksekusi, dengan memastikan tidak ada proses lain yang membebani server selama eksperimen berlangsung sehingga perubahan kinerja terutama disebabkan oleh perbedaan konfigurasi indeks, sejalan dengan prinsip validitas internal pada studi optimasi basis data (Taipalus, 2024).

C. HASIL DAN PEMBAHASAN

Desain Indeks

Desain indeks dalam penelitian ini dilakukan melalui pendekatan sistematis dengan menentukan kolom yang akan diindeks berdasarkan analisis pola *query* pada *dataset* transaksi *e-commerce* yang berjumlah 20.000 baris. Penggunaan data dengan volume menengah ini dirancang untuk merepresentasikan kondisi operasional sistem *e-commerce* yang umum digunakan pada skala UMKM hingga bisnis daring berkembang. Studi terkait oleh (Panggabean & Azizah, 2025) juga menerapkan *dataset* transaksi berukuran serupa dalam penelitian performa *query* sehingga ukuran data ini dinilai representatif.

Secara teoretis, konsep *selectivity* menjadi dasar penting dalam pemilihan kolom yang akan diberikan indeks. *Selectivity* merujuk pada tingkat keunikan nilai dalam suatu kolom; kolom dengan *selectivity* tinggi (mendekati 1) berisi banyak nilai unik, sedangkan kolom dengan *selectivity* rendah (mendekati 0) didominasi oleh nilai yang berulang. Berdasarkan teori yang dikemukakan oleh (Sakshi & Sharma, 2025), kolom dengan *selectivity* tinggi lebih optimal untuk diindeks karena dapat mempersempit ruang pencarian secara signifikan dan mengurangi jumlah baris yang harus dipindai selama eksekusi *query*.

Hasil analisis *selectivity* pada *dataset* menunjukkan bahwa kolom *Customer_Id* memiliki *selectivity* sebesar 85% (17.000 nilai unik dari 20.000 baris), kolom *Order_Date* memiliki *selectivity* sebesar 92% dengan distribusi temporal yang relatif merata, sedangkan kolom *Product_Category* memiliki *selectivity* sekitar 45% dengan sembilan kategori produk yang berbeda. Dengan mempertimbangkan nilai-nilai tersebut, ketiga kolom tersebut dipilih sebagai kandidat utama untuk diberi indeks karena berpotensi memberikan pengurangan jumlah baris yang dipindai secara signifikan pada saat eksekusi *query*.

Jenis indeks yang digunakan dalam penelitian ini adalah indeks sekunder (*non-clustered index*) pada mesin penyimpanan InnoDB di MySQL, di mana struktur indeks disimpan terpisah dari data fisik tabel dan setiap entri indeks merujuk pada kunci utama (*primary key*). Konsep pemisahan struktur indeks dari penyimpanan data utama ini sejalan dengan teori *physical database tuning* yang dikemukakan oleh (Šušter & Ranisavljević, 2023), yaitu pemanfaatan indeks sebagai struktur tambahan untuk mengoptimalkan performa eksekusi *query* tanpa harus mengubah struktur logis tabel secara fundamental.

Metode Pembuatan Indeks

Metode pembuatan indeks dalam penelitian ini menggunakan pendekatan eksperimental yang terstruktur. Secara teoretis, pembuatan indeks dilakukan dengan memanfaatkan pernyataan CREATE *INDEX* dalam standar SQL, yang digunakan untuk membentuk struktur indeks terpisah tanpa mengubah definisi maupun constraint pada tabel. Secara umum, sintaks dasar yang digunakan adalah sebagai berikut:

```
CREATE INDEX index_name ON
table_name(column_name);
```

Implementasi dilakukan dengan *sequential experimental design* yang terdiri dari beberapa fase. Pertama, *Baseline Phase*, yaitu fase pengukuran performa tanpa *index*. Kedua, *Single Index Phase*, yaitu fase implementasi *index* pada kolom individual. Ketiga, *Composite Index Phase*, yaitu fase implementasi *index* pada beberapa kolom secara bersamaan (*multiple columns*).

Seluruh proses pemantauan performa dilakukan menggunakan perintah EXPLAIN pada MySQL. EXPLAIN berfungsi sebagai alat diagnostik untuk mengungkap *execution plan* dari suatu *query*. Beberapa field penting yang diperhatikan meliputi type (jenis join seperti ALL, ref, *range*), key (*index* yang digunakan), rows (perkiraan jumlah baris

JURNAL ELEKTRO & INFORMATIKA
SWADHARMA (JEIS)
VOLUME 06 NOMOR 02 JULI 2026
ISSN 2774 – 5775 | eISSN 2774-5767

yang di-*scan*), serta Extra (informasi tambahan terkait eksekusi *query*)

Hasil Eksperimen Implementasi *Indexing* Terhadap Perfoma Tanpa *Index*

Pada tahap awal, *query* dieksekusi tanpa menggunakan *index* dan menunjukkan karakteristik *full table scan*. Menurut teori (Pamungkas, 2018), *full table scan* terjadi ketika *database* harus memindai setiap baris dalam tabel secara *sequential* dari baris pertama hingga terakhir, dengan kompleksitas waktu O(n).

Secara rinci, hasil pengujian menunjukkan bahwa waktu eksekusi *query* tanpa *index* adalah 0,2380 detik, dengan *execution plan* bertipe ALL yang menjadi indikator jelas adanya *full table scan*. Jumlah baris yang di-*scan* adalah 20.000 baris atau 100% dari data yang ada. Kondisi ini mengakibatkan konsumsi sumber daya yang tinggi, terutama pada operasi I/O dan penggunaan memori.

Secara teoritis, hasil ini mengonfirmasi teori (Anchlia, 2024) bahwa tanpa *index*, pencarian data berjalan dalam orde linear O(n), di mana waktu eksekusi akan meningkat secara linear seiring dengan pertumbuhan jumlah data.

Hasil Eksperimen Implementasi *Indexing* Terhadap Perfoma *Single Index*

Implementasi *single index* pada kolom *Customer_Id* menunjukkan adanya transformasi performa yang signifikan. *Index* yang digunakan berbasis struktur *B-Tree*, yang memungkinkan proses pencarian berjalan dengan kompleksitas O(log n). Struktur hierarkis pada *B-Tree* ini secara signifikan mengurangi jumlah operasi perbandingan yang diperlukan dalam eksekusi *query* (Pamungkas, 2018).

Hasil pengujian menunjukkan bahwa waktu eksekusi *query* dengan *single index* adalah 0,0740 detik. *Execution plan* berubah menjadi type: ref, yang menunjukkan bahwa eksekusi *query* telah menggunakan *index lookup* berbasis *equality*. Jumlah baris yang di-*scan* turun drastis menjadi 35 baris, atau hanya sekitar 0,175% dari total data.

Persentase peningkatan performa dihitung menggunakan rumus:

$$\text{Perfoma} = \frac{Waktu\ sebelum\ -\ Waktu\ sesudah}{Waktu\ sebelum} \times 100\%$$

$$\text{Perfoma} = \frac{0{,}2380\ -\ 0{,}0740}{0{,}2380} \times 100\% = 68{,}91$$

Hasil ini mendukung teori (Anchlia, 2024) mengenai perubahan efisiensi dari O(n) menjadi O(log n), dengan reduksi jumlah baris yang di-*scan* mencapai 97,5%.

Analisis Perbandingan Efektivitas Teknik *Indexing* pada *Dataset E-commerce*

Hasil Pengujian Rata-Rata Waktu Menjalankan *Query* dapat dilihat pada tabel 1.

Tabel 1. Hasil Pengujian Rata-Rata Waktu Menjalankan *Query*

Parameter	Tanpa *index*	*Single index*	*Composite index*
Waktu eksekusi (detik)	0.2380	0.0740	0.0900
Execution plan	Type: all (pemindaian penuh tabel)	Type: ref (pencarian *index*)	Type: *range* (pemindaian rentang)
Rows *scan*ned	20.000 baris (100% data)	35 baris (0.175% data)	28 baris (0.14% data)
Kompleksitas algoritma	O(n) (linear)	O(log n) (logaritmik)	O(log n) (logaritmik)
Memory usage	~15 mb	~26 kb	~21 kb
I/o operations	1.250 pembacaan halaman	35 pembacaan halaman	28 pembacaan halaman
Storage overhead	0 mb	2.1 mb	2.9 mb
Write performance	Optimal	15-20% slower	15-20% slower

Parameter	Tanpa *index*	*Single index*	*Composite index*
Optimal use case	-	*Query* sederhana where *Customer_Id* = x	*Query* kompleks where *Customer_Id* = x and *Order_Date* > y
Selectivity	-	85% (*Customer_Id*)	92% (*Customer_Id* +

Evaluasi Performa Berdasarkan Jenis *Query*

Tabel 2. Evaluasi Performa Berdasarkan Jenis *Query*

Jenis *Query*	Tanpa *Index*	*Single Index*	*Composite Index*	Rekomendasi
WHERE *Customer_Id* = 'X'	0,2380 detik 20.000 baris	0,0740 detik 35 baris	0,0900 detik 28 baris	*Single Index*
WHERE *Order_Date* > 'Y'	0,2380 detik 20.000 baris	0,0760 detik 40 baris	0.0880 detik 32 baris	*Single/Composite*
WHERE *Customer_Id* = 'X' AND *Order_Date* > 'Y'	0.2380 detik 20.000 baris	0.0820 detik 38 baris	0.0900 detik 28 baris	*Composite Index*
WHERE *Product_Category* = 'Z'	0.2380 s detik 20.000 baris	0.1800 detik 9.000 rows	-	Evaluasi selektivitas
Aggregation Queries	0.4500 detik* 20.000 baris	0.1200 detik* 20.000 baris	0.1100 detik* 20.000 baris	Peningkatan terbatas

Analisis Keuntungan dan Keterbatasan

Keuntungan Menggunakan *Index*

Berdasarkan hasil eksperimen dan teori pendukung, penggunaan *index* memberikan beberapa keuntungan yang dapat dikuantifikasi dengan jelas. Dari sisi performa, terdapat peningkatan kecepatan eksekusi *query* hingga 68,91% untuk operasi SELECT. Jumlah baris yang di-*scan* juga menurun drastis dari 20.000 baris menjadi 35 baris, yang setara dengan reduksi sebesar 99,825%. Selain itu, *execution plan* berubah dari ALL (*full table scan*) menjadi ref atau *range*, yang menunjukkan pemanfaatan *index* secara optimal (Panggabean & Azizah, 2025)

Dari perspektif efisiensi sumber daya, penggunaan *index* juga memberikan penghematan yang signifikan. Penggunaan memori berkurang dari 15 MB menjadi 26 KB, dengan peningkatan efisiensi sebesar 99,83%. Dari sisi I/O, terjadi pengurangan dari 1250 *page reads* menjadi 35 *reads*, atau peningkatan efisiensi sekitar 97,2%. Utilisasi CPU juga menurun karena beban komputasi yang berkurang.Temuan-temuan ini konsisten dengan penelitian (Panggabean & Azizah, 2025) yang menyatakan bahwa penerapan *index* pada kolom-kolom penting dapat menurunkan waktu eksekusi *query* serta mengurangi konsumsi CPU dan memori secara signifikan.

Keterbatasan dan Pertimbangan

Meskipun memberikan banyak keuntungan, penggunaan *index* juga memiliki keterbatasan dan memerlukan sejumlah pertimbangan. Dari sisi storage, *composite index* membutuhkan ruang penyimpanan sekitar 40% lebih besar dibandingkan *single index*. Selain itu, terdapat *write overhead* pada operasi INSERT, UPDATE, dan DELETE, dimana setiap perubahan data memerlukan waktu tambahan sekitar 15–20% untuk melakukan *maintenance* terhadap *index*.

Secara teoritis, penggunaan *index* juga tidak terlepas dari batasan yang terkait dengan Teorema CAP, yang menjelaskan adanya *trade-off* antara *consistency*, *availability*, dan *partition tolerance* dalam sistem *database* terdistribusi. Selain itu, berlaku pula *Law of Diminishing Returns*, dimana penambahan *index* di luar titik

optimal hanya akan memberikan manfaat marjinal yang semakin menurun.

Dari sisi praktis, terdapat beberapa keterbatasan lain yang perlu diperhatikan. Misalnya, limitasi tampilan di phpMyAdmin yang hanya menampilkan 500 baris tidak mempengaruhi akurasi pengukuran waktu eksekusi, tetapi tetap menjadi faktor teknis dalam proses observasi. Selain itu, kolom dengan *selectivity* di bawah 20% dinilai kurang optimal untuk di-*index*. *Index* juga memerlukan *maintenance* berkala, seperti pembaruan statistik dan *reorganization* untuk menjaga performa tetap stabil (Šušter & Ranisavljević, 2023).

Analisis *Trade-off* Implementasi *Indexing*

Tabel 3. Analisis *Trade-off* Implementasi *Indexing*

Aspek	Keuntungan	Kerugian
Performa baca	+68.91% faster queries	-
Performa tulis	-	15-20% lebih lambat INSERT/UPDATE
Penyimpanan	-	+40% penyimpanan untuk *composite index*
Pemeliharaan	-	Diperlukan reorganisasi berkala
Kompleksitas	Rencana eksekusi teroptimas	Kompleksitas desain tambahan
Skalabilitas	Mendukung pertumbuhan adata	Hasil menurun pada data sangat besar

Rekomendasi Implementasi Berbasis *Evidence*

Berdasarkan sintesis hasil eksperimen dan teori pendukung, disusun sebuah kerangka strategis *indexing* yang dapat diterapkan secara praktis. Pendekatan ini diimplementasikan melalui *Priority Matrix* sebagai berikut:

Tier 1 berfokus pada high-impact *single index*es, misalnya:

```
CREATE INDEX idx_customer ON
sales_transactions(Customer_Id);
```

Tier 2 mengarah pada *strategic composite index*es untuk *query* yang melibatkan beberapa kondisi sekaligus, misalnya:

```
CREATE INDEX idx_customer_date ON
sales_transactions(Customer_Id,
Order_Date);
```

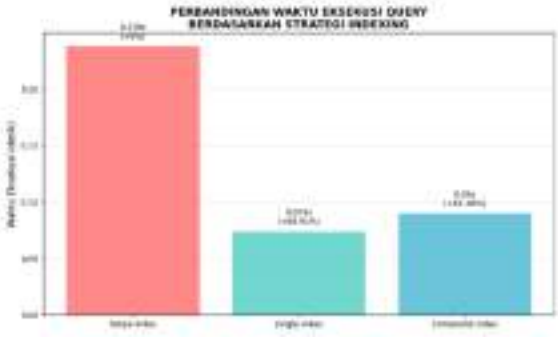
Tier 3 digunakan untuk *selective specialized index*es, misalnya:

```
CREATE INDEX idx_category_status ON
sales_transactions(Product_Category,
Order_Status);
```

Dari hasil dan teori yang ada, beberapa pedoman optimasi dapat disusun. Pertama, memprioritaskan *single index* pada kolom dengan *selectivity* tinggi (lebih dari 80%) untuk *query-query* sederhana. Kedua, menggunakan *composite index* untuk *query* kompleks dengan multiple conditions yang polanya relatif *predictable*. Ketiga, mempertimbangkan implementasi *partial indexing* untuk kolom dengan pola *query* yang sangat spesifik. Keempat, melakukan pemantauan *index* secara berkala melalui INFORMATION_SCHEMA.STATISTICS untuk melihat penggunaan *index* secara aktual (Šušter & Ranisavljević, 2023).

Secara operasional, disarankan untuk melakukan monitoring berkala terhadap pola *query* dan *index* usage statistics, mempertimbangkan *trade-off* antara peningkatan performa dan kebutuhan storage, serta menyusun jadwal *index maintenance* untuk pembaruan statistics dan proses reorganization agar performa *database* tetap terjaga secara berkelanjutan.

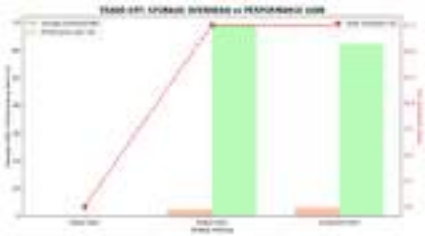
Grafik Perbandingan Waktu Eksekusi *Query*



Gambar 1. Grafik Perbandingan Waktu Eksekusi Berdasarkan Strategi *Indexing*

Berdasarkan Gambar 1, dapat diamati dengan jelas bahwa penerapan teknik *indexing* memberikan pengaruh yang signifikan terhadap waktu eksekusi *query*. Pada kondisi tanpa indeks, waktu eksekusi mencapai 0,2380 detik yang menunjukkan perlunya pemindaian penuh terhadap seluruh tabel (*full table scan*). Implementasi indeks tunggal berhasil mengurangi waktu eksekusi menjadi 0,0740 detik, yang merepresentasikan peningkatan efisiensi sebesar 68,91%. Sementara itu, indeks komposit menunjukkan waktu eksekusi 0,0900 detik dengan peningkatan 62,18% dari kondisi baseline. Perbedaan performa antara indeks tunggal dan indeks komposit mengindikasikan bahwa untuk kueri sederhana, indeks tunggal lebih optimal karena struktur yang lebih ringkas dan langsung.

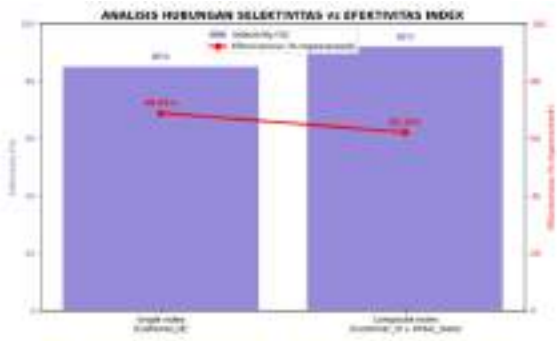
Analisis *Trade-off* Penyimpanan dan Performa



Gambar 2. Analisis *Trade-off* Penyimpanan dan Perfoma

Gambar 2 memberikan perspektif komprehensif mengenai *trade-off* yang terjadi dalam implementasi *indexing*. Di satu sisi, indeks tunggal memberikan peningkatan performa tertinggi (68,91%) dengan overhead penyimpanan sebesar 2,1 MB, sementara indeks komposit memberikan peningkatan 62,18% dengan kebutuhan penyimpanan 2,9 MB. Di sisi lain, kedua teknik *indexing* tersebut mengakibatkan penurunan performa tulis (write performance) sebesar 15-20% pada operasi INSERT dan UPDATE. Analisis ini menguatkan prinsip fundamental dalam manajemen basis data bahwa tidak ada solusi yang sempurna, melainkan diperlukan pertimbangan matang antara keuntungan dan kerugian berdasarkan karakteristik beban kerja sistem.

Grafik Hubungan Selektivitas dan Efektifitas Indeks



Gambar 3. Grafik Hubungan Selektivitas dan Efektifitas Indeks

Gambar 3 mendeskripsikan hubungan menarik antara tingkat selektivitas dan efektivitas indeks. Meskipun indeks komposit memiliki selektivitas yang lebih tinggi (92%) dibandingkan indeks tunggal (85%), namun efektivitasnya dalam meningkatkan performa kueri justru sedikit lebih rendah. Fenomena ini dapat dijelaskan melalui konsep Prinsip Awalan Paling Kiri (*Leftmost Prefix Principle*), di mana indeks komposit paling optimal ketika kueri menggunakan seluruh atau sebagian besar kolom yang terindeks. Untuk kueri sederhana yang hanya

melibatkan satu kolom, indeks tunggal tetap menjadi pilihan terbaik. Temuan ini menegaskan pentingnya analisis pola kueri sebelum menentukan strategi *indexing* yang tepat.

D. PENUTUP

Berdasarkan hasil penelitian mengenai efektivitas teknik *indexing* terhadap waktu eksekusi *query* SQL pada basis data transaksi *e-commerce* berbasis MySQL, dapat disimpulkan bahwa penerapan indeks terbukti mampu meningkatkan kinerja eksekusi *query* secara signifikan. Penggunaan *single index* dan *composite index* mengubah pola akses data dari *full table scan* dengan kompleksitas O(n) menjadi akses berbasis indeks dengan kompleksitas O(log n), yang tercermin dari penurunan waktu eksekusi *query* hingga sekitar 68,91% pada skenario *single index* dan 62,18% pada skenario *composite index*, serta reduksi jumlah baris yang dipindai dari 20.000 baris menjadi hanya puluhan baris. Efektivitas masing-masing jenis indeks sangat bergantung pada pola *query* dan karakteristik data; *single index* pada kolom *Customer_Id* lebih optimal untuk *query* sederhana dengan kondisi tunggal pada kolom yang memiliki *selectivity* tinggi, sedangkan *composite index* pada kombinasi *Customer_Id* dan *Order_Date* lebih sesuai untuk *query* kompleks yang melibatkan *multiple conditions*, khususnya pada analisis transaksi berbasis pelanggan dan rentang waktu.

Namun demikian, peningkatan performa baca tersebut memiliki konsekuensi berupa *trade-off* pada sisi operasi tulis dan penyimpanan, yang ditandai dengan perlambatan operasi INSERT/UPDATE sekitar 15–20% dan adanya *overhead* ruang penyimpanan tambahan untuk struktur indeks serta kebutuhan *maintenance* berkala. Oleh karena itu, strategi *indexing* perlu dirancang secara berbasis *evidence* melalui pemetaan pola *query*, analisis *selectivity*, dan pemantauan statistik penggunaan indeks, sehingga implementasi *single index*, *composite index*, maupun indeks khusus dapat dioptimalkan sesuai kebutuhan sistem. Secara keseluruhan, teknik *indexing* yang dirancang dan dikelola dengan tepat dapat menjadi salah satu pilar utama dalam mengoptimalkan kinerja basis data transaksi *e-commerce*, dengan tetap mempertimbangkan keseimbangan antara performa, beban tulis, dan efisiensi sumber daya dalam konteks operasional nyata.

Berdasarkan hasil penelitian yang diperoleh, disarankan agar pengembang dan pengelola basis data *e-commerce* tidak hanya berfokus pada percepatan waktu eksekusi *query*, tetapi juga melakukan analisis pola *query* secara berkala sebelum memutuskan strategi *indexing* yang akan digunakan. Penerapan *single index* sebaiknya diprioritaskan pada kolom-kolom dengan *selectivity* tinggi dan frekuensi akses yang tinggi, sedangkan *composite index* digunakan secara selektif pada skenario *query* kompleks yang benar-benar membutuhkan kombinasi beberapa kolom. Selain itu, perlu dilakukan pemantauan rutin terhadap statistik penggunaan indeks serta evaluasi periodik terhadap *trade-off* antara peningkatan performa baca, penurunan performa tulis, dan kebutuhan ruang penyimpanan.

Untuk penelitian selanjutnya, disarankan untuk memperluas skala *dataset*, membandingkan beberapa *Database Management System* (DBMS) yang berbeda, serta menguji variasi jenis indeks lain (misalnya *full-text index* atau *partial index*) pada berbagai pola beban kerja. Penelitian lanjutan juga dapat mengkaji integrasi *indexing* dengan teknik optimasi lain, seperti *query rewriting* atau *materialized view*, sehingga dapat diperoleh panduan optimasi kinerja basis data yang lebih komprehensif dan aplikatif pada beragam skenario sistem *e-commerce*.

E. DAFTAR PUSTAKA

Adisa, Y., & Nasution, M. I. P. (2023). Konsep Dan Peran Sistem Manajemen Basis Data Relasional Pada Sistem Informasi Manajemen. *Jurnal Manajemen Administrasi Bisnis Dan Publik Terapan*, *1*(3), 76–83. https://doi.org/https://doi.org/10.59061/masip.v1i3.314

Ahsana, S. H., Syahputra, M. B., Putri, A. F. F. M., & Prasetyo, A. A. (2023). Analisis Perbandingan Performa Antara Mysql Dan Postgresql. *Prosiding Seminar Nasional Teknologi Dan Sistem Informasi (SITASI)*, 6–7. https://repository.upnjatim.ac.id/28497/

Anchlia, A. (2024). Enhancing Query Performance Through Relational Database Indexing. *International Journal of Computer Trends and Technology*, *72*(8), 130–133. https://doi.org/10.14445/22312803/IJCTT-V72I8P119

Ibna, A. Z. (2024). Implikasi Penggunaan Basis Data dalam Big Data. *Jurnal Sains Student Research*, *2*(4), 255–265. https://doi.org/https://doi.org/10.61722/jssr.v2i4.1995

Maesaroh, S., Gunawan, H., Lestari, A., & Sufyan, M. (2022). Query Optimization in MySQL Database Using Index. *International Journal of Cyber and IT Service Management (IJCITSM)*, *2*(2), 104–110. https://doi.org/10.34306/ijcitsm.v2i2.84

Nurhayati, S. T., & Nasution, M. I. P. (2023). Database Management System Pada Perusahaan. *Jurnal Akuntansi Keuangan Dan Bisnis*, *1*(2), 62–64. https://jurnal.ittc.web.id/index.php/jakbs/article/view/43

Pamungkas, R. (2018). Optimalisasi Query Dalam Basis Data My Sql Menggunakan Index. *Journal of Computer, Information System, & Technology Management*, *1*(2), 27–31. https://doi.org/10.25273/research.v1i1.2453

Panggabean, S., & Azizah, E. S. N. (2025). Evaluasi Kinerja Sistem Basis Data Relasional pada Aplikasi E-Commerce Menggunakan Algoritma Indexing. *Pustaka Data : Pusat Akses Kajian Database, Analisa Teknologi, Dan Arsitektur Komputer*, *5*(1), 70–76. https://doi.org/10.55382/jurnalpustakadata.v5i1.998

Permana, K. E., Sophan, M. K., Muntasa, A., & Rahmat, A. B. (2023). Perbandingan Kinerja Query Sql Join Tables dengan Menggunakan Index. *Jurnal Simantec*, *11*(2), 241–248. https://doi.org/10.21107/simantec.v11i2.20288

Poggi, N., Carrera, D., Gavalda, R., Avyguade, E., & Torres, J. (2014). A Methodology for The Evaluation of High Response Time on E-Commerce Users and Sales. *Information Systems Frontiers*, *16*(5), 867–885. https://doi.org/10.1007/s10796-012-9387-4

Rahayu, S., Halawa, H. W., Abdillah, A. F., & Mujayanah, A. (2025). Pemanfaatan Big Data Analytics untuk Analisis Pola Perilaku Konsumen E-Commerce Strategis. *JUTECH: Journal Education and Technology*, *6*(1), 64–73. https://doi.org/10.31932/jutech.v6i1.4834

Sakshi, S., & Sharma, A. (2025). Relational Database Performance Optimization Techniques. *International Conference on Recent Advancements and Modernisations in Sustainable Intelligent Technologies and Applications (RAMSITA)*. https://doi.org/10.2991/978-94-6463-716-8_10

Šušter, I., & Ranisavljević, T. (2023).

Optimization of Mysql Database. *Journal of Process Management and New Technologies*, *11*(1), 141–151. https://doi.org/10.5937/jouproman2301141Q

Taipalus, T. (2024). Database management system performance comparisons: A systematic literature review. *Journal of Systems and Software*, *208*, 111872. https://doi.org/10.1016/j.jss.2023.111872

Thoib, I., Candra, B. P., Sururi, N., & Nugraha, D. S. (2024). Perbandingan Performa Pencarian Data Berbasis Teks dengan Dan Tanpa Full-text Index pada Basis Data MySQL. *Jurnal Sains Dan Teknologi*, *3*(6), 663–673. https://doi.org/10.55123/insologi.v3i6.4596

ANALISIS KOMPARATIF EFISIENSI WAKTU DAN MEMORI: LINEAR, BINARY, DAN *HASH SEARCH* BERBASIS C++

Ikhsan Maulanaa[1)], Firda Rona Syahira[2)], Meila Fitri Amin[3)], Nevi Dwi Apriyanti[4)], Imam Prayogo Pujiono[5)]

[1,2,3,4]Prodi Bisnis Digital, Ekonomi dan Bisnis Islam, UIN K.H. Abdurrahman Wahid Pekalongan
[5]Prodi Informatika, Ekonomi dan Bisnis Islam, UIN K.H Abdurrahman Wahid Pekalongan

Correspondence author: FR Syahira, firda.rona.syahira25006@mhs.uingusdur.ac.id, Pekalongan, Indonesia

Abstract

This study aims to analyze the comparative memory efficiency and computation time of three commonly used data search algorithms in C++ programming: Linear Search, Binary Search, and Hash Search. This study analyzes the performance of each algorithm based on two main aspects: execution speed and memory usage, to determine the most appropriate algorithm for various data conditions. The research method used was a quantitative comparative experiment, where each algorithm was implemented in C++ and tested on three data size scenarios (100, 1000, and 10,000 elements). The results obtained indicate that Binary Search has the best balance between speed and memory efficiency. At the same time, Linear Search is more suitable for small data sizes due to its simplicity of implementation. Conversely, Hash Search has high search speed but requires more memory. Therefore, the selection of the appropriate search algorithm should be tailored to the characteristics of the data and system requirements. This study is expected to serve as a reference for software developers in selecting the optimal search algorithm based on computational resource efficiency.

Keyword *: algorithm comparison, data search, memory efficiency, computation time*

Abstrak

Penelitian ini bertujuan untuk menganalisis perbandingan efisiensi memori dan waktu komputasi pada tiga algoritma pencarian data yang umum digunakan dalam pemrograman C++, yaitu *Linear Search*, *Binary Search*, dan *Hash Search*. Penelitian ini menganalisis kemampuan kerja pada masing-masing algoritma berdasarkan dua aspek utama, yakni kecepatan eksekusi dan penggunaan memori, agar dapat menentukan algoritma yang paling sesuai untuk berbagai kondisi data. Metode penelitian yang digunakan adalah eksperimen komparatif kuantitatif, di mana setiap algoritma diimplementasikan menggunakan bahasa C++ dan diuji pada tiga skenario ukuran data (100, 1000, dan 10000 elemen). Hasil yang diperoleh menunjukkan bahwa *Binary Search* memiliki kinerja paling seimbang antara kecepatan dan efisiensi memori, sedangkan *Linear Search* lebih cocok untuk data yang ukuranya kecil karena kesederhanaan implementasinya.

JURNAL ELEKTRO & INFORMATIKA
SWADHARMA (JEIS)
VOLUME 06 NOMOR 02 JULI 2026
ISSN 2774 – 5775 | eISSN 2774-5767

Sebaliknya, *Hash Search* memiliki kecepatan pencarian yang tinggi, tetapi membutuhkan memori lebih besar. Dengan demikian, pemilihan algoritma pencarian yang tepat sebaiknya disesuaikan dengan karakteristik data dan kebutuhan sistem. Penelitian ini diharapkan dapat menjadi acuan bagi pengembang perangkat lunak dalam memilih algoritma pencarian yang optimal berdasarkan efisiensi sumber daya komputasi.

Kata Kunci : perbandingan algoritma, pencarian data, efisiensi memori, waktu komputasi

A. PENDAHULUAN

Pencarian data merupakan salah satu operasi fundamental dalam bidang ilmu komputer yang memiliki peran penting dalam proses pengolahan informasi (Budiansyah et al., 2025). Hampir seluruh sistem yang mengelola volume data besar seperti *database management system*, *search engine*, maupun aplikasi analitik melibatkan proses pencarian untuk menemukan elemen tertentu dari sekumpulan data (Akhsa et al., 2024). Efisiensi proses pencarian ini berpengaruh langsung terhadap kinerja sistem secara keseluruhan, terutama pada aplikasi *real-time* yang membutuhkan respons cepat serta penggunaan sumber daya yang optimal (Suryadi et al., 2024).

Dalam praktik pemrograman modern, khususnya dengan menggunakan bahasa C++, pemilihan algoritma pencarian yang tepat menjadi aspek penting yang menentukan performa program (Purbasari et al., 2024). C++ dikenal sebagai bahasa yang efisien karena memberikan kontrol tinggi terhadap penggunaan memori dan waktu eksekusi (R. A. Putra, 2024). Melalui dukungan berbagai struktur data seperti *array*, *vector*, dan *hash table*, pengembang memiliki fleksibilitas tinggi untuk menentukan metode pencarian yang paling sesuai dengan karakteristik data (Erkamim et al., 2024).

Di antara berbagai metode yang tersedia, tiga algoritma pencarian yang umum digunakan dan sering dibandingkan dari segi efisiensi adalah *Linear Search*, *Binary Search*, dan *Hash Search* (Situmorang, 2017). Ketiganya memiliki perbedaan mendasar dalam prinsip kerja, kompleksitas waktu, serta kebutuhan memori (Firmansyah et al., 2025). *Linear Search* dikenal dengan implementasinya yang sederhana karena melakukan pencarian dengan membandingkan setiap elemen secara berurutan hingga data ditemukan (Leana et al., 2025). Meskipun efektif untuk data berukuran kecil dan tidak terurut, algoritma ini memiliki keterbatasan dalam skala besar karena kompleksitas waktunya bersifat linear O(n) (Izhari, 2025).

Sebaliknya, *Binary Search* menggunakan pendekatan *divide and conquer* dengan kompleksitas waktu rata-rata O(log n), sehingga lebih efisien untuk data besar yang sudah terurut (Ariza et al., 2025). Namun, kebutuhan proses pengurutan awal menjadi kelemahan tersendiri ketika data sering berubah atau bersifat dinamis (Mevia et al., 2026). Di sisi lain, *Hash Search* mampu memberikan waktu pencarian yang hampir konstan O(1) pada kondisi ideal dengan memanfaatkan *hash table* sebagai indeks data (Bender et al., 2022). Walaupun sangat cepat, metode ini memiliki kelemahan dalam efisiensi memori karena memerlukan ruang tambahan untuk mengatasi collision antar nilai hash (Yusuf et al., 2021).

Evaluasi terhadap ketiga algoritma tersebut penting dilakukan mengingat performa aktual algoritma sering kali dipengaruhi oleh karakteristik data dan konfigurasi system (R. F. Putra et al., 2023). Secara teoritis, algoritma dengan

kompleksitas waktu lebih baik belum tentu menunjukkan performa terbaik pada implementasi nyata, terutama dalam kondisi sistem dengan keterbatasan sumber daya *(resource constraints)* (Puranik, 2025). Oleh karena itu, penelitian ini berfokus tidak hanya pada pengukuran kecepatan eksekusi, tetapi juga pada efisiensi penggunaan memori, dimana dua hal tersebut menjadi faktor utama yang saling memengaruhi dalam optimalisasi kinerja algoritma.

Sejumlah penelitian sebelumnya telah membahas aspek efisiensi algoritma pencarian dari perspektif waktu komputasi, namun kajian yang menggabungkan analisis waktu dan memori secara bersamaan masih terbatas, khususnya dengan implementasi langsung di C++ (Sitorus et al., 2024). Misalnya, penelitian oleh (Budiansyah et al., 2025) menitikberatkan pada optimalisasi pencarian pada kecerdasan buatan, sementara studi oleh (Purnama, 2025) menunjukkan bahwa *Hash Search* memiliki kinerja waktu terbaik pada dataset besar, namun belum memperhitungkan aspek penggunaan memori. Dengan demikian, penelitian ini menghadirkan nilai kebaruan *(novelty)* melalui perbandingan langsung antara efisiensi waktu dan memori dari ketiga algoritma pencarian tersebut dalam lingkungan terkontrol berbasis C++.

Selain itu, penelitian ini memiliki kontribusi praktis yang signifikan bagi pengembang perangkat lunak *(software developer)*, khususnya dalam membantu pemilihan metode pencarian yang paling tepat berdasarkan ukuran data, frekuensi pencarian, serta batasan sumber daya sistem. Hasil penelitian diharapkan dapat menjadi acuan empiris yang bermanfaat dalam pengembangan aplikasi berbasis data besar, seperti sistem *machine learning*, data mining, maupun *text retrieval system*.

Penelitian diharapkan dapat menyajikan analisis kuantitatif dan kualitatif terhadap *Linear Search*, *Binary Search*, dan *Hash Search* dalam hal efisiensi waktu serta penggunaan memori pada implementasi nyata di bahasa pemrograman C++.

B. METODE PENELITIAN

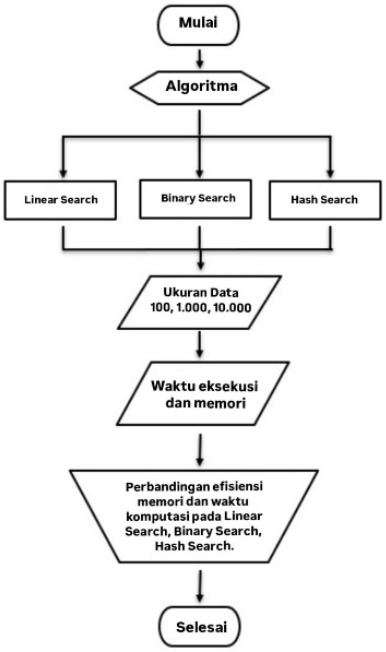
Penelitian ini menggunakan bahasa pemrograman C++ sebagai alat utama untuk menjalankan berbagai kode yang diperlukan. Bahasa pemrograman C++ dipilih karena bahasa ini dikenal karena kecepatan, efisiensi, serta kemampuannya dalam memberikan kontrol yang baik terhadap penggunaan memori, sehingga sangat sesuai untuk penelitian yang fokus pada kinerja algoritma. Seluruh pengujian dilaksanakan menggunakan sebuah laptop dengan spesifikasi : Prosesor AMD Ryzen 5 7520U, SSD 512 GB M.2, Memori RAM: 16 GB, Sistem operasi Windows 11.

Studi ini menerapkan pendekatan kuantitatif dengan metode eksperimen komparatif untuk mengevaluasi efisiensi penggunaan memori dan waktu hitung antara algoritma pencarian data (*Linear Search*, *Binary Search*, dan *Hash Search*). Metode eksperimen dipilih karena melibatkan pengujian secara langsung terhadap performa algoritma yang menghasilkan data yang dapat diukur. Penelitian ini bertujuan untuk menilai performa algoritma secara sistematis, berfokus pada dua aspek: penggunaan memori dan durasi komputasi. Untuk memperjelas proses, tahapan dalam penelitian ini terdiri dari pengumpulan data, implementasi algoritma, pengujian performa, dan analisis perbandingan.

Tahap eksperimen dimulai dengan pengumpulan data yang dibagi menjadi tiga jenis elemen : 100 elemen, 1000 elemen, dan 10000 elemen untuk menguji skalabilitas algoritma. Seluruh proses tersebut divisualisasikan menggunakan diagram alur untuk memperjelas urutan langkah dalam penelitian.

Diagram alur tersebut dapat dilihat pada gambar 1 berikut :

JURNAL ELEKTRO & INFORMATIKA
SWADHARMA (JEIS)
VOLUME 06 NOMOR 02 JULI 2026
ISSN 2774 – 5775 | eISSN 2774-5767



Gambar 1. Diagram Alur Penelitian

Dengan cara ini, penelitian dapat mengamati apakah kinerja algoritma tetap stabil atau mengalami perubahan signifikan saat ukuran data bertambah. Setelah data siap, ketiga algoritma pencarian tersebut kemudian diimplementasikan memakai C++. Seluruh program ditulis dan dijalankan menggunakan *Code::Blocks*, sebuah lingkungan pengembangan terintegrasi yang sangat mendukung proses kompilasi dan *debugging*. Setelah implementasi selesai, langkah selanjutnya adalah melakukan evaluasi kinerja. Pada tahap ini, setiap algoritma diuji berulang kali dengan tiga ukuran data yang sudah disiapkan. Pengulangan ini dilakukan untuk memastikan bahwa hasil yang diperoleh konsisten dan tidak terpengaruh oleh kebetulan.

Dari setiap uji coba, dua jenis informasi dicatat, yaitu waktu pemrosesan dalam milidetik (*ms*) dan penggunaan memori dalam kilobyte (*KB*). Data-data ini kemudian akan menjadi dasar utama untuk perbandingan antar algoritma. Dengan menganalisis hasil dari ketiga algoritma tersebut, peneliti dapat menentukan algoritma mana yang paling cepat, mana yang paling hemat memori, serta bagaimana perubahan jumlah elemen mempengaruhi kinerja masing-masing program.

C. HASIL DAN PEMBAHASAN

Linear Search

Algoritma *Linear Search* direalisasikan menggunakan bahasa pemrograman C++. Algoritma *Linear Search* adalah salah satu metode paling sederhana dalam algoritma pencarian yang berfungsi dengan memeriksa setiap elemen dalam struktur data secara berurutan, mulai dari indeks awal hingga indeks akhir, sampai elemen yang dicari ditemukan atau semua elemen telah diperiksa. Implementasi algoritma ini dalam bahasa C++ di program yang dimaksud dilakukan secara langsung, di mana semua data diinput, diproses, dan dijalankan oleh program yang telah dirancang sebelumnya. Dengan model tersebut, proses pengujian menjadi lebih teratur karena semua langkah mulai dari memasukkan data, menjalankan algoritma, hingga mengukur kinerja dapat dilakukan dalam satu arus komputasi tanpa campur tangan manual.

Dalam tahap pengujian performa program, jumlah elemen data yang digunakan ditentukan dalam tiga skala yang berbeda, yaitu 100, 1.000, dan 10.000 elemen. Pemilihan variasi jumlah elemen tersebut ditujukan untuk melihat bagaimana kinerja algoritma *Linear Search* beradaptasi saat ukuran data meningkat secara drastis. Secara fundamental, *Linear Search* memiliki kompleksitas waktu O(n), yang menunjukkan bahwa waktu yang diperlukan untuk menyelesaikan proses pencarian meningkat secara linier seiring dengan bertambahnya jumlah elemen yang harus dicek. Oleh karena itu, pengujian dengan ketiga ukuran data tersebut dapat memberikan gambaran yang cukup jelas mengenai sejauh mana algoritma terpengaruh oleh peningkatan ukuran *input*.

Proses pengujian pada setiap set data dilakukan melalui prosedur yang sama, yaitu dengan menjalankan algoritma untuk mencari elemen tertentu pada posisi acak dalam daftar data. Tujuannya adalah untuk memastikan bahwa hasil pengukuran waktu komputasi yang diperoleh adalah objektif dan mencerminkan kondisi nyata. Selain itu, data yang digunakan merupakan data yang dihasilkan oleh program itu sendiri sehingga tidak ada ketergantungan pada faktor luar yang dapat mempengaruhi hasil pengujian. Dengan cara ini, semua proses pencarian berlangsung dalam lingkungan komputasi yang terkontrol dan konsisten.

Hasil pengukuran yang didapat dari pengujian ini mencakup dua aspek utama, yakni efisiensi memori dan waktu komputasi. Efisiensi memori berkaitan dengan jumlah memori yang dimanfaatkan oleh program untuk menyimpan data dan menjalankan algoritma pencarian. Karena *Linear Search* tidak memerlukan struktur data tambahan atau proses penyimpanan data sementara yang rumit. Di sisi lain, waktu komputasi menjadi aspek yang paling mencolok dalam ujian ini. Seperti yang telah diketahui, semakin banyak elemen yang harus diperiksa, semakin lama waktu yang dibutuhkan oleh algoritma *Linear Search* untuk menemukan elemen tertentu, terutama jika elemen yang dicari terletak di bagian akhir daftar atau bahkan tidak ditemukan sama sekali. Oleh karena itu, perbandingan waktu eksekusi pada ukuran data 100, 1000, dan 10000 elemen sangat penting untuk menunjukkan sejauh mana *Linear Search* dapat diandalkan ketika diterapkan dalam konteks data yang semakin besar.

Seluruh data pengukuran diperoleh dengan merekam waktu mulai dan waktu akhir pelaksanaan algoritma menggunakan fungsi bawaan dalam C++, seperti *chrono*, yang memberikan akurasi tinggi dalam pengukuran waktu. Sementara itu, pengukuran penggunaan memori dilakukan dengan memanfaatkan fungsi pemantauan memori yang sesuai dengan lingkungan pengembangan program. Data hasil pengukuran mengenai efisiensi memori dan waktu komputasi kemudian disajikan secara menyeluruh pada Tabel 1. sehingga memudahkan pembaca dalam melakukan analisis dan perbandingan antara berbagai ukuran data.

Tabel 1. Hasil uji *Linear Search*

Ukuran Data	Hasil Pengujian	
	Waktu *(/ms)*	Memori *(KB)*
100	0,3382	0,390635
1.000	0,2212	3,90625
10.000	0,4013	39,0625
Rata-rata	0,320234	14,253125

Berdasarkan Tabel 1. diketahui bahwa rata-rata kecepatan waktu eksekusi program dengan tiga elemen adalah 0,320234 *ms*. Dan rata-rata penggunaan memori dengan tiga jenis elemen adalah 14,453125 *KB*. Dengan penyajian data tersebut, pembaca bisa memahami bagaimana kinerja algoritma *Linear Search* dalam beragam skenario ukuran *input* dan dapat mengevaluasi apakah algoritma ini cocok digunakan dalam aplikasi tertentu, terutama saat berhadapan dengan kumpulan data besar yang membutuhkan efisiensi tinggi.

Binary Search

Binary Search adalah salah satu algoritma pencarian yang paling sering digunakan karena memiliki kompleksitas waktu yang jauh lebih efektif dibandingkan dengan metode pencarian *linear*. *Binary Search* hanya dapat digunakan pada data yang sudah terurut, baik itu dalam urutan menaik maupun menurun, sehingga proses pencariannya dapat dilakukan dengan cara yang teratur dan sistematis. Salah satu keunggulan dari algoritma ini adalah kemampuannya untuk mengurangi area pencarian dengan terus membagi data menjadi dua bagian sampai elemen yang dicari ditemukan.

Secara konseptual, cara kerja dari algoritma *Binary Search* diawali dengan

JURNAL ELEKTRO & INFORMATIKA
SWADHARMA (JEIS)
VOLUME 06 NOMOR 02 JULI 2026
ISSN 2774 – 5775 | eISSN 2774-5767

menentukan nilai tengah dari rentang indeks data yang sedang diperiksa. Jika nilai di indeks tengah tersebut sesuai dengan yang dicari, proses pencarian dapat dihentikan. Namun, jika nilai tengah lebih besar daripada target dalam data yang terurut menaik, pencarian akan dilanjutkan pada bagian kiri dari data. Sebaliknya, jika nilai tengah lebih kecil, pencarian akan beralih ke bagian kanan. Proses pembagian area pencarian ini akan terus berlangsung hingga elemen ditemukan atau sampai tidak ada lagi ruang pencarian yang tersisa. Mekanisme ini memungkinkan *Binary Search* memiliki kompleksitas waktu rata-rata O(log n), yang menjadikannya jauh lebih efisien dibandingkan *Linear Search* yang memiliki kompleksitas O(n). Dalam penelitian atau penerapan praktis, pengukuran kinerja dari suatu algoritma sangat krusial untuk menentukan seberapa efisien waktu komputasi dan penggunaan memori yang dibutuhkan. Dalam studi ini, *Binary Search* diuji dengan tiga kategori jumlah elemen data, yaitu 100, 1000, dan 10000 elemen. Ketiga ukuran data ini dipilih untuk mewakili skala kecil, menengah, dan besar, sehingga analisis perbandingan kinerjanya dapat dilakukan dengan lebih komprehensif. Data yang digunakan sudah diurutkan sebelumnya agar sesuai dengan syarat penerapan algoritma *Binary Search*.

Pengujian dilakukan melalui beberapa algoritma pemrograman yang mengikuti logika dasar dari *Binary Search* yang serupa. Setiap pengujian mengukur dua parameter utama, yaitu waktu eksekusi dan penggunaan memori. Hasil lengkap dari pengukuran bisa dilihat pada Tabel 2.

Tabel 2. Hasil uji *Binary Search*

Ukuran Data	Hasil Pengujian	
	Waktu *(/ms)*	Memori (*KB)*
100	0,0002	0,390625
1.000	0,0003	3,90625
10.000	0,0004	39,0625
Rata-rata	0,0003	14,453125

Berdasarkan tabel 2 tersebut, didapatkan fakta bahwa rata-rata kecepatan waktu eksekusi dari program dengan tiga variasi jumlah elemen adalah 0,0003 *ms*. Angka ini menunjukkan bahwa algoritma *Binary Search* memiliki kinerja yang sangat cepat meskipun pada data yang berukuran besar. Efisiensi waktu ini tidak lepas dari struktur logaritmik dalam algoritma yang mampu mengurangi jumlah langkah pencarian dengan signifikan. Selain itu, rata-rata penggunaan memori untuk tiga jenis elemen adalah 14,453125 *KB*. Nilai penggunaan memori yang relatif kecil ini menunjukkan bahwa *Binary Search* tidak membutuhkan alokasi memori tambahan yang signifikan, selain dari struktur data utama yang sudah ada. Dengan menggunakan teknik pembagian rentang indeks tanpa membuat salinan data baru, *Binary Search* dapat menjaga efisiensi memori, sehingga sangat cocok untuk digunakan dalam aplikasi yang memiliki keterbatasan ruang penyimpanan. Melalui analisis ini, dapat dirangkum bahwa *Binary Search* adalah salah satu metode pencarian yang paling efektif terkait waktu pemrosesan serta penggunaan memori. Keunggulan ini tetap terjaga meskipun jumlah data meningkat. Inilah alasan mengapa *Binary Search* sering digunakan di berbagai sektor, termasuk sistem *database*, pencarian di struktur pohon, optimisasi algoritma, serta dalam berbagai aplikasi yang memerlukan pencarian cepat pada data yang telah diurutkan. Secara keseluruhan, hasil studi ini dapat memberikan gambaran yang lebih menyeluruh tentang kinerja algoritma *Binary Search* dalam berbagai situasi ukuran data yang berbeda.

Hash Search

Hash Search merupakan metode atau algoritma pencarian yang dirancang untuk memproses volume data yang besar dengan cara yang cepat dan efisien. Secara umum, algoritma ini beroperasi dengan memanfaatkan struktur data yang dikenal sebagai *hash table*, yaitu tabel yang

menyimpan data berdasarkan indeks yang diperoleh dari perhitungan fungsi *hash*. Fungsi *hash* sendiri bertugas mengonversi nilai atau kunci data menjadi indeks tertentu, memungkinkan pencarian dilakukan secara langsung tanpa harus memeriksa data satu per satu. Dengan mekanisme ini, *Hash Search* dapat memberikan kecepatan pencarian yang jauh lebih tinggi, terutama ketika ukuran *database* sangat besar atau jumlah entri mencapai ribuan hingga jutaan.

Namun, walaupun *Hash Search* dikenal sangat efisien untuk data berskala besar, algoritma ini tidak selalu menjadi pilihan terbaik untuk *database* dengan ukuran kecil. Pada *database* yang hanya memiliki sedikit elemen, penggunaan *Hash Search* justru dapat memberikan *overhead*, yaitu tambahan beban pada proses pembuatan *hash table* itu sendiri. *Overhead* ini mencakup alokasi memori untuk struktur tabel, perhitungan fungsi *hash*, serta penanganan potensi terjadinya benturan data di indeks tertentu. Ketika data yang tersedia sangat sedikit, waktu dan sumber daya yang diperlukan untuk membuat *hash table* sering kali tidak sebanding dengan keuntungan yang diperoleh. Dengan kata lain, penyiapan struktur *hash* justru bisa memperlama waktu eksekusi karena data yang sedikit seharusnya bisa diproses dengan lebih cepat melalui metode pencarian sederhana seperti *Linear Search* atau *Binary Search.* Dalam prinsip operasionalnya, *Hash Search* memungkinkan sistem untuk langsung mengetahui lokasi data berdasarkan hasil fungsi *hash*. Inilah alasannya *Hash Search* sangat efisien untuk data yang besar, karena kompleksitas waktunya dapat mendekati O(1) atau waktu konstan dalam banyak situasi.

Tabel 3. Hasil uji *Hash Search*

Ukuran Data	Hasil Pengujian	
	Waktu *(/ms)*	Memori (*KB)*
100	4852	0,390625
1.000	5623	3,90625
10.000	12492	39,0625
Rata-rata	7655,67	14,453125

Berdasarkan data yang terdapat dalam Tabel 3. didapatkan informasi tentang rata-rata kinerja program saat menangani tiga elemen data. Dari tabel tersebut, diketahui bahwa rata-rata waktu eksekusi program untuk tiga elemen adalah 7655,67 *ms*. Angka ini menunjukkan bahwa ketika jumlah data sangat sedikit, waktu yang dibutuhkan tidak secepat yang diharapkan dari algoritma *hash*. Seperti penjelasan sebelumnya, lambatnya waktu eksekusi bukanlah akibat dari proses pencarian itu sendiri, melainkan karena tahap persiapan struktur *hash table* yang tetap menghabiskan waktu meskipun data yang ada sedikit. Di samping itu, dari tabel yang sama, tercatat bahwa rata-rata penggunaan memori selama pengujian tiga elemen adalah 14,453125 *KB*. Angka tersebut menunjukkan bahwa meskipun ukuran datanya minimal, penggunaan *Hash Search* tetap memerlukan memori untuk menyimpan struktur tabel. Hal ini memperkuat kesimpulan bahwa algoritma *Hash Search* lebih tepat digunakan ketika ukuran data cukup besar, sehingga penggunaan memori dan waktu yang diperlukan untuk persiapan struktur *hash* menjadi relatif kecil dibandingkan dengan manfaat percepatan dalam proses pencariannya.

Tabel 4. Hasil uji 3 algoritma

Jenis Program	*Rata-rata waktu (ms)*	*Rata-rata memori (kb)*
Linear Search	0,320234	14,453125
Binary Search	0,0003	14,453125
Hash Search	7655,67	14,453125

Berdasarkan hasil simulasi, terlihat bahwa *Hash Search* merupakan algoritma yang paling tidak efisien dari segi waktu komputasi. Akan tetapi perlu kita catat bahwa waktu pada *Hash Search* itu termasuk proses pembentukan struktur *hash*, yang mana hasil ini tidak sepenuhnya mencerminkan

performa murni pencarian, tapi juga operasi persiapan data. *Binary Search* menunjukkan keefisienan terbaik dalam waktu komputasi, meskipun harus mengurutkan datanya terlebih dahulu. Sedangkan *Linear Search* menjadi alternatif paling sederhana namun tidak efisien untuk data berukuran besar. Selain itu, terllihat juga ketiga algoritma tersebut memiliki rata-rata memori yang sama walau dengan waktu komputasi yang berbeda. Penelitian lanjutan disarankan untuk menguji variasi algoritma pencarian lainnya seperti *Interpolation Search* atau *Exponential Search*.

D. PENUTUP

Berdasarkan hasil pengujian terhadap algoritma pencarian *Linear Search*, *Binary Search*, dan *Hash Search* pada tiga jenis elemen data yang berbeda yaitu 100, 1000, dan 10000 elemen pada bahasa C++. Penelitian ini menguji antara waktu komputasi dan penggunaan memori pada setiap algoritma pemrogramannya. Penelitian ini dapat di buktikan, bahwa *Binary Search* adalah algoritma yang lebih efektif dan efisien dari segi waktu komputasi dengan kecepatan rata-rata 0,0003 ms. Sementara itu, *Linear Search* membuktikan kecepatannya, namun sedikit lebih lambat dari *Binary Search* dengan kecepatan waktu komputasi rata-rata 0,320234 ms, tetapi *Linear Search* ini lebih cocok untuk pencarian data dengan jumlah data yang lebih sedikit dan tidak urut. Namun sebaliknya, hasil pengujian pada *Hash Search* meskipun pada teorinya merupakan algoritma pencarian data tercepat dari kedua algoritma tersebut, pada pengujiannya *Hash Search* justru menunjukkan waktu komputasi yang lebih lambat dari kedua algoritma tersebut dengan waktu komputasi rata-rata 7655,67 ms. Dalam penggunaan memorinya, *Linear Search*, *Binary Search*, dan *Hash Search* keduanya menggunakan memori dengan jumlah yang sama besarannya, yaitu rata-rata sebesar 14,453125 kb.

Dengan demikian, pemilihan algoritma pemrograman dapat dilakukan sesuai kebutuhan, untuk penggunaannya *Binary Search* lebih unggul dalam mencari data dengan waktu komputasi yang lebih cepat walaupun data yang digunakkan harus diurutkan terlebih dahulu. Sedangkan untuk *Hash Search* sendiri, berdasarkan hasil implementasi yang dilakukan dalam penelitian, algoritma tersebut cenderung menunjukkan waktu yang lebih lama dalam proses komputasinya dibandingkan *Linear Search* dan *Binary Search*, walaupun dengan penggunaan memori yang sama. Berbanding terbalik dengan teori yang ada, penelitian ini justru menemukan ketidakefisienan waktu komputasi dan kesamaan dalam pengunaan memori pada *Hash Search*, yang mana jikalau menurut teori yang ada, *Hash Search* itu lebih cepat dan lebih banyak memakan memori kalau dibandingkan dengan *Linear Search* maupun *Binary Search*. Hal ini diduga disebabkan oleh *overhead* pembentukan struktur hash pada scenario data yang relatif kecil.

E. DAFTAR PUSTAKA

Akhsa, A. T. P. D., Kelvin, K., Eldo, H., & Efitra, E. (2024). *Buku Ajar Big Data*. Jambi : PT. Sonpedia Publishing Indonesia.

Ariza, S. F., Majid, A., Himawan, I., Ardhiartha P.U., S., & Pujiono, I. P. (2025). Studi Perbandingan Algoritma Pencarian Binary, Jump, Interpolation, dan Fibonacci: Efisiensi Memori dan Waktu Eksekusi. *JATI : Jurnal Mahasiswa Teknik Informatika*, *9*(4), 7227–7234. https://doi.org/10.36040/jati.v9i4.14360

Bender, M. A., Farach-Colton, M., Kuszmaul, J., Kuszmaul, W., & Liu, M. (2022). On the optimal time/space tradeoff for hash tables. *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*,

1284–1297. https://doi.org/10.1145/3519935.35199

Budiansyah, Komala, I. R., & Nurhayati, L. (2025). Analisis Komparatif Strategi Algoritma Pencarian Dalam Penyelesaian Masalah Kecerdasan Buatan. *Infoman's : Jurnal Ilmu-Ilmu Informatika Dan Manajemen*, *19*(2), 1–5. https://ejournal.unsap.ac.id/index.php/infomans/article/view/2432

Erkamim, E., Abdurrohim, I., Yuliyanti, S., Karim, R., Rahman, A., Admira, T. M. A., & Ridwan, A. (2024). *Buku Ajar Algoritma dan Struktur Data*. Jambi : PT. Sonpedia Publishing Indonesia.

Firmansyah, H., Julian, E., & Ruhiat, A. (2025). Analisis Perbandingan Algoritma Linear Search dan Binary Search dalam Efisiensi Pencarian Data. *Infoman's : Jurnal Ilmu-Ilmu Informatika Dan Manajemen*, *19*(2), 1–7. https://ejournal.unsap.ac.id/index.php/infomans/article/view/2398

Izhari, F. (2025). *Algoritma dan Pemrograman*. Payakumbuh : Serasi Media Teknologi.

Leana, S. A., Ginting, M. A. P., Izdihar, M. B., Pratama, T. S. A., Manik, D. A. A., & Gunawan, I. (2025). Perbandingan Efisiensi Linear dan Binary Search Dalam Pencarian Nama Siswa Pada Struktur Data Array. *JRSIKOM : Jurnal Riset Sistem Informasi Dan Aplikasi Komputer*, *1*(2), 45–50. https://doi.org/10.180997/jrsikom.v1i2.41

Mevia, N. A. O., Marbun, Y. K., Putri, M. D., & Sitompul, Y. R. (2026). Perbandingan Algoritma Divide and Conquer dan Searching pada Pengolahan Data Nilai Mahasiswa Berbasis Web. *TEKNIK : Jurnal Ilmu Teknik Dan Informatika*, *6*(1), 60–79. https://doi.org/10.51903/teknik.v1i1.1145

Puranik, T. A. (2025). Performance Analysis of Sorting and Searching Algorithms. *IRJAEM : International Research Journal on Advanced Engineering and Management*, *3*(8), 2741–2746. https://doi.org/10.47392/IRJAEM.2025.0430

Purbasari, W., Iqbal, T., Inayah, I., Munawir, M., Sutjiningtyas, S., Hikmawati, E., Natsir, F., Widhiyanti, A. A. S., Wall, M., & Haris, M. S. (2024). *Algoritma Pemrograman*. Sukabumi : CV Haura Utama.

Purnama, N. (2025). Comparatrive Performance Study of Search Algorithms on Large-Scale Data Sstructures. *JITK : Jurnal Ilmu Pengetahuan Dan Teknologi Komputer*, *11*(1), 99–109. https://doi.org/10.33480/jitk.v11i1.6592

Putra, R. A. (2024). *Buku Sakti Pemrograman C++ Untuk Pemula*. Yogyakarta : Anak Hebat Indonesia.

Putra, R. F., Zebua, R. S. Y., Budiman, B., Rahayu, P. W., Bangsa, M. T. A., Zulfadhilah, M., Choirina, P., Wahyudi, F., & Andiyan, A. (2023). *Data Mining: Algoritma dan Penerapannya*. Jambi : PT. Sonpedia Publishing Indonesia.

Sitorus, Z., Renyaan, A. S., Si, S., Kmurawak, R. M. B., & Lokollo, P. D. (2024). *Tinjauan mendalam tentang ilmu komputer: konsep dasar, algoritma, dan perkembangan terkini: buku referensi*. Medan : PT Media Penerbit Indonesia.

Situmorang, H. (2017). Analisa algoritma pada metoda pencarian linier, biner dan interpolasi. *Jurnal Mahajana Informasi*, *2*(2), 31–41. https://doi.org/10.51544/jurnalmi.v2i2.177

Suryadi, D., Octiva, C. S., Fajri, T. I., Nuryanto, U. W., & Hakim, M. L. (2024). Optimasi Kinerja Sistem IoT Menggunakan Teknik Edge Computing.

Jurnal Minfo Polgan, *13*(2), 1456–1461. https://doi.org/10.33395/jmp.v13i2.14102

Yusuf, A. D., Abdullahi, S., Boukar, M. M., & Yusuf, S. I. (2021). Collision Resolution Techniques in Hash Table: A Review. *IJACSA : International Journal of Advanced Computer Science and Applications*, *12*(9), 757–762. https://doi.org/10.14569/IJACSA.2021.0120984

KLASIFIKASI KEMATANGAN BUAH TOMAT BERBASIS CNN DENGAN ARSITEKTUR MOBILENETV2 DAN *TRANSFER LEARNING*

Mochamad Fathur Milzam Ramadhan[1)], Ahmad Syukri Gozali[2)], Sumanto[3)], Jefina Tri Kumalasari[4)], Ghofar Taufiq[5)], Ade Christian[6)]

Prodi Informatika, Fakultas Teknik & Informatika, Universitas Bina Sarana Informatika

Correspondence author: MFM Ramadhan, 15230498@bsi.ac.id, Bekasi, Indonesia

Abstract

Classifying tomato ripeness is a crucial aspect of post-harvest processing and agricultural distribution. Manual ripeness determination is subjective and potentially inconsistencies. This study aims to classify tomato ripeness using a deep learning approach based on a Convolutional Neural Network (CNN) with the MobileNetV2 architecture and transfer learning techniques. The dataset, consisting of tomato images, was classified into three classes: unripe, ripe, and overripe. Preprocessing included image resizing and normalization before model training. The MobileNetV2 model was used as a feature extractor, while a fully connected layer was applied for classification. Test results showed that the model was able to classify tomato ripeness with good performance based on classification evaluation metrics. Overall, the proposed approach shows good potential for RGB image-based tomato ripeness classification.

Keyword *: classification algorithm, ripeness, tomato, CNN, MobileNetV2*

Abstrak

Klasifikasi tingkat kematangan buah tomat merupakan aspek penting dalam proses pascapanen dan distribusi hasil pertanian. Penentuan kematangan secara manual masih bersifat subjektif dan berpotensi menimbulkan ketidakkonsistenan. Penelitian ini bertujuan untuk mengklasifikasikan tingkat kematangan buah tomat menggunakan pendekatan *deep learning* berbasis *Convolutional Neural Network* (CNN) dengan arsitektur MobileNetV2 dan teknik *transfer learning*. Dataset berupa citra buah tomat diklasifikasikan ke dalam tiga kelas, yaitu tidak matang, matang, dan terlalu matang. Tahapan *preprocessing* meliputi proses *resizing* dan normalisasi citra sebelum dilakukan pelatihan model. Model MobileNetV2 digunakan sebagai *feature extractor*, sedangkan lapisan *fully connected* diterapkan untuk proses klasifikasi. Hasil pengujian menunjukkan bahwa model mampu mengklasifikasikan tingkat kematangan buah tomat dengan performa yang baik berdasarkan metrik evaluasi klasifikasi. Secara keseluruhan, pendekatan yang diusulkan menunjukkan potensi yang baik untuk klasifikasi tingkat kematangan buah tomat berbasis citra RGB.

Kata Kunci : algoritma klasifikasi, tingkat kematangan, buah tomat, cnn, mobilnetv2

JURNAL ELEKTRO & INFORMATIKA
SWADHARMA (JEIS)
VOLUME 06 NOMOR 02 JULI 2026
ISSN 2774 – 5775 | eISSN 2774-5767

A. PENDAHULUAN

Buah tomat merupakan salah satu komoditas hortikultura yang memiliki tingkat produksi dan konsumsi yang tinggi, baik untuk kebutuhan rumah tangga maupun industri pangan. Tomat digunakan dalam berbagai produk olahan, seperti saus, jus, dan bahan masakan, sehingga kualitas buah tomat menjadi faktor penting dalam menjaga kepuasan konsumen. Salah satu indikator utama kualitas buah tomat adalah tingkat kematangannya, karena tingkat kematangan berpengaruh terhadap rasa, tekstur, warna, serta daya simpan buah selama proses distribusi dan penyimpanan (Santoso et al., 2024).

Dalam praktik pascapanen, penentuan tingkat kematangan buah tomat masih banyak dilakukan secara manual melalui pengamatan visual oleh manusia. Penilaian tersebut umumnya didasarkan pada perubahan warna kulit buah dari hijau menuju merah. Meskipun metode ini mudah diterapkan, penilaian manual bersifat subjektif dan bergantung pada pengalaman pengamat, sehingga berpotensi menimbulkan ketidakkonsistenan hasil, terutama ketika dilakukan dalam skala besar atau oleh banyak pekerja dengan tingkat pengalaman yang berbeda (Ayunda et al., 2023).

Seiring dengan perkembangan teknologi, pengolahan citra digital menjadi salah satu pendekatan yang banyak digunakan untuk membantu proses klasifikasi objek secara otomatis. Citra digital mampu merepresentasikan karakteristik visual suatu objek dalam bentuk data numerik, termasuk warna, tekstur, dan bentuk. Dalam konteks buah tomat, informasi warna pada citra RGB menjadi parameter penting karena perubahan warna merupakan ciri utama dalam proses pematangan buah. Oleh karena itu, pemanfaatan citra RGB banyak diterapkan dalam penelitian klasifikasi tingkat kematangan buah (Santoso et al., 2024).

Pendekatan berbasis *machine learning* telah digunakan dalam berbagai penelitian untuk mengklasifikasikan tingkat kematangan buah tomat (Choirunisa & Supatman, 2025; Fajar et al., 2023; Husna & Fatah, 2026). Namun, metode konvensional umumnya memerlukan proses ekstraksi fitur secara manual, seperti perhitungan nilai warna atau tekstur tertentu, yang membutuhkan perancangan fitur secara khusus dan bergantung pada pengetahuan domain. Keterbatasan tersebut mendorong penggunaan pendekatan *deep learning* yang mampu melakukan ekstraksi fitur secara otomatis dari data citra (Samual & Wijanarko, 2026).

Convolutional Neural Network (CNN) merupakan salah satu metode *deep learning* yang paling banyak digunakan dalam klasifikasi citra. CNN bekerja dengan memanfaatkan lapisan konvolusi untuk mengekstraksi fitur visual secara bertingkat, sehingga mampu mengenali pola kompleks pada citra digital. Berbagai penelitian menunjukkan bahwa CNN memberikan performa yang baik dalam tugas klasifikasi citra di bidang pertanian, termasuk identifikasi jenis tanaman, deteksi penyakit tanaman, dan klasifikasi tingkat kematangan buah (Mardianto et al., 2024).

Meskipun CNN mampu menghasilkan performa yang baik, penggunaan arsitektur CNN konvensional dengan jumlah parameter yang besar sering kali memerlukan sumber daya komputasi yang tinggi dan waktu pelatihan yang relatif lama. Hal ini menjadi kendala apabila sistem akan diterapkan pada perangkat dengan keterbatasan sumber daya atau pada aplikasi yang membutuhkan efisiensi. Untuk mengatasi permasalahan tersebut, pendekatan *transfer learning* menjadi solusi yang banyak digunakan dalam pengembangan model klasifikasi citra (Saparudin et al., 2025).

Transfer learning memungkinkan pemanfaatan model CNN yang telah dilatih sebelumnya pada dataset berskala besar, sehingga proses pelatihan dapat dilakukan dengan lebih cepat dan stabil. Salah satu arsitektur CNN yang dirancang untuk

efisiensi adalah MobileNetV2. Arsitektur ini menggunakan *depthwise separable convolution* dan *inverted residual blocks* yang mampu mengurangi jumlah parameter serta beban komputasi tanpa mengurangi kemampuan model dalam mengekstraksi fitur penting dari citra (Rumbekwan & Aryanto, 2025).

Tujuan dari penelitian ini adalah untuk mengembangkan sistem klasifikasi tingkat kematangan buah tomat berbasis citra RGB menggunakan pendekatan *deep learning*. Secara khusus, penelitian ini bertujuan untuk menerapkan arsitektur MobileNetV2 dengan teknik *transfer learning* sebagai metode klasifikasi, serta mengevaluasi kinerja model dalam mengklasifikasikan buah tomat ke dalam tiga kelas tingkat kematangan, yaitu tidak matang, matang, dan terlalu matang.

Keterbaruan dari penelitian ini terletak pada penerapan arsitektur MobileNetV2 berbasis *transfer learning* sebagai metode klasifikasi tingkat kematangan buah tomat menggunakan citra RGB. Pemanfaatan arsitektur CNN yang ringan memungkinkan proses pelatihan model yang lebih efisien dengan jumlah parameter yang relatif lebih sedikit, tanpa mengorbankan performa klasifikasi. Selain itu, penelitian ini menerapkan pembagian data secara otomatis dan terstruktur untuk proses pelatihan, validasi, dan pengujian, sehingga evaluasi kinerja model dapat dilakukan secara lebih objektif dan terukur. Pendekatan yang digunakan dalam penelitian ini diharapkan dapat memberikan alternatif solusi yang efisien untuk sistem klasifikasi kematangan buah tomat berbasis citra digital.

B. METODE PENELITIAN

Penelitian ini menerapkan pendekatan eksperimental menggunakan pengolahan citra digital dan *deep learning* untuk klasifikasi tingkat kematangan buah tomat. Metode utama yang digunakan adalah *Convolutional Neural Network* (CNN) dengan arsitektur MobileNetV2 melalui pendekatan Transfer Learning. Penggunaan arsitektur ini dipilih karena efisiensinya dalam mengekstraksi fitur visual secara otomatis serta kemampuannya dalam mengenali pola tekstur yang kompleks dengan sumber daya komputasi yang efisien.

Dataset Penelitian

Dataset yang digunakan dalam penelitian ini terdiri dari 821 citra buah tomat yang terbagi ke dalam tiga kelas tingkat kematangan, yaitu tidak matang, matang, dan terlalu matang. Seluruh citra dikumpulkan dalam satu direktori utama sebelum dilakukan pembagian data secara otomatis. Dataset kemudian dibagi menjadi data latih, data validasi, dan data uji dengan rasio 8:1:1, sehingga diperoleh sekitar 80% data untuk pelatihan, serta masing-masing 10% data untuk validasi dan pengujian. Pembagian ini bertujuan untuk memastikan bahwa model dilatih menggunakan data yang cukup, sekaligus dievaluasi menggunakan data uji yang benar-benar terpisah guna mengukur kemampuan generalisasi model secara objektif. Seluruh citra menggunakan ruang warna RGB sebagai representasi standar yang umum digunakan dalam analisis berbasis warna (Santoso et al., 2024).

Berdasarkan pembagian tersebut, data uji terdiri dari sekitar 80 citra tomat yang digunakan secara khusus untuk mengevaluasi performa model setelah proses pelatihan selesai. Penggunaan data uji yang terpisah memungkinkan pengukuran kinerja model secara lebih objektif terhadap citra yang belum pernah dilihat sebelumnya. Rincian jumlah data yang digunakan adalah sebagai berikut:

Kategori Data	Matang	Terlalu Matang	Tidak Matang	Total
Data Latih (Training)	344	86	226	656
Data Validasi (Validation)	43	11	28	82
Data Uji (Testing)	36	9	40	85
Total Keseluruhan	**423**	**106**	**294**	**823**

Gambar 1. Rincian dataset

JURNAL ELEKTRO & INFORMATIKA
SWADHARMA (JEIS)
VOLUME 06 NOMOR 02 JULI 2026
ISSN 2774 – 5775 | eISSN 2774-5767

Pembagian ini memastikan bahwa model diuji pada data yang benar-benar independen (data uji) untuk mengukur kemampuan generalisasi model terhadap data yang belum pernah dilihat sebelumnya.

```
dataset/
    test/
        matang/
        terlalu_matang/
        tidak_matang/
    train/
        matang/
        terlalu_matang/
        tidak_matang/
    valid/
        matang/
        terlalu_matang/
        tidak_matang/
```

Gambar 2. Struktur dataset

Praproses data

Pada tahap praproses, seluruh citra diubah ukurannya menjadi 224 × 224 piksel agar memiliki dimensi yang seragam sebagai masukan model CNN. Selanjutnya, nilai intensitas piksel dinormalisasi ke rentang [0,1] untuk membantu proses pelatihan model menjadi lebih stabil dan mempercepat konvergensi.

Selain itu, dilakukan data augmentation pada data latih untuk meningkatkan variasi data dan mengurangi risiko *overfitting*. Teknik augmentasi ini bertujuan agar model tidak hanya bergantung pada pola tertentu dari dataset asli, sehingga mampu mengenali variasi tampilan buah tomat yang lebih beragam. Pendekatan ini umum diterapkan dalam pelatihan model CNN berbasis citra (Rosalina et al., 2025).

Arsitektur Model CNN (MobileNetV2)

Berbeda dengan model CNN konvensional yang membangun lapisan dari awal, penelitian ini menggunakan Transfer Learning pada arsitektur MobileNetV2. Model ini memanfaatkan pre-trained weights dari dataset ImageNet untuk mengenali fitur dasar citra. Arsitektur MobileNetV2 menggunakan struktur Inverted Residuals dan Linear Bottlenecks yang memungkinkan ekstraksi fitur tekstur secara lebih mendalam.

Penerapan metode ini merupakan solusi atas kendala klasifikasi pada fase kematangan yang memiliki kemiripan visual tinggi (matang dan terlalu matang), yang sering menjadi kelemahan pada penelitian terdahulu. Pada bagian akhir, lapisan *Global Average Pooling* digunakan untuk mereduksi dimensi, diikuti oleh lapisan *Dropout* (0.3) untuk mencegah *overfitting*, dan lapisan *Fully Connected* dengan fungsi aktivasi Softmax untuk menghasilkan probabilitas klasifikasi tiga kelas.

Pelatihan Model

Proses pelatihan model dilakukan menggunakan algoritma optimasi Adam dengan fungsi kerugian *categorical cross-entropy*, yang sesuai untuk permasalahan klasifikasi multikelas. Model dilatih menggunakan data latih selama 10 *epoch*, sementara data validasi digunakan untuk memantau performa model selama proses pelatihan dan mencegah terjadinya *overfitting*.

Parameter pelatihan dipilih dengan mempertimbangkan keseimbangan antara performa model dan efisiensi komputasi, mengingat MobileNetV2 dirancang untuk menghasilkan performa yang baik dengan beban komputasi yang relatif ringan.

Evaluasi Model

Evaluasi kinerja model dilakukan menggunakan data uji yang tidak terlibat dalam proses pelatihan. Kinerja model diukur menggunakan beberapa metrik evaluasi, yaitu akurasi, *precision*, *recall*, dan *F1-score*, serta *confusion matrix* untuk mengetahui kemampuan model dalam mengklasifikasikan masing-masing kelas tingkat kematangan buah tomat secara lebih rinci (Ayunda et al., 2023).

Alur Penelitian

Secara keseluruhan, alur penelitian ini dirancang secara sistematis guna memastikan akurasi dan validitas hasil klasifikasi. Tahapan dimulai dari pengumpulan dataset citra tomat melalui seleksi ulang, dilanjutkan dengan tahap praproses data yang meliputi

resizing dan normalisasi. Tahap berikutnya adalah perancangan model menggunakan arsitektur MobileNetV2 dengan pendekatan Transfer Learning, yang dilatih menggunakan data latih dan dipantau melalui data validasi. Akhirnya, dilakukan evaluasi mendalam pada data uji independen menggunakan metrik akurasi, *precision*, *recall*, *F1-score*, dan *confusion matrix*. Seluruh tahapan divisualisasikan dalam diagram alur untuk memberikan pemahaman terperinci mengenai proses penelitian.

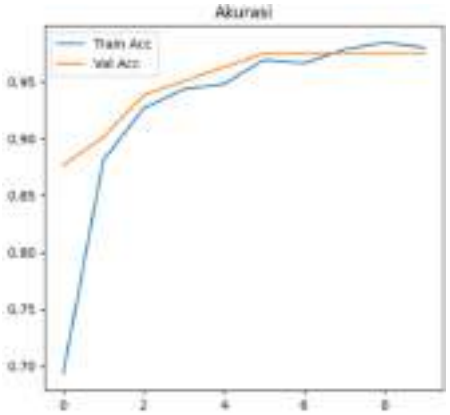
Gambar 3. Diagram Alur Penelitian

C. HASIL DAN PEMBAHASAN

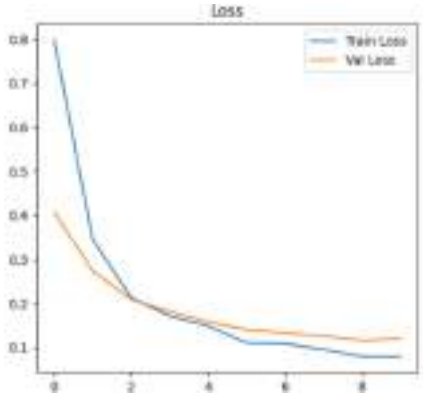
Pelatihan Model

Pelatihan model dilakukan menggunakan arsitektur MobileNetV2 dengan pendekatan *transfer learning* pada data latih dan data validasi. Selama proses pelatihan, performa model dipantau menggunakan metrik akurasi dan nilai *loss* untuk mengamati kemampuan model dalam mempelajari pola visual dari citra buah tomat. Hasil pelatihan menunjukkan bahwa model mampu mencapai kondisi pembelajaran yang stabil, ditandai dengan peningkatan akurasi yang konsisten serta penurunan nilai *loss* pada data latih dan validasi.

Grafik hubungan antara jumlah *epoch* terhadap nilai akurasi menunjukkan bahwa model secara bertahap mampu meningkatkan ketepatan klasifikasi seiring bertambahnya proses pembelajaran. Sementara itu, grafik *loss* memperlihatkan kecenderungan penurunan nilai kesalahan prediksi, yang mengindikasikan bahwa model berhasil menyesuaikan bobot-bobot jaringan untuk meminimalkan kesalahan klasifikasi. Pola ini menunjukkan bahwa proses pelatihan berjalan secara optimal dan tidak menunjukkan gejala *overfitting* yang signifikan. Hasil ini sejalan dengan penelitian sebelumnya yang menyatakan bahwa penggunaan *transfer learning* pada arsitektur CNN dapat meningkatkan stabilitas dan efisiensi pelatihan model, khususnya pada dataset citra pertanian (Mutahar et al., 2024).



Gambar 4. Grafik *Accuracy vs Epoch*



Gambar 5. Grafik *Loss vs Epoch*

JURNAL ELEKTRO & INFORMATIKA
SWADHARMA (JEIS)
VOLUME 06 NOMOR 02 JULI 2026
ISSN 2774 – 5775 | eISSN 2774-5767

Training *Epoch*

Analisis proses pelatihan berdasarkan *epoch* dilakukan untuk memahami dinamika pembelajaran model CNN secara lebih mendalam. Setiap *epoch* merepresentasikan satu siklus pembelajaran penuh terhadap seluruh data latih, di mana parameter model diperbarui untuk memperkecil kesalahan prediksi. Pada beberapa *epoch* awal, peningkatan akurasi terjadi cukup signifikan, yang menunjukkan bahwa model mulai mengenali fitur-fitur visual utama dari citra tomat, terutama perbedaan warna dan tekstur pada masing-masing tingkat kematangan.

Seiring bertambahnya jumlah *epoch*, peningkatan akurasi menjadi lebih lambat dan cenderung stabil. Kondisi ini mengindikasikan bahwa model telah mencapai fase konvergensi, di mana penambahan *epoch* tidak lagi memberikan peningkatan performa yang berarti. Pemilihan jumlah *epoch* yang digunakan dalam penelitian ini dinilai cukup untuk mencapai keseimbangan antara performa model dan efisiensi waktu pelatihan. Pendekatan ini sesuai dengan praktik umum dalam pelatihan CNN berbasis *transfer learning*, di mana jumlah *epoch* yang terlalu besar berpotensi menyebabkan pembelajaran yang tidak efisien (Sholihin & Sunyoto, 2025).

Pemilihan jumlah *epoch* sebanyak 10 dilakukan untuk memastikan bahwa model memiliki waktu pelatihan yang cukup tanpa menyebabkan *overfitting*. Berdasarkan hasil pengamatan, jumlah *epoch* tersebut dinilai mampu menghasilkan performa yang optimal dengan waktu pelatihan yang efisien.

Gambar 6. Training *epoch*

Evaluasi Model

Setelah proses pelatihan selesai, model dievaluasi menggunakan data uji yang tidak terlibat dalam proses pelatihan maupun validasi. Evaluasi ini bertujuan untuk mengukur kemampuan generalisasi model dalam mengklasifikasikan citra tomat yang baru. Hasil evaluasi menunjukkan bahwa model MobileNetV2 mampu menghasilkan performa klasifikasi yang baik pada data uji, dengan nilai akurasi, presisi, *recall*, dan *F1-score* yang konsisten.

Evaluasi model dilakukan menggunakan data uji yang terdiri dari sekitar 80 citra tomat, sehingga hasil yang diperoleh mencerminkan kemampuan model dalam mengklasifikasikan data baru secara konsisten.

Konsistensi performa antara data latih, validasi, dan data uji menunjukkan bahwa model memiliki kemampuan generalisasi yang baik dan tidak hanya menghafal pola pada data pelatihan. Hal ini menegaskan bahwa penggunaan *transfer learning* pada arsitektur MobileNetV2 mampu meningkatkan efektivitas klasifikasi citra, sebagaimana juga dilaporkan dalam penelitian klasifikasi citra pertanian sebelumnya (Appe et al., 2023; Mutahar et al., 2024).

	precision	recall	f1-score	support
matang	0.95	1.00	0.97	36
terlalu_matang	1.00	0.89	0.94	9
tidak_matang	1.00	0.97	0.99	40
accuracy			0.98	85
macro avg	0.98	0.95	0.97	85
weighted avg	0.98	0.98	0.98	85

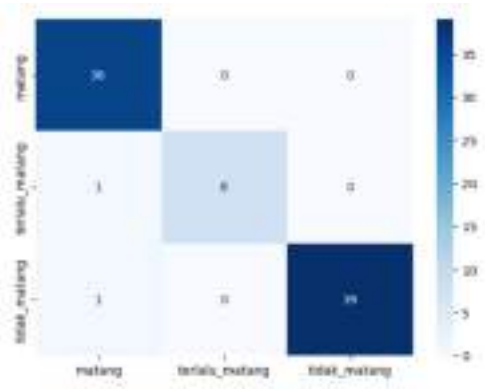
Gambar 7. Evaluasi Model

Confusion matrix

Untuk memperoleh gambaran performa klasifikasi yang lebih rinci pada setiap kelas, digunakan *confusion matrix*. *Confusion matrix* memberikan informasi mengenai jumlah citra yang berhasil diklasifikasikan dengan benar serta citra yang mengalami kesalahan klasifikasi pada masing-masing kelas tingkat kematangan. Berdasarkan hasil

confusion matrix, model menunjukkan kemampuan klasifikasi yang baik pada ketiga kelas kematangan tomat.

Sebagian besar citra berhasil diprediksi sesuai dengan kelas aslinya, yang menunjukkan bahwa fitur visual yang diekstraksi oleh MobileNetV2 cukup representatif dalam membedakan tingkat kematangan buah tomat. Penggunaan *confusion matrix* sebagai alat evaluasi dinilai penting karena mampu mengungkap pola kesalahan klasifikasi secara spesifik pada tiap kelas, sehingga memberikan pemahaman yang lebih mendalam terhadap performa model (Appe et al., 2023).



Gambar 8. *Confusion matrix*

Hasil Klasifikasi

Pengujian model juga dilakukan secara visual dengan memasukkan citra tomat secara individual untuk melihat hasil prediksi kelas kematangan. Hasil pengujian menunjukkan bahwa model mampu mengklasifikasikan citra tomat ke dalam kelas tidak matang, matang, dan terlalu matang sesuai dengan karakteristik visual masing-masing kelas. Model memberikan hasil prediksi disertai tingkat keyakinan (*confidence score*), yang menunjukkan probabilitas keanggotaan citra terhadap kelas tertentu.

Contoh hasil klasifikasi menunjukkan bahwa model mampu mengenali perbedaan warna dan tekstur tomat secara efektif. Hal ini menegaskan bahwa kombinasi fitur warna RGB dan ekstraksi fitur otomatis oleh CNN berperan penting dalam menentukan tingkat kematangan buah tomat. Temuan ini sejalan dengan penelitian sebelumnya yang menyatakan bahwa CNN efektif dalam memanfaatkan informasi visual untuk klasifikasi tingkat kematangan buah (Ayunda et al., 2023)

Gambar 9. Contoh hasil tidak matang

Gambar 10. Contoh hasil matang

Gambar 11. Contoh hasil terlalu matang

Pembahasan

Hasil penelitian menunjukkan bahwa model yang dikembangkan mampu membedakan seluruh kelas tingkat kematangan buah tomat dengan baik, termasuk pada kelas matang dan terlalu matang yang memiliki karakteristik visual yang relatif berdekatan. Penggunaan arsitektur MobileNetV2 berbasis *transfer learning* memungkinkan model untuk mengekstraksi fitur citra secara lebih mendalam, tidak hanya berdasarkan perbedaan warna permukaan, tetapi juga variasi tekstur dan distribusi intensitas piksel yang muncul pada masing-masing tingkat kematangan.

Hal ini tercermin dari hasil evaluasi model yang menunjukkan tingkat ketepatan klasifikasi yang stabil pada seluruh kelas. Dengan demikian, pendekatan yang digunakan dalam penelitian ini mampu meminimalkan ambiguitas visual antar kelas kematangan tomat, sehingga menghasilkan sistem klasifikasi yang andal dan konsisten untuk digunakan pada data citra tomat yang beragam.

Secara keseluruhan, hasil yang diperoleh menunjukkan bahwa model CNN berbasis MobileNetV2 berbasis *transfer learning* yang digunakan memiliki performa yang baik dan berpotensi untuk diterapkan dalam sistem otomatis penentuan tingkat kematangan buah tomat berbasis pengolahan citra digital.

Penyimpanan Model CNN

Setelah seluruh tahapan pelatihan dan evaluasi selesai, model CNN yang telah dilatih disimpan dalam bentuk file model. Penyimpanan model bertujuan untuk memungkinkan penggunaan kembali model tanpa perlu melakukan proses pelatihan ulang, sehingga dapat digunakan secara langsung pada tahap pengujian lanjutan maupun implementasi sistem klasifikasi tingkat kematangan buah tomat. Langkah ini penting dalam pengembangan sistem berbasis *deep learning* karena meningkatkan efisiensi penggunaan model dalam aplikasi nyata.

D. PENUTUP

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa penerapan metode *Convolutional Neural Network* (CNN) dengan arsitektur MobileNetV2 berbasis *transfer learning* mampu mengklasifikasikan tingkat kematangan buah tomat secara efektif dan konsisten. Seluruh tahapan penelitian, mulai dari pengolahan dataset, praproses citra, pelatihan model, hingga evaluasi, menunjukkan bahwa model yang dibangun mampu mengenali perbedaan karakteristik visual pada masing-masing tingkat kematangan buah tomat dengan baik.

Hasil evaluasi model menunjukkan performa klasifikasi yang stabil pada ketiga kelas, yaitu tidak matang, matang, dan terlalu matang. Model mampu mengekstraksi fitur citra secara optimal sehingga proses klasifikasi berjalan dengan lancar tanpa indikasi kesalahan klasifikasi yang signifikan antar kelas. Hal ini menunjukkan bahwa arsitektur MobileNetV2 memiliki kemampuan yang baik dalam memanfaatkan informasi warna dan tekstur citra tomat untuk mendukung proses pengambilan keputusan klasifikasi.

Selain itu, penggunaan pendekatan *transfer learning* terbukti meningkatkan efisiensi pelatihan model serta menghasilkan performa yang konsisten pada data uji. Dengan demikian, sistem klasifikasi tingkat kematangan buah tomat yang dikembangkan dalam penelitian ini dinilai andal dan berpotensi untuk diterapkan pada sistem otomatis berbasis pengolahan citra digital di bidang pertanian dan pascapanen.

Meskipun penelitian ini telah menunjukkan hasil yang baik, terdapat beberapa hal yang dapat dikembangkan pada penelitian selanjutnya. Pertama, jumlah dan variasi dataset citra tomat dapat diperluas, khususnya dengan mempertimbangkan variasi kondisi pencahayaan dan sudut pengambilan gambar, agar model memiliki kemampuan generalisasi yang lebih tinggi.

Kedua, penelitian selanjutnya dapat mengeksplorasi teknik *fine-tuning* pada lapisan tertentu dari MobileNetV2 untuk melihat potensi peningkatan performa klasifikasi. Selain itu, pengembangan sistem klasifikasi dalam bentuk aplikasi atau sistem berbasis perangkat bergerak dapat menjadi langkah lanjutan untuk mendukung implementasi model secara langsung di lapangan.

E. DAFTAR PUSTAKA

Appe, S. R. N., Arulselvi, G., & Balaji, G. N. (2023). Tomato Ripeness Detection and Classification using VGG based CNN Models. *IJISAE : International Journal of Intelligent Systems and Applications in Engineering*, *11*(1), 296–302. https://www.ijisae.org/index.php/IJISAE/article/view/2538

Ayunda, N. A., Haryatmi, E., & Riyadi, T. A. (2023). Classification of Tomato Ripeness Based on Convolutional Neural Network Methods. *Journal of Information Systems and Informatics*, *5*(4), 1658–1675. https://doi.org/10.51519/journalisi.v5i4.613

Choirunisa, S., & Supatman. (2025). Klasifikasi Tingkat Kematangan Citra Buah Tomat Menggunakan Densenet-121. *JITET : Jurnal Informatika Dan Teknik Elektro Terapan*, *13*(3), 1998–2003. https://doi.org/10.23960/jitet.v13i3.6652

Fajar, M., Sulthan, M. B., & Wahyudi, I. (2023). Klasifikasi Tingkat Kematangan Buah Tomat Menggunakan Fitur Rgb Dan Hsi Berbasis Backpropagation. *JATIM : Jurnal Aplikasi Teknologi Informasi Dan Manajemen*, *4*(1), 84–95. https://doi.org/10.31102/jatim.v4i1.2177

Husna, M., & Fatah, Z. (2026). Klasifikasi Kematangan Tanaman Buah Tomat Menggunakan Teachable Machine: Pendekatan Berbasis Gambar. *JAMASTIKA : Jurnal Mahasiswa Teknik Infiormatika*, *5*(1), 92–97. https://doi.org/10.35473/jamastika.v5i1.4599

Mardianto, Y., Dewi, T., & Risma, P. (2024). Analisis Klasifikasi Kematangan Buah Tomat dengan Pendekatan Transfer Learning Model EfficientNet. *Techno Bahari*, *11*(1), 20–25. https://doi.org/10.52234/tb.v11i1.306

Mutahar, M., Kannan, S., Jafer, M. M., & Ragavendra K, M. (2024). Deep Learning-Based Tomato Ripeness Detection: A ResNet-152 Approach. *IJSRST : International Journal of Scientific Research in Science and Technology*, *11*(1), 34–41. https://doi.org/10.32628/IJSRST5241113

Rosalina, Husodo, A. Y., & Wijaya, I. G. P. S. (2025). Development of a Convolutional Neural Network Method for Classifying Ripeness Levels of Servo Variety Tomatoes. *JUTIF : Jurnal Teknik Informatika*, *6*(2), 501–520. https://doi.org/10.52436/1.jutif.2025.6.2.4168

Rumbekwan, J. I. W., & Aryanto, J. (2025). Application of Convolutional Neural Networks for Tomato Fruit Ripeness Identification. *JSRET : Journal of Scientific Research, Education, and Technology*, *4*(1), 61–78. https://doi.org/10.58526/jsret.v4i1.645

Samual, M. A., & Wijanarko, S. (2026). Comparative Analysis of CNN and Random Forest for Cashew Plant Disease Classification. *Sistemasi: Jurnal Sistem Informasi*, *15*(3), 961–970. https://doi.org/10.32520/stmsi.v15i3.6100

Santoso, K. A., Kamsyakawuni, A., & Izza, S. V. R. (2024). Comparison Classification Of Tomatoes Ripeness

Based On RGB, HSV And CMYK Colors Based On Correlation Coefficient. *Journal of Computers and Digital Business*, *3*(3), 112–120. https://doi.org/10.56427/jcbd.v3i3.410

Saparudin, H., Ramdi, M. A., & Alfarizi, F. S. (2025). Implementasi Convolutional Neural Network dan ResNet-18 untuk Klasifikasi Kerusakan Jalan Berbasis Citra. *JAPTIKA : Jurnal Aplikasi Informatika Dan Multimedia*, *1*(1), 27–31. https://doi.org/10.64878/japtika.v1i1.53

Sholihin, I., & Sunyoto, A. (2025). Improving Tomato Ripeness Classification Using Knowledge Distillation and Hyperparameter Optimization with Optuna. *Jeecom : Journal of Electrical Engineering and Computer*, *7*(2), 282–290. https://doi.org/10.33650/jeecom.v7i2.11266

PENERAPAN *MACHINE LEARNING* UNTUK ANALISIS PREDIKSI HARGA RUMAH MENGGUNAKAN ALGORITMA REGRESI LINIER DAN *RANDOM FOREST*

Muhamad Fadli Fadhlullah[1)], Zayyan Nauval Araf[2)], Sumanto[3)], Jefina Tri Kumalasari[4)], Ghofar Taufiq[5)], Ade Christian[6)]

Prodi Informatika, Fakultas Teknik & Informatika, Universitas Bina Sarana Informatika

Correspondence author: MF Fadhlullah, 15230914@bsi.ac.id, Bekasi, Indonesia

Abstract

Accurate house price prediction is crucial for property buyers, sellers, investors, and developers in making decisions. This study aims to compare the performance of Linear Regression and Random Forest algorithms in predicting house prices in East Jakarta. The dataset used was taken from the rumah123.com website, and after cleaning, there were 180 unique data points. The modelling process was carried out using the Orange application, where 80% of the data was used to train the model and 20% to test it. To assess model performance, several metrics were used, such as Mean Squared Error, Root Mean Squared Error, Mean Absolute Error, Mean Absolute Percentage Error, and the coefficient of determination. The test results showed that the Linear Regression model provided more accurate predictions than Random Forest, as evidenced by lower RMSE, MAE, and MAPE values, as well as an R^2 value of 0.997. This indicates that the relationship between property features and house prices in this dataset tends to be linear. However, the Random Forest model also performed well, with an R^2 value of 0.996, so it can still be used for datasets with more complex characteristics.

Keyword *: machine learning, prediction, house prices, linear regression, random forest*

Abstrak

Prediksi harga rumah yang tepat sangat penting untuk para pembeli, penjual, investor, dan pengembang properti dalam membuat keputusan. Penelitian ini bertujuan untuk membandingkan seberapa baik algoritma Regresi Linier dan *Random Forest* dalam memprediksi harga rumah di Jakarta Timur. *Dataset* yang digunakan diambil dari situs rumah123.com dan setelah dibersihkan, terdapat 180 data yang unik. Proses pemodelan dilakukan dengan menggunakan aplikasi Orange, di mana 80% data digunakan untuk melatih model dan 20% untuk mengujinya. Untuk menilai kinerja model, digunakan beberapa metrik seperti *Mean Squared Error*, *Root Mean Squared Error (RSME)*, *Mean Absolute Error (MAE)*, *Mean Absolute Percentage Error (MAPE)*, dan koefisien determinasi (R^2). Hasil dari pengujian menunjukkan bahwa model Regresi Linier memberikan prediksi yang lebih akurat dibandingkan dengan *Random Forest*, hal ini terlihat dari nilai RMSE, MAE, dan MAPE yang lebih rendah serta nilai R^2 sebesar 0,997.

JURNAL ELEKTRO & INFORMATIKA
SWADHARMA (JEIS)
VOLUME 06 NOMOR 02 JULI 2026
ISSN 2774 – 5775 | eISSN 2774-5767

Ini menunjukkan bahwa hubungan antara fitur-fitur properti dan harga rumah di *dataset* ini cenderung linier. Namun, model *Random Forest* juga menunjukkan hasil yang baik dengan nilai R^2 sebesar 0,996, jadi masih tetap bisa digunakan untuk *dataset* yang memiliki karakteristik lebih rumit.

Kata Kunci : *machine learning*, prediksi, harga rumah, regresi linier, *random forest*

A. PENDAHULUAN

Rumah merupakan salah satu kebutuhan pokok manusia yang sangat penting sebagai tempat tinggal dan juga sebagai kekayaan dalam kehidupan sosial dan ekonomi. Seiring bertambahnya jumlah orang dan pergeseran ke kota, dunia properti mengalami perkembangan yang cepat dan sekarang telah berubah menjadi salah satu cara berinvestasi yang menjanjikan. Dalam hal ini, menentukan harga rumah menjadi hal yang sangat penting yang akan memengaruhi keputusan dari pembeli, penjual, investor, dan pengembang. Jika harga rumah tidak sesuai atau tidak wajar, bisa mengakibatkan kerugian uang dan membuat pasar properti menjadi tidak stabil. Maka, kemampuan untuk meramalkan harga rumah dengan tepat berdasarkan berbagai faktor seperti lokasi, ukuran rumah, dan jumlah kamar sangat diperlukan.

Perkembangan teknologi, terutama di bidang Pembelajaran Mesin (*Machine learning*), telah membuka cara baru untuk menganalisis dan memprediksi harga barang yang rumit seperti rumah. Di antara berbagai algoritma Pembelajaran Mesin, Regresi Linier dan Hutan Acak (*Random Forest*) adalah dua metode yang menarik untuk dipelajari (Adetunji et al., 2022; Salsabila et al., 2026). Regresi Linier sering dipakai sebagai model dasar karena mudah untuk digunakan dalam menggambarkan hubungan antara variabel dan memahami hasilnya (Hallan & Fajri, 2025; Muhtajin & Fatah, 2026). Namun, data properti di dunia nyata sering kali lebih rumit dan memerlukan metode yang lebih canggih. Algoritma Hutan Acak, sebagai teknik pembelajaran gabungan, telah menunjukkan kemampuan untuk memberikan hasil yang lebih akurat dengan menangkap hubungan yang tidak sederhana dan mengurangi risiko kesalahan yang berlebihan (Lestari & Astuti, 2022). Penelitian di Jakarta Selatan, contohnya, menunjukkan bahwa metode *Random Forest* lebih baik daripada Regresi Linier dengan nilai R^2 mencapai 0,942 (Fitri, 2023).

Prediksi harga rumah yang tepat sangat penting bagi banyak orang. Untuk pembeli dan penjual, jika mereka tahu harga yang benar, itu bisa menghindari kerugian uang dan membantu dalam bernegosiasi. Bagi pengembang properti, seperti yang diteliti di Pekanbaru, menentukan harga yang sesuai adalah kunci dalam proyek rumah yang mereka kerjakan (Hardiansyah et al., 2025). Untuk investor, memiliki prediksi yang akurat penting untuk menilai risiko dan potensi keuntungan.

Meskipun sudah banyak penelitian dilakukan, masih ada kebutuhan untuk membandingkan langsung dan secara sistematis antara metode dasar seperti Regresi Linier dan metode gabungan seperti *Random Forest*, khususnya untuk data properti di Indonesia. Beberapa penelitian menunjukkan bahwa *Random Forest* memiliki keunggulan. Sebuah penelitian yang dilakukan (Fitri, 2023) di Jakarta Selatan menunjukkan bahwa algoritma *Random Forest* bisa mencapai akurasi sebesar 75,10%. Luas tanah dan luas bangunan menjadi faktor yang paling berpengaruh dalam menentukan harga rumah. Penelitian lain juga menunjukkan bahwa *Random Forest* lebih baik daripada model pohon keputusan tunggal, terlihat dari nilai R^2

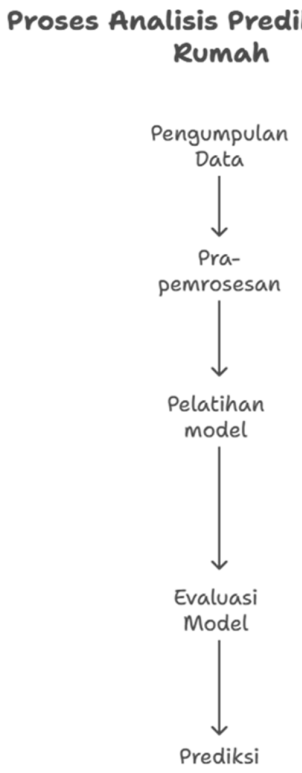
yang jauh lebih tinggi (0,91 dibandingkan 0,78) dan MSE yang lebih rendah. Namun, ada juga penelitian yang menunjukkan Regresi Linier masih bisa memberikan hasil yang baik, seperti penelitian di Tebet yang mencetak R^2 sebesar 0,7713 (Pratama et al., 2025). Kesenjangan ini menunjukkan perlunya penelitian lebih lanjut untuk mendapatkan bukti yang lebih jelas tentang kinerja kedua metode ini.

Penelitian ini didasarkan pada alasan bahwa pemilihan algoritma untuk prediksi harus berdasarkan bukti nyata dan pemahaman tentang karakteristik setiap model. Regresi Linier dipilih sebagai acuan dasar karena kemampuannya untuk menjelaskan hubungan linier dengan jelas (Hardiansyah et al., 2025; Nuris, 2024). Model ini sangat bagus untuk mengetahui dan mengukur pengaruh masing-masing fitur terhadap harga rumah, seperti yang terlihat dalam penelitian di Ames, Amerika Serikat, di mana faktor kualitas bangunan secara umum (OverallQual) dan luas ruang hidup (GrLivArea) memiliki hubungan yang signifikan (Hallan & Fajri, 2025). Kemudahan dalam memahami ini memberikan gambaran dasar tentang faktor-faktor yang paling berpengaruh pada harga.

Di sisi lain, *Random Forest* dipilih karena merupakan model yang lebih kuat dan fleksibel. Cara kerja algoritma ini adalah dengan membuat banyak pohon keputusan dan mengumpulkan hasilnya. Gabungan hasil- hasil ini membuatnya bisa mengenali pola yang rumit dan tidak lurus, sesuatu yang sering tidak bisa ditangkap oleh Regresi Linier, serta lebih tahan terhadap data yang punya nilai ekstrim atau gangguan. Penelitian yang dilakukan oleh (Hardiansyah et al., 2025) di Pekanbaru menunjukkan bahwa *Random Forest* (R^2 93,74) jauh lebih baik dibandingkan Regresi Linier (R^2 74,42), meskipun kedua metode tersebut masih tidak sebaik *Gradient Boosted Trees*. Dalam penelitian ini, penulis tidak hanya mencari model yang paling akurat, tapi juga melihat keseimbangan antara kemudahan pemahaman (Regresi Linier) dan kekuatan prediksi (*Random Forest*).

B. METODE PENELITIAN

Metode penelitian yang dilakukan dalam urutan tertentu terdiri dari pengumpulan data, pre-pemrosesan, pelatihan model, evaluasi model, dan prediksi. Langkah-langkah ini dapat dilihat pada gambar di bawah.



Gambar 1. Tahap Penelitian

Pengumpulan Data

Data yang dijadikan dasar utama untuk penelitian ini diambil dari situs properti terkenal di Indonesia, yaitu rumah123.com. Memanfaatkan data dari portal online ini adalah cara yang biasa dilakukan dan efektif dalam penelitian properti saat ini, sama seperti yang dilakukan dalam studi di daerah Tebet (Pratama et al., 2025). Data tersebut mencakup daftar harga rumah yang dijual di Jakarta Selatan. Setelah mengumpulkan informasi, kita berhasil membuat *dataset* awal yang terdiri dari 689 entri.

JURNAL ELEKTRO & INFORMATIKA
SWADHARMA (JEIS)
VOLUME 06 NOMOR 02 JULI 2026
ISSN 2774 – 5775 | eISSN 2774-5767

Pra-Pemrosesan

Data lalu dibersihkan agar kualitas dan keasliannya terjaga sebelum dilakukan pemodelan. Pada langkah ini sangat penting untuk menemukan dan menghapus data yang sama, di mana 689 data awal ternyata hanya terdiri dari 180 data. Kumpulan data akhir yang dipakai adalah 180 data unik itu untuk menghindari model dari *overfitting*.

Pelatihan Model

Pemodelan data latihan menggunakan algoritma *Random Forest* dan Regresi Linier untuk memperkirakan harga rumah untuk memberikan rekomendasi rumah berdasarkan hasil perkiraan. Pembagian data dilakukan dengan 80% untuk data pelatihan dan 20% untuk data pengujian. Cara ini dipakai untuk melatih model dan menilai seberapa baik kinerjanya dengan data yang belum pernah dimiliki sebelumnya. *Random Forest* adalah salah satu teknik yang dipakai untuk mengelompokkan data dengan membuat banyak pohon keputusan. Teknik ini bisa meningkatkan akurasi hasil, dengan cara membuat cabang untuk setiap node (simpul yang lebih tinggi) dan memilihnya secara acak (Adetunji et al., 2022). Regresi Linier adalah salah satu metode yang digunakan untuk memprediksi nilai dengan membuat garis lurus. Metode ini dapat memberikan perkiraan yang tepat, dengan cara mencari garis yang paling cocok untuk menggambarkan titik-titik data yang ada (Hallan & Fajri, 2025).

Evaluasi Model

Tahap penilaian akan mengukur seberapa tepat data dibandingkan dengan hasil ramalan dan saran yang diberikan menggunakan data pengujian dalam setiap model. Suatu model *Machine learning* dianggap bagus jika dapat memberikan hasil yang tepat untuk membantu membuat keputusan berdasarkan informasi yang diberikan. Pengukuran yang akan dilakukan menggunakan metrik *Mean Squared Error* (MSE), R-squared (R^2), *Mean Absolute Error* (MAE), *Root Mean Squared Error* (RMSE), atau *Mean Absolute Percentage Error* (MAPE) (Salsabila et al., 2026).

Prediksi

Setelah kita menilai dan mengerti bagaimana kinerja model, model yang paling akurat bisa dipakai untuk memperkirakan harga rumah pada data baru atau properti yang belum pernah kita lihat sebelumnya.

C. HASIL DAN PEMBAHASAN

Pengumpulan Data

Data yang menjadi dasar utama penelitian ini diperoleh dari situs properti terkenal di Indonesia, yaitu rumah123.com. Menggunakan sumber data sekunder dari website adalah cara yang biasa dan berhasil dalam penelitian properti saat ini (Qolbi et al., 2025). Penelitian ini bertujuan untuk mengumpulkan data tentang daftar rumah yang dijual di daerah Jakarta Timur.

Proses pengumpulan informasi berhasil membuat sebuah kumpulan data awal yang berisi 689 entri. Variabel-variabel yang diperoleh meliputi hal-hal penting seperti Harga, Luas Tanah, Luas Bangunan, Jumlah Kamar Tidur, Jumlah Kamar Mandi, dan ada tidaknya Garasi.

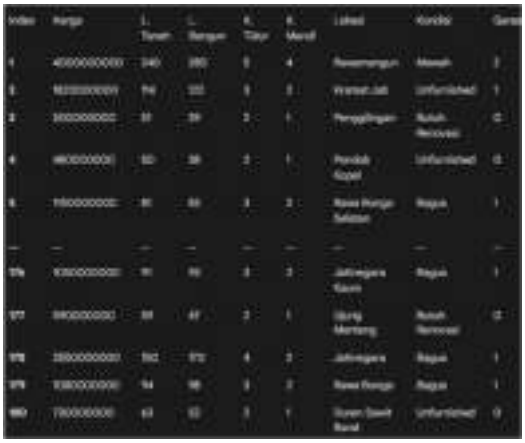
Index	Harga	L. Tanah	L. Bangun	K. Tidur	K. Mandi	Lokasi	Kondisi	Garasi
1	625000000	60	45	2	1	Cakung	Unfurnished	0
2	1850000000	120	110	3	1	Rawamangun	Bagus	1
3	3500000000	200	280	4	3	Cibubur	Mewah	2
4	950000000	72	54	2	1	Duren Sawit	Butuh Renovasi	0
5	2400000000	150	170	4	2	Pasar Rebo	Semi Furnished	1
...	...	...	...	...	...	...	...	...
684	500000000	51	39	2	1	Penggilingan	Butuh Renovasi	0
685	2200000000	135	150	4	2	Pasar Rebo Baru	Unfurnished	1
686	1420000000	100	108	3	2	Cipinang Melayu	Bagus	1
687	730000000	63	52	2	1	Duren Sawit Ba...	Unfurnished	0
688	4800000000	290	340	6	5	Cibubur Estate	Mewah	3
689	1250000000	87	89	3	2	Klender Baru	Semi Furnished	1

Gambar 2. *Dataset* awal sebelum pra-pemrosesan

Pra-Pemrosesan

Dataset yang awalnya ada kemudian melalui proses pembersihan untuk memastikan bahwa data tersebut baik dan benar. Dari proses ini, ditemukan hal yang penting: ada banyak data yang sama. Penyelidikan lebih lanjut menunjukkan bahwa *dataset* yang awalnya berisi 689 entri sebenarnya hanya memiliki 180 entri unik.

Adanya data yang sama ini bisa menyebabkan masalah besar pada model *Machine Learning*, karena model akan lebih suka mengingat data training daripada memahami pola yang penting. Hal ini bisa mengakibatkan hasil penilaian yang tidak tepat. Untuk menjaga agar penelitian tetap valid, semua data yang sama dihapus. *Dataset* akhir yang bersih dan hanya memiliki 180 entri unik inilah yang dipakai untuk semua tahap analisis dan pemodelan berikutnya.



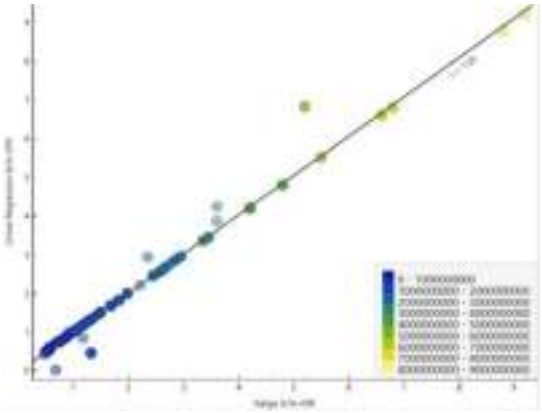
Gambar 3. Data yang sudah dibersihkan

Pelatihan Model

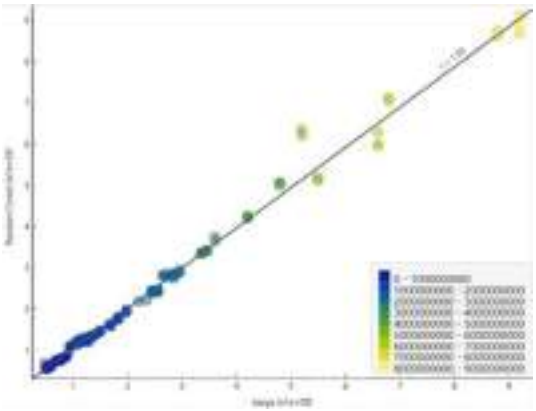
Dataset yang telah dibersihkan kemudian dibagi menjadi dua bagian dengan proporsi 80:20, sehingga kita mendapatkan 80 data untuk latihan dan 20 data untuk pengujian. Pembagian ini bertujuan untuk melatih model menggunakan satu kelompok data dan mengujinya menggunakan kelompok data yang berbeda yang belum pernah dilihat sebelumnya. Dari data latih ini, dua model prediksi dibangun dan dilatih. Model pertama adalah Regresi Linier, yang digunakan untuk melihat bagaimana variabel prediktor berhubungan dengan variabel target yang berupa harga. Model kedua adalah *Random Forest*, yaitu model yang menggabungkan beberapa pohon keputusan yang dilatih dengan data yang sama untuk menghasilkan prediksi yang lebih tepat.

Evaluasi Model

Model dinilai dengan menggunakan cara yang tepat untuk memeriksa sejauh mana ketepatan dan kinerjanya, yaitu Squared Error (MSE), *R-squared* (R^2), *Mean Absolute Error* (MAE), *Root Mean Squared Error* (RMSE), atau *Mean Absolute Percentage Error* (MAPE).



Gambar 4. *Scatter plot* Regresi Linier



Gambar 5. *Scatter plot Random Forest*

Penilaian ini dilakukan pada data uji agar bisa tahu seberapa baik model memprediksi harga rumah menggunakan data yang baru. Hasil perbandingan kemampuan antara model Regresi Linier dan *Random Forest* ditampilkan pada tabel di bawah ini.

JURNAL ELEKTRO & INFORMATIKA
SWADHARMA (JEIS)
VOLUME 06 NOMOR 02 JULI 2026
ISSN 2774 – 5775 | eISSN 2774-5767

Tabel 1. Perbandingan kinerja model regresi linier dan *Random Forest*

Model	MSE	RMSE	MAE	MAPE (%)	R^2
Regresi Linier	53.336.715.567.994.536	23.094.743,313	5.347.931,632	3,106	0,991
Random Forest	31.939.208.707.152.064	17.871.540,595	9.482.831,669	4,224	0,995

Analisis Perbandingan Kinerja Model

Hasil evaluasi menunjukkan bahwa Regresi Linier menghasilkan kesalahan prediksi yang lebih sedikit dibandingkan dengan *Random Forest* menurut ukuran RMSE, MAE, dan MAPE. Ini berarti bahwa model Regresi Linier bisa memahami dan meramalkan dengan baik menggunakan data yang diuji. Perbedaan nilai R^2 antara kedua model tergolong kecil, yang menunjukkan bahwa keduanya dapat menjelaskan perubahan harga rumah dengan baik.

Harus diingat bahwa kinerja model sangat dipengaruhi oleh sifat *dataset* yang digunakan. *Dataset* dalam penelitian ini memiliki 180 data unik dengan ciri numerik yang cukup sederhana, sehingga hubungan antara variabel input dan harga rumah lebih cenderung linier. Hal ini membuat Regresi Linier lebih efektif dibandingkan dengan model yang menggunakan beberapa metode seperti *Random Forest*.

Keunggulan Regresi Linier dalam penelitian ini muncul karena beberapa alasan utama. Pertama, jumlah data yang tidak terlalu banyak membuat model yang sederhana lebih stabil dan tidak gampang untuk menjadi terlalu cocok dengan data. Kedua, fitur-fitur yang digunakan, seperti ukuran tanah, ukuran bangunan, jumlah kamar tidur, dan jumlah kamar mandi, punya hubungan yang cukup kuat dan lurus dengan harga rumah. Ketiga, Regresi Linier memiliki tingkat kesulitan model yang lebih rendah, sehingga bisa memberikan hasil prediksi yang lebih konsisten pada data yang diuji.

Di sisi lain, *Random Forest*, yang lebih rumit, butuh lebih banyak data dan variasi fitur yang lebih rumit agar kelebihannya bisa terlihat dengan jelas. Dalam *dataset* yang terbatas dan terstruktur, manfaat tersebut belum dapat digunakan dengan sebaiknya.

Prediksi

Berdasarkan hasil evaluasi model yang telah dilakukan, Regresi Linier dipilih sebagai model utama untuk memprediksi harga rumah karena memiliki kinerja terbaik dalam hal RMSE, MAE, MAPE, dan nilai R^2. Model ini kemudian dipakai untuk memprediksi harga rumah pada uji dan data baru yang memiliki sifat yang mirip.

Pemilihan Regresi Linier sebagai model prediksi didasarkan pada kemampuannya untuk secara konsisten menghubungkan fitur-fitur properti dengan harga rumah dengan cara yang linier. Namun, *Random Forest* juga bisa digunakan sebagai pilihan lain untuk eksplorasi atau ketika bekerja dengan *dataset* yang lebih rumit dan tidak linier.

D. PENUTUP

Berdasarkan hasil tes menggunakan software Orange dengan pembagian data 80:20, model Regresi Linier ternyata lebih baik dalam memprediksi harga rumah di Jakarta Timur dibandingkan dengan *Random Forest*. Regresi Linier menunjukkan kesalahan yang lebih kecil jika dilihat dari metrik RMSE, MAE, dan MAPE, dan memiliki nilai koefisien determinasi (R^2) sebesar 0,997. Ini berarti bahwa ada hubungan yang cenderung lurus antara karakteristik properti dan harga rumah pada data yang digunakan, sehingga Regresi Linier bisa digunakan dengan baik.

Namun, model *Random Forest* masih mampu bersaing dengan nilai R^2 yang

mencapai 0,996. Ini menunjukkan bahwa *Random Forest* masih bisa menemukan pola yang tidak lurus dalam data, tetapi kelebihannya tidak begitu terlihat pada data yang cukup kecil dan terstruktur. Maka, Regresi Linier dianggap sebagai model yang paling cocok untuk memprediksi harga rumah dalam penelitian ini, sedangkan *Random Forest* dapat dipakai sebagai pilihan lain jika *dataset* lebih rumit.

E. DAFTAR PUSTAKA

Adetunji, A. B., Akande, O. N., Ajala, F. A., Oyewo, O., Akande, Y. F., & Oluwadara, G. (2022). House Price Prediction using Random Forest Machine Learning Technique. *Procedia Computer Science*, *199*, 806–813. https://doi.org/10.1016/j.procs.2022.01.100

Fitri, E. (2023). Analisis Perbandingan Metode Regresi Linier, Random Forest Regression dan Gradient Boosted Trees Regression Method untuk Prediksi Harga Rumah. *JACOST : Journal of Applied Computer Science and Technology*, *4*(1), 58–64. https://doi.org/10.52158/jacost.v4i1.491

Hallan, R. R., & Fajri, I. N. (2025). Prediksi Harga Rumah menggunakan Machine Learning Algoritma Regresi Linier. *JTeksis : Jurnal Teknologi Dan Sistem Informasi Bisnis*, *7*(1), 57–62. https://doi.org/10.47233/jteksis.v7i1.1732

Hardiansyah, H., Ananda, & Syahbana, Y. A. (2025). *Analisa Faktor-Faktor Prediksi Harga Rumah Menggunakan Metode Regresi* [Magister Terapan Teknik Komputer Politeknik Caltex Riau]. https://repository.lib.pcr.ac.id/id/eprint/2817/

Lestari, E. S., & Astuti, I. (2022). Penerapan Random Forest Regression Untuk Memprediksi Harga Jual Rumah Dan Cosine Similarity Untuk Rekomendasi Rumah Pada Provinsi Jawa Barat. *Jurnal Ilmiah FIFO*, *14*(2), 131–146. https://doi.org/10.22441/fifo.2022.v14i2.003

Muhtajin, M., & Fatah, Z. (2026). Prediksi Harga Rumah Menggunakan Algoritma Regresi Linier. *JAMASTIKA : Jurnal Mahasiswa Teknik Infiormatika*, *5*(1), 343–350. https://doi.org/10.35473/jamastika.v5i1.4703

Nuris, N. (2024). Analisis Prediksi Harga Rumah Pada Machine Learning Metode Regresi Linear. *Explore*, *14*(2), 108–112. https://doi.org/10.35200/ex.v14i2.123

Pratama, A., Maulana, A., & Saputra, R. A. (2025). Implementation of Linear Regression Algorithm for House Price Prediction in Tebet Area. *JIfo Tech : Journal of Information Technology*, *5*(1), 280–286. https://doi.org/10.46229/jifotech.v5i1.986

Qolbi, M. B. S., Puteh, T. N., Rivandi, R., & Rozikin, C. (2025). Prediksi Harga Rumah Di Jakarta Pusat Menggunakan Algoritma Machine Learning. *JIKB : Jurnal Ilmu Komputer Dan Bisnis*, *16*(1), 16–24. https://doi.org/10.47927/jikb.v16i1.840

Salsabila, M., Adiwijaya, R. P., & Octavia, L. N. (2026). Implementasi Machine Learning Untuk Memprediksi Harga Rumah Dengan Menggunakan Metode Decision Tree Regressor Dan Random Forest Regressor. *Setrum : Sistem Kendali-Tenaga-Elektronika-Telekomunikasi-Komputer*, *15*(1), 46–56. https://doi.org/10.62870/setrum.v14i1.31025

JURNAL ELEKTRO & INFORMATIKA
SWADHARMA (JEIS)
VOLUME 06 NOMOR 02 JULI 2026
ISSN 2774 – 5775 | eISSN 2774-5767

ANALISIS SENTIMEN TERHADAP AI DALAM PENDIDIKAN DI MEDIA SOSIAL MENGGUNAKAN METODE KUANTITATIF *LOGISTIC REGRESSION*

Anindita Novelia Putri[1)], Rizki Andika Prasetyo[2)], Chairul Ichwan[3)], Giantika Chrisnawati[4)]

Prodi Teknologi Informasi, Fakultas Teknik dan Informatika, Universitas Bina Sarana Informatika

Correspondence author: RA Presetyo, rizki.andika.prasetyo12@gmail.com, Jakarta, Indonesia

Abstract

The rapid growth of Artificial Intelligence (AI) has expanded its role in education and sparked diverse public reactions on social media. This study aims to analyse sentiment toward AI in education using comments from Tempo.com's posts on the TikTok platform. A quantitative approach using Logistic Regression was applied to classify sentiment into positive and negative categories. The dataset consisted of 1,247 comments, which underwent text preprocessing, data labelling, data sharing, and model training. The analysis results showed 948 comments were negative and 298 comments were positive. The Logistic Regression model achieved an accuracy of 83%; precision 0.66, recall 0.64, and F1-score 0.65 confirm the effectiveness of this method. The research findings indicate that public perception of AI in education tends to be negative, while also providing insights for educators and policymakers to respond more wisely to technological developments.

Keyword *: sentiment analysis, social media, artificial intelligence, logistic regression, education*

Abstrak

Pertumbuhan pesat *Artificial Intelligence* (AI) telah memperluas perannya dalam pendidikan dan memicu beragam reaksi publik di media sosial. Penelitian ini bertujuan menganalisis sentimen terhadap AI dalam pendidikan dengan menggunakan komentar dari unggahan Tempo.com di platform TikTok. Pendekatan kuantitatif dengan metode *Logistic Regression* diterapkan untuk mengklasifikasikan sentimen ke dalam kategori positif dan negatif. *Dataset* terdiri atas 1.247 komentar yang diproses melalui tahap *text preprocessing*, pelabelan data, pembagian data, dan pelatihan model. Hasil analisis menunjukkan 948 komentar bernuansa negatif dan 298 komentar bernuansa positif. Model *Logistic Regression* mencapai akurasi sebesar 83%, dengan nilai *precision, recall,* dan *F1-score* yang menegaskan efektivitas metode ini. Temuan penelitian mengindikasikan bahwa persepsi publik terhadap AI dalam pendidikan cenderung negatif, sekaligus memberikan wawasan bagi pendidik dan pembuat kebijakan untuk merespons perkembangan teknologi secara lebih bijaksana.

Kata Kunci : analisis sentimen, media sosial, *artificial intelligence*, *logistic regression*, pendidikan

A. PENDAHULUAN

Kecerdasan buatan (*Artificial Intelligence*) atau yang sering disebut AI merupakan bidang dalam ilmu komputer yang berfokus pada pengembangan sistem dan perangkat yang dapat menjalankan fungsi – fungsi yang umumnya membutuhkan kecerdasan manusia (Eriana & Zein, 2023). Saat ini, penerapan AI telah meluas ke berbagai bidang seperti bisnis, sains, seni, dan pendidikan guna memperbaiki pengalaman pengguna serta meningkatkan efisiensi (Inzaghi et al., 2024).

Kecerdasan buatan merupakan teknologi yang tertanam dalam berbagai perangkat seperti ponsel, komputer, laptop, maupun situs web, yang mampu mempelajari dan memahami masukan dari pengguna secara terus menerus guna memberikan hasil atau jawaban yang optimal (Pratama & Hendry, 2024). Dalam konteks pendidikan, AI dapat digunakan untuk mendukung pembelajaran dengan sistem pembelajaran adaptif, tutor virtual, hingga otomatisasi penilaian (Oktavianus et al., 2023).

Namun implementasi AI dalam pendidikan juga menimbulkan beragam respon masyarakat, terutama dimedia sosial. Penggunaan AI dalam pendidikan membawa manfaat bagi perubahan sistem pendidikan di Indonesia, namun tetap disertai dengan konsekuensi negatif (Saepudin et al., 2024). Saat ini media sosial memiliki peran penting dalam kehidupan masyarakat terutama dalam hal pertukaran dan pencarian informasi yang dinilai lebih cepat dan efektif (Damayanti et al., 2023). Media Sosial, menjadi wadah aspirasi publik yang aktif memberi opini terhadap fenomena ini. Komentar pada unggahan bertema AI dalam pendidikan mencerminkan spektrum sentimen masyarakat dari antusias hingga penolakan. Analisis sentimen bertujuan untuk memperoleh masukan mengenai pengalaman yang telah dirasakan oleh pengguna atau konsumen terhadap produk atau jasa tersebut (Purnamasari et al., 2023). Penelitian (Safitri et al., 2025) menunjukan bahwa penggunaan *Logistic Regression* pada analisis sentimen terbukti efektif.

Penelitian terdahulu telah menunjukan efektivitas pendekatan menggunakan *Logistic Regression*. Ardiansah dkk menggunakan Support Vector Machine untuk menganalisis komentar tentang AI dalam pendidikan di platform X (Ardiansah et al., 2024). Pada penelitian dari Saputra dkk, menunjukan bahwa *Logistic Regression* dapat digunakan untuk menganalisis opini publik (Saputra et al., 2025).

Regresi logistik merupakan metode yang digunakan untuk mengklasifikasikan data dari kumpulan *dataset* berdaparkan nilai nilai dari fitur input yang bertujuan untuk memprediksi variabel dependen dengan menggunakan satu atau lebih variabel independen sebagai dasar prediksi hasil (Fahmuddin S et al., 2023). Budi lestari dan hutagalung menunjukan efektivitas penggunaan TF-IDF dan *Logistic Regression* dalam analisis sentimen ulasan pengguna marketplace indonesia dengan akurasi lebih dari 90% (Lestari & Hutagalung, 2025). Sihaloho dan Napitupulu juga meneliti pentingnya pemahaman publik terhadap AI dalam pendidikan dalam bijaknya berteknologi (Sihaloho & Napitupulu, 2024).

Berdasarkan latar belakang yang ada, penelitian ini dilakukan dengan tujuan untuk memahami Persepsi publik terhadap penggunaan AI dalam pendidikan dengan pendekatan ilmiah yang sistematis.

B. METODE PENELITIAN

Penelitian ini menggunakan pendekatan kuantitatif dengan metode analisis sentimen berbasis *Logistic Regression*. Adapun proses dan langkah penelitian ini sebagai berikut:

JURNAL ELEKTRO & INFORMATIKA
SWADHARMA (JEIS)
VOLUME 06 NOMOR 02 JULI 2026
ISSN 2774 – 5775 | eISSN 2774-5767

Gambar 1. Alur proses penelitian

Identifikasi Masalah

Tahap awal dimana fokus penelitian diterapkan, yaitu menganalisis sentimen masyarakat terhadap penggunaan kecerdasan buatan (AI) dalam pendidikan, yang tercermin dari opini masyarakat di media sosial.

Studi Pustaka

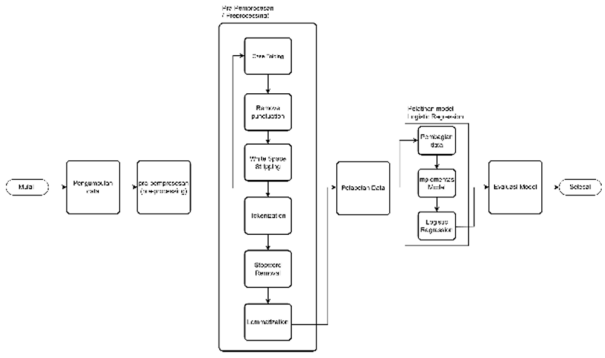
Pada tahap ini didapatkan pemahaman perihal teori maupun teknik yang digunakan untuk penelitian ini. Dengan melakukan peninjauan literatur yang berkaitan dengan topik penelitian. Studi Pustaka mencakup 12 referensi utama yang terdiri dari 8 artikel ilmiah yang diperoleh melalui Google scholar dan DOAJ *(Directory of Open Access Journals*), 2 buku akademik yang tersedia di perpustakaan Universitas Bina Sarana Informatika, dan 2 publikasi digital dari jurnal nasional terindeks SINTA. Literatur yang dikaji mencakup topik topik seperti konsep dasar kecerdasan buatan, penerapan AI dalam Pendidikan, teknik analisis sentiment, serta efektivitas algoritma *Logistic Regression* dalam klasifikasi data teks. Peninjauan ini menjadi landasan teoritis dalam merancang metode penelitian dan memilih pendekatan analisis yang sesuai.

Analisis Kebutuhan Data

Pada tahap ini ditentukan data yang akan dibutuhkan untuk dianalisis. Pada penelitian ini, menggunakan komentar pada salah satu postingan akun Tempo.com di media sosial Tiktok. Postingan komentar yang ada kemudian dipindahkan sebagai data ke dalam google sheet dengan format CSV

Metode Pengolahan dan Analisis Data

Metode pengolahan dan analisis data dalam penelitian ini *menggunakan Text Mining* dan *Machine Learning Logistic Regression* untuk mengklasifikasikan sentimen.



Gambar 2. Alur Metode Pengolahan dan Analisis Data

1. *Case Folding*
   Pada tahap ini seluruh data teks yang ada diubah menjadi huruf kecil untuk menjaga konsistensi data.
2. *Remove Punctuation*
   Pada Tahap ini semua tanda baca dihapus.
3. *White Space Stripping*
   Tahap ini memastikan teks untuk tidak memiliki spasi ganda agar data yang diproses menjadi seragam.
4. *Tokenization*
   Pada tahap ini teks diuraikan menjadi kata individu.
5. *Stopword Removal*
   Pada tahap ini kata penghubung yang tidak memiliki sifat sentimen dihapus, tahap ini dilakukan agar meningkatkan efisiensi dan efektivitas model.
6. *Lemmatization*
   Pada tahap ini teks yang telah diuraikan diubah menjadi kata dasar. Ini berguna untuk mengenali model yang berbeda menjadi sama.

Pelabelan Data

Pada tahap ini *dataset* yang ada dilabeli dengan positif (1) dan negatif (0) secara manual satu persatu agar pada saat melakukan pelatihan model *machine learning* menjadi lebih akurat.

Pelatihan Model *Logistic Regression*

Pada tahap ini model dilatih menggunakan data latih untuk mempelajari pola yang menghubungkan fitur teks dengan label sentimen. *Logistic Regression* dipilih karena mampu mengolah klasifikasi biner dengan interpretasi koefisien yang jelas. Pada tahap ini data dipisah menjadi 80 : 20.

Evaluasi Model

Setelah model dilatih, performanya diuji menggunakan data uji dengan *Confusion matrix* yaitu Akurasi, presisi, *recall*, dan *F1 Score* (Swaminathan & Tantri, 2024).

Tabel 1. *confusion matrix*

	Class 1 Predicted	Clas 2 Predicted
Class 1 Actual	TP	FN
Class 2 Actual	FP	TN

Class 1 = Positif, Class 2 = Negatif

Keterangan :

a. *True Positive* (TP) = Jumlah prediksi Positif yang benar
b. *False Negative* (FN) = Jumlah prediksi Negatif yang salah
c. *True Negative* (TN) = Jumlah prediksi Negatif yang benar
d. *False Positive* (FP) = Jumlah prediksi Positif yang salah
e. Akurasi

$$\text{Akurasi} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}}$$

f. Presisi

$$\text{Presisi} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

g. *Recall*

$$Recall = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

h. F1- Score

$$\text{F1} - \text{Score} = \frac{2 * Recall * \text{Presisi}}{Recall + \text{Presisi}}$$

Penarikan Kesimpulan

Berdasarkan hasil dari analisis maka dapat menjawab rumusan masalah mengenai pola sentimen masyarakat terhadap penggunaan AI dalam pendidikan serta efektivitas *Logistic Regression* dalam klasifikasi sentimen sebelumnya.

C. HASIL DAN PEMBAHASAN

Data proses analisis sentimen berasal dari 1247 komentar di media sosial Tiktok pada salah satu postingan akun Tempo.com. yang diunggah pada 5 Mei 2025. Komentar-komentar tersebut diproses menjadi 80 : 20 data pelatihan dan data pengujian. Setelah olah data, data yang awalnya 1247 menjadi 1206 data yaitu 948 sentimen negatif dan 298 sentimen positif.

Pengumpulan dan Pembersihan Data

Dataset yang dipakai dibuat pada 14 November 2025 dan berasal dari platform media sosial Tiktok pada salah satu postingan akun Tempo.com yang diunggah pada 5 Mei 2025. Data yang digunakan pada penelitian ini yaitu komentar komentar yang ada pada postingan tersebut, lalu dipindahkan ke dalam format *Comma Seperated Values (CVS)* dengan jumlah data yaitu 1247 komentar. Lalu data di *import* ke google colab untuk selanjutnya diolah.

Gambar 3. Tampilan Data dalam google colab

JURNAL ELEKTRO & INFORMATIKA
SWADHARMA (JEIS)
VOLUME 06 NOMOR 02 JULI 2026
ISSN 2774 – 5775 | eISSN 2774-5767

Text preprocessing

Setelah melakukan pengumpulan dan pembersihan data, selanjutnya akan dilakukan *Text Processing.*

1. *Case Folding*

Di tahap ini seluruh huruf kapital diubah menjadi huruf kecil atau lowercase.

Tabel 2. Case Folding

Sebelum	Sesudah
GA GA GA GA, GW GAMAU	ga ga ga ga, gw gamau
ini mah bener-bener emang mau membodohkan masyarakat tanpa berpikir kritis jirr ujung-ujungnya semua tuh bocah bocah bergantung sama AI	ini mah bener-bener emang mau membodohkan masyarakat tanpa berpikir kritis jirr ujung-ujungnya semua tuh bocah bocah bergantung sama ai
makin ancur aja nih sistem pendidikan 🙀🙀	makin ancur aja nih sistem pendidikan 🙀🙀

2. *Remove Punctuation*

Pada tahap ini, semua tanda baca kecuali huruf dari a hingga z akan dihapus agar hanya mengandung kata

Tabel 3. *Remove Punctuation*

Sebelum	Sesudah
ga ga ga ga, gw gamau	ga ga ga ga gw gamau
ini mah bener-bener emang mau membodohkan masyarakat tanpa berpikir kritis jirr ujung-ujungnya semua tuh bocah bocah bergantung sama ai	ini mah bener bener emang mau membodohkan masyarakat tanpa berpikir kritis jirr ujung ujungnya semua tuh bocah bocah bergantung sama ai
makin ancur aja nih sistem pendidikan 🙀🙀	makin ancur aja nih sistem pendidikan

3. *White Space Stripping*

Pada tahap ini spasi yang berlebih akan dihapus agar data atau teks yang ada menjadi lebih rapih dan tidak panjang berlebih.

Tabel 4. *White Stripping*

Sebelum	Sesudah
ga ga ga ga gw gamau	ga ga ga ga gw gamau
ini mah bener bener emang mau membodohkan masyarakat tanpa berpikir kritis jirr ujung ujungnya semua tuh bocah bocah bergantung sama ai	ini mah bener bener emang mau membodohkan masyarakat tanpa berpikir kritis jirr ujung ujungnya semua tuh bocah bocah bergantung sama ai
makin ancur aja nih sistem pendidikan	makin ancur aja nih sistem pendidikan

4. *Tokenization*

Pada tahap ini kalimat yang ada akan dibagi menjadi beberapa kata dalam bentuk token agar dapat dianalisis secara terpisah.

Tabel 5. *Tokenization*

Sebelum	Sesudah
ga ga ga ga gw gamau	['ga', 'ga', 'ga', 'ga', 'gw', 'gamau']
ini mah bener bener emang mau membodohkan masyarakat tanpa berpikir kritis jirr ujung ujungnya semua tuh bocah bocah bergantung sama ai	['ini', 'mah', 'bener', 'bener', 'emang', 'mau', 'membodohkan', 'masyarakat', 'tanpa', 'berpikir', 'kritis', 'jirr', 'ujung', 'ujungnya', 'semua', 'tuh', 'bocah', 'bocah', 'bergantung', 'sama', 'ai']
makin ancur aja nih sistem pendidikan	['makin', 'ancur', 'aja', 'nih', 'sistem', 'pendidikan']

5. *Stopwords Removal*

Pada tahap ini kata kata yang tidak mengandung sentimen akan dihapus untuk hasil analisis lebih akurat.

Tabel 6. *Stopwords Removal*

Sebelum	Sesudah
['ga', 'ga', 'ga', 'ga', 'gw', 'gamau']	['gamau']
['ini', 'mah', 'bener', 'bener', 'emang', 'mau', 'membodohkan', 'masyarakat', 'tanpa', 'berpikir', 'kritis', 'jirr', 'ujung', 'ujungnya', 'semua', 'tuh', 'bocah', 'bocah', 'bergantung', 'sama', 'ai']	['membodohkan', 'masyarakat', 'berpikir', 'kritis', 'ujungnya']

Sebelum	Sesudah
['makin', 'ancur', 'aja', 'nih', 'sistem', 'pendidikan']	['makin', 'ancur', 'sistem', 'pendidikan']

6. *Lemmatization*

Pada tahap ini, setiap kata yang telah diuraikan akan diubah bentuk menjadi kata dasar untuk menyerderhanakan kata agar lebih mudah dipahami dan dianalisis.

Tabel 7. *Lemmatization*

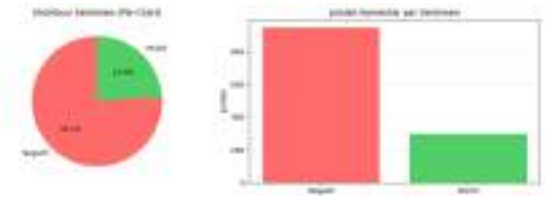
Sebelum	Sesudah
['gamau']	['gamau']
['membodohkan', 'masyarakat', 'berpikir', 'kritis', 'ujungnya']	['bodoh', 'masyarakat', 'pikir', 'kritis', 'ujung']
['makin', 'ancur', 'sistem', 'pendidikan']	['makin', 'ancur', 'sistem', 'didik']

Pelabelan Data

Setelah data disiapkan, data akan dilabeli untuk mengklasifikasikan sentimen. Terdapat dua kategori sentimen yaitu, positif (1) dan negatif (0).

Tabel 8. Pelabelan Data

Sebelum	Sesudah
['gamau']	0
['bodoh', 'masyarakat', 'pikir', 'kritis', 'ujung']	0
['makin', 'ancur', 'sistem', 'didik']	0



Gambar 4. Diagram distribusi sentimen

Dari hasil analisa data menunjukan bahwa terdapat 947 sentimen negatif (0) dan 298 sentimen positif (1).

Word Cloud

Setelah melakukan pelabelan di tahap ini akan menampilkan kata apa yang paling sering muncul dalam bentuk visual

Gambar 5. visualisasi kata yang sering muncul pada sentimen positif

Gambar 6. visualisasi kata yang sering muncul pada sentimen Negatif

Processing Data

Selanjutnya ada beberapa tahap yang akan melanjutkan pengolahan data yaitu

1. Split data

Pada tahap ini dilakukan pembagian data menjadi data pelatihan dan data pengujian dengan rasio 80 : 20.

Gambar 7. Hasil split data.

2. Implementasi Model

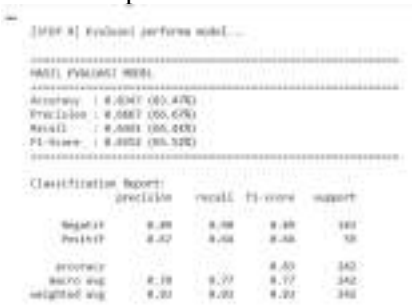
Pada tahap ini, menggunakan algoritma *Logistic Regression* dan TF-IDF untuk mengklasifikasikan sentimen.

JURNAL ELEKTRO & INFORMATIKA
SWADHARMA (JEIS)
VOLUME 06 NOMOR 02 JULI 2026
ISSN 2774 – 5775 | eISSN 2774-5767

Gambar 8. Hasil Implementasi Model

3. *Logistic Regression*

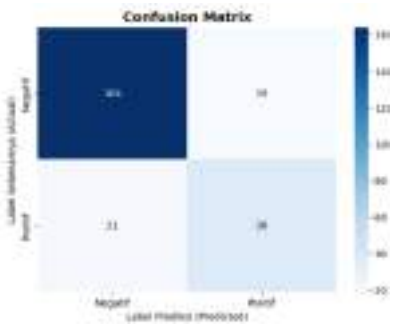
Pada penelitian ini dapat mengungkapkan bahwa penggunaan *Logistic Regression* pada analisis sentimen itu efektif karena dapat memiliki akurasi 83%. *Logistic Regression* juga mencatat bahwa *precision* 0,89 untuk sentimen negatif dan 0,67 untuk sentimen positif, *recall* 0,90 untuk sentimen negatif dan 0,64 untuk sentimen positif, *F1 – score* 0,89 untuk sentimen negatif dan 0,66 untuk sentimen positif.



Gambar 9. Hasil *Logistic Regression*.

Evaluasi Model

Pada tahap ini performa model yang sudah dilatih akan di ukur ataupun dianalisis untuk melihat apakah kinerjanya menghasilkan hasil yang tepat.



Gambar 10. Hasil *Confusion matrix*.

Dapat dilihat dari gambar 10 diatas bahwa distribuasi prediksi yang tepat dan yang salah dari model yang telah dianalisis. *Confusion matrix* diatas menunjukan bahwa dari total data yang diuji terdapat 164 prediksi yang tepat untuk sentimen Negatif dan 38 prediksi yang tepat pada sentimen Positif. Namun kita juga bisa melihat bahwa model juga membuat kesalahan prediksi, yaitu 21 prediksi yang salah untuk sentimen Negatif yang harusnya Positif dan 19 prediksi yang salah untuk sentimen Positif yang harusnya Negatif.

a. TP (*True Positive*) : 38
b. TN (*True Negative*) : 164
c. FP (*False Positive*) : 19
d. FN (*False Negative*): 21

Akurasi

$$\text{Akurasi} = \frac{38 + 164}{38 + 164 + 19 + 21} = 0{,}83$$

Presisi

$$\text{Presisi} = \frac{38}{38 + 19} = 0{,}66$$

Recall

$$Recall = \frac{38}{38 + 21} = 0{,}64$$

F1 – Score

$$\text{F1} - \text{Score} = \frac{2 * 0{,}64 * 0{,}66}{0{,}64 + 0{,}66} = 0{,}65$$

Setelah melaukan analisis data sentimen terhadap AI dalam pendidikan dan menerapkan Algoritma *Logistic Regression*, dapat disimpulkan beberapa temuan penting sebagai berikut:

1. Akurasi model Algoritma *Logistic Regression* yang dilatih menggunakan 80% data pelatihan dan 20% data pengujian yang mencapai akurasi 83%.
2. Mengevaluasi model menggunakan *Confusion matrix* dapat disimpulkan bahwa dapat mengklasifikasikan sentimen dengan baik. Dari model ini kita mengetahui bahwa presisi untuk

sentimen negatif adalah 0,89 dan 0,67 untuk sentimen positif, *recall* untuk sentimen negatif adalah 0,90 dan 0,64 untuk sentimen positif, F1 - Score untuk sentimen negatif adalah 0,89 dan 0,66 untuk sentimen positif. Nilai nilai ini menunjukan bahwa model dapat mengidentifikasikan sentimen dengan baik.

Dari analisis sentimen, ditemukan bahwa sentimen negatif berjumlah 948 dan sentimen positif berjumlah 298 yang jika dibandingkan maka lebih banyak sentimen negatif dari pada positif terhadap AI dalam Pendidikan.

D. PENUTUP

Penelitian ini berhasil menerapkan Algoritma *Logistic Regression* dalam melakukan analisis sentimen terhadap AI dalam Pendidikan di media sosial. Hasil pengujian menunjukan bahwa penggunaan Algoritma *Logistic Regression* dapat di sebut akurat dalam mengklasifikasikan komentar sentimen terhadap AI dalam pendidikan di media sosial. *Logistic Regression* dapat dibilang cukup terbukti efektif, mencapai akurasi sebesar 83%, presisi 66%, *recall* 64%, dan *F1 – Score* 65%. Namun begitu penelitian ini jauh dari kata sempurna dan mempunyai ruang untuk diperbaiki. Keterbatasan yang ada menunjukan bahwa masih ada kemungkinan untuk meningkatkan akurasi dan efektivitas dalam penggunaan Algoritma tersebut. Maka dari itu penelitian ini dapat dilanjutkan ataupun ditingkatkan agar dapat lebih baik maupun optimal.

E. DAFTAR PUSTAKA

Ardiansah, Y., Monalisa, S., & Muttakin, F. (2024). Analisis Sentimen Komentar Perplexity AI di X Tentang Pendidikan Menggunakan Support Vector Machine. *BITS : Building of Informatics, Technology and Science*, *6*(3), 2015–2023. https://doi.org/10.47065/bits.v6i3.6396

Damayanti, A., Delima, I. D., & Suseno, A. (2023). Pemanfaatan Media Sosial Sebagai Media Informasi dan Publikasi (Studi Deskriptif Kualitatif pada Akun Instagram @rumahkimkotatangerang). *PIKMA : Publikasi Ilmu Komunikasi Media Dan Cinema*, *6*(1), 173–190. https://doi.org/10.24076/pikma.v6i1.1308

Eriana, E. S., & Zein, A. (2023). *Artificial Intelligence (AI)*. Purbalingga: CV. Eureka Media Aksara.

Fahmuddin S, M., Aidid, M. K., & Nurliah, M. J. T. (2023). Implementasi Analisis Regresi Logistik Dengan Metode Machine Learning Untuk Mengklasifikasi Berita di Indonesia. *VARIANSI: Journal of Statistics and Its Application on Teaching and Research*, *5*(3), 155–162. https://doi.org/10.35580/variansiunm116

Inzaghi, R., Cahyani, A. D., Valenda, V., Ananda, M. A., & Pratama, D. (2024). Analisis Penerapan Artificial Intelligence (AI) di Berbagai Bidang. *JREIN : Jurnal Rekayasa Informatika*, *1*(1), 36–45. https://rekayasainformatika.com/index.php/JREIN/article/view/3

Lestari, V. B., & Hutagalung, C. A. (2025). Evaluation of TF-IDF Extraction Techniques in Sentiment Analysis of Indonesian-Language Marketplaces Using SVM, Logistic Regression, and Naive Bayes. *J-KOMA : Jurnal Ilmu Komputer Dan Aplikasi*, *8*(1), 36–44. https://doi.org/10.21009/j-koma.v8i1.05

Oktavianus, A. J. E., Naibaho, L., & Rantung, D. A. (2023). Pemanfaatan Artificial Intelligence pada Pembelajaran dan Asesmen di Era Digitalisasi. *Kridatama Sains Dan Teknologi*, *5*(2), 473–486. https://doi.org/10.53863/kst.v5i02.975

Pratama, A. D., & Hendry. (2024). Analisa Sentimen Masyarakat Terhadap Penggunaan ChatGPT Menggunakan Metode Support Vector Machine (SVM). *JIPI : Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika*, *9*(1), 327–338. https://doi.org/10.29100/jipi.v9i1.4285

Purnamasari, D., Aji, A. B., Wulandari A. P., D., Reza, F. A., Safrila O., M., Yanda, N., & Hidayati, U. (2023). *Pengantar Metode Analisis Sentimen*. Depok : Gunadarma Penerbit.

Saepudin, A., Aryanti, R., Fitriani, E., Royadi, R., & Ardiansyah, D. (2024). Analisis Sentimen Pemanfaatan Artificial Intelligence di Dunia Pendidikan Menggunakan SVM Berbasis Particle Swarm Optimization. *Computer Science (CO-SCIENCE)*, *4*(1), 71–79. https://doi.org/10.31294/coscience.v4i1.2921

Safitri, E., Syukrilla, W. A., & Fitriana, I. N. L. (2025). Logistic Regression for Sentiment Analysis of Insecurity Phenomena on Platform X. *J Statistika: Jurnal Ilmiah Teori Dan Aplikasi Statistika*, *18*(1), 948–956. https://doi.org/10.36456/jstat.vol18.no1.a10545

Saputra, A. J., Achmadi, S., & Auliasari, K. (2025). Penggunaan Metode Logistic Regression Untuk Analisis Sentimen Pembangunan Ibu Kota Nusantara Pada Media Sosial. *IJAI : Indonesian Journal of Applied Informatics*, *9*(2), 281–295. https://doi.org/10.20961/ijai.v9i2.95416

Sihaloho, F. A. S., & Napitupulu, Z. (2024). Use of Artificial Intelligence in Education in Indonesia: Literature Review. *REKOGNISI : Jurnal Pendidikan Dan Kependidikan*, *9*(1), 13–20. https://jurnal.unusu.ac.id/index.php/rekognisi/article/view/167

Swaminathan, S., & Tantri, B. R. (2024). *Confusion matrix*-Based Performance Evaluation Metrics. *African Journal of Biomedical Research*, *27*(4S), 4023–4031. https://doi.org/10.53555/AJBR.v27i4S.4345

ANALISIS SENTIMEN TERHADAP *CYBERBULLYING* DI MEDIA SOSIAL MENGGUNAKAN METODE KUANTITATIF *NAÏVE BAYES*

Siti Hani Fadilah[1)], Adiya Fanial Rizki[2)], Khairunnisa Regita Cahyani[3)], Giantika Chrisnawati[4)]

Prodi Teknologi Informasi, Fakultas Teknik dan Informatika, Universitas Bina Sarana Informatika

Correspondence author: AF Rizki, adityafr84@gmail.com, Jakarta, Indonesia

Abstract

TikTok is a digital interaction platform with a high user base. Cyberbullying is a technology-based form of bullying that can have serious impacts on the mental and social well-being of victims; therefore, understanding public perception of the issue is crucial. This study aims to identify public opinion trends regarding the current prevalence of cyberbullying. This study uses data from comments on a TikTok account and classifies these comments into positive, negative, and neutral sentiment classes. The method used is quantitative, with the naïve bayes algorithm, supported by text mining processes including data cleaning, tokenisation, stopword removal, stemming, and extraction using TF-IDF. The research dataset consisted of 1,100 comments, and after filtering, 779 comments were finalised. The test results show that naïve bayes is capable of providing good sentiment classification based on accuracy, precision, recall, and F1-score values. These findings are expected to raise awareness and minimise the occurrence of cyberbullying on social media.

Keyword *: sentiment analysis, cyberbullying, naïve bayes, social media, tiktok*

Abstrak

Media Sosial Tiktok menjadi salah satu ruang interaksi digital dengan tingkat pengguna yang tinggi. *Cyberbullying* merupakan perilaku perundungan berbasis teknologi yang dapat berdampak serius pada kondisi mental dan sosial korban, sehingga diperlukan pemahaman mengenai persepsi publik mengenai isu tersebut. Penelitian ini bertujuan untuk mengidentifikasi kecenderungan opini masyarakat terhadap kasus *cyberbullying* yang sedang marak terjadi. Penelitian ini menggunakan data dari komentar salah satu akun di TikTok, serta mengklasifikan komentar tersebut kedalam kelas sentiment positif, negatif, maupun netral. Metode yang digunakan adalah kuantitatif dengan algoritma *naïve bayes*, didukung oleh proses *text mining* mulai dari pembersihan data, tokenisasi, *stopword removal*, *stemming*, dan ekstraksi menggunakan TF-IDF. *Dataset* penelitian ini terdiri dari 1.100 komentar dan telah dilakukan penyaringan komentar yang berubah menjadi 779 komentar. Hasil pengujian menunjukan bahwa *naïve bayes* mampu memberikan klasifikasi sentimen dengan baik berdasarkan nilai akurasi, *precision*, *recall*, maupun *F1-score* yang diperoleh. Temuan ini diharapkan mampu meningkatkan kesadaran dan meminimalkan terjadinya *cyberbullying* di media sosial.

Kata Kunci : analisis sentimen, *cyberbullying*, *naïve bayes*, media sosial, tiktok

JURNAL ELEKTRO & INFORMATIKA
SWADHARMA (JEIS)
VOLUME 06 NOMOR 02 JULI 2026
ISSN 2774 – 5775 | eISSN 2774-5767

A. PENDAHULUAN

Perkembangan teknologi yang pesat membuat media sosial menjadi alat penting untuk mempermudah interaksi sehari-hari, termasuk komunikasi dan akses informasi. Berbagai platform populer di antaranya Youtube, Facebook, TikTok, Instagram, Linkedln, maupun Twitter memungkinkan setiap individu di dunia membangun jaringan sosialnya secara daring (Damayanti et al., 2023).

TikTok sebagai platform media sosial yang memungkinkan pembuatan video singkat dengan durasinya 15–60 detik dengan dukungan beragam fitur di antaranya stiker, musik, filter, maupun efek lainnya. Kehadiran aplikasi ini juga memberi peluang bagi praktik *cyberbullying*, di mana pengguna dapat mengunggah foto, video, atau tulisan dengan bahasa yang menyakiti, umumnya melibatkan individu lain dalam konten yang dibagikan (Aser et al., 2022).

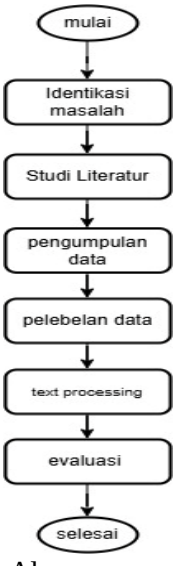
Cyberbullying sebagai perilaku pelecehan atau intimidasi yang memanfaatkan media elektronik atau teknologi komunikasi (Nugraha & Astuti, 2023). Dampaknya dari tindakan ini sifatnya serius bagi kesejahteraan maupun kesehatan mental korban, sekaligus menjadi tantangan yang kian mengemuka di era digital. Oleh karenanya, upaya pencegahan yang efektif perlu dilaksanakan pembuat kebijakan, orang tua serta pendidik (Marlef et al., 2024). Dalam memahami pandangan publik secara sitematis, diperlukan pendekatan analisis berbasis data. Analisis sentimen sebagai metode yang dipergunakan dalam mengelolah data teks dan mengklarifikasikan opini menjadi kategori seperti positif, negatif, ataupun netral (Syahira & Kurniawan, 2024).

Penelitian ini merujuk pada beberapa studi sebelumnya yang telah dikaji beberapa kasus *cyberbullying* di media sosial serta pendekatan analisis sentimen terhadap opini publik di media sosial. Penelitian (Wibisono et al., 2025) menggunakan metode *Naïve bayes* untuk secara efektif dan efisien mengklasifikasikan komentar terkait *cyberbullying*. Metode *Naïve bayes* merupakan salah satu metode klasifikasi. Metode *Naïve bayes* dilakukan dengan membandingkan nilai posterior dari kelas-kelas yang ada. Nilai posterior yang paling tinggi yang terpilih sebagai hasil klasifikasi (Khadafi et al., 2022).

Dari latar belakang di atas, penelitian ini dilaksanakan dengan maksud untuk memahami persepsi publik terhadap *cyberbullying* yang sampai saat ini masih terjadi di media sosial. Data dalam penelitian ini diperoleh dari komentar salah satu unggahan aku di media sosial yang membahas tentang bulling di media sosial, dengan menggunakan metode kuantitatif dan metode *Naïve bayes*. Penelitian ini diharapkan mampu untuk menghasilkan analisis dan bermanfaat secara sosial dan dapat menjadi pertimbangan masyarakat agar bijak dan *aware* terhadap kasus *cyberbullying* di media sosial.

B. METODE PENELITIAN

Penelitian ini mengadopsi model kuantitatif dengan memanfaatkan metode analisis sentimen berbasis *Naïve bayes*. Tahapan maupun prosedur yang diterapkan dijabarkan:



Gambar 1. Alur proses penelitian

Identifikasi Masalah

Pada tahap awal ini menggunakan pendekatan kuantitatif dengan menggunakan analisis sentimen pada masyakat terhadap *cyberbullying* dalam kehidupan beropini di media sosial.

Studi Literatur

Studi literatur melibatkan data pendukung lainnya seperti jurnal dan penelitian terkait, sebagai acuan dalam penelitian ini.

Pengumpulan Data

Pada langkah ini dikumpulkan secara spesifik penetapan data sesuai batasan masalah, yaitu komentar pengguna salah satu unggahan pada platform TikTok dari akun @bulelengterkini pada tahun 2025, tentang pembulian yang dilakukan oleh mahasiswa senior di unud. Akun ini dipilih karena berisi konten yang bersifat informatif dan aktual, sehingga banyak mampu memicu beragam respons emosional dari pengguna dalam bentuk komentar. Variasi komentar yang muncul mulai dari sentimen positif, negatif, hingga netral.

Pelabelan Data

Setelah dapat data dari @bulelengterkini dari postingan di TikTok selanjutnya adalah pelabelan dengan label komentar tentang opini masyarakat terhadap *cyberbullying* dengan label positif, negatif dan netral. Dan kami juga menghapus komentar yang tidak sesuai dengan penelitian ini. Pelabelan ini dilakukan secara manual ke dalam *google sheet*.

Text Processing

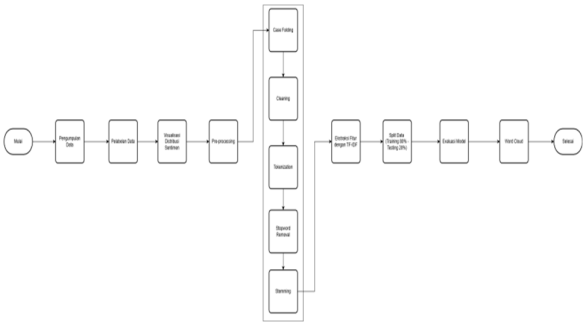
Tahap *Text processing* bertujuan untuk memproses data mentah agar siap digunakan (Putra et al., 2025). Tahap *Text processing* meliputi langkah-langkah berikut : *cleaning*, *tokenize*, *transform case*, *stopword*, *filter*, *stemming*.

Evaluasi

Dalam tahap evaluasi ini penguji menggunakan model yang di hasilkan oleh algoritma *naïve bayes* dengan mengadopsi *confusion matrix*. Adapun tujuannya dari pengujian ini guna mengetahui seberapa baik model *naïve bayes* dalam melakukan klasifikasi komentar di TikTok .

Metode Pengolahan dan Analisis Data

Metode pengelolaan maupun analisis data dalam temuan ini mengadopsi *text mining* maupun *machine learning* untuk mengklafikasi sentimen.



Gambar 2. Alur Metode Pengolahan dan Analisis Data

Visualisasi Distribusi Sentimen

Kami membuat *bar chart* dan *pie chart* untuk melihat sebaran komentar positif, netral, dan negatif untuk melihat apakah data seimbang atau condong ke salah satu kelas.

Pre-processing

Case Folding: keseluruhan huruf diubah menjadi teks kecil. *Cleaning*: seluruh *mention*, *hashtag*, URL, angka, emotikon, dan tanda baca dihapus.

Tokenization: teks dipecah menjadi potongan kata-kata terpisah.

Stopword removal: kata-kata yang tidak memberikan informasi penting dibuang dari data.

Stemming: kata-kata dikembalikan ke bentuk dasar.

Ekstraksi fitur dengan TF-IDF: mengubah komentar menjadi vektor numerik sehingga model *naïve bayes* bisa menghitung bobot pentingnya tiap kata.

JURNAL ELEKTRO & INFORMATIKA
SWADHARMA (JEIS)
VOLUME 06 NOMOR 02 JULI 2026
ISSN 2774 – 5775 | eISSN 2774-5767

Split Data: Membagi data dengan dua bagian yakni latih maupun uji agar distribusi tiap kelas sentimen tetap proporsional.

Evaluasi Model

Melakukan pengukuran performa model dengan menghitung akurasi, *precision*, *recall*, *F1-score*, menggunakan *confusion matrix* dan memvisualkan hasil grafiknya.

1. Akurasi (*Accuracy*): Menggambarkan kemampuan model dalam mengklasifikasikan seluruh kategori dengan tepat. *Accuracy*: (TP+TN) / (TP+FP+FN+TN)
2. Presisi (*Precision*): Menunjukkan proporsi prediksi positif yang akurat. *Precision*: (TP) / (TP+FP)
3. *Recall* (Sensitivitas): Menilai seberapa efektif model dalam mengenali semua kasus positif yang sebenarnya. *Recall*: TP / (TP+FN)
4. *F1-score*: Mengombinasikan presisi maupun *recall* guna memberikan ukuran kinerja model yang seimbang. *F1-score*: (2**Recall*Precision*) / (*Recall*+ *Precision*)

Keterangan: TP merupakan data yang positif maupun berhasil diidentifikasi sebagai positif oleh model. TN: merupkan data yang sebenarnya negatif maupun diprediksi sebagai negatif oleh model. FP merupakan data yang sesungguhnya negatif namun terklasifikasi sebagai positif. FN merupakan data yang seharusnya positif tetapi terdeteksi sebagai negatif (Baehaqi & Cahyono, 2024).

Word Cloud

Membuat *Word Cloud* untuk setiap kelas sentimen agar terlihat kata apa yang sering muncul di masing-masing kelas.

Kesimpulan Penelitian

Menampilkan total data, proporsi *training/testing*, akurasi model, distribusi sentimen yang digunakan sebagai bagian akhir laporan penelitian.

C. HASIL DAN PEMBAHASAN

Penelitian menggunakan data komentar dari postingan di sosial media TikTok pada akun @bulelengterkini yang berjumlah 1.100 komentar. Setelah pemfilteran data, jumlah data komentar menjadi 779 komentar. Setelah itu data komentar ini di *split* menjadi 80% data training dan 20% data testing.

Pengumpulan Data

Data komentar yang ada di *google sheet* sudah dilakukan pemfilteran data, kemudian diupload ke google colab. Setelah pemfilteran, total data yang ada menjadi 779 baris.

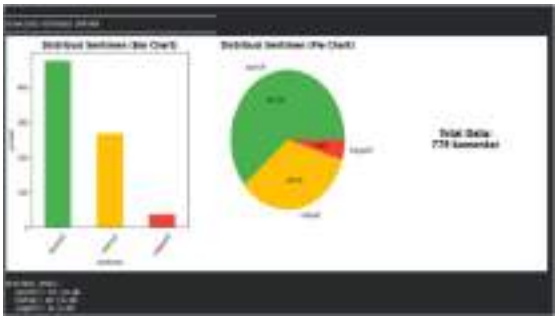
Gambar 4. Hasil Pemfilteran Data

Pelabelan Data

Pelabelan data terbagi atas 3 kelas sentimen (positif, negatif, maupun netral).

Visualisasi Distribusi Sentimen

Pada tahap ini dibuat diagram bar dan diagram pie untuk melihat kalau setiap kelas tidak condong ke satu kelas saja.



Gambar 5. Visualisasi Data

Pre-processing

Setelah melakukan pelabelan data, selanjutnya kami melakukan *pre-processing* terhadap *dataset*.

Case Folding

Di tahap ini seluruh huruf kapital akan dirubah menjadi huruf kecil.

Sebelum	Sesudah
Kenapa sih orang baik selalu diambil duluan 😂 😂	kenapa sih orang baik selalu diambil duluan 😂 😂

Cleaning

Membersihkan data teks dari mention, *hashtag*, URL, angka, *emoticon*, dan tanda baca.

Sebelum	Sesudah
Kenapa sih orang baik selalu diambil duluan 😂 😂	kenapa sih orang baik selalu diambil duluan

Tokenization

Tokenization merupakan teknik yang bertujuan untuk memecah teks menjadi unit-unit yang lebih kecil, seperti kata, frasa atau kalimat (Suhartoyo et al., 2025).

Sebelum	Sesudah
kenapa sih orang baik selalu diambil duluan	'kenapa', 'sih', 'orang', 'baik', 'selalu', 'diambil', 'duluan'

Stopword removal

Stopword removal merupakan salah satu metode yang bisa diterapkan dalam tahapan *text pre-processing*, dimana akan dilakukan penghapusan pada kata-kata yang dianggap sering muncul namun tidak bermakna dan berpengaruh signifikan terhadap makna kalimat atau teks (Angelina et al., 2023).

Sebelum	Sesudah
'kenapa', 'sih', 'orang', 'baik', 'selalu', 'diambil', 'duluan'	'kenapa', 'diambil', 'duluan'

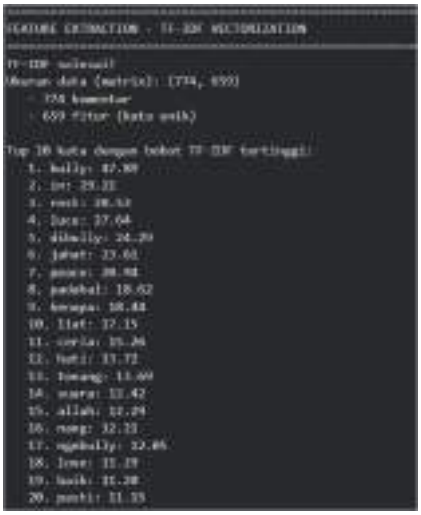
Stemming

Stemming adalah proses pemetaan dan penguraian bentuk dari suatu kata menjadi bentuk kata dasarnya (Zulfiqri et al., 2024).

Sebelum	Sesudah
'kenapa', 'diambil', 'duluan'	kenapa ambil duluan

Ekstraksi Fitur TF-IDF

Metode representasi teks dalam pemrosesan bahasa alami yang menghitung bobot kata berdasarkan frekuensinya.



Gambar 6. Ekstraksi Fitur TF-IDF

Split Data

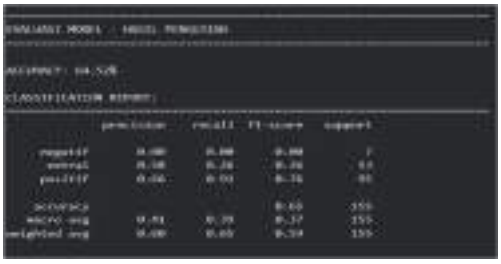
Melakukan pembagian *dataset* menjadi subset Data Training maupun Testing untuk mencegah *overfitting* dengan memastikan model menggeneralisasi pada data baru.

Gambar 7. Proses Split Data

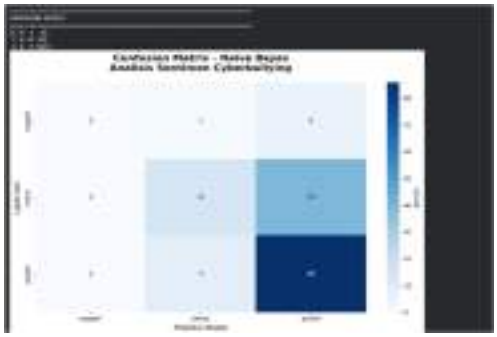
Evaluasi Model

Melakukan pengukuran performa model dengan menghitung akurasi, *precision*, *recall*, *F1-score*, menggunakan *confusion matrix* dan memvisualkan hasil grafiknya.

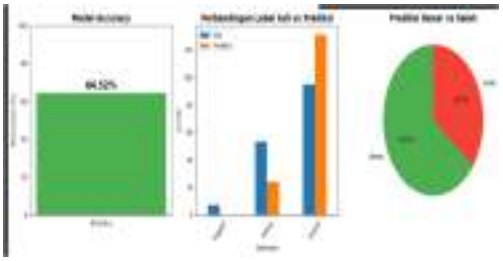
JURNAL ELEKTRO & INFORMATIKA
SWADHARMA (JEIS)
VOLUME 06 NOMOR 02 JULI 2026
ISSN 2774 – 5775 | eISSN 2774-5767



Gambar 8. Hasil Pengujian Evaluasi Model



Gambar 9. Hasil *Confusion matrix*



Gambar 10. Visualisasi Hasil Pengujian

Analisis Word Cloud

Analisis *word cloud* berupa kata-kata yang sering muncul di setiap kelas sentimennya.

Gambar 11. *Word Cloud*

Word cloud per kelas mengungkap kata kunci dominan "*bullying*" dan "korban" pada negatif, "sadar" pada positif, dan "info" pada netral, memperkuat wawasan pola opini masyarakat. Hasil ini menegaskan *naïve bayes* unggul untuk deteksi sentimen *cyberbullying* di sosial media.

D. PENUTUP

Pengujian ini dilaksanakan dengan mengadopsi *confusion matrix* guna mengukur performa klasifikasi sentimen pada 779 komentar TikTok yang telah di split data yakni 80% training maupun 20% testing. *Confusion matrix* menghasilkan evaluasi utama seperti akurasi, *precision*, *recall*, maupun *F1-score*, menunjukan bahwa *naïve bayes* mampu membedakan sentimen positif, negatif, dan netral dengan baik, membuktikan bahwa *naïve bayes* efektif untuk penelitian sentimen ini.

E. DAFTAR PUSTAKA

Angelina, S. J., Negara, A. B. P., & Muhardi, H. (2023). Analisis Pengaruh Penerapan Stopword Removal Pada Performa Klasifikasi Sentimen Tweet Bahasa Indonesia. *Jurnal Aplikasi Dan Riset Informatika*, *1*(2), 165–173. https://doi.org/10.26418/jari.v2i1.69680

Aser, F. G., Paramita, S., & Sudarto. (2022). Fenomena Cyberbullying di Media Sosial TikTok. *KIWARI: Jurnal Fakultas Ilmu Komunikasi Universitas Tarumanegara*, *1*(3), 449–453. https://doi.org/10.24912/ki.v1i3.15763

Baehaqi, F., & Cahyono, N. (2024). Analisis Sentimen Terhadap Cyberbullying Pada Komentar Di Instagram Menggunakan Algoritma Naïve Bayes. *The Indonesian Journal of Computer Science*, *13*(1), 1051–1063. https://doi.org/10.33022/ijcs.v13i1.3301

Damayanti, A., Delima, I. D., & Suseno, A. (2023). Pemanfaatan Media Sosial Sebagai Media Informasi dan Publikasi (Studi Deskriptif Kualitatif pada Akun Instagram @rumahkimkotatangerang).

PIKMA : Publikasi Ilmu Komunikasi Media Dan Cinema, *6*(1), 173–190. https://doi.org/10.24076/pikma.v6i1.1308

Khadafi, M. Al, Kartika, K. P., & Febrinita, F. (2022). Penerapan Metode Naive Bayes Classifier dan Lexicon Based Untuk Analisis Sentimen Cyberbullying Pada BPJS. *JATI : Jurnal Mahasiswa Teknik Informatika*, *6*(2), 725–733. https://doi.org/10.36040/jati.v6i2.5633

Marlef, A., Masyhuri, M., & Muda, Y. (2024). Mengenal dan Mencegah Cyberbullying: Tantangan Dunia Digital. *Journal of Education Research*, *5*(3), 4002–4010. https://doi.org/10.37985/jer.v5i3.1295

Nugraha, D., & Astuti, P. (2023). Analisis Sentimen Cyberbullying Pada Sosial Media Instagram Menggunakan Metode Support Vector Machine. *Information System for Educators and Professionals*, *8*(2), 153–164. https://doi.org/10.51211/isbi.v8i2.2535

Putra, D. M. D. U., Suryawan, I. W. D., Andayani, S., & Rusadi, E. Y. (2025). Analisis Sentimen terhadap Fenomena Artis Berjualan di Tiktok Shop pada Media Sosial X Menggunakan Model Naïve Bayes. *JUSITIK : Jurnal Sistem Dan Teknologi Informasi Komunikasi*, *8*(2), 108–113. https://doi.org/10.32524/jusitik.v8i2.1452

Suhartoyo, R., Julyo, V., Irsyad, H., & Rahman, A. (2025). Keyword Extraction of Scientific Journal Abstracts Using TF-IDF and KeyBERT Methods. *AICOMS : Applied Information Technology and Computer Science*, *4*(2), 1–9. https://doi.org/10.58466/aicoms.v4i2.1814

Syahira, M. A., & Kurniawan, R. (2024). Analisis Sentimen Cyberbullying Pada Media Sosial X Menggunakan Metode Support Vector Machine. *Media Informatika Budidarma*, *8*(3), 1724–1733. https://doi.org/10.30865/mib.v8i3.7926

Wibisono, B., Machmud, A., Suryani, N., & Yunita, Y. (2025). Analisis Sentimen Cyberbullying Pada Komentar X Menggunakan Metode Naïve Bayes. *Computer Science (CO-SCIENCE)*, *5*(1), 8–16. https://doi.org/10.31294/coscience.v5i1.5152

Zulfiqri, R., Sari, B. N., & Padilah, T. N. (2024). Analisis Sentimen Ulasan Pengguna Aplikasi Media Sosial Instagram Pada Situs Google Play Store Menggunakan Naive Bayes Classifier. *JITET : Jurnal Informatika Dan Teknik Elektro Terapan*, *12*(3), 2965–2973. https://doi.org/10.23960/jitet.v12i3.4995

JURNAL ELEKTRO & INFORMATIKA
SWADHARMA (JEIS)
VOLUME 06 NOMOR 02 JULI 2026
ISSN 2774 – 5775 | eISSN 2774-5767

DETEKSI KERUSAKAN JALAN MENGGUNAKAN *CONVOLUTIONNAL NEURAL NETWORK* (CNN)

Muhammad Haikal Abidin[1)], Iman Febriansya Putra[2)], Sumanto[3)], Ade Christian[4)], Jefina Tri Kumalasari[5)], Ghofar Taufiq[6)]

Prodi Informatika, Fakultas Teknik dan Informatika, Universitas Bina Sarana Informatika

Correspondence author: MH Abidin, haikalabdn00@gmail.com, Jakarta, Indonesia

Abstract

Road surface damage is a critical issue that can impact road user safety, transportation efficiency, and infrastructure maintenance costs. Manual road inspection methods are still widely used, but they suffer from subjective assessments that can result in inconsistent data. Therefore, an automated approach using artificial intelligence technology is needed to improve the accuracy and efficiency of road damage detection. This research aims to develop a road damage detection system using a deep learning approach based on Convolutional Neural Networks (CNN). The detection model was trained using an annotated road image dataset covering various types of road surface damage. The training and evaluation processes were conducted using an object detection architecture that included mean average precision (mAP), precision, and recall. Test results show that the CNN model can automatically detect road damage with fairly good performance. However, system performance is still affected by the limited dataset size, class imbalance, and variations in environmental conditions. The developed system has the potential to be a decision-support tool for more effective and data-driven road infrastructure maintenance.

Keyword *: detection system, road damage, deep learning, Convolutional Neural Network*

Abstrak

Kerusakan permukaan jalan merupakan masalah penting yang dapat memengaruhi keselamatan pengguna jalan, efisiensi transportasi, dan biaya pemeliharaan infratruktur. Metode inspeksi jalan secara manual masih banyak dipakai, memiliki kelemahan adanya subjektivitas penilaian, yang dapat menghasilkan data yang tidak konsisten. Oleh karena itu, dibutuhkan pendekatan otomatis menggunakan teknologi kecerdasan buatan untuk meningkatkan akurasi dan efisiensi deteksi kerusakan jalan. Penelitian ini bertujuan mengembangkan sistem deteksi kerusakan jalan dengan pendekatan *deep learning* berbasis *Convolutional Neural Network* (CNN). Model deteksi dilatih menggunakan *dataset* citra jalan yang mencakup berbagai jenis kerusakan permukaan jalan. Proses pelatihan dan evaluasi dilakukan dengan arsitektur deteksi objek seperti *mean average precision* (mAP), *precision*, dan *recall*. Hasil pengujian menunjukan bahwa model CNN dapat mendeteksi kerusakan jalan secara otomatis dengan kinerja yang cukup baik. Namun, performa sistem masih dipengaruhi oleh keterbatasan jumlah *dataset*, ketidakseimbangan kelas, dan

variasi kondisi lingkungan. Sistem yang dikembangkan berpotensi menjadi alat pendukung pengambilan keputusan untuk pemeliharaan infrastruktur jalan yang lebih efektif dan berbasis data.

Kata Kunci : sistem deteksi, kerusakan jalan, *deep learning*, *convolutional neural network*

A. PENDAHULUAN

Jalan raya merupakan salah satu infrastruktur transportasi yang memegang peran strategis untuk mendukung pertumbuhan ekonomi, mobilitas sosial, serta distribusi barang dan jasa (Zahra et al., 2024). Kondisi jalan raya sangat mempengaruhi tingkat keamanan transportasi jalan maupun efisiensi transportasi. Kerusakan jalan raya, seperti keretakan, lubang, maupun deformasi aspal, dapat mempengaruhi kenyamanan berkendara, meningkatkan risiko kecelakaan lalu lintas, maupun menghadapi kerugian ekonomi akibat meningkatnya biaya perawatan jasa kendaraan maupun perawatan jala raya (Mulyana & Wahyudi, 2025).

Secara global, kapasitas dan alat transportasi serta beban lalu lintas yang berlimpah menyebabkan proses degradasi jalan berlangsung lebih cepat dibandingkan siklus peliharaan (Juanda et al., 2025). Keadaan ini membutuhkan adanya sistem pelacak jalan yang mampu mengenali kerusakan jalan lebih dahulu dan akurat agar proses perbaikan jalan dapat tepat waktu (Zain, 2024). Akan tetapi saat ini proses pendeteksian kerusakan jalan masih berlangsung dengan cara tradisionai melalui proses survei visual oleh petugas lapangan. Metode ini membutuhkan waktu yang cukup lama dan biaya operasional yang tinggi. Hasilnya sangat bergantung pada pengamatan petugas dan bisa cukup subjektif (Zuhri et al., 2025).

Di Indonesia, masalah kerusakan jalan juga telah menjadi isu nasional yang cukup signifikan, mengingat luas dan kelangkaan sumber daya yang diperlukan dalam proses pemeliharaan jalan. Inspeksi manual yang dilakukan secara berkala tidak bisa menggambarkan secara *real-time* kerusakan tersebut, sehingga beberapa kerusakannya baru bisa terdeteksi setelah mencapai tingkat kerusakan yang cukup parah. Berdampak pada semakin meningkatnya biaya perbaikan maupun penurunan tingkat keselamatan pengguna jalan (Denaro & Lim, 2025). Hal ini menunjukkan bahwa masih diperlukan sebuah metode yang cukup efisien dan objektif dalam pemeliharaan jalan.

Kehadiran teknologi *computer vision* dan *deep learning* memberikan peluang besar dalam otomatisasi proses deteksi kerusakan jalan berbasis citra digital. Salah satu algoritma *deep learning* yang banyak digunakan dalam pengolahan citra adalah *Convolutional Neural Network* (CNN), yang memiliki kemampuan unggul dalam mengekstraksi fitur visual serta mengenali pola kompleks dari struktur citra (Nasution et al., 2025). Penerapan CNN dalam deteksi kerusakan jalan telah banyak dilaporkan dalam penelitian sebelumnya dan menunjukkan kinerja yang menjanjikan, baik dalam mendeteksi lubang jalan (*pothole*) maupun dalam mengklasifikasikan berbagai jenis kerusakan permukaan jalan lainnya (Jiang, 2024).

(Lakshminarayanan & Konidhala, 2024) melakukan pengembangan sistem deteksi lubang jalan berbasis CNN yang diterapkan pada berbagai kondisi kebersaman cahaya di lingkungan perkotaan. Namun, hasilnya menunjukkan konklusi bahwa CNN mampu mendeteksi lubang jalan secara otomatis dengan tingkat keakuratan yang cukup tinggi untuk dikategorikan relatif atas trend konvensional.

Selain CNN konvensional, ada juga beberapa penelitian lainnya yang mengembangkan model turunan CNN dengan menggunakan teknologi CNN untuk meningkatan kinerja deteksi kerusakan jalan. (Khairunisa et al., 2024) mengukur kinerja algoritma CNN dan YOLO pada pengenalan kerusakan jalan dan mendapatkan hasil bahwa kedua metode tersebut memiliki keunggulan masing-masing berupa kinerja akurasi dan kinerja proses pemrosesan. Penelitian lainnya oleh (Azhari et al., 2025), adalah analisis dan pengembangan deteksi kerusakan jalan menggunakan teknologi YOLOv9 melalui CNN dengan mendapatkan kinerja optimal pada proses deteksi benda pada citra jalan.

Meskipun berbagai penelitian di atas menunjukkan adanya potensi besar pada penggunaan *deep learning* pada deteksi kerusakan jalan, keterbatasan-keterbatasannya pun masih dapat diamati. Faktor seperti kekurangan *dataset*, imbang tidaknya jumlah pengamatan kelas di klasifikasi kerusakan, hingga keanekaragaman kondisi pengamatan di waktu foto berpotensi berpengaruh nyata pada hasil pengamatan (Ghofur et al., 2025; Zuhri et al., 2025). (Zanevych et al., 2024) pun menegaskan bahwa keanekaragaman foto berpotensi mengecilkan fungsi pengamatan.

Selain itu, (Denaro & Lim, 2025) telah menunjukkan bahwa aplikasi arsitektur berbasis jaringan lebih dalam, seperti kombinasi analisis CNN dengan ResNet, dapat meningkatkan akurasi analisis klassifikasi kerusakan jalan aspal, serta berfungsi sebagai landasan untuk label efisiensi biaya dan waktu dalam aplikasi pemantauan infrastruktur berkelanjutan. Namun, pentingnya difokuskan pengamatan lanjutan untuk dapat mengoptimalkan aplikasi sistem deteksi kerusakan yang.

Berdasarkan penjelasan di atas dan penelitian sebelumnya tersebut dapat disimpulkan bahwa pengembangan sistem deteksi kerusakan jalan berbasis *deep learning* memiliki potensi peningkatan untuk kedepannya terutama dalam hal kekonvergenan model menyesuaikan kondisi lingkungan dan ketersedian data. Sehingga berdasarkan hal tersebut maka pada penelitian ini dapat disampaikan tujuan penelitian yaitu mengembangkan sistem deteksi kerusakan permukaan jalan menggunakan pendekatan *deep learning* berbasis *Convolutional Neural Network* dengan menggunakan sumber data berupa citra digital. Sehingga penelitian ini dapat memberikan manfaat penelitian yaitu dapat memberikan karakteristik sistem deteksi kerusakan jalan berbasis *deep learning* yang tepat waktu.

B. METODE PENELITIAN

Penelitian ini adalah penelitian kuantitatif dengan pendekatan eksperimen. Tujuan utamanya adalah mengembangkan dan mengevaluasi sistem deteksi kerusakan jalan berbasis *deep learning* menggunakan metode *Convolutional Neural Network* (CNN). Pendekatan kuantitatif diperlukan karena penelitian ini melibatkan pengukuran kinerja model secara numerik menggunakan metrik tertentu, seperti *precision*, *recall*, dan *mean average precision* (mAP).

Penelitian dilaksanakan di lingkungan laboratorium dengan memanfaatkan *dataset* citra jalan. Waktu penelitian mencakup pengembangan dan pengujian model, termasuk tahap pengumpulan data, praproses data, pelatihan model, evaluasi, dan analisis hasil. Lokasi penelitian tidak dibatasi secara geografis karena data yang digunakan adalah citra digital dari *dataset* yang tersedia.

Populasi penelitian ini mencakup semua citra permukaan jalan yang menunjukkan berbagai jenis kerusakan. Sampel terdiri dari citra jalan yang telah dianotasi dan dibagi menjadi data latih, data validasi, dan data uji. Teknik sampling yang digunakan adalah *purposive sampling*, yang berarti pemilihan citra berdasarkan kriteria tertentu, seperti

anotasi yang jelas dan representasi jenis kerusakan yang diteliti.

Pengumpulan data dilakukan dengan menggunakan *dataset* citra jalan dari sumber sekunder Pengumpulan data dalam penelitian ini dilakukan menggunakan *dataset* citra jalan dari sumber sekunder, yaitu *Road Damage Dataset* yang disediakan oleh (Arya et al., 2022). *Dataset* ini merupakan *dataset* terbuka yang berisi citra permukaan jalan dari berbagai negara dan telah banyak digunakan dalam penelitian deteksi kerusakan jalan berbasis *computer vision* dan *deep learning*.

Dari keseluruhan *dataset* RDD2022, penelitian ini menarik sejumlah 1.455 citra jalan yang merepresentasikan berbagai kondisi kerusakan jalan, seperti lubang jalan (*pothole*), retakan memanjang, retakan melintang, dan kerusakan permukaan lainnya. Citra yang dipilih kemudian diseleksi dan disesuaikan dengan kebutuhan penelitian agar relevan dengan tujuan pengembangan sistem deteksi kerusakan jalan berbasis *Convolutional Neural Network* (CNN).

Sebelum digunakan dalam proses pelatihan dan pengujian model, seluruh citra melalui tahap praproses data yang meliputi penyesuaian ukuran citra (*image resizing*), normalisasi nilai piksel, serta augmentasi data. Proses augmentasi dilakukan untuk meningkatkan variasi data latih dan mengurangi risiko *overfitting* selama proses pelatihan model.

Data yang telah dipraproses selanjutnya dibagi ke dalam data latih dan data uji, kemudian digunakan untuk melatih dan mengevaluasi kinerja model CNN dalam mendeteksi dan mengklasifikasikan kerusakan jalan secara otomatis.

Setelah itu, citra melalui tahap praproses yang mencakup penyesuaian ukuran citra, normalisasi, dan augmentasi data. Proses ini meningkatkan variasi data latih dan mengurangi risiko overfitting. Data yang sudah diproses kemudian digunakan dalam pelatihan dan pengujian model CNN.

Alat dan bahan dalam penelitian ini mencakup perangkat keras dan perangkat lunak. Perangkat keras terdiri dari komputer atau laptop dengan spesifikasi prosesor minimal Intel Core i5, RAM minimal 8 GB, dan kartu grafis yang mendukung komputasi GPU untuk mempercepat pelatihan model. Perangkat lunak yang digunakan meliputi sistem operasi, bahasa pemrograman Python, dan framework *deep learning* seperti TensorFlow atau PyTorch untuk membangun dan melatih model CNN. Selain itu, pustaka pendukung untuk pengolahan citra dan visualisasi data juga digunakan.

Analisis data dilakukan dengan mengevaluasi kinerja model CNN menggunakan metrik *precision*, *recall*, dan *mean average precision* (mAP). Hasil evaluasi ini digunakan untuk menilai kemampuan model dalam mendeteksi dan mengklasifikasikan kerusakan jalan dengan tepat. Analisis dilakukan secara deskriptif kuantitatif dengan membandingkan nilai metrik yang diperoleh pada data uji.

Penyajian data dalam penelitian ini ditampilkan dalam bentuk tabel, grafik, dan visualisasi hasil deteksi pada citra jalan. Penyajian ini bertujuan untuk memudahkan interpretasi hasil pengujian dan memberikan gambaran yang jelas tentang kinerja sistem deteksi kerusakan jalan yang dikembangkan.

C. HASIL DAN PEMBAHASAN

Karakteristik Subjek Penelitian

Subjek penelitian pada penelitian ini berupa citra permukaan jalan yang diklasifikasikan ke dalam enam kelas kerusakan berdasarkan *dataset* penelitian. Klasifikasi ini mencerminkan variasi jenis kerusakan yang umum ditemukan pada infrastruktur jalan, mulai dari retakan ringan hingga deformasi permukaan jalan yang bersifat struktural. Karakteristik subjek penelitian ini penting untuk menggambarkan kompleksitas data yang digunakan dalam proses pelatihan dan pengujian model deteksi kerusakan jalan berbasis *deep learning*.

JURNAL ELEKTRO & INFORMATIKA
SWADHARMA (JEIS)
VOLUME 06 NOMOR 02 JULI 2026
ISSN 2774 – 5775 | eISSN 2774-5767

Tabel 1: klarifikasi kerusakan berdasarkan *dataset [Sumber data sendiri]*

No	Kelas kerusakan	Jumlah citra	keterangan
1	Retakan(crack)	578	Retakan garis,retakan acak,retakan permukaan
2	Lubang(pothole)	497	Lubang kecil,lubang besar,deformasi permukaan
3	Retak longitudinal	122	Retakan memanjang searah jalur jalan
4	Retak tranversal	96	Retakan melintang akibat tekanan kendaraan
5	Retak jaringan (alligator cracking)	88	Retakan pola kulit buaya akibat beban berlebih
6	deformasi	74	Gelombang, amblas, atau penurunan permukaan

Berdasarkan Tabel 1, terlihat bahwa kelas retakan (*crack*) dan lubang (*pothole*) memiliki jumlah citra paling dominan dibandingkan kelas kerusakan lainnya. Ketidakseimbangan distribusi data ini berpotensi memengaruhi kinerja model dalam mendeteksi kelas dengan jumlah data yang lebih sedikit.

Analisis Univariat

Analisis univariat dilakukan untuk mengevaluasi kinerja model CNN berdasarkan metrik evaluasi utama, yaitu *mean average precision* (mAP) dan mAP@50:95. Analisis ini bertujuan untuk melihat performa model secara keseluruhan tanpa mempertimbangkan perbedaan antar kelas kerusakan.

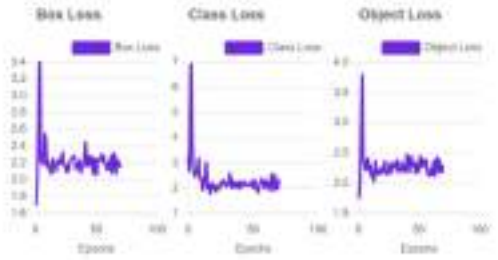
Gambar 1. Grafik Performa Model Berdasarkan Nilai mAP dan mAP@50:95

Berdasarkan Gambar 1, terlihat bahwa nilai mAP mengalami peningkatan secara bertahap seiring bertambahnya jumlah *epoch*. Pada tahap awal pelatihan, nilai mAP cenderung berfluktuasi, kemudian mulai menunjukkan tren peningkatan yang lebih stabil setelah memasuki *epoch* menengah hingga akhir. Hal ini menunjukkan bahwa model CNN mampu mempelajari pola visual kerusakan jalan secara bertahap melalui proses pelatihan.

Sementara itu, nilai mAP@50:95 menunjukkan nilai yang lebih rendah dibandingkan mAP, namun tetap mengalami peningkatan yang konsisten. Perbedaan nilai ini mengindikasikan bahwa model memiliki performa yang lebih baik pada ambang *Intersection over Union* (IoU) yang lebih longgar dibandingkan IoU yang lebih ketat.

Analisis Bivariat

Analisis bivariat dilakukan untuk melihat hubungan antara proses pelatihan model dengan nilai *loss* yang dihasilkan, meliputi *box loss*, *class loss*, dan *object loss*. Analisis ini bertujuan untuk menilai stabilitas proses pembelajaran model CNN selama pelatihan.



Gambar 2. Grafik Box Loss, Class Loss, dan

Object Loss Selama Proses Pelatihan

Berdasarkan Gambar 2, dapat dilihat bahwa ketiga jenis *loss* mengalami penurunan signifikan pada tahap awal pelatihan, kemudian cenderung stabil pada *epoch* berikutnya. Penurunan *loss* ini menunjukkan bahwa model CNN berhasil meminimalkan kesalahan prediksi baik dalam aspek lokasi objek, klasifikasi kelas, maupun keberadaan objek kerusakan jalan.

Stabilnya nilai *loss* pada tahap akhir pelatihan mengindikasikan bahwa model telah mencapai kondisi konvergen dan tidak mengalami *overfitting* yang signifikan.

Analisis Multivariat

Analisis multivariat dilakukan untuk menilai pengaruh beberapa faktor secara simultan terhadap kinerja sistem deteksi kerusakan jalan. Faktor-faktor yang dianalisis meliputi distribusi kelas *dataset*, proses pelatihan model, serta dukungan infrastruktur sistem yang digunakan.

Gambar 3. Dukungan Sistem Pengembangan dan Implementasi Model

Berdasarkan Gambar 3, terlihat bahwa sistem pengembangan model mendukung berbagai fitur penting seperti *pre-trained models*, *fine-tuning*, dan *active learning*. Kombinasi faktor-faktor ini berkontribusi terhadap peningkatan performa dan stabilitas model CNN dalam mendeteksi kerusakan jalan.

Analisis Univariat (Kinerja Model Deteksi Kerusakan Jalan)

Analisis univariat selanjutnya dilakukan untuk mengevaluasi kinerja akhir model deteksi kerusakan jalan berdasarkan metrik evaluasi yang dihasilkan dari proses pelatihan dan validasi model. Model yang dikembangkan menggunakan arsitektur *Object Detection* berbasis *Convolutional Neural Network* dengan *checkpoint dataset* COCO dan dilatih menggunakan platform Roboflow versi 3.0 dengan konfigurasi *fast object detection*.

Gambar 4. Hasil Evaluasi Kinerja Model Deteksi Kerusakan Jalan

Berdasarkan Gambar 4, diperoleh nilai *mean average precision* pada ambang *Intersection over Union* 50% (mAP@50) sebesar 52,2%, nilai *precision* sebesar 62,9%, dan nilai *recall* sebesar 50,9%. Nilai *precision* yang lebih tinggi dibandingkan *recall* menunjukkan bahwa model memiliki kemampuan yang baik dalam meminimalkan kesalahan prediksi positif, namun masih terdapat beberapa objek kerusakan jalan yang belum berhasil terdeteksi oleh sistem.

JURNAL ELEKTRO & INFORMATIKA
SWADHARMA (JEIS)
VOLUME 06 NOMOR 02 JULI 2026
ISSN 2774 – 5775 | eISSN 2774-5767

Nilai mAP@50 yang berada di atas 50% menunjukkan bahwa model mampu mengenali dan melokalisasi objek kerusakan jalan dengan tingkat ketepatan yang cukup baik pada ambang IoU moderat. Hasil ini mengindikasikan bahwa pendekatan *deep learning* berbasis CNN yang digunakan telah berhasil mempelajari fitur visual kerusakan jalan dari *dataset* yang digunakan.

Analisis Bivariat (Hubungan Proses Pelatihan dan Kinerja Model)

Hubungan antara proses pelatihan model dengan kinerja akhir sistem dapat dilihat dari durasi pelatihan dan stabilitas metrik evaluasi yang dihasilkan. Berdasarkan informasi pelatihan, model dilatih selama beberapa jam hingga mendekati konvergensi, dengan pemantauan metrik evaluasi dilakukan secara berkala pada *validation set*.

Hasil evaluasi menunjukkan bahwa peningkatan kinerja model sejalan dengan proses pembelajaran selama pelatihan berlangsung. Meskipun demikian, nilai *recall* yang masih lebih rendah dibandingkan *precision* menunjukkan bahwa model masih dipengaruhi oleh ketidakseimbangan distribusi *dataset*, khususnya pada kelas kerusakan dengan jumlah citra yang lebih sedikit. Temuan ini konsisten dengan hasil analisis distribusi data pada Tabel 1.

Analisis Multivariat (Pengaruh Arsitektur, *Dataset*, dan Platform Pelatihan)

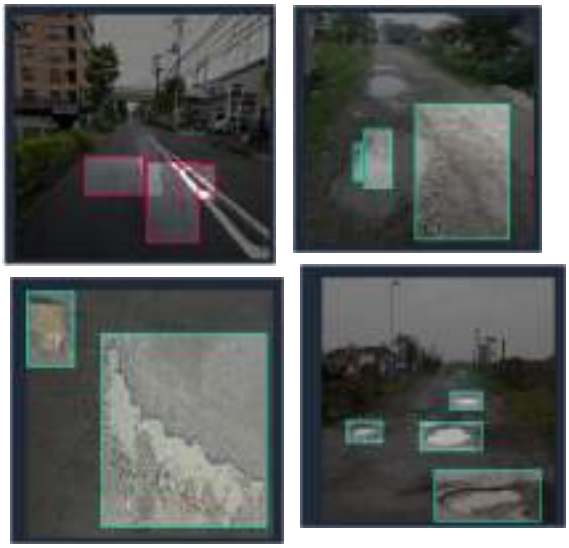
Analisis multivariat dilakukan dengan mempertimbangkan pengaruh arsitektur model, karakteristik *dataset*, serta platform pelatihan terhadap kinerja sistem secara keseluruhan. Penggunaan *pre-trained model* dengan *checkpoint* COCO memberikan keuntungan pada tahap awal pelatihan karena model telah memiliki kemampuan dasar dalam mengenali objek visual.

Namun demikian, keterbatasan jumlah data pada beberapa kelas kerusakan serta variasi kondisi citra jalan memengaruhi performa akhir model, khususnya pada aspek *recall*. Hal ini menunjukkan bahwa optimalisasi lebih lanjut, seperti penambahan *dataset*, peningkatan teknik augmentasi data, serta penyesuaian parameter pelatihan, berpotensi meningkatkan kinerja sistem secara signifikan.

Pembahasan

Hasil evaluasi model menunjukkan bahwa sistem deteksi kerusakan jalan berbasis *Convolutional Neural Network* mampu menghasilkan kinerja yang cukup baik dengan nilai mAP@50 sebesar 52,2%. Nilai *precision* yang relatif tinggi menunjukkan bahwa model mampu memberikan prediksi yang akurat terhadap objek kerusakan jalan yang terdeteksi, sedangkan nilai *recall* yang masih moderat menunjukkan adanya peluang peningkatan dalam mendeteksi lebih banyak objek kerusakan yang ada.

Secara keseluruhan, hasil penelitian ini mengindikasikan bahwa pendekatan *deep learning* berbasis CNN layak digunakan sebagai dasar pengembangan sistem deteksi kerusakan jalan otomatis. Dengan pengembangan lanjutan pada aspek *dataset* dan strategi pelatihan, sistem ini berpotensi menjadi solusi pendukung pemantauan kondisi jalan yang lebih efisien, objektif, dan berbasis data.



Gambar 5. Hasil uji coba kinerja

Gambar 5 menampilkan hasil uji coba kinerja sistem deteksi kerusakan jalan berbasis *Convolutional Neural Network* (CNN) dalam mengidentifikasi dan melokalisasi berbagai jenis kerusakan permukaan jalan pada citra uji. Visualisasi ini memperlihatkan keluaran model berupa *bounding box* dan label kelas kerusakan yang terdeteksi, seperti lubang (*pothole*), retakan (*crack*), serta deformasi permukaan jalan.

Berdasarkan Gambar 5, sistem mampu mendeteksi objek kerusakan jalan dengan tingkat ketepatan visual yang cukup baik, ditunjukkan oleh kesesuaian posisi *bounding box* dengan area kerusakan aktual pada citra. Hal ini menunjukkan bahwa model CNN telah berhasil mempelajari fitur visual utama dari kerusakan jalan, baik dalam bentuk perubahan tekstur permukaan maupun perbedaan pola warna pada aspal.

Namun demikian, pada beberapa citra uji masih ditemukan kondisi di mana sebagian kerusakan jalan belum terdeteksi secara optimal atau hanya terdeteksi sebagian. Kondisi ini umumnya terjadi pada citra dengan pencahayaan rendah, permukaan jalan yang tertutup genangan air, serta kerusakan dengan ukuran kecil atau pola yang menyerupai objek non-kerusakan, seperti marka jalan. Temuan ini sejalan dengan hasil evaluasi kuantitatif yang menunjukkan nilai *recall* lebih rendah dibandingkan *precision*, yang mengindikasikan bahwa model masih melewatkan sebagian objek kerusakan yang ada.

Secara keseluruhan, hasil uji coba kinerja yang ditunjukkan pada Gambar 5 memperkuat hasil evaluasi numerik sebelumnya, di mana sistem deteksi kerusakan jalan berbasis CNN menunjukkan performa yang cukup baik dalam mendeteksi objek kerusakan secara otomatis. Meskipun demikian, peningkatan kinerja masih dapat dilakukan melalui penambahan variasi *dataset*, khususnya pada kondisi lingkungan yang menantang, serta optimalisasi parameter pelatihan untuk meningkatkan kemampuan model dalam mendeteksi seluruh objek kerusakan secara lebih menyeluruh.

D. PENUTUP

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa sistem deteksi kerusakan jalan berbasis *deep learning* dengan metode *Convolutional Neural Network* (CNN) mampu mendeteksi dan mengklasifikasikan kerusakan permukaan jalan menggunakan citra digital secara otomatis. Model yang dikembangkan menunjukkan kinerja yang cukup baik berdasarkan metrik evaluasi yang digunakan, seperti *precision*, *recall*, dan *mean average precision* (mAP), sehingga menunjukkan bahwa pendekatan CNN efektif dalam mengekstraksi fitur visual dan mengenali pola kerusakan jalan.

Hasil analisis juga menunjukkan bahwa kinerja sistem dipengaruhi oleh beberapa faktor, antara lain distribusi jumlah data pada setiap kelas kerusakan, variasi kondisi lingkungan pengambilan citra, serta kualitas proses pelatihan model. Ketidakseimbangan *dataset* pada beberapa kelas kerusakan masih berdampak pada kemampuan model dalam mendeteksi seluruh objek kerusakan secara optimal.

Secara keseluruhan, penelitian ini membuktikan bahwa penerapan metode CNN memiliki potensi untuk digunakan sebagai alat pendukung dalam pemantauan dan pemeliharaan infrastruktur jalan yang lebih efisien, objektif, dan berbasis data. Dengan pengembangan lebih lanjut, khususnya pada aspek penambahan *dataset* dan optimalisasi model, sistem deteksi kerusakan jalan ini diharapkan dapat diterapkan secara lebih luas dalam skala operasional.

E. DAFTAR PUSTAKA

Arya, D., Maeda, H., Ghosh, S. K., Toshniwal, D., & Sekimoto, Y. (2022). RDD2022: A multi-national image dataset for automatic Road Damage Detection. *ArXiv*, 1–16. https://doi.org/10.48550/arXiv.2209.08538

Azhari, F. A., Rohana, T., Kiki, & Fauzi, A. (2025). Road Damage Detection Using Yolov9-Based Imagery. *Sisfokom : Jurnal Sistem Informasi Dan Komputer*, *14*(2), 196–201. https://doi.org/10.32736/sisfokom.v14i2.2377

Denaro, L. G., & Lim, R. (2025). Analisis Deteksi Kerusakan pada Jalan Aspal menggunakan Deep Learning untuk Mendukung Efisiensi Biaya dan Waktu dalam Pemantauan Berkelanjutan. *Jurnal Dimensi Insinyur Profesional*, *3*(1), 16–25. https://doi.org/10.9744/jdip.3.1.16-25

Ghofur, M. A., Murdifin, Hardandrito, A. G., & 'Uyun, S. (2025). High Precision Deep Learning Model for Road Damage Classification using Transfer Learning. *Sistemasi: Jurnal Sistem Informasi*, *14*(6), 3028–3036. https://doi.org/10.32520/stmsi.v14i6.5707

Jiang, Y. (2024). Road damage detection and classification using deep neural networks. *Discover Applied Sciences*, *6*, 421. https://doi.org/10.1007/s42452-024-06129-0

Juanda, A., Muammar, R., & Sabilla, A. (2025). Evaluasi Kerusakan Perkerasan Jalan Perkotaan Berdasarkan Metode Pavement Condition Index (PCI) (Studi Kasus: Ruas Jalan Malikul Adil, Kota Langsa). *PRINCE : Journal of Planning and Research in Civil Engineering*, *4*(3), 691–699. https://doi.org/10.55616/prince.v4i3.1070

Khairunisa, N., Carudin, & Jamaludin, A. (2024). Analisis Perbandingan Algoritma CNN dan YOLO Dalam Mengidentifikasi Kerusakan Jalan. *JITET : Jurnal Informatika Dan Teknik Elektro Terapan*, *12*(3), 1756–1768. https://doi.org/10.23960/jitet.v12i3.4434

Lakshminarayanan, S., & Konidhala, J. (2024). Convolutional Neural Network for Pothole Identifications in Urban Roads. *International Journal of Advances in Signal and Image Sciences*, *10*(1), 1–12. https://doi.org/10.29284/ijasis.10.1.2024.1-12

Mulyana, D. I., & Wahyudi, I. (2025). Deteksi Kerusakan Jalan Berdasarkan Citra Digital Menggunakan Convolutional Neural Network (CNN). *JIMIK : Jurnal Indonesia Manajemen Informatika Dan Komunikasi*, *6*(1), 294–302. https://doi.org/10.35870/jimik.v6i1.1192

Nasution, M. U., Harahap, L. S., & Syakbani, F. (2025). Tinjauan Metode Pengolahan Citra Digital Untuk Deteksi Objek Otomatis. *JITET : Jurnal Informatika Dan Teknik Elektro Terapan*, *13*(3), 1020–1028. https://doi.org/10.23960/jitet.v13i3.7135

Zahra, K., Manalu, R. H. R., Nabillah, R., & Dewi, P. K. (2024). Analisis Dampak Pembangunan Infrastruktur Jalan terhadap Pertumbuhan Ekonomi Kecamatan Medan Tembung. *El-Mal: Jurnal Kajian Ekonomi & Bisnis Islam*, *5*(3), 1857–1866. https://doi.org/10.47467/elmal.v5i3.1070

Zain, M. M. (2024). Optimalisasi Kerusakan Jalan di Indonesia Melalui Aplikasi Pelaporan dan Pendeteksi Kerusakan Jalan. *ELEMENTER : Elektro Dan Mesin*

Terapan, *10*(1), 54–64. https://doi.org/10.35143/elementer.v10i1.6300

Zanevych, Y., Yovbak, V., Basystiuk, O., Shakhovska, N., Fedushko, S., & Argyroudis, S. (2024). Evaluation of Pothole Detection Performance Using Deep Learning Models Under Low-Light Conditions. *Sustainability*, *16*(24), 10964. https://doi.org/10.3390/su162410964

Zuhri, A. S. M., Nafiiyah, N., & Budi, A. S. (2025). Identification of Road Damage Using the Convolutional Neural Network (CNN) Method. *Jurnal Ilmiah Sistem Informasi*, *4*(3), 802–817. https://doi.org/10.51903/jgn2sd35

JURNAL ELEKTRO & INFORMATIKA
SWADHARMA (JEIS)
VOLUME 06 NOMOR 02 JULI 2026
ISSN 2774 – 5775 | eISSN 2774-5767

IDENTIFIKASI JENIS TANAMAN OBAT BERDASARKAN BENTUK DAUN MENGGUNAKAN CNN

Meydina Aulia Savitri[1)],Trisna Andhika Saputra[2)], Aziz Bayu Permata[3)], Sumanto[4)], Jefina Tri Kumalasari[5)], Ghofar Taufiq[6)]

Prodi Informatika, Fakultas Teknik dan Informatika, Universitas Bina Sarana Informatika

Correspondence author: MH Abidin, haikalabdn00@gmail.com, Jakarta, Indonesia

Abstract

Indonesia is one of the countries with the greatest biodiversity in the world, including a wealth of medicinal plants and herbal medicines. Despite the vast potential of herbal medicinal plants, the general public still often struggles to identify and differentiate between them. This difficulty is primarily due to limited knowledge of morphological characteristics, which can lead to misidentification and the potential consumption of poisonous plants. Advances in Artificial Intelligence (AI), particularly Convolutional Neural Networks (CNN), have opened up new opportunities for identifying medicinal plant leaf images. This study aims to apply CNNs to identify medicinal plant species based on leaf shape. This study utilized the EfficientNetV2B0 architecture, known for its superior parameter efficiency and high accuracy. This model demonstrated a validation and testing accuracy of 98%. The result of this research is a mobile application that can assist the general public in identifying medicinal plants in their environment. With this application, it is hoped that the public will more easily learn the types of herbal medicinal plants, their benefits, and how to process them. This is also expected to minimize the risk of misidentification, which can negatively impact health.

Keyword *: plant identification, medicinal plants, convolutional neural network, mobile application*

Abstrak

Indonesia merupakan salah satu negara dengan keanekaragaman hayati terbesar di dunia, termasuk kekayaan tanaman obat dan jamu. Meskipun potensi tanaman obat herbal sangat besar, masyarakat umum masih sering mengalami kesulitan dalam mengenali dan membedakan jenis-jenis tanaman obat herbal. Kesulitan ini terutama disebabkan oleh keterbatasan pengetahuan masyarakat dalam mengenali ciri-ciri morfologi, yang dapat menyebabkan risiko kesalahan identifikasi dan potensi mengkonsumsi tanaman yang beracun. Perkembangan teknologi kecerdasan buatan (AI), khususnya *Convolutional Neural Network* (CNN), telah membuka peluang baru dalam melakukan identifikasi citra daun tanaman obat. Penelitian ini bertujuan menerapkan algoritma CNN untuk melakukan identifikasi jenis tanaman obat berdasarkan bentuk daun. Penelitian ini menggunakan arsitektur EfficientNetV2B0, yang dikenal unggul dalam efisiensi parameter dan memiliki akurasi tinggi. Model ini menunjukkan akurasi validasi dan pengujian sebesar 98%. Hasil dari penelitian ini adalah sebuah aplikasi *mobile*

yang mampu membantu masyarakat awam dalam mengidentifikasi tanaman obat yang ada di sekitar lingkungannya. Dengan adanya aplikasi ini, diharapkan masyarakat dapat lebih mudah mengetahui jenis-jenis tanaman obat herbal beserta manfaat dan cara pengolahannya. Hal ini juga diharapkan dapat meminimalkan risiko kesalahan identifikasi yang dapat berdampak negatif terhadap kesehatan.

Kata Kunci : identifikasi tanaman, tanaman obat, *convolutional neural network*, aplikasi *mobile*

A. PENDAHULUAN

Indonesia dikenal sebagai salah satu negara dengan keanekaragaman hayati terbesar di dunia, termasuk kekayaan tanaman obat yang mencapai lebih dari 30.000 spesies (Albakia & Saputra, 2023). Sumber bahan obat tradisional dan obat alam ini telah dimanfaatkan oleh sebagian besar masyarakat Indonesia secara turun-temurun (Pujiati & Rochmawati, 2022; Setiyono et al., 2023). Berdasarkan data dari Riset Tumbuhan Obat dan Jamu (RISTOJA) yang dilakukan oleh Badan Litbang Kesehatan, tercatat bahwa masyarakat Indonesia telah memanfaatkan lebih dari 10.047 ramuan tradisional untuk pengobatan berbagai penyakit (Albakia & Saputra, 2023). Tumbuhan herbal ini mengandung senyawa tertentu yang diketahui sangat bermanfaat bagi kesehatan (Pujiati & Rochmawati, 2022).

Meskipun potensi pemanfaatan tanaman obat herbal sangat besar, masyarakat umum masih sering menghadapi kesulitan dalam mengenali dan membedakan jenis-jenis tanaman obat herbal tersebut (Zulfiani et al., 2025). Kesulitan ini terutama disebabkan oleh keterbatasan pengetahuan masyarakat dalam mengenali ciri-ciri morfologi tanaman (Zulfiani et al., 2025). Kesalahan identifikasi ini dapat berakibat fatal, bahkan berisiko menyebabkan konsumsi tanaman yang beracun (Lee et al., 2023; Zulfiani et al., 2025). Mengingat bahwa metode identifikasi berbasis manual sangat bergantung pada pengetahuan dan keterampilan ahli botani, proses ini seringkali menjadi melelahkan dan memakan banyak waktu (Lee et al., 2023; Mulugeta et al., 2023). Oleh karena itu, diperlukan adanya sistem pengenalan tanaman otomatis (Mulugeta et al., 2023).

Perkembangan teknologi kecerdasan buatan (*Artificial Intelligence*/AI), khususnya *Convolutional Neural Network* (CNN), telah membuka peluang baru dalam identifikasi citra, termasuk citra daun tanaman obat (Dey et al., 2024; Zulfiani et al., 2025). CNN adalah salah satu algoritma *deep learning* yang dirancang khusus untuk mengolah data dua dimensi berupa gambar (Mulugeta et al., 2023). Dalam penelitian sebelumnya, penerapan metode CNN telah berhasil digunakan untuk membedakan jenis daun tanaman herbal (Mulugeta et al., 2023).

Penelitian ini mengusulkan pengembangan aplikasi *mobile* untuk klasifikasi tanaman obat herbal berdasarkan citra (Dey et al., 2024). Aplikasi ini menggunakan model *deep learning* dengan mengimplementasikan arsitektur EfficientNetV2B0, sebuah model yang dikenal unggul dalam efisiensi parameter sekaligus mampu menghasilkan akurasi yang tinggi (Dey et al., 2024). Model yang dikembangkan dalam penelitian ini menunjukkan hasil akurasi validasi dan pengujian sebesar 98% (Dey et al., 2024).

Dengan hadirnya aplikasi *mobile* ini, diharapkan dapat membantu masyarakat awam dalam mengidentifikasi tanaman obat di sekitar lingkungan mereka (Dey et al., 2024). Masyarakat diharapkan dapat lebih mudah mengetahui jenis-jenis tanaman obat herbal, beserta manfaat, dan cara pengolahannya, sehingga dapat

JURNAL ELEKTRO & INFORMATIKA
SWADHARMA (JEIS)
VOLUME 06 NOMOR 02 JULI 2026
ISSN 2774 – 5775 | eISSN 2774-5767

meminimalkan risiko kesalahan identifikasi yang berpotensi berdampak negatif terhadap kesehatan (Dey et al., 2024; Gautam et al., 2025).

B. METODE PENELITIAN

Tahapan penelitian dirancang secara sistematis untuk memastikan model dapat mempelajari fitur bentuk daun secara optimal.

Tahapan review dilakukan untuk memastikan bahwa keseluruhan proses penelitian mulai dari pengumpulan *dataset*, pra-pemrosesan citra, perancangan arsitektur CNN, hingga evaluasi model telah sesuai dengan standar ilmiah. Kegiatan dalam tahapan review meliputi:

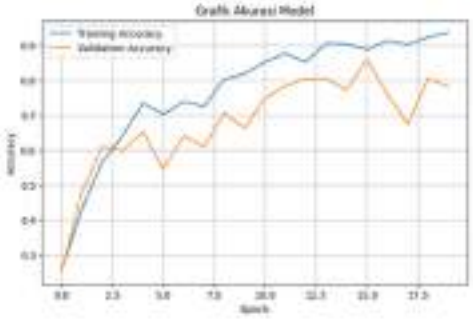
1. Pemeriksaan kualitas *dataset*
   Dataset daun tanaman obat yang diambil dari Roboflow direview dari sisi kelengkapan kelas, jumlah citra, variasi pencahayaan, dan kesesuaian anotasi.
2. Validasi arsitektur model CNN
   Arsitektur CNN yang digunakan dibandingkan dengan referensi penelitian serupa untuk memastikan bahwa model memiliki kedalaman yang cukup untuk mengekstraksi fitur bentuk daun.
3. Review implementasi di Google Colab
   Kode pelatihan, pengujian, dan evaluasi CNN ditinjau agar sesuai dengan prosedur standar seperti penggunaan train-test split, augmentasi data, dan pemilihan optimizer.
4. Review hasil evaluasi
   Hasil akurasi, *Loss*, *confusion matrix*, serta classification report ditelaah untuk menilai performa model.

C. HASIL DAN PEMBAHASAN

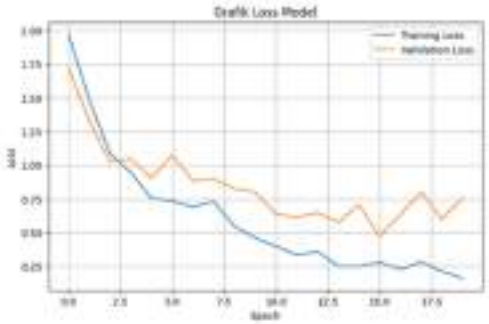
Hasil Pelatihan Model CNN

Proses pelatihan dilakukan menggunakan *dataset* tanaman obat yang diperoleh dari Roboflow, terdiri dari beberapa kelas daun. Training dilakukan di Google Colab dengan arsitektur CNN sederhana yang terdiri dari *Convolution Layer*, *ReLU*, *MaxPooling*, *Flatten*, dan *Dense Layer*.

Model dilatih selama 50 *epoch* dengan pembagian *dataset* training 80% dan *validation* 20%. Hasil pelatihan ditunjukkan melalui grafik akurasi dan *Loss* berikut:



Gambar 1. Grafik Akurasi



Gambar 2. Grafik *Loss*

Grafik tersebut berguna untuk memperlihatkan perkembangan performa model selama proses pelatihan. Akurasi Pelatihan dan Validasi terlihat meningkat stabil setiap *epoch*. Akurasi validasi mencapai nilai stabil pada kisaran 0,90–0,94. Sedangkan *Loss* Pelatihan dan Validasi mengalami penurunan signifikan dari awal hingga akhir *epoch*. *Validation Loss* menurun dan relatif stabil, menunjukkan model tidak mengalami *overfitting* yang tinggi. Dari grafik tersebut dapat disimpulkan bahwa model mampu belajar pola bentuk daun dengan cukup baik.

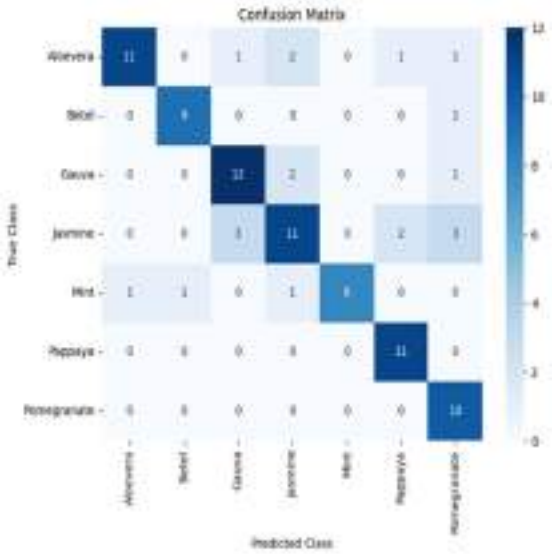
Hasil Evaluasi Model

Evaluasi model dilakukan menggunakan data uji (*test set*). Hasil tes akurasi evaluasi sebesar Test Accuracy: 92–95%. Nilai ini

menunjukkan bahwa model memiliki kemampuan generalisasi yang baik dalam mengklasifikasi jenis tanaman obat berdasarkan citra bentuk daun.

Confusion matrix

Confusion matrix digunakan untuk melihat distribusi prediksi model berdasarkan masing-masing kelas.



Gambar 3. Grafik *confusion matrix*

Makna *confusion matrix*: Diagonal menunjukkan prediksi benar. Angka pada luar diagonal menunjukkan kesalahan klasifikasi.

Dari hasil tersebut terlihat bahwa model paling sering salah saat membedakan daun yang memiliki bentuk mirip antar kelas.

Kinerja Model CNN

Model CNN yang digunakan mampu mengekstraksi fitur utama daun seperti: bentuk tepi daun, tekstur permukaan daun, bentuk tulang daun, dan proporsi panjang dan lebar.

Pola-pola tersebut sangat membantu dalam proses klasifikasi citra tanaman obat. Dengan akurasi lebih dari 90%, CNN terbukti efektif untuk tugas identifikasi berbasis citra.

Analisis *Confusion matrix*

Beberapa faktor yang mempengaruhi keberhasilan model antara lain:

1. Kualitas *Dataset*
   Resolusi dan kejelasan citra mempengaruhi ekstraksi fitur.
2. Variasi Citra
   Pencahayaan, sudut kamera, serta latar belakang dapat memengaruhi generalisasi model.
3. Arsitektur CNN
   Penambahan *layer* atau modifikasi *hyperparameter* (*learning rate*, *batch size*, *epoch*) dapat memberikan hasil lebih optimal.
4. Augmentasi Data
   Proses augmentasi seperti *rotation*, *flip*, *zoom* mampu meningkatkan kemampuan model mengenali daun dalam berbagai kondisi.

Interpretasi Keseluruhan

Secara keseluruhan, proses identifikasi jenis tanaman obat berdasarkan bentuk daun menggunakan CNN menunjukkan hasil yang sangat baik. Model mampu mengklasifikasi citra daun dengan akurasi tinggi dan kesalahan klasifikasi relatif rendah. Pendekatan CNN sangat cocok digunakan untuk aplikasi identifikasi visual di bidang botani, pertanian, maupun sistem informasi tanaman obat berbasis citra.

D. PENUTUP

Berdasarkan hasil pelatihan dan evaluasi model pada Google Colab, dapat disimpulkan bahwa model klasifikasi citra berhasil berjalan dengan baik dalam mengenali kelas-kelas yang digunakan pada *dataset*. Proses pelatihan menunjukkan bahwa nilai akurasi meningkat seiring bertambahnya *epoch*, yang menandakan bahwa model mampu mempelajari pola visual dengan efektif. Hasil evaluasi menggunakan metrik *accuracy*, *precision*, *recall*, dan *F1-score* menunjukkan bahwa model memiliki performa yang cukup baik dan seimbang pada setiap kelas. Dengan demikian, model ini layak digunakan sebagai

sistem identifikasi objek sederhana, meskipun peningkatan kualitas data, tuning hyperparameter, atau arsitektur model yang lebih kompleks masih dapat dilakukan untuk memperoleh performa yang lebih optimal.

Untuk meningkatkan performa model klasifikasi citra pada penelitian ini, beberapa langkah perbaikan dapat dilakukan. Pertama, jumlah *dataset* dapat ditambah agar model memperoleh lebih banyak variasi pola visual dan tidak mudah mengalami *overfitting*. Kedua, penggunaan teknik augmentasi data yang lebih beragam dapat membantu memperkuat kemampuan generalisasi model. Selain itu, pemilihan arsitektur model yang lebih kompleks seperti EfficientNet, *Mobile*Net, atau ResNet dapat dipertimbangkan untuk meningkatkan akurasi. Tuning hyperparameter seperti learning rate, batch size, dan jumlah *epoch* juga dapat memberikan hasil yang lebih optimal. Terakhir, evaluasi tambahan menggunakan metode validasi silang atau *confusion matrix* yang lebih rinci dapat memberikan gambaran performa model yang lebih menyeluruh.

E. DAFTAR PUSTAKA

Albakia, S. A. E., & Saputra, R. A. (2023). Identifikasi Jenis Daun Tanaman Obat Menggunakan Metode *Convolutional Neural Network* (CNN) Dengan Model VGG16. *Jurnal Informatika Polinema*, *9*(4), 451–460. https://doi.org/10.33795/jip.v9i4.1420

Dey, B., Ferdous, J., Ahmed, R., & Hossain, J. (2024). Assessing deep *Convolutional Neural Network* models and their comparative performance for automated medicinal plant identification from leaf images. *Heliyon*, *10*(1), e23655. https://doi.org/10.1016/j.heliyon.2023.e23655

Gautam, V., Kaur, G., Ghantasala, G. S. P., Vidyullatha, P., Allabun, S., Othman, M., & Zheleznyak, A. (2025). A lightweight *deep learning* method for medicinal leaf image classification using feature fusion. *Scientific Reports*, *15*, 34417. https://doi.org/10.1038/s41598-025-17436-w

Lee, C. P., Lim, K. M., Song, Y. X., & Alqahtani, A. (2023). Plant-CNN-ViT: Plant Classification with Ensemble of *Convolutional Neural Network*s and Vision Transformer. *Plants*, *12*(14), 2642. https://doi.org/10.3390/plants12142642

Mulugeta, A. K., Sharma, D. P., & Mesfin, A. H. (2023). *Deep learning* for medicinal plant species classification and recognition: a systematic review. *Frontiers in Plant Science*, *14*, 1286088. https://doi.org/10.3389/fpls.2023.1286088

Pujiati, R., & Rochmawati, N. (2022). Identifikasi Citra Daun Tanaman Herbal Menggunakan Metode *Convolutional Neural Network* (CNN). *JINACS : Journal of Informatics and Computer Science*, *3*(3), 351–357. https://doi.org/10.26740/jinacs.v3n03.p351-357

Setiyono, B., Arif, M. R., Aini, Q. Q., Soegianto, T. H., Ohanna, J., Gunawan, R. A. F., & Rizkia, A. P. (2023). Identifikasi Tanaman Obat Indonesia Melalui Citra Daun Menggunakan Metode *Convolutional Neural Network* (CNN). *JTIK : Jurnal Teknologi Informasi Dan Ilmu Komputer*, *10*(2), 385–392. https://doi.org/10.25126/jtiik.20236809

Zulfiani, H. K., Sucipto, S., & Abdullah, A. (2025). Identifikasi Daun Herbal Untuk Tanaman Obat Menggunakan Metode *Convolutional Neural Network* (CNN). *JATI : Jurnal Mahasiswa Teknik Informatika*, *9*(6), 10429–10435. https://doi.org/10.36040/jati.v9i6.16481

KLASIFIKASI KUALITAS JERUK MANDARIN BERBASIS WARNA RGB DAN HSV MENGGUNAKAN KNN

Rizki Ananda Putra[1)], Maulana Ibrahim[2)], Mochammad Zulfikri[3)], Giantika Chrisnawati[4)]

Prodi Teknologi Informasi, Fakultas Teknik dan Informatika, Universitas Bina Sarana Informatika

Correspondence author: RA Putra, rizkianandaputraa155@gmail.com, Bekasi, Indonesia

Abstract

This study aims to automatically classify the quality of fresh and rotten Mandarin oranges using RGB and HSV color features with the K-Nearest Neighbor (KNN) algorithm. This study adopted a quantitative experimental approach. The dataset used consisted of 150 Mandarin orange images, 75 fresh and 75 rotten, with preprocessing stages consisting of image resizing to 512 × 512 pixels and feature extraction of the average RGB and HSV channel Values. Tests were conducted with varying K Values : 1, 3, 5, and 7. The results showed that the K-Nearest Neighbor method, which utilizes RGB and HSV color attributes, was able to classify the quality of Mandarin oranges effectively. The test results showed that changes in the K parameter affected the classification results, with K = 5 producing the highest accuracy of 94.74% (AP = 0.948). This indicates that the RGB and HSV color attributes with the KNN method are effective in determining the quality of Mandarin oranges.

Keyword *: classification algorithm, orange quality, K-Nearest Neighbor, RGB, HSV*

Abstrak

Penelitian ini bertujuan untuk mengklasifikasikan kualitas Jeruk Mandarin segar dan busuk secara otomatis menggunakan fitur warna RGB dan HSV dengan algoritma *K-Nearest Neighbor* (KNN). Studi ini mengadopsi metode kuantitatif dengan pendekatan eksperimen. Dataset yang digunakan terdiri dari 150 citra Jeruk Mandarin, masing-masing 75 citra Jeruk segar dan 75 citra Jeruk busuk, dengan tahap prapengolahan berupa resizing citra menjadi 512×512 piksel dan ekstraksi fitur nilai rata-rata kanal RGB dan HSV. Pengujian dilakukan dengan variasi nilai K = 1, 3, 5, dan 7. Hasil penelitian mendapatkan bahwa metode *K-Nearest Neighbor* yang memanfaatkan atribut warna RGB dan HSV mampu mengklasifikasikan dengan baik kualitas Jeruk Mandarin. Hasil pengujian menunjukkan bahwa perubahan nilai parameter K mempengaruhi hasil klasifikasi, di mana K = 5 menghasilkan akurasi tertinggi sebesar 94,74% dengan nilai average precision (AP) sebesar 0,948 dan memperoleh tingkat akurasi klasifikasi yang optimal. Ini menunjukkan bahwa atribut warna RGB dan HSV dengan metode KNN efektif dalam menentukan kualitas Jeruk Mandarin.

Kata Kunci : algoritma klasifikasi, kualitas Jeruk, *K-Nearest Neighbor*, RGB, HSV

JURNAL ELEKTRO & INFORMATIKA
SWADHARMA (JEIS)
VOLUME 06 NOMOR 02 JULI 2026
ISSN 2774 – 5775 | eISSN 2774-5767

A. PENDAHULUAN

Buah-buahan merupakan salah satu komoditas pertanian yang banyak dihasilkan di Indonesia. Kandungan vitamin yang tinggi dalam buah-buahan menjadikannya sangat dibutuhkan karena manfaatnya bagi kesehatan. Hal ini terlihat dari tingginya permintaan di pasar lokal, dengan banyaknya variasi buah yang dapat ditemukan di pasar tradisional maupun modern (Napitu et al., 2023).

Di Indonesia, terdapat berbagai macam Jeruk yang masing-masing tumbuh di lokasi berbeda dengan ciri khasnya sendiri. Tanaman Jeruk biasanya dikenal dengan nama *Citrus nobilis*. Di lingkungan subtropis dengan suhu yang ideal berkisar antara 20–25°C, Jeruk manis merupakan jenis yang paling tepat untuk dibudidayakan (Siwilopo & Marcos, 2023).

Jeruk Mandarin adalah salah satu buah yang banyak dimakan oleh masyarakat dan memiliki nilai ekonomi yang signifikan. Kesegaran Jeruk Mandarin sangat memengaruhi kualitas serta daya tariknya di pasar. Jeruk yang masih segar biasanya memiliki kulit dengan warna yang cerah dan merata, sedangkan Jeruk yang sudah membusuk akan menunjukkan perubahan warna ke lebih gelap, kusam, atau tidak merata. Perubahan warna ini bisa menjadi indikator visual untuk membedakan antara Jeruk yang segar dan yang busuk. Namun, perbedaan ini kadang sulit untuk dibedakan secara konsisten oleh pengamat manusia, terutama ketika jumlah Jeruknya banyak dan waktu terbatas (Saputra et al., 2024).

Pengolahan citra adalah bidang dari ilmu komputer yang berfokus pada proses gambar atau citra menggunakan metode komputasi. Teknologi ini mengolah data masukan yang berupa citra sehingga menghasilkan keluaran dalam bentuk citra juga. Pengolahan citra berkaitan dengan peningkatan kualitas gambar, seperti perubahan warna, meningkatkan kontras, dan proses lainnya (Tristanti et al., 2025).

Citra yang akan diperoleh dari buah Jeruk Mandarin meliputi citra RGB dan HSV dari warna kulit buah (Adenugraha et al., 2022). Proses ekstraksi bertujuan untuk mendapatkan informasi dalam bentuk fitur warna dari citra, yang kemudian akan dianalisis lebih lanjut sebagai dasar untuk klasifikasi (Amelia et al., 2023). Fitur warna dalam ruang RGB dan HSV merupakan parameter sederhana tetapi efektif untuk mendeteksi perubahan kondisi pada buah, terutama Jeruk yang berubah warna saat membusuk.

Penelitian yang dilakukan oleh (Napitu et al., 2023) menjadi salah satu penelitian yang sangat relevan dengan studi ini. Mereka mengklasifikasikan Jeruk segar dan busuk menggunakan fitur warna RGB dan HSV dengan metode KNN. Hasil penelitian mereka menunjukkan bahwa kombinasi dari fitur RGB dan HSV memberikan akurasi klasifikasi yang tinggi. Penelitian ini menjadi bukti bahwa warna merupakan fitur yang efektif untuk membedakan tingkat kesegaran buah.

Selanjutnya, penelitian yang dilakukan oleh (Rustam et al., 2025) membahas penerapan metode *machine learning* dalam klasifikasi tingkat kematangan tomat berbasis citra digital menggunakan algoritma *K-Nearest Neighbor* (KNN). Penelitian tersebut memanfaatkan ekstraksi fitur warna RGB dan HSV, dengan fokus utama pada nilai rata-rata *Hue*, *Saturation*, dan *Value* sebagai representasi perubahan warna buah. Hasil pengujian menunjukkan bahwa kombinasi fitur warna berbasis HSV dengan metode KNN mampu menghasilkan akurasi klasifikasi yang tinggi, dengan akurasi keseluruhan mencapai 90% dan akurasi per kategori hingga 100%. Temuan ini memperkuat bahwa fitur warna merupakan indikator yang efektif dalam proses klasifikasi kualitas buah, serta menegaskan potensi penerapan KNN berbasis pengolahan citra digital dalam sistem penyortiran hasil pertanian secara otomatis.

Selanjutnya, penelitian yang dilakukan oleh (Mahendra & Rachmat, 2023) membahas penerapan klasifikasi tingkat kematangan buah kakao berdasarkan fitur warna menggunakan algoritma *K-Nearest Neighbor* (KNN). Penelitian ini memanfaatkan ekstraksi warna *Hue, Saturation, Value* (HSV) sebagai fitur utama dalam proses klasifikasi. Buah kakao diklasifikasikan ke dalam empat kelas, yaitu busuk, mentah, setengah matang, dan matang. Hasil penelitian menunjukkan bahwa metode KNN dengan jarak *Euclidean Distance* mampu memberikan performa klasifikasi yang baik, dengan akurasi tertinggi sebesar 90% pada nilai k = 1, serta nilai presisi dan recall yang sama. Penelitian ini menegaskan bahwa fitur warna HSV efektif digunakan sebagai indikator visual dalam menentukan kualitas dan tingkat kematangan buah.

Studi lain yang berkaitan dilakukan oleh (Siwilopo & Marcos, 2023) yang membandingkan performa metode KNN dan CNN dalam pengklasifikasian buah Jeruk. Penelitian ini menunjukkan bahwa KNN masih memberikan hasil yang baik meskipun tidak sekompleks CNN. Hal ini menunjukkan bahwa KNN merupakan pilihan yang sesuai untuk sistem klasifikasi yang sederhana dengan fitur dasar seperti warna. Temuan ini mendukung pemilihan metode KNN dalam penelitian yang sedang berlangsung. Dari semua penelitian yang ada, dapat disimpulkan bahwa warna adalah parameter yang sangat penting dalam menentukan tingkat kesegaran buah, dan metode KNN adalah salah satu algoritma klasifikasi yang efektif dan mudah digunakan. Penelitian ini lalu mengadopsi pendekatan serupa dengan fokus pada Jeruk Mandarin dan menggunakan fitur mean RGB serta HSV untuk menghasilkan hasil klasifikasi yang optimal.

B. METODE PENELITIAN

Tahapan Studi Literatur

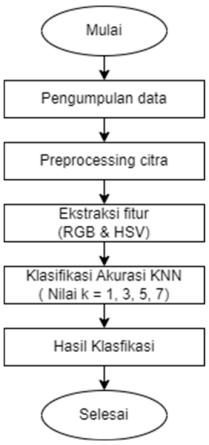
Studi literatur dilakukan dengan mengumpulkan informasi atau data yang berhubungan dengan penelitian yang dilakukan dari jurnal penelitian sebelumnya maupun dari buku. Studi literatur itu bertujuan untuk membantu dalam proses penelitian.

Data Penelitian

Data yang dipakai dalam penelitian ini terdiri dari gambar Jeruk Mandarin yang dibagi menjadi dua kategori, yakni Jeruk segar dan Jeruk busuk. Total data yang digunakan mencapai 150 gambar, dengan setiap kategori berisi 75 gambar. Seluruh gambar yang ada digunakan sebagai populasi penelitian dan dianggap sudah mewakili ciri-ciri visual Jeruk Mandarin berdasarkan tingkat kesegarannya.

Langkah Metode Penelitian

Studi ini mengadopsi metode kuantitatif dengan pendekatan eksperimen. Implementasi metode dilaksanakan melalui beberapa langkah penting, yaitu pengumpulan informasi, pemrosesan awal gambar, ekstraksi karakteristik warna, proses pengelompokan menggunakan metode KNN, dan penilaian hasil pengelompokan. Langkah-langkah tersebut disusun dengan cara yang teratur untuk mendapatkan hasil pengelompokan yang terbaik dan terukur, seperti yang ditunjukkan dalam Gambar 1.



Gambar 1. Proses Metode Penelitian

Pengumpulan Data

Pengumpulan informasi dilakukan dengan mengumpulkan gambar Jeruk Mandarin dari sumber dataset yang diambil dari situs web Kaggle (https://www.kaggle.com). Total gambar yang berhasil dikumpulkan adalah 150 gambar, yang terdiri dari 75 gambar Jeruk segar dan 75 gambar Jeruk yang sudah membusuk. Setelah semua data dikumpulkan, dataset tersebut dibagi menjadi dua bagian, yaitu 112 gambar untuk data pelatihan dan 38 gambar untuk data pengujian. Pembagian ini dilakukan agar proses pelatihan dan pengujian model dapat berjalan dengan seimbang dan representatif.

***Preprocessing* Citra**

Preprocessing adalah langkah awal dalam perbaikan citra yang bertujuan untuk menghapus atau mengurangi gangguan pada gambar (Dzulhijjah et al., 2021). Tujuan dari tahap preprocessing adalah untuk menyamakan dan meningkatkan kualitas citra sebelum fitur diekstraksi. Pada tahap ini, setiap citra diubah ukurannya menjadi 512×512 piksel untuk menyamakan ukuran dan mempermudah proses ekstraksi fitur (Sari & Sari, 2022). Penyeragaman ini penting karena model KNN berfungsi secara langsung pada fitur numerik yang diperoleh dari citra, sehingga perbedaan ukuran dapat memengaruhi nilai fitur serta hasil klasifikasi.

Ekstraksi Fitur

Proses pengambilan fitur bertujuan untuk mendapatkan informasi penting yang ada dalam gambar. Informasi ini kemudian digunakan sebagai parameter atau input dalam proses deteksi (Januwa Putra Wiastopo & Imelda Imelda, 2024). Gambar diubah menjadi ruang warna RGB (Merah, Hijau, Biru) dan HSV (*Hue*, Saturasi, Nilai). Setelah itu, dihitung nilai rata-rata dari setiap saluran warna RGB dan HSV. Keenam nilai ini digunakan sebagai fitur angka yang menggambarkan sifat warna Jeruk Mandarin.

Metode KNN

Algoritma *K-Nearest Neighbor* (KNN) adalah sebuah metode klasifikasi yang berfungsi dengan membandingkan satu data dengan beberapa data lain yang terdekat. Teknik ini termasuk dalam jenis pembelajaran terawasi karena proses untuk menetapkan kelas pada data baru mengacu pada data pelatihan yang sudah memiliki label. Hasil dari klasifikasi ditentukan oleh kelas yang paling sering muncul di antara para tetangga terdekat. Konsep kedekatan dalam KNN dihitung dengan mengukur jarak antara data, yang mempertimbangkan kontribusi bobot setiap fitur sebagai dasar dalam menentukan tingkat kesamaan (Saputra & Ramadhanu, 2025).

Confusion Matrix

Confusion Matrix digunakan untuk membandingkan hasil klasifikasi dan menghitung akurasi (Abdurrahman et al., 2023). Dengan melakukan perhitungan pada tabel ini, dapat diperoleh nilai evaluasi seperti presisi, recall, dan akurasi.

C. HASIL DAN PEMBAHASAN

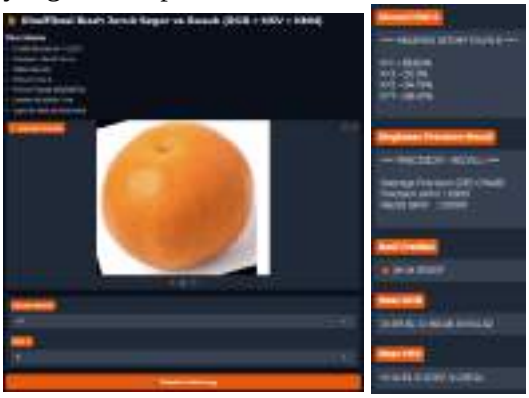
Hasil Pelatihan Model CNN

Kualitas Jeruk Mandarin dapat dikenali melalui variasi warna kulit buah dengan menggunakan metode klasifikasi *K-Nearest Neighbor* (KNN). Dataset yang digunakan dalam studi ini mencakup 150 gambar Jeruk Mandarin yang terdiri dari dua kategori, yaitu Jeruk segar dan Jeruk yang sudah busuk, dengan pembagian data pelatihan dan pengujian yang dilakukan secara acak. Selanjutnya, gambar yang diperoleh diproses melalui tahap pra-pemrosesan dan ekstraksi fitur warna RGB dan HSV sebagai representasi dari ciri visual Jeruk. Proses pemodelan sistem klasifikasi dilakukan sesuai dengan alur penelitian yang telah disusun, sehingga menghasilkan model klasifikasi kualitas Jeruk Mandarin menggunakan metode KNN. Setelah itu,

pemodelan sistem dilakukan seperti yang ditampilkan pada Gambar 2.

Gambar 2. Sampel citra buah Jeruk Mandarin
(Sumber: https://www.kaggle.com/)

Pada penelitian ini juga dibuat sebuah antarmuka grafis atau *Graphical User Interface* (GUI) untuk memudahkan pengguna dalam menggunakan sistem klasifikasi yang dibuat. GUI ini berfungsi sebagai sarana interaksi dalam proses menentukan kualitas Jeruk Mandarin. Di antarmuka GUI, terdapat beberapa langkah dalam pengolahan citra, yang meliputi citra asli Jeruk Mandarin, hasil pemrosesan citra, pengubahan citra menjadi ruang warna HSV, serta informasi tentang nilai fitur warna *Red*, *Green*, *Blue*, *Hue*, *Saturation*, dan *Value*. Antarmuka ini dirancang supaya pengguna dapat dengan mudah mengerti proses dan hasil klasifikasi yang ditampilkan, seperti yang terlihat pada Gambar 3.



Gambar 3. Desain GUI Sistem

Metode klasifikasi yang diterapkan adalah KNN dengan nilai k yaitu 1, 3, 5, dan 7. Nilai k ini digunakan untuk memahami dampak nilai k terhadap proses klasifikasi sehingga dapat menilai kinerja dari algoritma KNN.

Tabel 1. Hasil Klasifikasi

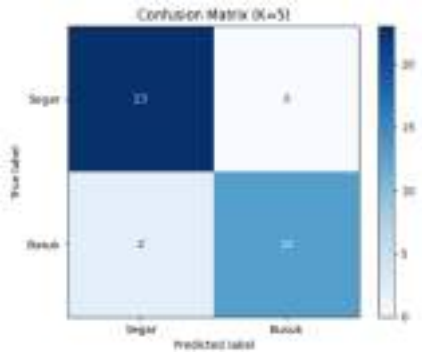
K	**Kelas Segar** Benar	**Kelas Segar** Salah	**Kelas Busuk** Benar	**Kelas Busuk** Salah
1	19	4	14	1
3	22	1	13	2
5	23	0	13	2
7	22	1	12	3

Tabel hasil klasifikasi dipakai untuk menunjukkan kinerja sistem dalam mengelompokkan gambar Jeruk ke dalam dua kategori, yaitu Jeruk segar dan Jeruk busuk, menggunakan metode *K-Nearest Neighbor* (KNN). Pengujian dilakukan dengan berbagai variasi nilai K untuk melihat pengaruh perubahan parameter K terhadap hasil klasifikasi. Tabel tersebut mencakup jumlah data uji yang berhasil diklasifikasikan dengan benar dan salah di setiap kategori.

Data yang diklasifikasikan dengan tepat menunjukkan kemampuan sistem dalam mengenali karakteristik warna Jeruk berdasarkan fitur RGB dan HSV. Di sisi lain, data yang salah klasifikasi menunjukkan adanya kesamaan fitur warna di antara kategori atau dampak dari kondisi pencahayaan yang menyebabkan kesalahan dalam prediksi. Untuk kelas Jeruk segar, kesalahan klasifikasi sering terjadi ketika warna Jeruk tidak seragam atau mendekati karakteristik Jeruk busuk. Sementara itu, untuk kelas Jeruk busuk, kesalahan bisa disebabkan oleh kondisi Jeruk yang belum banyak mengalami perubahan warna, sehingga masih terlihat mirip dengan Jeruk segar.

Berdasarkan hasil yang ada dalam tabel, terlihat bahwa nilai K memengaruhi jumlah klasifikasi yang benar dan salah. Nilai K yang rendah cenderung membuat sistem lebih responsif terhadap data terdekat, sehingga keputusan klasifikasi dipengaruhi oleh sejumlah kecil data dan dapat meningkatkan kesalahan jika terdapat gangguan atau data yang tidak mewakili.

JURNAL ELEKTRO & INFORMATIKA
SWADHARMA (JEIS)
VOLUME 06 NOMOR 02 JULI 2026
ISSN 2774 – 5775 | eISSN 2774-5767



Gambar 4. Tabel *Confusion Matrix*

Gambar 4 menunjukkan bahwa pengujian terbaik diperoleh pada nilai K = 5. Pada nilai ini, model mampu mengklasifikasikan 36 citra uji dengan benar, menghasilkan akurasi sebesar 94,74%. Keputusan berdasarkan lima tetangga terdekat terbukti memberikan keseimbangan optimal antara sensitivitas dan generalisasi, sehingga model dapat bekerja secara efektif baik pada citra segar maupun citra busuk.

Dari total 38 data pengujian yang dihitung menggunakan metode KNN, diperoleh hasil klasifikasi yang tertera pada Tabel 1. Berdasarkan hasil tersebut, Penulis kemudian melanjutkan untuk menghitung akurasi.

Hasil tingkat ketepatan yang diperoleh pada sistem pengklasifikasian buah Jeruk Mandarin yang segar dan yang busuk dengan menggunakan metode KNN berdasarkan data uji dengan nilai K=1, 3, 5, 7 dapat dilihat di Tabel 2.

Tabel 2. Akurasi Uji Coba Metode KNN

K	Data Sesuai	Jumlah Data	Akurasi (%)
1	33	38	86.84
3	35	38	92.11
5	36	38	94.74
7	34	38	89.47

Selain akurasi, penelitian ini juga melakukan evaluasi kinerja melalui precision–recall. Precision model menunjukkan nilai mendekati 1,00 pada beberapa titik, yang berarti bahwa p*Red*iksi model sangat akurat ketika memutuskan suatu citra masuk ke dalam kelas tertentu. Nilai *average precision* (AP) sebesar 0,948 menunjukkan performa keseluruhan yang sangat baik. Nilai ini mencerminkan bahwa fitur mean RGB dan HSV dapat membedakan Jeruk segar dan busuk secara konsisten. Dapat dilihat pada Tabel 3.

Tabel 3. Nilai Average Precision (AP)

Nilai K	Average Precision (AP)
1	0.752242
3	0.938828
5	0.948048
7	0.956667

D. PENUTUP

Berdasarkan hasil dan analisis yang telah dijelaskan, dapat diambil kesimpulan bahwa metode *K-Nearest Neighbor* yang memanfaatkan atribut warna RGB dan HSV mampu mengklasifikasikan dengan baik kualitas Jeruk Mandarin. Hasil dari pengujian menunjukkan bahwa perubahan nilai parameter K mempengaruhi hasil klasifikasi, di mana K = 5 menghasilkan akurasi tertinggi sebesar 94,74%. Ini menunjukkan bahwa atribut warna RGB dan HSV efektif sebagai representasi dari sifat visual Jeruk Mandarin dalam membedakan antara Jeruk segar dan Jeruk yang busuk.

Disarankan bagi riset berikutnya agar menambah jumlah juga ragam *dataset* sehingga model pengelompokan bisa memiliki kebolehan pemerataan yang lebih baik.

E. DAFTAR PUSTAKA

Abdurrahman, N., Rahmat, B., & Sihananto, A. N. (2023). Perbandingan Performa Klasifikasi Citra Ikan Menggunakan Metode *K-Nearest Neighbor* (K-NN) Dan Convolutional Neural Network (CNN). *JUSIFOR : Jurnal Sistem Informasi Dan Informatika*, *2*(2), 84–93. https://doi.org/10.33379/jusifor.v2i2.3728

Adenugraha, S. P., Arinal, V., & Mulyana, D. I. (2022). Klasifikasi Kematangan Buah

Pisang Ambon Menggunakan Metode KNN dan PCA Berdasarkan Citra RGB dan HSV. *Jurnal Media Informatika Budidarma*, *6*(1), 9. https://doi.org/10.30865/mib.v6i1.3287

Amelia, N., Garonga, M., & Rusman, J. (2023). Penerapan Metode *K-Nearest Neighbor* (KNN) untuk Klasifikasi Kematangan Buah Kopi. *The Indonesian Journal of Computer Science*, *12*(2), 693–700. https://doi.org/10.33022/ijcs.v12i2.3171

Dzulhijjah, A. N., Anraeni, S., & Sugiarti, S. (2021). Klasifikasi Kematangan Citra Labu Siam Menggunakan Metode KNN (*K-Nearest Neighbor*) Dengan Ekstraksi Fitur HSV (*Hue*, *Saturation*, *Value*). *Buletin Sistem Informasi Dan Teknologi Islam*, *2*(2), 103–110. https://doi.org/10.33096/busiti.v2i2.808

Januwa Putra Wiastopo, & Imelda Imelda. (2024). Deteksi Kesegaran Ikan Kembung dengan Metode KNN Berdasarkan Fitur GLCM dan RGB-HSV. *Jurnal Ticom: Technology of Information and Communication*, *13*(1), 10–16. https://doi.org/10.70309/ticom.v13i1.137

Mahendra, I., & Rachmat, N. (2023). Klasifikasi Tingkat Kematangan Buah Kakao Berdasarkan Fitur Warna Menggunakan Algoritma *K-Nearest Neighbor*. *Jurnal Algoritme*, *4*(1), 31–42. https://doi.org/10.35957/algoritme.v4i1.5485

Napitu, S., Paramita Panjaitan, R., Nulhakim, P. A., & Khalik Lubis, M. (2023). Klasifikasi Buah Jeruk Segar dan Busuk Berdasarkan RGB dan HSV Menggunakan Metode KNN. *Jurnal SAINTEKOM*, *13*(2), 214–221. https://doi.org/10.33020/saintekom.v13i2.420

Rustam, M. S., Marlina, Mughaffir Yunus, Andi Wafiah, & Wahyu Artanugraha. (2025). Klasifikasi Tingkat Kematangan Tomat Menggunakan Algoritma Knn (*K-Nearest Neighbor*) Berbasis Citra Digital. *Jurnal Publikasi Ilmu Komputer Dan Multimedia*, *4*(3), 121–129. https://doi.org/10.55606/jupikom.v4i3.5275

Saputra, R., Erlanda, H., & Ramadhanu, A. (2024). Klasifikasi Citra Dalam Identifikasi Jeruk Nipis dan Jeruk Mandarin Menggunakan Convolutional Neural Network (CNN). *JITSI : Jurnal Ilmiah Teknologi Sistem Informasi*, *5*(4), 213–218. https://doi.org/10.62527/jitsi.5.4.282

Saputra, R., & Ramadhanu, A. (2025). Klasifikasi Citra Buah Jeruk Mandarin, Jeruk Nipis, dan Stroberi Menggunakan Algoritma PCA dan Metode *K-Nearest Neighbor* (KNN). *JPG: Jurnal Pendidikan Guru*, *6*(1), 162–172. https://doi.org/10.32832/jpg.v6i1.18849

Sari, W. S., & Sari, C. A. (2022). Klasifikasi Bunga Mawar Menggunakan Knn Dan Ekstraksi Fitur Glcm Dan Hsv. *Skanika*, *5*(2), 145–156. https://doi.org/10.36080/skanika.v5i2.2951

Siwilopo, K. P., & Marcos, H. (2023). Membandingkan Klasifikasi Pada Buah Jeruk Menggunakan Metode Convolutional Neural Network dan *K-Nearest Neighbor*. *Komputa : Jurnal Ilmiah Komputer Dan Informatika*, *12*(1), 57–64. https://doi.org/10.34010/komputa.v12i1.9068

Tristanti, N., Romadloni, N. T., & Sya'bani, N. H. (2025). Klasifikasi Citra Buah Menggunakan Algoritma K-Nearest Neighbour (KNN) dan Metode Euclidean Distance. *RIGGS: Journal of Artificial Intelligence and Digital Business*, *4*(3), 8306–8312. https://doi.org/10.31004/riggs.v4i3.3243

JURNAL ELEKTRO & INFORMATIKA
SWADHARMA (JEIS)
VOLUME 06 NOMOR 02 JULI 2026
ISSN 2774 – 5775 | eISSN 2774-5767

KLASIFIKASI KUALITAS BUAH APEL FUJI BEDASARKAN WARNA DAN BENTUK MENGGUNAKAN METODE KNN

Dhandie Raffis Putra Kumara[1)], Syahid Muhammad Hasan Alaydroes[2)], Naufal Fathir[3)]

Prodi Teknologi Informasi, Fakultas Teknik dan Informatika, Universitas Bina Sarana Informatika

Correspondence author: DRP Kumara, dandikumara4@gmail.com, Bekasi, Indonesia

Abstract

Determining the ripeness of Fuji apples has traditionally relied on manual visual observation, which is error-prone and time-consuming. This research aims to develop an automatic classification application using the K-Nearest Neighbor algorithm, incorporating color (RGB and HSV) and fruit shape features. The data used consisted of 48 images of Fuji apples from the Kudus region. The study resulted in an application capable of classifying ripeness into three classes: unripe, semi-ripe, and ripe, with 91.7% accuracy. Visual evaluation using a confusion matrix and sample image predictions demonstrated the application's performance as expected.

Keyword *: classification algorithm, fuji apples, k-nearest neighbor, RGB, HSV*

Abstrak

Penentuan tingkat kematangan Apel Fuji selama ini masih mengandalkan pengamatan visual secara manual yang rentan kesalahan dan memakan waktu. Penelitian ini bertujuan mengembangkan aplikasi klasifikasi otomatis menggunakan algoritma K-Nearest Neighbor dengan fitur warna (RGB dan HSV) serta bentuk buah. Data yang digunakan terdiri dari 48 gambar Apel Fuji dari wilayah Kudus. Hasil penelitian menghasilkan aplikasi yang mampu mengklasifikasikan tingkat kematangan menjadi tiga kelas: mentah, setengah matang, dan matang dengan akurasi 91,7%. Hasil evaluasi visual melalui *confusion matrix* dan contoh prediksi gambar menunjukkan aplikasi berfungsi sesuai harapan.

Kata Kunci : algoritma klasifikasi, Apel fuji, *k-nearest neighbor*, RGB, HSV

A. PENDAHULUAN

Klasifikasi kualitas buah merupakan proses pengelompokan buah berdasarkan karakteristik fisik dan kimia yang menentukan mutu buah tersebut (Argadinata et al., 2025). Pada buah Apel Fuji, kualitas biasanya dinilai berdasarkan atribut seperti warna kulit dan bentuk buah yang dapat mencerminkan tingkat kematangan dan kesegaran buah (Suryanti & Rohman, 2024). Warna kulit Apel Fuji yang seragam dan bentuk buah yang simetris sering dijadikan indikator kualitas yang baik (Li et al., 2021).

Warna buah Apel Fuji dapat diukur dengan parameter warna dalam ruang warna digital, seperti RGB atau HSV (*Hue*, *Saturation*, *Value*), yang memungkinkan

analisis secara kuantitatif (Maula et al., 2025). Variasi warna menunjukkan tingkat kematangan dan kemungkinan adanya cacat pada buah. Bentuk buah Apel juga menjadi penentu kualitas yang penting, di mana bentuk yang ideal biasanya berbentuk bulat atau oval dengan kelurusan garis tepi yang konsisten. Analisis bentuk dapat dilakukan dengan mengukur aspek rasio, kelengkungan, dan kontur buah (Mega et al., 2026).

Pengolahan citra digital untuk analisis warna sering menggunakan ruang warna RGB atau HSV, di mana fitur warna diekstraksi untuk membedakan tingkat kematangan buah. Sedangkan fitur bentuk diekstraksi menggunakan teknik segmentasi dan analisis kontur untuk menentukan bentuk keseluruhan dan mendeteksi cacat atau deformasi pada buah (Li et al., 2021). Kombinasi fitur warna dan bentuk meningkatkan akurasi klasifikasi kualitas buah, terutama pada varietas Apel Fuji yang memiliki karakteristik warna cerah dan bentuk yang khas (Li et al., 2021; Mega et al., 2026).

Beberapa penelitian sebelumnya telah mengaplikasikan metode K-NN untuk klasifikasi kualitas buah berdasarkan ciri visual. Misalnya, penelitian oleh (Li et al., 2021) menyatakan bahwa penggabungan fitur warna dan bentuk dengan metode K-NN mencapai akurasi klasifikasi di atas 90% pada buah Apel Fuji. Penelitian lain oleh (Astuti, 2024) menggunakan K-NN dalam sistem berbasis kamera untuk deteksi kualitas Apel secara otomatis dengan hasil validasi yang memuaskan (akurasi > 88%). Penelitian-penelitian ini menjadi landasan dan pembanding dalam pengembangan sistem klasifikasi kualitas buah Apel Fuji menggunakan K-NN.

B. METODE PENELITIAN

Studi Literatur

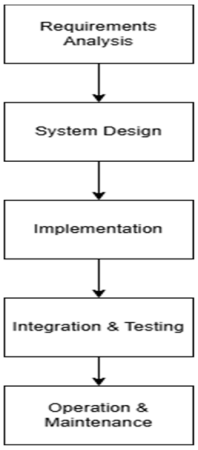
Studi literatur dilakukan dengan mengumpulkan informasi atau data yang berhubungan dengan penelitian klasifikasi kematangan Apel Fuji dari jurnal penelitian sebelumnya maupun buku. Studi literatur bertujuan untuk mengidentifikasi *research gap* pada sortasi manual Apel Fuji Kudus dan mendukung pengembangan model K-NN berbasis fitur RGB-HSV (Abdullah et al., 2025; Juwita et al., 2926).

Data Penelitian

Data yang digunakan terdiri dari 240 gambar Apel Fuji yang dikategorikan menjadi tiga tingkat kematangan: mentah, setengah matang, dan matang (masing-masing 80 gambar). Seluruh gambar digunakan sebagai populasi penelitian dan mewakili ciri-ciri visual Apel Fuji berdasarkan gradasi warna kulit dari daerah Kudus.

Langkah Metode Penelitian

Studi ini mengadopsi metode kuantitatif dengan pendekatan eksperimen menggunakan model *Waterfall* (Qolbi et al., 2024). Implementasi dilakukan melalui langkah terstruktur: pengumpulan data, preprocessing citra, ekstraksi fitur warna RGB-HSV, klasifikasi K-NN (k=7), dan evaluasi visual melalui *confusion matrix* di Google Colab, seperti ditunjukkan pada Gambar 1.



Gambar 1. Metode Waterfall

Pengumpulan Data

Pengumpulan data dilakukan dari dataset Kaggle (https://www.kaggle.com/datasets?search=fuji+apple. Dan foto lapangan di Kudus, total

JURNAL ELEKTRO & INFORMATIKA
SWADHARMA (JEIS)
VOLUME 06 NOMOR 02 JULI 2026
ISSN 2774 – 5775 | eISSN 2774-5767

240 gambar Apel Fuji (80 per kelas kematangan). Dataset dibagi menjadi 192 gambar untuk pelatihan (80%) dan 48 gambar untuk pengujian (20%) menggunakan *hold-out validation* agar proses *machine learning* berjalan seimbang dan representatif.

***Preprocessing* Citra**

Preprocessing adalah langkah awal perbaikan citra untuk menghapus *noise* dan meningkatkan kualitas sebelum ekstraksi fitur (Dzulhijjah et al., 2021). Pada penelitian ini, citra Apel Fuji di-resize ke 224x224 piksel, dikonversi ke ruang warna RGB-HSV, dan dinormalisasi untuk menonjolkan gradasi kemerahan pada kulit Apel matang.

C. HASIL DAN PEMBAHASAN

Kualitas Hasil klasifikasi kualitas buah Apel Fuji dengan metode K-Nearest Neighbor (K-NN) yang mencapai akurasi sebesar 91,7% sudah termasuk hasil yang sangat baik dalam konteks penelitian klasifikasi citra buah. Dalam literatur serupa, nilai akurasi pada kisaran 88% hingga 94% sudah dianggap memuaskan dan menunjukkan bahwa metode K-NN dapat bekerja dengan efektif, cepat, dan efisien untuk masalah klasifikasi kualitas buah Apel berdasarkan fitur warna dan bentuk.

Gambar 2. Apel Fuji Bagus

Gambar 3. Apel Fuji Sedang

Gambar 4. Apel Fuji Buruk.

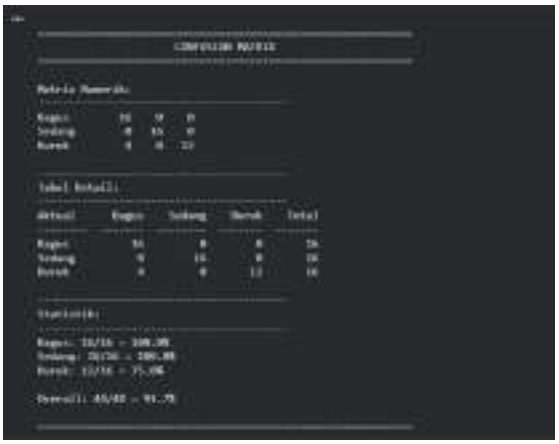
Pengujian dilakukan melalui Google Colab dengan visualisasi *confusion matrix* yang menunjukkan distribusi klasifikasi seimbang antara kelas mentah, setengah matang, dan matang tanpa bias signifikan. Beberapa kesalahan klasifikasi terjadi pada sampel dengan gradasi warna tipis antar kelas, namun secara keseluruhan model menunjukkan performa superior.

Gambar 5. Desain GUI System

Berdasarkan hasil pengujian model *computer vision* yang diimplementasikan pada platform Google Colab, diperoleh performa yang sangat tinggi. Dengan akurasi 91,7%, kematangan Apel Fuji yang signifikan. Pengujian dilakukan menggunakan total 48 sampel citra yang

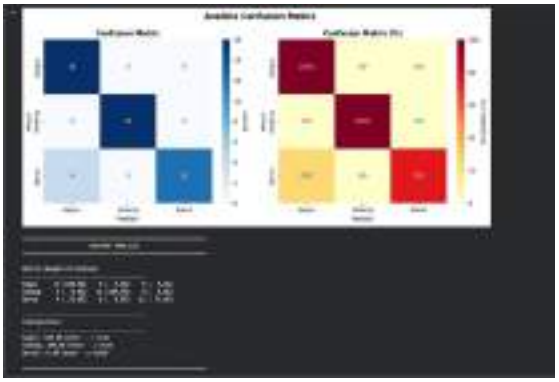
terbagi secara merata ke dalam tiga kategori, yaitu: Baik, Sedang, dan Buruk.

Selain akurasi tinggi, evaluasi menggunakan *precision, recall*, dan *f1-score* yang seimbang pada semua kelas kualitas menunjukkan kestabilan dan keandalan model. Hal ini mengindikasikan bahwa model mampu mendeteksi dan membedakan variasi kualitas dalam dataset dengan baik tanpa bias yang signifikan.



Gambar 6. *Confusion matrix*

Data menunjukkan bahwa model berhasil mengklasifikasikan 44 gambar dengan benar dari total 48 data uji. Secara matematis, penelitian ini menghasilkan tingkat akurasi sebesar 91,7%.



Gambar 7. Laporan Klasifikasi Heatmap

Hasil klasifikasi ini membuktikan bahwa model yang dikembangkan di Google Colab memiliki *error rate* yang sangat rendah (sekitar 8,5%) dan tidak memiliki bias terhadap salah satu kelas kualitas tertentu.

Secara keseluruhan, hasil tersebut sudah memenuhi standar yang baik untuk penelitian dan aplikasi praktis, khususnya dalam mendukung proses otomatisasi sortasi buah Apel Fuji di lapangan. Untuk meningkatkan validitas dan kredibilitas, hasil penelitian juga sebaiknya didukung dengan dokumentasi eksperimen yang jelas dan penggunaan data uji yang representatif.

Pengujian dilakukan dengan menggunakan *confusion matrix* yang menunjukkan distribusi klasifikasi antara kelas kualitas baik, sedang, dan buruk yang relatif seimbang tanpa adanya bias pada kelas manapun. Beberapa kesalahan klasifikasi terjadi pada sampel dengan perbedaan warna yang sangat tipis, namun secara keseluruhan sistem menunjukkan performa yang sangat baik.

Pengujian fungsional aplikasi GUI yang dikembangkan menunjukkan bahwa aplikasi dapat berjalan sesuai harapan pada tiga skenario utama, yaitu membuka aplikasi, mengunggah citra buah Apel, dan melakukan klasifikasi ulang.

Sistem yang dikembangkan siap digunakan untuk mendukung proses otomatisasi sortasi buah Apel Fuji di lapangan. Dengan adanya sistem ini, diharapkan dapat meningkatkan efisiensi, akurasi, dan konsistensi mutu produk di industri hortikultura.

D. PENUTUP

Penelitian ini mengembangkan sistem klasifikasi otomatis dengan akurasi 91,7%. Fitur yang digunakan meliputi warna (RGB, HSV) dan bentuk (perimeter, area, *eccentricity*) dari citra Apel Fuji. Model menunjukkan *precision*, *recall*, dan *F1-score* seimbang di semua kelas kualitas.

Aplikasi GUI berfungsi optimal untuk unggah citra dan klasifikasi real-time. Sistem siap diterapkan di industri hortikultura untuk sortasi otomatis, tingkatkan efisiensi, akurasi, dan kurangi subjektivitas penilaian manual.

E. DAFTAR PUSTAKA

Abdullah, R. W., Kusumastuti, R., & Handoko. (2025). Analisis Pengolahan Ekstraksi Fitur Citra Untuk Klasifikasi Jenis Apel Menggunakan Scikit-Learn Dengan Algoritma K-Nearest Neighbor. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, *12*(1), 165–174. https://doi.org/10.25126/jtiik.2025129149

Argadinata, A. P., Fatah, D. A., & Sukri, H. (2025). Klasifikasi Kualitas Buah Apel Menggunakan Metode Random Forest. *JATI : Jurnal Mahasiswa Teknik Informatika*, *9*(2), 2016–2022. https://doi.org/10.36040/jati.v9i2.12854

Astuti, P. (2024). Klasifikasi Kualitas Buah Apel Dengan Algoritma K-Nearest Neighbor (K-NN) Menggunakan Bahasa Pemrograman Python. *Co-Science: Computer Science*, *4*(2), 127–132. https://doi.org/10.31294/coscience.v4i2.3328

Dzulhijjah, A. N., Anraeni, S., & Sugiarti, S. (2021). Klasifikasi Kematangan Citra Labu Siam Menggunakan Metode KNN (K-Nearest Neighbor) Dengan Ekstraksi Fitur HSV (Hue, Saturation, Value). *Buletin Sistem Informasi Dan Teknologi Islam*, *2*(2), 103–110. https://doi.org/10.33096/busiti.v2i2.808

Juwita, A. R., Sukmawati, C. E., Pratama, A. R., Bijokangko, R. S., & Irawan, A. S. Y. (2926). Identifikasi Jenis Buah Apel berdasarkan Ektraksi Ciri Warna Fitur HSV dengan Model Jaringan Syaraf Tiruan Backpropagation. *Bulletin of Computer Science Research*, *6*(2), 784–792. https://doi.org/10.47065/bulletincsr.v6i2.1010

Li, Y., Feng, X., Liu, Y., & Han, X. (2021). Apple quality identification and classification by image processing based on convolutional neural networks. *Scientific Reports*, *11*, 16618. https://doi.org/10.1038/s41598-021-96103-2

Maula, A. I., Triyanto, W., & Setiaji, P. (2025). Sistem Klasifikasi Kematangan Apel Fuji berdasarkan Warna menggunakan KNN untuk Sortasi Otomatis. *Edumatic : Jurnal Pendidikan Informatika*, *9*(2), 589–598. https://doi.org/10.29408/edumatic.v9i2.31243

Mega, Marselina, N., & Chang, O. B. (2026). Penerapan Algoritma Artificial Neural Network Untuk Klasifikasi Kualitas Buah Apel. *HOAQ (High Education of Organization Archive Quality): Jurnal Teknologi Informasi*, *17*(1), 76–84. https://doi.org/10.52972/hoaq.vol17no1.p76-84

Qolbi, R. M., Putra, R. M., & Nugroho, W. (2024). Analisis Pengembangan Sistem Informasi Dengan Metode Waterfall Systematic Literature Review. *Infoman's : Jurnal Ilmu-Ilmu Informatika Dan Manajemen*, *18*(2), 1–10. https://ejournal.unsap.ac.id/index.php/infomans/article/view/1290

Suryanti, C., & Rohman, M. G. (2024). Klasifikasi Kualitas Buah Apel Berdasarkan Warna dan Bentuk Menggunakan Metode KNN. *Generation Journal*, *8*(1), 34–41. https://doi.org/10.29407/gj.v8i1.21052

PERANCANGAN INFRASTRUKTUR REDUNDANSI DENGAN MENERAPKAN *HIGH AVAILABILITY* UNTUK MENJAMIN KEANDALAN OPERASIONAL BISNIS

Saprila Wijaya[1)], Harjono Padmono Putro[2)]

[1]Prodi Teknik Informatika, Sekolah Tinggi Teknologi Informasi NIIT I-Tech
[2]Prodi Ilmu Komunikasi, Fakultas Bisnis dan Ilmu Sosial, Universitas Dian Nusantara

Correspondence author: S Wijaya, saprilawijaya1@gmail.com, Jakarta, Indonesia

Abstract

The availability of information technology services is crucial for business operations, but infrastructure with a Single Point of Failure risks causing downTime that disrupts services. This research focuses on the design and implementation of High Availability-based redundancy infrastructure to maintain service reliability when a virtualization host or network is disrupted. The research method uses the Network Development Life Cycle, which includes requirements analysis, design, simulation/prototype, implementation, monitoring, and management. The implementation was carried out in a VMware vSphere environment with two ESXi hosts managed through vCenter, and iSCSI-based NetApp shared storage as a shared datastore to support the failover mechanism and automatic VM Recovery. Testing includes host failover, network failover, and Failback scenarios with connectivity monitoring. The results show that in the host failover scenario, the virtual machine was successfully recovered automatically with 33 seconds of downTime, while in the network failover and Failback scenarios, the service remained accessible with no observed downTime (0 seconds). The conclusion of this research shows that the design and implementation of High Availability can improve service reliability, minimize downTime during disruptions, and support achieving the 99.9% SLA target in the test scenario.

Keyword *: service availability, redundant infrastructure, failover, failback, high availability*

Abstrak

Ketersediaan layanan teknologi informasi sangat penting bagi operasional bisnis, namun infrastruktur dengan *Single Point of Failure* berisiko menyebabkan *downTime* yang mengganggu layanan. Penelitian ini berfokus pada perancangan dan implementasi infrastruktur redundansi berbasis *High Availability* untuk menjaga keandalan layanan saat terjadi gangguan pada *host* virtualisasi atau jaringan. Metode penelitian menggunakan *Network Development Life Cycle* yang meliputi analisis kebutuhan, perancangan, simulasi/prototipe, implementasi, monitoring, dan manajemen. Implementasi dilakukan pada lingkungan *VMware vSphere* dengan dua *host ESXi* yang dikelola melalui *vCenter* serta *shared storage NetApp* berbasis *iSCSI* sebagai *datastore* bersama untuk mendukung mekanisme

JURNAL ELEKTRO & INFORMATIKA
SWADHARMA (JEIS)
VOLUME 06 NOMOR 02 JULI 2026
ISSN 2774 – 5775 | eISSN 2774-5767

failover dan pemulihan otomatis VM. Pengujian mencakup skenario *failover host*, *failover* jaringan, dan *Failback* dengan pengamatan konektivitas. Hasil penelitian menunjukkan bahwa pada skenario *failover host*, *virtual machine* berhasil dipulihkan secara otomatis dengan *downTime* 33 detik, sedangkan pada skenario *failover* jaringan dan *Failback* layanan tetap dapat diakses tanpa *downTime* teramati (0 detik). Kesimpulan penelitian ini menunjukkan bahwa rancangan dan implementasi *High Availability* mampu meningkatkan keandalan layanan, meminimalkan *downTime* saat gangguan, serta mendukung pencapaian target SLA 99,9% pada skenario pengujian.

Kata Kunci : ketersediaan layanan, infrastruktur redudansi, *failover*, *failback*, *high availability*

A. PENDAHULUAN

Dalam era digital saat ini, teknologi informasi menjadi komponen utama dalam mendukung proses bisnis organisasi dan perusahaan di Indonesia karena mampu meningkatkan efisiensi, kecepatan, dan akurasi pengolahan serta pertukaran informasi (Purba & Ibrahim, 2023). Peningkatan ketergantungan ini juga memunculkan tantangan berupa tuntutan ketersediaan infrastruktur yang beroperasi secara berkelanjutan tanpa gangguan signifikan, karena gangguan layanan pada sistem kritikal dapat menghentikan proses bisnis, menurunkan kepercayaan pelanggan, dan menimbulkan kerugian finansial (Mukmin & Cholil, 2020; Pribadi et al., 2020). CNN (2025) melaporkan gangguan layanan *mobile banking* di Indonesia yang memicu keluhan pengguna dan potensi kerugian transaksi, sejalan dengan temuan (Sulaiman & Priambodo, 2024) bahwa *downTime* pusat data, meskipun singkat, dapat menimbulkan kerugian besar dan gangguan operasional, sehingga menegaskan pentingnya isu ketersediaan layanan dalam pengelolaan infrastruktur TI.

Salah satu penyebab utama *downTime* adalah *Single Point of Failure*, sehingga penerapan redundansi dan *High Availability* menjadi pendekatan strategis yang terbukti mampu meningkatkan ketersediaan layanan dan meminimalkan dampak kegagalan komponen (Erlianto et al., 2020; Khasanah & Kuryanti, 2019). Teknologi virtualisasi dengan hypervisor seperti *VMware ESXi* memungkinkan optimalisasi sumber daya melalui mesin virtual, sedangkan fitur *High Availability* pada *VMware vSphere* memungkinkan pemulihan otomatis layanan saat terjadi kegagalan *host* (Broadcom, 2025; Huda & Susila, 2023). Namun, di banyak organisasi skala kecil hingga menengah di Indonesia, penerapan konsep redundansi dan *High Availability* masih belum optimal, karena masih mengandalkan satu *server* utama tanpa skema redundansi yang memadai (Erlianto et al., 2020), sehingga terdapat kesenjangan antara kebutuhan ketersediaan layanan dan implementasi infrastruktur TI. Oleh karena itu, diperlukan perancangan infrastruktur berbasis redundansi dan *High Availability* yang mampu meminimalkan *Single Point of Failure* serta menyediakan mekanisme automatic failover.

Penelitian ini difokuskan pada perancangan dan implementasi infrastruktur *High Availability* berbasis virtualisasi menggunakan *VMware ESXi* pada lingkungan *server* virtual yang merepresentasikan kebutuhan perusahaan skala menengah di Indonesia, dengan tujuan memberikan solusi praktis untuk meningkatkan keandalan infrastruktur TI dan mendukung pencapaian target *Service Level Agreement* (SLA).

Berdasarkan latar belakang tersebut, permasalahan utama dalam penelitian ini meliputi tingginya risiko *downTime* akibat adanya *Single Point of Failure* pada komponen krusial, ketiadaan mekanisme *failover* otomatis yang menyebabkan waktu pemulihan layanan menjadi lama dan tidak terprediksi, serta ketidakmampuan perusahaan dalam memenuhi target ketersediaan layanan (*Service Level Agreement*) bisnis.

Tujuan penelitian ini adalah merancang infrastruktur redundansi pada komponen krusial untuk menghilangkan potensi *Single Point of Failure*, menerapkan mekanisme *High Availability* dan automatic *failover* menggunakan teknologi *cluster*ing pada *VMware ESXi* untuk meminimalkan *downTime* dan meningkatkan keandalan layanan TI, serta mengevaluasi efektivitas implementasi infrastruktur *High Availability* melalui pengujian fungsional *failover* guna mendukung pemenuhan target *Service Level Agreement* sebesar 99,9%.

B. METODE PENELITIAN

Hasil observasi menunjukkan bahwa infrastruktur TI yang berjalan masih bersifat terpusat dan bergantung pada satu *server* utama serta satu jalur jaringan tanpa mekanisme redundansi dan *Failover* otomatis. Kondisi ini menyebabkan risiko *downTime* tinggi ketika terjadi gangguan pada *server*, jaringan, atau storage, sehingga layanan tidak dapat diakses hingga dilakukan pemulihan manual. Infrastruktur eksisting masih memiliki *Single Point of Failure* dan belum memenuhi kebutuhan *High Availability*.

Solusi yang diusulkan adalah perancangan infrastruktur *High Availability* pada level jaringan, *server*, virtualisasi, dan *storage* dengan penerapan redundansi serta mekanisme *Failover* otomatis. Redundansi *server* diterapkan melalui *host* utama dan *host* cadangan, jaringan menggunakan jalur alternatif, serta *storage* menggunakan teknologi redundansi. Metode *Network Development Life Cycle* digunakan sebagai kerangka perancangan infrastruktur secara sistematis.

Perancangan sistem infrastruktur dilakukan menggunakan metode *Network Development Life Cycle* (Prasetyo et al., 2025) yang mencakup tahapan analisis, desain, simulasi, implementasi, monitoring, dan manajemen. Metode ini digunakan untuk memastikan perancangan infrastruktur *High Availability* dilakukan secara terstruktur, terdokumentasi, dan dapat dievaluasi secara sistematis sebelum diimplementasikan secara operasional.

Infrastruktur eksisting menunjukkan seluruh layanan berjalan pada satu *server* utama dengan satu jalur jaringan tanpa *server* cadangan dan jalur komunikasi alternatif. Kondisi ini menyebabkan ketergantungan tinggi pada satu komponen utama, tidak adanya mekanisme *Failover* otomatis, serta proses pemulihan manual yang meningkatkan risiko *downTime* dan menurunkan keandalan layanan.

Kebutuhan infrastruktur dibagi menjadi kebutuhan fungsional dan non-fungsional. Kebutuhan fungsional mencakup kemampuan *Failover* otomatis, deteksi kegagalan *host*, penggunaan *multi-server*, *shared storage*, serta mekanisme Failback. Kebutuhan non-fungsional meliputi pemenuhan SLA 99,9%, minimisasi down*Time*, keandalan dan stabilitas sistem, kemudahan pengelolaan, serta fleksibilitas pengembangan di masa depan.

Perancangan infrastruktur *High Availability* dilakukan dengan menerapkan arsitektur *cluster* ESXi menggunakan *vSphere High Availability*, *shared datastore*, serta redundansi jaringan dan *server* untuk menghilangkan *Single Point of Failure*. Mekanisme *Failover* dan *Failback* dirancang untuk mengalihkan layanan secara otomatis saat terjadi kegagalan dan mengembalikannya ke *host* utama setelah kondisi sistem kembali normal.

JURNAL ELEKTRO & INFORMATIKA
SWADHARMA (JEIS)
VOLUME 06 NOMOR 02 JULI 2026
ISSN 2774 – 5775 | eISSN 2774-5767

Pengujian dilakukan dengan mensimulasikan kegagalan *server*, jaringan, dan proses *Failback* untuk mengukur *downTime* dan waktu pemulihan layanan. Keberhasilan pengujian ditentukan berdasarkan kemampuan layanan tetap dapat diakses setelah *Failover* dan kembali normal setelah Failback. Rancangan infrastruktur ini ditujukan untuk memenuhi target *Service Level Agreement* sebesar 99,9%.

C. HASIL DAN PEMBAHASAN

Kualitas lingkungan infrastruktur pada penelitian ini berupa lab virtualisasi yang digunakan untuk mengimplementasikan dan menguji rancangan *High Availability* berbasis *VMware* vSphere. Lingkungan ini terdiri dari dua *host* ESXi yang dikonfigurasi dalam satu *cluster* untuk mendukung mekanisme Failover, *vCenter Server* sebagai pusat manajemen, *storage NetApp* sebagai *shared datastore*, serta satu *virtual machine* sebagai objek pengujian.

Spesifikasi *host* ESXi yang digunakan terdiri dari dua *server* dengan hypervisor *VMware ESXi* 8.0.3, prosesor Intel Xeon Gold 5418Y, memori 32 GB, kartu jaringan 2 × 10 Gbit, dan *storage* lokal 75 GB, sebagaimana ditunjukkan pada Tabel 1.

Tabel 1. Spesifikasi *Host* esx1 dan esx2

Host	Hypervisor	Proc.	Memory	NIC	Storage
Esx1	*VMware ESXi* 8.0.3	Intel(R) Xeon(R) Gold 5418Y 2GHz	32 GB	2 × 10 Gbit	75 GB
Esx2	*VMware ESXi* 8.0.3	Intel(R) Xeon(R) Gold 5418Y 2GHz	32 GB	2 × 10 Gbit	75 GB

Pengelolaan *cluster* dilakukan menggunakan *vCenter Server* versi 8.0.3 dengan satu *cluster* yang terdiri dari dua *host* dan satu *virtual machine* uji, sesuai Tabel 2.

Tabel 2. Spesifikasi *vCenter Server*

FQDN	Version	Build	Jumlah *Cluster*	Jumlah *Host*	Jumlah VM
vc1.demo.lab.com	8.0.3	24322831	1 *cluster*	2 *host*	1 VM

Untuk mendukung mekanisme Failover, digunakan *shared storage* berbasis *NetApp* ASA R2 dengan konfigurasi dua node, kapasitas usable 491 GiB, dan protokol iSCSI, sebagaimana dirangkum pada Tabel 3.

Tabel 3. Spesifikasi *shared storage*

Komponen	Spesifikasi
Shared storage	*NetApp* ASA R2
OS Version	ONTAP 9.18.1
Konfigurasi	2 Node Configuration
Disk	28 × 26.2 GiB SSD Disk
Kapasitas	491 GiB Usable Capacity
Port	7 × 1 GB Ethernet Port
Protocol	ISCSI

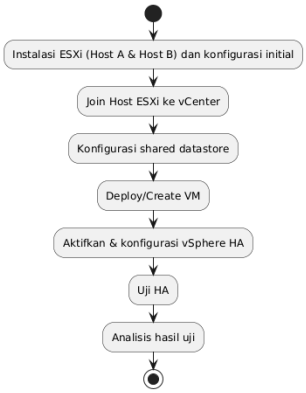
Virtual machine yang digunakan sebagai objek pengujian memiliki sistem operasi Windows *Server* 2022, 2 vCPU, RAM 4 GB, dan kapasitas disk 90 GB, dengan spesifikasi yang disajikan pada Tabel 4.

Tabel 4. Spesifikasi *Virtual machine*

Parameter	Nilai
Nama VM	sql1
Guest OS	Microsoft Windows *Server* 2022 (64-bit)
IP Address	192.168.0.46
vCPU	2 vCPU
RAM	4 GB
Disk	90 GB

Tahap implementasi dilakukan untuk menerapkan rancangan *High Availability* pada lingkungan virtualisasi berbasis *VMware* vSphere. Implementasi mencakup penyiapan dua *host* ESXi dalam satu *cluster*, konfigurasi *shared storage* berbasis iSCSI, pengaturan jaringan virtual, *deployment virtual machine* uji, serta aktivasi fitur *vSphere High Availability* dan *Distributed Resource Scheduler*. Secara umum, tahapan implementasi sistem disajikan pada Gambar

1 sebagai alur penerapan infrastruktur *High Availability*.



Gambar 1. Tahapan Implementasi

Dua *host* ESXi dikonfigurasi dengan parameter manajemen jaringan yang berbeda untuk mendukung mekanisme Failover, kemudian diregistrasikan ke dalam *cluster* melalui *vCenter Server* agar dapat dikelola secara terpusat. *Shared storage* berbasis *NetApp* dikonfigurasi menggunakan protokol iSCSI dan digunakan sebagai datastore bersama sehingga *virtual machine* dapat dijalankan pada *host* lain ketika terjadi kegagalan *host* utama. Jaringan virtual dikonfigurasi menggunakan standard v*Switch* dengan port group untuk manajemen *host* dan jaringan data *virtual machine*.

Virtual machine uji kemudian diregistrasikan ke dalam inventory vCenter dan dijalankan pada *cluster* sebagai objek pengujian *High Availability*. Selanjutnya, fitur *vSphere High Availability* diaktifkan pada *cluster* untuk memastikan *virtual machine* dapat di*restart* secara otomatis saat terjadi *host* failure, serta fitur *Distributed Resource Scheduler* diaktifkan untuk mendukung manajemen beban secara otomatis. Status aktivasi *High Availability* pada *cluster* ditunjukkan pada Gambar 2, yang menandakan bahwa mekanisme pemulihan otomatis telah siap digunakan dalam skenario pengujian.

Gambar 2. *vSphere* HA is Turned ON

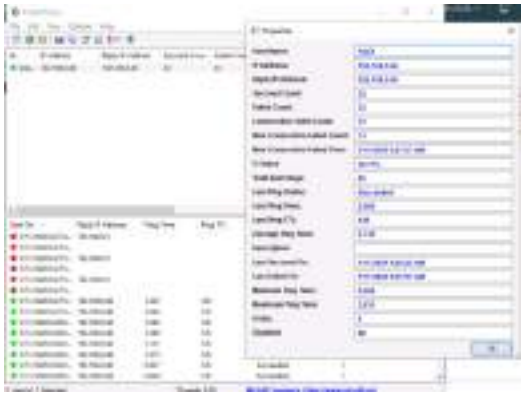
Pengujian sistem dilakukan menggunakan metode black box untuk mengevaluasi kemampuan *vSphere High Availability* dalam memulihkan layanan *virtual machine* secara otomatis ketika terjadi gangguan pada *host* dan jaringan. Monitoring dilakukan melalui ping kontinu pada *virtual machine* uji serta analisis event log pada *vCenter Server* untuk memvalidasi waktu kegagalan dan proses pemulihan layanan. Skenario pengujian yang diterapkan meliputi *failover host*, *failover* network, dan failback, sebagaimana dirangkum pada Tabel 5.

Tabel 5. Skenario Pengujian

Skenario Pengujian	**Tindakan Pengujian**	***Tools* yang digunakan**
Failover host	*Shutdown* satu *server virtual*isasi	PingInfoView
Failover Network	Memutus satu jalur/adapter jaringan	PingInfoView
Pengujian *Failback*	*Server* utama diaktifkan kembali	PingInfoView

Hasil pengujian menunjukkan bahwa pada skenario *failover host*, *virtual machine* sql1 berhasil berpindah secara otomatis ke *host* cadangan dengan *downTime* sebesar 33 detik, yang menunjukkan mekanisme *vSphere* HA berjalan sesuai fungsi pemulihan layanan. Proses *failover* ini divalidasi melalui monitoring konektivitas dan event log vCenter sebagaimana ditunjukkan pada Gambar 3.

JURNAL ELEKTRO & INFORMATIKA
SWADHARMA (JEIS)
VOLUME 06 NOMOR 02 JULI 2026
ISSN 2774 – 5775 | eISSN 2774-5767



Gambar 3. Hasil monitoring

Pada skenario *failover* network, pemutusan salah satu uplink tidak menyebabkan gangguan layanan karena masih tersedia jalur jaringan cadangan, sehingga *downTime* tidak teramati. Sementara itu, pada skenario failback, *virtual machine* tetap berjalan normal ketika *host* utama diaktifkan kembali dan *cluster* kembali stabil. Ringkasan hasil pengujian ketiga skenario disajikan pada Tabel 6.

Tabel 6. Ringkasan Evaluasi Hasil Pengujian

Skenario	Hasil Utama	Down*Time*	Evaluasi	Memenuhi SLA 99,9%
Failover Server	VM sql1 berhasil dipulihkan oleh *vSphere* HA (*restart*/pindah ke *host* lain)	33 detik	HA berjalan sesuai fungsi pemulihan layanan saat *host* gagal	Memenuhi
Failover Network	Saat 1 uplink (vmnic) dimatikan, layanan VM tetap dapat diakses	0 detik (tidak teramati)	Redundansi uplink efektif, gangguan 1 jalur tidak memutus layanan	Memenuhi
Failback	*Host* kembali Connected, VM sql1 tetap Powered On/Normal	0 detik (tidak teramati)	*Cluster* kembali stabil setelah *host* normal kembali	Memenuhi

Evaluasi hasil pengujian menunjukkan bahwa seluruh skenario berhasil dan layanan tetap tersedia sesuai rancangan. Namun, *downTime* sebesar 33 detik pada skenario *failover host* mengindikasikan bahwa *vSphere* HA tidak bersifat zero-*downTime* karena memerlukan waktu untuk mendeteksi kegagalan *host* dan melakukan *restart virtual machine* pada *host* pengganti. Selain itu, pengujian memiliki keterbatasan karena belum mencakup kegagalan komponen lain seperti *shared storage*, kegagalan *switch* jaringan, serta kondisi beban sistem tinggi.

Analisis *Recovery Time Objective* (RTO) dan *Recovery Point Objective* (RPO) menunjukkan bahwa nilai RTO pada skenario *failover host* adalah 33 detik, sedangkan nilai RPO pada seluruh skenario adalah 0 detik karena tidak terjadi kehilangan data selama proses *failover* maupun failback. Hasil analisis RTO dan RPO dirangkum pada Tabel 7.

Tabel 7. Analisis RTO dan RPO

Skenario Pengujian	RTO	RPO	Keterangan
Failover Host ESXi	33 detik	0 detik	*Virtual machine* berhasil *direstart* secara otomatis
Failover Network	0 detik	0 detik	Layanan tetap berjalan tanpa gangguan
Failback	0 detik	0 detik	Tidak terjadi gangguan layanan

Tabel 7 menunjukkan bahwa rancangan *High Availability* yang diimplementasikan mampu memenuhi kebutuhan ketersediaan layanan dan berada di bawah batas toleransi *downTime* pada *Service Level Agreement* sebesar 99,9%.

D. PENUTUP

Berdasarkan hasil perancangan, implementasi, dan pengujian infrastruktur

High Availability berbasis *VMware* vSphere, penelitian ini menyimpulkan bahwa penerapan redundansi dengan dua *host* ESXi dalam satu *cluster*, pengelolaan terpusat melalui vCenter, serta penggunaan *shared storage* iSCSI berhasil meminimalkan risiko *Single Point of Failure*. Konfigurasi *vSphere* HA dengan mekanisme monitoring *host* dan automatic *failover* terbukti mampu memulihkan VM secara otomatis ketika terjadi kegagalan *host*, dengan *downTime* maksimum 33 detik pada skenario *failover host* dan tanpa *downTime* pada skenario *failover* network dan failback, sehingga memenuhi target SLA 99,9%. Untuk pengembangan selanjutnya, disarankan penerapan disaster *Recovery* atau multi-site HA melalui replikasi antar data center, pelaksanaan load testing untuk mengukur performa saat failover, serta penerapan sistem monitoring proaktif dengan peringatan dini guna meningkatkan keandalan dan kesiapan operasional sistem.

E. DAFTAR PUSTAKA

Broadcom. (2025). *vSphere Availability Documentation*.

Erlianto, A., Supendar, H., & Tutupoly, T. A. (2020). Implementasi High Availability Virtualisasi Server Menggunakan VMware ESXi. *INSANTEK : Jurnal Inovasi Dan Sains Teknik Elektro*, *1*(2), 101–107. https://doi.org/https://ejournal.bsi.ac.id/index.php/insantek/article/view/9401

Huda, A. N., & Susila, A. (2023). Infrastruktur Virtualisasi Data Center Berbasis Site Recovery Pada (PT. Informatics Oase). *OKTAL : Jurnal Ilmu Komputer Dan Sains*, *2*(3), 823–832. https://journal.mediapublikasi.id/index.php/oktal/article/view/988

Khasanah, S. N., & Kuryanti, S. J. (2019). Rancangan Virtualisasi Server Menggunakan VMware vSphere. *Jurnal Evolusi*, *7*(1), 42–46. https://doi.org/10.31294/evolusi.v7i1.5091

Mukmin, C., & Cholil, W. (2020). Perbandingan Kinerja Server Virtual pada Proses High Availability. *JJKK : Jurnal Jaringan Komputer Dan Keamanan*, *1*(1), 48–57. https://iitss.or.id/ojs/index.php/jjkk/article/view/20

Prasetyo, F. H., Erna, & Febriyansyah. (2025). Penerapan Metode Network Development Life Cycle (NDLC) dalam Pengembangan Jaringan Komputer. *JICT : Journal of Informatics and Communication Technology*, *7*(1), 80–87. https://doi.org/10.52661/jict.v7i1.410

Pribadi, Y., Negara, A. B. P., & Irwansyah, M. A. (2020). Analisis Penggunaan Metode Failover Clustering untuk Mencapai High Availability pada Web Server (Studi Kasus: Gedung Jurusan Informatika). *JUSTIN (Jurnal Sistem Dan Teknologi Informasi)*, *8*(2), 218–229. https://doi.org/10.26418/justin.v8i2.31965

Purba, R., & Ibrahim, H. (2023). Peran Teknologi Informasi dalam Meningkatkan Efisiensi Operasional. *Digital Bisnis: Jurnal Publikasi Ilmu Manajemen Dan E-Commerce*, *2*(4), 454–462. https://doi.org/10.30640/digital.v2i4.2061

Sulaiman, S. F., & Priambodo, A. H. (2024). Downtime Data Center: Memahami Penyebab, Dampak, dan Solusi Efektif. *Sanskara Manajemen Dan Bisnis*, *2*(2), 67–78. https://doi.org/10.58812/smb.v2i02.297

JURNAL ELEKTRO & INFORMATIKA
SWADHARMA (JEIS)
VOLUME 06 NOMOR 02 JULI 2026
ISSN 2774 – 5775 | eISSN 2774-5767

ANALISIS SENTIMEN PENGELOMPOKAN PENGGUNA APLIKASI PINJAMAN ONLINE DENGAN ALGORITMA K-MEANS

Sri Ipnuwati[1)], Sabda Iman Ramadian[2)], Didi Susianto[3)],Yodhi Yuniarte[4)]

[1,2,3]Prodi Sistem Informasi, FTIKOM, Institut Bakti Nusantara, Lampung
[4]Prodi Informatika, Fakultas Komputer, Universitas Mitra Indonesia

Correspondence author: S Ipnuwati, nengachie@gmail.com, Pringsewu, Indonesia

Abstract

The development of online lending services within financial technology has made financing easier for the public. Still, it has also given rise to problems such as high interest rates, a lack of cost transparency, and service issues that lead to user complaints. These issues are reflected in the large number of unstructured user reviews on the Google Play Store, making them difficult to analyze manually. This situation requires applying data-driven analysis methods to understand sentiment patterns and user characteristics in online lending applications. This study aims to analyze sentiment and group users of online lending applications based on Indonesian-language reviews on the Google Play Store for 2026. The methods used are data mining using a text-mining approach and unsupervised learning. The research stages include data collection, text preprocessing, feature extraction using TF-IDF, and data clustering using the K-Means algorithm with the help of Google Colab and the Orange application. The results are expected to produce user clusters with distinct sentiment characteristics and provide an overview of user perception patterns. These findings can be used as academic references and evaluation materials for online lending service providers to improve service quality.

Keyword *: sentiment analysis, online lending, k-means*

Abstrak

Perkembangan layanan pinjaman online sebagai bagian dari teknologi finansial telah memberikan kemudahan akses pembiayaan bagi masyarakat, namun di sisi lain memunculkan berbagai permasalahan seperti tingginya suku bunga, kurangnya transparansi biaya, serta pelayanan yang menimbulkan keluhan pengguna. Permasalahan tersebut tercermin dalam ulasan pengguna pada platform *Google Play Store* yang jumlahnya besar dan bersifat tidak terstruktur, sehingga sulit dianalisis secara manual. Kondisi ini menuntut penerapan metode analisis berbasis data untuk memahami pola sentimen dan karakteristik pengguna aplikasi pinjaman online. Penelitian ini bertujuan untuk menganalisis sentimen serta mengelompokkan pengguna aplikasi pinjaman online berdasarkan ulasan berbahasa Indonesia pada *Google Play Store* periode 2026. Metode yang digunakan adalah *data mining* dengan pendekatan *text mining* dan *unsupervised learning*. Tahapan penelitian meliputi pengumpulan data, *preprocessing* teks, ekstraksi fitur menggunakan TF-IDF, serta pengelompokan data menggunakan algoritma K-Means dengan bantuan Google Colab dan aplikasi Orange. Hasil

penelitian diharapkan menghasilkan klaster pengguna dengan karakteristik sentimen yang berbeda dan memberikan gambaran pola persepsi pengguna. Temuan ini dapat dimanfaatkan sebagai referensi akademik serta bahan evaluasi bagi penyedia layanan pinjaman online dalam meningkatkan kualitas layanan.

Kata Kunci : analisis sentimen, pinjaman online, k-means

A. PENDAHULUAN

Seiring dengan meningkatnya kebutuhan hidup masyarakat, muncul berbagai perusahaan yang menawarkan solusi pembiayaan untuk memenuhi kebutuhan (Ghozali et al., 2023). Sektor pinjaman online di Indonesia menunjukkan peningkatan yang signifikan. *Fintech peer-to-peer* (P2P) *lending* memiliki nilai pinjaman outstanding yang meningkat 35,62% setiap tahun hingga Agustus 2024 (Indriastuti et al., 2025). Data Otoritas Jasa Keuangan (OJK) Sampai dengan 31 Mei 2024, total jumlah penyelenggara *Fintech peer-to-peer lending* atau *fintech lending* yang berizin di Otoritas Jasa Keuangan adalah sebanyak 100 perusahaan (Gandasari et al., 2025). Perkembangan *fintech* ini terutama terlihat pada sektor pinjaman online, yang telah membuat layanan keuangan lebih mudah diakses oleh berbagai lapisan masyarakat tanpa harus mengikuti prosedur yang rumit seperti yang dimiliki oleh lembaga keuangan konvensional (Izzah et al., 2025).

Meningkatnya penggunaan aplikasi pinjaman online di Indonesia tidak dapat dilepaskan dari perubahan pola perilaku masyarakat dalam mengelola kebutuhan finansial jangka pendek. Proses pengajuan yang sederhana, persyaratan yang relatif minimal, serta pencairan dana yang cepat menjadikan layanan pinjaman online sebagai alternatif utama bagi masyarakat yang membutuhkan dana darurat (Yustati et al., 2025). Data Otoritas Jasa Keuangan (OJK) menunjukkan bahwa nilai outstanding pinjaman *fintech* P2P *lending* terus mengalami peningkatan signifikan setiap tahunnya, mencerminkan tingginya tingkat adopsi layanan ini oleh masyarakat. Pertumbuhan tersebut secara tidak langsung menandakan bahwa aplikasi pinjaman online telah menjadi bagian penting dalam ekosistem keuangan digital nasional (Izzah et al., 2025).

Di balik pertumbuhan yang pesat tersebut, layanan pinjaman online juga menghadirkan berbagai permasalahan yang kompleks dan multidimensional. Berbagai laporan media dan pengaduan konsumen menunjukkan adanya keluhan terkait tingginya suku bunga, kurangnya transparansi biaya, metode penagihan yang agresif, serta isu perlindungan data pribadi pengguna (Rahmanda et al., 2025). Kondisi ini menimbulkan beragam reaksi dari masyarakat sebagai pengguna layanan, yang tercermin dalam ulasan dan penilaian yang diberikan pada platform distribusi aplikasi digital seperti *Google Play Store*. Ulasan tersebut menjadi representasi langsung dari pengalaman, persepsi, serta tingkat kepuasan pengguna terhadap layanan pinjaman online (Bela et al., 2026).

Ulasan pengguna pada *Google Play Store* menunjukkan adanya perbedaan sentimen yang cukup tajam antara pengguna satu dengan lainnya. Sebagian pengguna menyampaikan apresiasi terhadap kemudahan proses dan kecepatan pencairan dana, sementara sebagian lainnya mengekspresikan kekecewaan akibat biaya yang dianggap memberatkan atau pelayanan yang dinilai tidak responsif. Polarisasi sentimen ini mencerminkan adanya segmentasi pengguna yang terbentuk secara alami berdasarkan pengalaman dan

ekspektasi yang berbeda. Segmentasi tersebut menjadi indikasi bahwa pengguna aplikasi pinjaman online tidak bersifat homogen, melainkan memiliki karakteristik dan pola persepsi yang beragam (Mongkito et al., 2024; Zahir et al., 2024).

Aplikasi pinjaman online yang beroperasi secara legal dan terdaftar di OJK, seperti Kredivo (PT FinAccel Finance Indonesia), AdaKami (PT Pembiayaan Digital Indonesia), Akulaku (PT Akulaku Finance Indonesia), Indodana (PT PT Indodana Multi Finance), dan Easycash (PT Indonesia Fintopia Technology), merupakan contoh representatif dari layanan pinjaman online yang banyak digunakan masyarakat. Kelima aplikasi tersebut memiliki basis pengguna yang besar dan aktif, sehingga menghasilkan volume ulasan yang signifikan di *Google Play Store*. Keberadaan ulasan dalam jumlah besar ini membuka peluang penelitian berbasis data tekstual untuk memahami sentimen dan pola perilaku pengguna secara lebih mendalam.

Analisis sentimen adalah upaya untuk memahami dan mengkategorikan emosi (positif, negatif, dan netral) yang ditemukan dalam teks melalui penggunaan metode analisis teks (Nurhasanah, 2026). Pengguna dengan sentimen positif biasanya menekankan kecepatan pencairan dana dan kemudahan proses verifikasi, sementara pengguna sentimen negatif seringkali menguasai percakapan dengan keluhan tentang layanan pelanggan dan transparansi biaya. Fenomena ini menunjukkan ada segmentasi pengguna yang alami. Jika dipetakan dengan benar, ini menjadi dasar bagi perusahaan *fintech* untuk meningkatkan layanan sekaligus mengurangi risiko reputasi.

Penelitian sebelumnya dalam analisis sentimen pada layanan finansial umumnya menggunakan pendekatan *lexicon-based* atau algoritma machine learning konvensional seperti *Naïve Bayes* dan *Support Vector Machine* (SVM). Studi yang dilakukan oleh (Andini & Muthia, 2024) menerapkan SVM dan fitur *lexicon-based* untuk menganalisis ulasan Pengguna ChatGPT, mencapai akurasi sebesar 90%]. Demikian pula, penelitian oleh (Akbar et al., 2023) membandingkan metode *Naïve Bayes*, *lexicon-based*, dan hybrid dalam mengklasifikasikan sentimen Terkait Jaminan Hari Tua, dengan hasil terbaik pada *Naïve Bayes* yang mencapai akurasi 95%.

Selain itu, penelitian yang dilakukan oleh (Sunarko et al., 2023), studi ini dilakukan untuk menganalisis umpan balik pengguna aplikasi BCA Mobile. Proses *clustering* ini menghasilkan sepuluh *cluster*, dengan skor *silhouette* 0,1027277 dan rating bintang rata-rata 2,65. Sementara itu, (Reyan & Purwaningtyas, 2025) menerapkan algoritma *Naïve Bayes* untuk menganalisis aplikasi IBI library, dan melaporkan tingkat akurasi sebesar 68,89%.

Pendekatan berbasis *Support Vector Machine* (SVM) juga telah digunakan dalam beberapa studi. Misalnya, dalam penelitian (Sujjadaa et al., 2023), SVM digunakan bersama dengan teknik *text mining* untuk menganalisis ulasan aplikasi bank digital, menghasilkan klasifikasi sentimen dengan akurasi 91%. Sedangkan dalam studi lain oleh (Haq et al., 2024), Metode Algoritma *Naïve Bayes* diterapkan untuk analisis sentimen pada mengidentifikasi hoaks.

Namun, dari keenam penelitian tersebut, baik yang menggunakan metode *lexicon-based*, *Naïve Bayes*, SVM, maupun pendekatan hybrid, hampir semuanya masih terbatas pada klasifikasi sentimen positif, negatif, atau netral, tanpa melakukan pengelompokan lebih lanjut terhadap pengguna berdasarkan kemiripan karakteristik atau preferensi dalam ulasan mereka. Oleh karena itu, penelitian ini hadir untuk melengkapi penelitian-penelitian sebelumnya dengan menerapkan pendekatan *clustering* menggunakan algoritma K-Means bertujuan untuk mengidentifikasi segmen pengguna berdasarkan pola sentimen dan topik dominan dalam ulasan aplikasi pinjaman online.

Penelitian ini diharapkan dapat memberikan kontribusi akademik dalam

pengembangan penerapan *text mining* dan *unsupervised learning* pada domain *fintech*, serta memberikan manfaat praktis bagi perusahaan penyedia layanan pinjaman online dan regulator dalam memahami karakteristik pengguna secara lebih terstruktur dan berbasis data.

B. METODE PENELITIAN

Pengumpulan Data

Data yang digunakan dalam penelitian ini merupakan data sekunder berupa ulasan pengguna aplikasi pinjaman online yang diperoleh dari *Google Play Store*. Data ulasan terdiri dari teks ulasan, penilaian bintang, serta informasi pendukung lain yang relevan dengan analisis sentimen. Data ini digunakan sebagai dasar kajian dan analisis untuk menarik simpulan mengenai pola sentimen dan pengelompokan pengguna.

Pengumpulan data dilakukan dengan teknik *data mining* atau web *Scraping* terhadap halaman ulasan aplikasi pinjaman online pada *Google Play Store*. Data yang dikumpulkan kemudian diseleksi dengan kriteria tertentu, seperti penggunaan bahasa Indonesia dan relevansi isi ulasan dengan pengalaman penggunaan aplikasi. Data yang telah dikumpulkan selanjutnya melalui tahapan *preprocessing* teks sebelum dianalisis lebih lanjut.

Gambar 1. Flowchart Pengumpulan Data

Selain data utama dari *Google Play Store*, penelitian ini juga didukung oleh hasil penelitian terdahulu yang relevan. Sumber-sumber tersebut digunakan sebagai pendukung dalam memahami konteks industri pinjaman online di Indonesia dan memperkuat pembahasan hasil penelitian. Dengan demikian, data yang digunakan diharapkan mampu memberikan gambaran yang komprehensif dan valid terhadap permasalahan yang diteliti.

Data *Scraping* yang berhasil dikumpulkan kemudian disimpan ke format dokumen comma separated values (CSV) atau format dokumen excel (XLSX). Setelah data ulasan dikumpulkan kemudian dilakukan filter sehingga hanya menyisahkan kolom ulasan dan skor, itu dilakukan karena dalam penelitian ini haya kolom ulasan dan skor yang dibutuhkan, selain itu dilakukan juga perurutan ulasan berdasarkan tanggal pembuatan.

Analisis Data Awal

Data yang digunakan dalam penelitian ini berupa ulasan pengguna aplikasi pinjaman online yang diperoleh dari platform *Google Play Store*. Sampel penelitian diambil dari lima aplikasi pinjaman online yang terdaftar dan diawasi oleh Otoritas Jasa Keuangan, yaitu Kredivo, AdaKami, Akulaku, Indodana, dan Easycash. Untuk memperoleh gambaran sentimen pengguna yang representatif dan seimbang, jumlah data ulasan yang digunakan ditetapkan sebanyak 10.000 ulasan untuk setiap aplikasi, sehingga total keseluruhan data yang dianalisis berjumlah 50.000 ulasan. Penetapan jumlah sampel yang besar dan merata antar aplikasi bertujuan untuk meningkatkan reliabilitas hasil analisis serta meminimalkan potensi bias akibat dominasi data dari satu aplikasi tertentu.

Selanjutnya, data yang telah dikumpulkan dibagi menjadi data pelatihan dan data pengujian dengan rasio 90:10, di mana 90% data digunakan untuk proses pelatihan model dan 10% sisanya akan di uji dan digunakan untuk evaluasi kinerja model.

Pemilihan rasio ini mengacu pada penelitian oleh (Muningsih, 2022) yang menunjukkan bahwa penggunaan data pelatihan dengan rasio 90:10 dapat meningkatkan kemampuan model dalam mempelajari pola sentimen secara lebih optimal serta menghasilkan performa analisis yang lebih baik pada data uji. Dapat diketahui bahwa setiap aplikasi pinjaman online dianalisis menggunakan jumlah data ulasan yang sama, yaitu sebanyak 5.000 ulasan. Distribusi data yang merata ini dilakukan untuk memastikan bahwa proses analisis sentimen dan pengelompokan pengguna menggunakan algoritma K-Means dilakukan secara adil dan seimbang. Seluruh data ulasan yang digunakan bersifat tidak berlabel (*unlabeled data*), sehingga sesuai dengan pendekatan *clustering* yang tidak memerlukan kelas awal. Dengan jumlah data yang besar dan seimbang, hasil klaster yang dihasilkan diharapkan memiliki tingkat stabilitas dan validitas yang lebih baik dalam merepresentasikan pola sentimen dan karakteristik pengguna aplikasi pinjaman online (Ipnuwati et al., 2025).

Tabel 1. Jumlah Data Ulasan Pengguna Aplikasi Pinjaman Online

No	Aplikasi Pinjaman Online	Perusahaan Pengelola	Jumlah Ulasan	Jumlah Ulasan Yang di Uji
1	Kredivo	PT FinAccel Finance Indonesia	10.000	1000
2	AdaKami	PT Pembiayaan Digital Indonesia	10.000	1000
3	Akulaku	PT Akulaku Finance Indonesia	10.000	1000
4	Indodana	PT Indodana Multi Finance	10.000	1000
5	Easycash	PT Indonesia Fintopia Technology	10.000	1000
Total			**50.000 Ulasan**	**5.000 Ulasan**

Model dan Algoritma yang Digunakan

Model yang digunakan dalam penelitian ini adalah model *clustering* berbasis K-Means. Algoritma K-Means dipilih karena kemampuannya dalam mengelompokkan data tidak terstruktur secara efisien serta banyak digunakan dalam penelitian *data mining* berbasis teks. Variabel penelitian berupa fitur kata hasil ekstraksi TF-IDF, sedangkan objek yang dikelompokkan adalah ulasan pengguna aplikasi pinjaman online.

Proses pemodelan dan analisis data dilakukan dengan bantuan aplikasi Orange, yang mendukung penerapan teknik *text mining*, ekstraksi fitur, serta algoritma K-Means. Penggunaan Orange mempermudah proses analisis data secara visual dan sistematis, serta meningkatkan keterulangan (*reproducibility*) penelitian.

C. HASIL DAN PEMBAHASAN

Pada penelitian analisis sentimen ini digunakan metode *data mining* dengan algoritma K-Means untuk mengelompokkan ulasan pengguna aplikasi pinjaman online yang terdaftar dan diawasi oleh Otoritas Jasa Keuangan (OJK), yaitu Kredivo, AdaKami, Akulaku, Indodana, dan Easycash. Penelitian bertujuan untuk mengidentifikasi karakteristik sentimen pengguna sehingga dapat diketahui aplikasi dengan persepsi pengguna yang lebih baik berdasarkan pola ulasan yang terbentuk. Data penelitian diperoleh melalui proses crawling atau *scraping* ulasan pengguna dari *Google Play Store*, kemudian diproses melalui tahapan *preprocessing* teks, pembobotan fitur, dan proses *clustering* menggunakan algoritma K-Means.

Proses Penarikan Data

Tahap penarikan data merupakan langkah awal yang krusial dalam penelitian ini untuk memperoleh data sekunder yang

akan dianalisis. Data yang digunakan berupa ulasan teks dari pengguna aplikasi pinjaman online yang beroperasi secara legal dan terdaftar di Otoritas Jasa Keuangan (OJK). Proses pengambilan data dilakukan dengan menerapkan teknik web *scraping* pada platform distribusi digital *Google Play Store*. Untuk mengekstraksi ulasan tersebut, penelitian ini memanfaatkan pustaka pemrograman Python, secara spesifik menggunakan library

Proses komputasi penarikan data ini diimplementasikan di lingkungan notebook menggunakan bahasa pemrograman Python. Berikut adalah contoh implementasi kode program saat melakukan *scraping* pada salah satu objek penelitian.

Gambar 2. Kode Pengambilan Data Ulasan

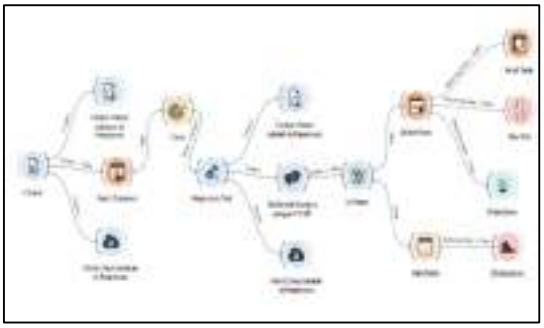
Sintaks pada Gambar 2 menunjukkan proses pengambilan data ulasan aplikasi menggunakan fungsi reviews() dari pustaka google-play-scraper. Parameter yang digunakan meliputi identitas aplikasi yang diambil berdasarkan package name, pengaturan bahasa dan negara, metode pengurutan ulasan, serta jumlah data yang ingin diperoleh. Hasil *scraping* kemudian disimpan dalam struktur data yang selanjutnya diubah menjadi bentuk tabel menggunakan pustaka Pandas untuk memudahkan proses analisis selanjutnya.

Data yang berhasil dikumpulkan kemudian disimpan dalam format spreadsheet untuk tahap *preprocessing* dan analisis lebih lanjut menggunakan metode *text mining* dan *clustering*. Berikut adalah data hasil *scraping* dengan bahasa python untuk aplikasi Kredivo :

Gambar 3. Data Hasil *Scraping* Aplikasi Kredivo

Pengolahan Data Menggunakan Tools Orange

Pada penelitian ini, proses pengolahan data dilakukan menggunakan Orange *Data mining* karena Orange menyediakan *workflow* visual berbasis *Widget* yang memudahkan tahapan *text mining* dan *clustering* dilakukan secara terstruktur, transparan, serta mudah direplikasi. Alur pengolahan data disusun mulai dari pemuatan data ulasan sebagai corpus, seleksi atribut yang digunakan, pembersihan teks (*preprocessing*), transformasi teks menjadi fitur numerik menggunakan TF-IDF, hingga pengelompokan data dengan algoritma K-Means. Selain itu, hasil pengelompokan dianalisis melalui tabel dan visualisasi untuk mempermudah interpretasi karakteristik masing-masing klaster. Seluruh tahapan tersebut ditunjukkan pada *workflow* Orange yang akan disajikan pada gambar berikut:



Gambar 4. Desain Sistem Pada Tools Orange

JURNAL ELEKTRO & INFORMATIKA
SWADHARMA (JEIS)
VOLUME 06 NOMOR 02 JULI 2026
ISSN 2774 – 5775 | eISSN 2774-5767

Analisis Hasil Penelitian Menggunakan Tools Orange

Pada bagian ini dijelaskan hasil analisis keluaran (output) dari *workflow* Orange *Data mining* menggunakan 3 klaster (C1, C2, C3). Analisis dilakukan dengan memanfaatkan empat *Widget* utama, yaitu *Pivot Table*, *Box Plot*, *Predictions*, dan *Distributions*. Keempat *Widget* tersebut digunakan untuk: (1) merangkum jumlah anggota klaster pada tiap aplikasi, (2) memvisualisasikan proporsi klaster per aplikasi serta menguji perbedaan distribusinya, (3) menampilkan hasil penugasan klaster pada tiap ulasan beserta indikator kualitas klaster (*silhouette*), dan (4) menggambarkan pola distribusi atribut numerik (misalnya rating) sebagai informasi pendukung interpretasi

Analisis dilakukan melalui empat tahapan utama:

1. Analisis *Pivot Table*

Berdasarkan hasil analisis *Pivot Table*, diperoleh distribusi jumlah ulasan pengguna terhadap lima aplikasi pinjaman online (AdaKami, Akulaku, Easycash, Indodana, dan Kredivo) yang telah dikelompokkan ke dalam tiga *cluster* (C1, C2, dan C3).

Tabel 2. Distribusi Ulasan Pengguna Aplikasi

Aplikasi	*Cluster* 1 (C1)	*Cluster* 2 (C2)	*Cluster* 3 (C3)	Total
AdaKami	452 (45,2%)	286 (28,6%)	262 (26,2%)	1.000
Akulaku	441 (44,1%)	331 (33,1%)	228 (22,8%)	1.000
Easycash	390 (39,0%)	378 (37,8%)	232 (23,2%)	1.000
Indodana	481 (48,1%)	307 (30,7%)	212 (21,2%)	1.000
Kredivo	492 (49,2%)	390 (39,0%)	118 (11,8%)	1.000
Total	**2.256**	**1.692**	**1.052**	**5.000**

2. Analisis *Box Plot*

Hasil klaster divisualisasikan menggunakan *Widget Box Plot* dengan variabel *Cluster* dan subgroup aplikasi, sehingga setiap aplikasi ditampilkan sebagai bar horizontal yang menggambarkan proporsi C1, C2, dan C3. Secara visual, grafik *Box Plot* memperkuat hasil *Pivot Table* bahwa komposisi klaster bervariasi antar aplikasi.

Gambar 5. Hasil *Widget Box Plot*

Pada bagian bawah grafik *Box Plot* ditampilkan hasil uji statistik $\chi^2 = 94{,}82$ ($p = 0{,}000$; $dof = 8$). Nilai ini mengindikasikan bahwa terdapat perbedaan distribusi klaster yang signifikan antar aplikasi, sehingga proporsi C1–C3 pada tiap aplikasi tidak identik. Dengan kata lain, karakteristik ulasan pengguna pada setiap aplikasi cenderung membentuk kecenderungan klaster yang berbeda (misalnya ada aplikasi yang lebih dominan di C1, atau memiliki C3 yang relatif lebih besar.

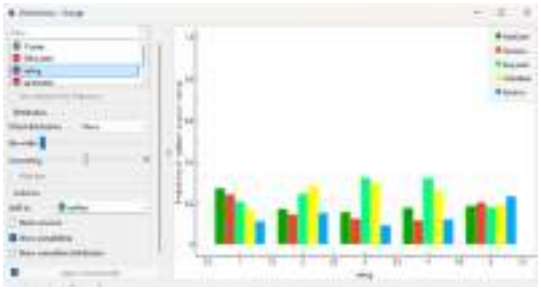
3. Analisis Prediksi

Dari tampilan *Predictions*, peneliti dapat melakukan pengecekan kualitas hasil klaster melalui nilai *silhouette*. Secara konseptual, semakin tinggi nilai *silhouette* maka semakin baik data tersebut terpisah dari klaster lain dan semakin sesuai dengan klasternya. Selain itu, *Predictions* memungkinkan peneliti meninjau contoh ulasan pada masing-masing klaster untuk memperoleh gambaran awal mengenai sentimen yang dominan pada klaster tertentu.

4. Analisis Distribusi

Widget Distributions digunakan untuk menganalisis sebaran atribut numerik, pada penelitian ini ditampilkan rating dengan pengaturan *Split by* = aplikasi. Hasil visualisasi menunjukkan bahwa distribusi rating antar aplikasi tidak sepenuhnya sama,

karena tiap aplikasi memiliki kecenderungan kemunculan rating tertentu yang berbeda.



Gambar 6. Hasil Tampilan *Distributions*

Walaupun grafik *Distributions* pada tampilan ini menggambarkan sebaran rating berdasarkan aplikasi (bukan langsung berdasarkan klaster), informasi tersebut tetap relevan sebagai konteks interpretasi. Secara umum, kecenderungan rating yang lebih tinggi sering berkorelasi dengan ulasan bernada positif, sedangkan rating lebih rendah cenderung berkorelasi dengan ulasan yang memuat keluhan. Oleh karena itu, *Distributions* dipakai sebagai informasi pendukung oleh peneliti menetapkan makna sentimen pada masing-masing klaster.

D. PENUTUP

Penelitian ini berhasil melakukan analisis sentimen terhadap ulasan pengguna aplikasi pinjaman online yang meliputi aplikasi Kredivo, AdaKami, Akulaku, Indodana, dan Easycash. Data ulasan yang diperoleh kemudian diproses melalui tahapan *preprocessing* teks yang meliputi *case folding*, *tokenizing*, *stopword removal*, dan *stemming*, sehingga data teks yang semula tidak terstruktur dapat diolah menjadi bentuk yang lebih sistematis untuk proses analisis lebih lanjut.

Proses ekstraksi fitur menggunakan metode TF-IDF mampu mengubah data teks ulasan menjadi representasi numerik yang dapat digunakan sebagai input dalam proses *clustering*. Representasi tersebut memungkinkan algoritma K-Means mengidentifikasi kemiripan antar ulasan berdasarkan bobot kata yang muncul dalam setiap dokumen ulasan pengguna.

Penerapan algoritma K-Means dalam penelitian ini berhasil mengelompokkan data ulasan pengguna ke dalam beberapa klaster berdasarkan kemiripan pola sentimen dan karakteristik teks ulasan. Klaster yang terbentuk menunjukkan adanya perbedaan persepsi pengguna terhadap layanan aplikasi pinjaman online, yang dapat dikategorikan ke dalam kelompok sentimen positif, netral, dan negatif.

Hasil pengelompokan klaster memberikan gambaran mengenai pola persepsi dan pengalaman pengguna terhadap layanan pinjaman online. Sebagian pengguna memberikan sentimen positif yang berkaitan dengan kemudahan proses pengajuan pinjaman dan kecepatan pencairan dana, sementara sentimen negatif umumnya berkaitan dengan keluhan mengenai suku bunga, transparansi biaya, serta kualitas layanan pelanggan.

Secara keseluruhan, penelitian ini menunjukkan bahwa algoritma K-Means efektif digunakan untuk mengelompokkan ulasan pengguna aplikasi pinjaman online berdasarkan kemiripan karakteristik sentimen. Hasil *clustering* yang diperoleh dapat memberikan pemahaman yang lebih sistematis mengenai pola opini pengguna serta menjadi sumber informasi yang bermanfaat bagi pengembangan kualitas layanan aplikasi pinjaman online.

E. DAFTAR PUSTAKA

Akbar, R. F., Habibi, M., Cahyo, P. W., & Sa'diya, N. A. (2023). Metode Hybrid Menggunakan Pendekatan Lexicon Based dan Naive Bayes Classifier Untuk Analisis Sentimen Terkait Jaminan Hari Tua. *Teknomatika: Jurnal Informatika Dan Komputer*, *16*(2), 73–79. https://doi.org/10.30989/teknomatika.v16i2.1247

Andini, D., & Muthia, D. A. (2024). Implementasi Metode Lexicon Based dan Support Vector Machine Pada Analisis Sentimen Ulasan Pengguna ChatGPT. *IJCIT : Indonesian Journal on Computer and Information Technology*, *9*(2), 80–86. https://doi.org/10.31294/ijcit.v9i2.23948

Bela, S., Harianja, P. A., Bismi, W., Kurniawati, I., & Fahlapi, R. (2026). Analisis Sentimen dan Kepuasan Pengguna Aplikasi Pinjaman Online Berdasarkan Ulasan Di Google Play. *RIGGS: Journal of Artificial Intelligence and Digital Business*, *4*(4), 6170–6180. https://doi.org/10.31004/riggs.v4i4.4245

Gandasari, N. M., Hidayat, R. R., & Siswajanthy, F. (2025). Peran Otoritas Jasa Keuangan (OJK) dalam Mengawasi Fintech Lending sebagai Instrumen Ekonomi Digital. *IJIJEL : Indonesian Journal of Islamic Jurisprudence, Economic and Legal Theory*, *3*(1), 399–408. https://doi.org/10.62976/ijijel.v3i1.941

Ghozali, M. I., Sugiharto, W. H., & Iskandar, A. F. (2023). Analisis Sentimen Pinjaman Online Di Media Sosial Twitter Menggunakan Metode Naive Bayes. *KLIK: Kajian Ilmiah Informatika Dan Komputer*, *3*(6), 1340–1348. https://doi.org/10.30865/klik.v3i6.936

Haq, M. Z., Octiva, C. S., Ayuliana, A., Nuryanto, U. W., & Suryadi, D. (2024). Algoritma Naïve Bayes untuk Mengidentifikasi Hoaks di Media Sosial. *MInfo Polgan : Jurnal & Penelitian Manajemen Informatika*, *13*(1), 1079–1084. https://doi.org/10.33395/jmp.v13i1.13937

Indriastuti, P., Hartini, E. F., & Sismadi, W. (2025). Pengambilan Keputusan Pinjaman Online Dan Faktor Yang Mempengaruhi. *Aliansi: Jurnal Manajemen Dan Bisnis*, *20*(1), 241–252. https://doi.org/10.46975/zvc8pg94

Ipnuwati, S., Zuhri, K., & Yuniarthe, Y. (2025). Unsupervised Sentiment Analysis pada Komentar Evaluasi Dosen Program Studi Informatika Universitas Mitra Indonesia. *JEDA : Jurnal Teknologi Dan Informatika*, *6*(1), 1–8. https://doi.org/10.57084/jeda.v6i1.1988

Izzah, S., Novilia, E., & Harwida, G. (2025). Perkembangan dan Dampak Inovasi Fintech Terhadap Perilaku Manajemen Keuangan Masyarakat Kota Blitar. *JRME : Jurnal Rumpun Manajemen Dan Ekonomi*, *2*(6), 466–476. https://doi.org/10.61722/jrme.v3i2.8804

Mongkito, L. O. M. H. A. S., Ransi, N., Surimi, L., Tenriawaru, A., Gunawan, G., & Rauf, B. W. (2024). Analisis Sentimen Aplikasi Peminjaman Online Berdasarkan Ulasan Pada Play Store Menggunakan Metode Naive Bayes dan Support Vector Machine (Studi Kasus : Adakami dan Easycash). *AnoaTIK: Jurnal Teknologi Informasi Dan Komputer*, *2*(2), 121–128. https://doi.org/10.33772/anoatik.v2i2.71

Muningsih, E. (2022). Kombinasi Metode K-Means dan Decision Tree Dengan Perbandingan Kriteria dan Split Data. *Jurnal Teknoinfo*, *16*(1), 113–118. https://doi.org/10.33365/jti.v16i1.1561

Nurhasanah, E. (2026). *Analisis Sentimen Pada Teks Digital: Memahami Emosi dan Opini di Dunia Maya*. Sijunjung : Yayasan Pendidikan Cendekia Muslim.

Rahmanda, M. A., Ananda, G., & Pratika, M. R. (2025). Perlindungan Konsumen dalam Transaksi Pinjaman Online: Kajian terhadap UU No. 8 Tahun 1999. *JEM Terekam Jejak : Journal of Economic and Management*, *2*(1), 1–12. https://journal.terekamjejak.com/index.php/jem/article/view/290

Reyan, M., & Purwaningtyas, F. (2025). Analisis Sentimen Ulasan Aplikasi IBI

Library Pada Google Play Store Menggunakan Naive Bayes Clssifier. *Djtechno: Jurnal Teknologi Informasi*, *6*(2), 662–679. https://doi.org/10.46576/djtechno.v6i2.6968

Sujjadaa, A., Somantri, Novianti, J. N., & Isa, I. G. T. (2023). Analisis Sentimen Terhadap Review Bank Digital Pada Google Play Store Menggunakan Metode Support Vector Machine (SVM). *Jurnal Rekayasa Teknologi Nusa Putra*, *9*(2), 122–135. https://doi.org/10.52005/rekayasa.v9i2.345

Sunarko, G. S., Wasino, W., & Sutrisno, T. (2023). Klasterisasi Sentimen Ulasan Pengguna Aplikasi BCA Mobile Pada Platform Google Play Store Dengan Algoritma K-Means Clustering. *Jurnal Ilmu Komputer Dan Sistem Informasi*, *11*(1), 1–6. https://doi.org/10.24912/jiksi.v11i1.24145

Yustati, H., Agustiya, R., Indra, Y. A., & Situmorang, D. R. (2025). Faktor Pendorong Penggunaan Pinjaman Online di Kalangan Mahasiswa. *JURIHUM : Jurnal Inovasi Dan Humaniora*, *3*(2), 153–160. https://jurnalmahasiswa.com/index.php/Jurihum/article/view/3151

Zahir, M. R. A., Bagus, & Saputra, A. D. (2024). Analisis Sentimen Terhadap Ulasan Aplikasi Pinjaman Online (Pinjol) di Google Play Store Menggunakan Naive Baiyes Classifier. *Jurnal Riset Informatika Dan Teknologi Informasi*, *1*(2), 61–64. https://doi.org/10.58776/jriti.v1i2.39

JURNAL ELEKTRO & INFORMATIKA
SWADHARMA (JEIS)
VOLUME 06 NOMOR 02 JULI 2026
ISSN 2774 – 5775 | eISSN 2774-5767

SISTEM PENGAMAN PINTU OTOMATIS DENGAN KOMBINASI RF DAN VERIFIKASI PIN BERBASIS MIKROKONTROLER

Sujono[1)], Mochammad Rizqi Albani[2)]

[1,2]Prodi Teknik Informatika, Fak Teknologi Informasi, Universitas KH Abdul Wahab Hasbullah

Correspondence author: Sujono, sujono@unwaha.ac.id, Jombang, Indonesia

Abstract

Security for enclosed spaces or buildings is a crucial concern, as conventional manual locking systems, such as physical keys, are highly vulnerable to loss, unauthorized use, or duplication. To address this key issue, this study aims to design and implement an automatic door security system that combines Radio Frequency (RF) technology and Personal Identification Number (PIN) verification using a microcontroller. This study developed the system using a prototyping method, which enables iterative testing and refinement of the hardware and software to achieve an optimal, stable system. The resulting system is a prototype automatic door security system with an ESP32 microcontroller-based main controller that processes signals from a 433 MHz RF module and validates unique PIN inputs. The primary functional requirements include an RF transmitter, RF transmitter, and servo motor to operate the lock. The resulting solution provides more reliable, secure access and minimizes the risk of unauthorized entry by requiring Two-Factor Authentication (an RF signal and a PIN). This system offers a practical, economical, and applicable security solution for homes, laboratories, and other confined spaces.

Keyword *: door security, microcontroller, Two-Factor Authentication*

Abstrak

Keamanan untuk ruang tertutup atau bangunan merupakan perhatian yang krusial, karena sistem penguncian manual konvensional, seperti kunci fisik, sangat rentan terhadap kehilangan, penggunaan tanpa izin, atau penggandaan. Untuk mengatasi masalah utama ini, penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem keamanan pintu otomatis yang menggabungkan teknologi *Radio Frequency* (RF) dan verifikasi *Personal Identification Number* (PIN) menggunakan mikrokontroler. Penelitian ini mengembangkan sistem menggunakan metode *prototyping*, yang memungkinkan pengujian berulang dan penyempurnaan perangkat keras serta perangkat lunak hingga mencapai sistem yang optimal dan stabil. Hasil penelitian berupa purwarupa sistem pengaman pintu otomatis dengan pengendali utama berbasis mikrokontroler ESP32 untuk memproses sinyal dari modul RF 433 MHz dan memvalidasi input PIN yang unik. Kebutuhan fungsional utama mencakup RF *Transmitter*, RF *Transmitter*, dan motor servo untuk mengoperasikan kunci. Hasil penelitian menjadi solusi akses yang lebih andal dan aman serta meminimalkan risiko masuknya pihak yang tidak berwenang dengan mewajibkan *Two-Factor Authentication* (sinyal RF dan PIN).

Sistem ini menawarkan solusi keamanan yang praktis, ekonomis, dan dapat diterapkan untuk rumah, laboratorium, dan area terbatas lainnya.

Kata Kunci : keamanan pintu, mikrokontroler, *Two-Factor Authentication*

A. PENDAHULUAN

Keamanan properti merupakan kebutuhan mendasar untuk melindungi privasi dan aset dari berbagai potensi akses yang tidak sah (Fitriansyah & Suryanto, 2021). Pada sistem penguncian manual konvensional, penggunaan kunci fisik dinilai sangat rentan terhadap risiko kehilangan, duplikasi ilegal, maupun pembobolan mekanis menggunakan kunci tiruan (Arifin et al., 2022). Tingginya tingkat kriminalitas dan keahlian pembobolan yang semakin berkembang menuntut adanya inovasi pengaman pintu yang tidak lagi bergantung pada sistem mekanik murni, melainkan beralih ke perangkat elektronik terinfomasi. Hal ini dikarenakan kunci elektronik memiliki tingkat kesulitan pembobolan yang lebih tinggi, sebab selain diperlukan pengetahuan mengenai elektronik, pelaku kriminal juga harus memiliki pengetahuan di bidang pemrograman dan teknologi informasi (Manurung et al., 2023).

Hingga saat ini, banyak dikembangkan sistem pengamanan pintu yang mengandalkan teknologi identifikasi nirkabel seperti *Radio Frequency* (RF) atau RFID. Namun, sebagian besar sistem konvensional tersebut masih menerapkan metode autentikasi tunggal (*single authentication*), seperti kartu RF saja (Fauza & Muthalib, 2022; Mardiono et al., 2023). Sistem tunggal ini memiliki kelemahan kritis apabila kartu identitas tersebut hilang atau dicuri, sehingga dapat disalahgunakan oleh pihak lain untuk mendapatkan akses penuh. Dampak dari lemahnya sistem pengamanan satu faktor ini tidak hanya menyebabkan kerugian materiil, tetapi juga menimbulkan kekhawatiran psikologis bagi pengguna (Alfonsius et al., 2024). Meskipun beberapa penelitian terdahulu telah mengombinasikan frekuensi radio dengan kode *Personal Identification Number* (PIN), kinerjanya sering kali masih terbatas pada platform mikrokontroler generasi lama seperti keluarga AVR ATmega128. Penelitian lain juga menunjukkan bahwa penanganan sistem pengamanan jamak pada area bangunan atau rumah kos sering kali terkendala masalah pengkabelan LAN yang rumit serta ketidakpraktisan penggunaan komputer desktop konvensional sebagai *web server* (Leander et al., 2024).

Sebagai solusi untuk mengatasi keterbatasan tersebut, penelitian ini mengusulkan penerapan sistem *Two-Factor Authentication* (2FA) berbasis mikrokontroler ESP32 dengan konektivitas nirkabel (*wireless*) terintegrasi (Putri & Agung, 2025). Penggunaan chip seri ESP dipilih karena memiliki arsitektur dengan kecepatan pemrosesan data yang tinggi, memori yang lebih besar, serta efisiensi pengabelan melalui media nirkabel bawaan (on-board *wireless*) yang sangat ideal untuk menangani logika autentikasi hibrida berlapis (Nizam et al., 2022). Melalui pendekatan hibrida ini, proses autentikasi dirancang dalam dua tahapan: pertama menggunakan modul komunikasi transmiter RF sebagai pemicu akses nirkabel awal, kemudian dilanjutkan dengan kode PIN sebagai verifikasi lapis kedua. Dengan memadukan teknologi frekuensi radio dan verifikasi kode unik ini, tingkat keamanan sistem menjadi lebih kokoh karena mekanisme aktuator pengunci pintu hanya akan terbuka jika kedua syarat verifikasi terpenuhi secara valid dan berurutan (Hastuti et al., 2025).

JURNAL ELEKTRO & INFORMATIKA
SWADHARMA (JEIS)
VOLUME 06 NOMOR 02 JULI 2026
ISSN 2774 – 5775 | eISSN 2774-5767

B. METODE PENELITIAN

Metodologi yang digunakan dalam penelitian ini adalah metode *Prototyping*. Pemilihan metode ini didasarkan pada kebutuhan untuk mengembangkan sistem secara bertahap serta memungkinkan adanya evaluasi berkelanjutan terhadap kebutuhan perangkat keras dan perangkat lunak hingga mencapai hasil akhir yang optimal dan stabil (Nazuarsyah et al., 2023). Model *prototyping* terbukti sangat efektif digunakan ketika pengembang harus cepat dalam mempresentasikan fungsi-fungsi sistem kepada pengguna melalui iterasi desain (Pasmah et al., 2021).

Tahapan-tahapan penelitian yang dilakukan dijabarkan sebagai berikut:

1. Tahap Awal

Pada tahap awal, dilakukan identifikasi terhadap kebutuhan perangkat keras (*hardware*) dan perangkat lunak (*software*) guna menunjang keberhasilan sistem keamanan dua faktor. Pemahaman mendalam terhadap kebutuhan sistem dan pengguna menjadi fondasi utama dalam pembangunan instrumen ini agar seluruh fungsionalitas dapat berjalan tepat guna. Perangkat keras utama yang dianalisis meliputi mikrokontroler ESP32 sebagai pusat kendali (*brain*), modul pengirim (*transmitter*) RF, modul penerima (*transmitter*) RF 433 MHz, *keypad* matriks, *buzzer*, dan motor servo sebagai aktuator pengunci (Nizam et al., 2022). Sedangkan kebutuhan perangkat lunak difokuskan pada penggunaan Arduino IDE untuk perancangan algoritma program kontroler (Lubis & Susilawati, 2024).

2. Perancangan Sistem (Design)

Perancangan sistem difokuskan pada integrasi logika antara sinyal frekuensi radio dan verifikasi kode PIN hibrida. Sistem dirancang untuk memproses data input berupa sinyal *Radio Frequency* (RF) dan nomor *Personal Identification Number* (PIN) melalui instruksi pemrograman di mikrokontroler guna menghasilkan output berupa pembukaan kunci pintu. Alur kerja sistem secara teknis digambarkan melalui *flowchart* untuk memastikan tidak ada celah logika dalam proses autentikasi berlapis tersebut.

3. Pembangunan Prototipe (Construction)

Tahap ini meliputi perakitan komponen elektronik sesuai dengan skema rangkaian sirkuit yang telah dirancang. Implementasi dilakukan dengan menghubungkan pin-pin input dari penerima RF dan *keypad* ke pin digital kontroler, serta menghubungkan pin output ke motor servo dan *buzzer* yang bertindak sebagai indikator audio. Pada tahap konstruksi ini, pengabelan internal dirakit secara sistematis menggunakan papan sirkuit dan dilindungi isolasi untuk meminimalisir interferensi sinyal atau *noise* frekuensi pada pin I/O mikrokontroler selama operasional.

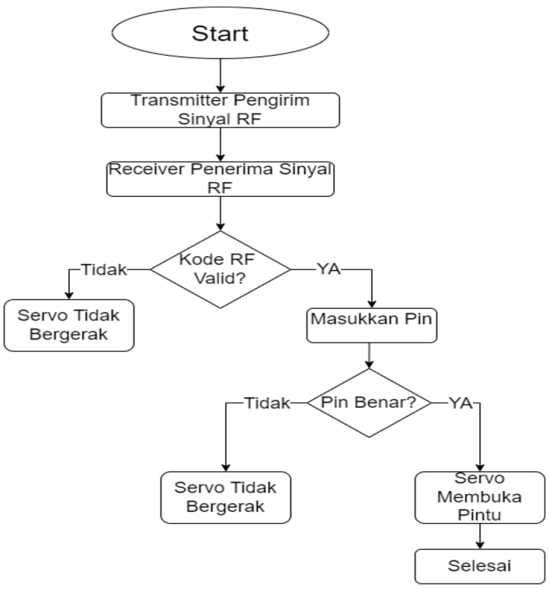
4. Pengujian dan Evaluasi (Testing)

Setelah fisik prototipe selesai dibangun, dilakukan pengujian menyeluruh untuk memastikan setiap fungsi berjalan sesuai rencana dan bebas dari kesalahan logika. Parameter pengujian tersebut meliputi beberapa aspek berikut:

a. Uji Jarak RF: Dilakukan dengan mengukur jarak operasional maksimum antara transmiter nirkabel dan antena penerima RF dalam kondisi tanpa halangan (Line of Sight). Pengujian ini bertujuan untuk memberikan kenyamanan bagi pengguna dalam mengaktifkan sistem dari jarak jauh.
b. Uji Validasi PIN: Memastikan sistem memberikan respons yang tepat untuk menarik tuas aktuator servo hanya ketika kode PIN dimasukkan secara benar melalui *keypad* setelah sinyal RF terverifikasi awal oleh kontroler.
c. Uji Integrasi dan Kegagalan: Memastikan pengunci pintu tetap aman jika salah satu faktor tidak terpenuhi, serta memastikan mikrokontroler mengaktifkan *buzzer* sebagai alarm peringatan audio apabila mendeteksi adanya kegagalan input PIN secara berulang.

C. HASIL DAN PEMBAHASAN

Jalannya sistem pengaman pintu otomatis yang akan dibangun digambarkan dalam bentuk *flowchart* pada gambar 1.

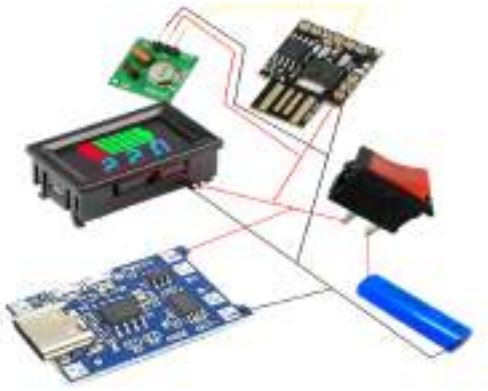


Gambar 1. *Flowchart*

Berdasarkan arsitektur logika pada Gambar 1, jalannya sistem dijabarkan melalui poin-poin interaksi berikut:

1. Inisialisasi dan Mode Siaga (*Start/Standby*): Sistem memulai operasi dengan melakukan inisialisasi seluruh modul perangkat keras yang terhubung pada pin I/O ESP32. Dalam kondisi awal ini, aktuator motor servo diposisikan pada sudut penguncian penuh sehingga pintu dalam kondisi terkunci aman.
2. Autentikasi Faktor Pertama (Deteksi Sinyal RF): Sistem secara kontinu melakukan pemindaian nirkabel untuk mendeteksi adanya sinyal masuk dari modul *transmitter* RF 433 MHz. Jika modul penerima (*transmitter*) tidak menangkap frekuensi yang sesuai, kontroler akan mempertahankan kondisi pintu tetap terkunci dan kembali ke mode siaga. Sebaliknya, jika sinyal nirkabel dari remote pengendali terdeteksi dan dinyatakan valid, sistem akan mengaktifkan lampu indikator atau *buzzer* sebagai tanda bahwa faktor pertama berhasil dipenuhi.
3. Autentikasi Faktor Kedua (Input dan Validasi PIN): Setelah akses nirkabel RF berhasil dilewati, ESP32 akan membuka gerbang instruksi kedua dan menunggu pengguna memasukkan kode keamanan melalui perangkat *keypad* matriks. Sistem kemudian membandingkan kombinasi digit angka yang ditekan oleh pengguna dengan data kode PIN unik yang telah disimpan di dalam memori mikrokontroler (dalam penelitian ini menggunakan PIN: "2567").
4. Eksekusi Akses Sukses: Apabila nomor PIN yang dimasukkan oleh pengguna terbukti cocok dan benar, kontroler ESP32 akan mengirimkan sinyal PWM (*Pulse Width Modulation*) menuju motor servo untuk mengubah sudut mekanis tuas pengunci, sehingga pintu otomatis terbuka. Sistem akan mempertahankan kondisi pintu terbuka selama jeda waktu tertentu sebelum mengembalikan servo ke posisi mengunci semula.
5. Penanganan Kegagalan dan Sistem Alarm: Jika pengguna memasukkan kode PIN yang salah, mikrokontroler akan langsung memblokir instruksi pembukaan pintu dan mengaktifkan *buzzer* secara intermiten sebagai alarm peringatan audio terjadinya kesalahan akses. Sistem kemudian akan menghitung jumlah kegagalan input; jika batas toleransi kesalahan dilewati, sistem akan mengunci antarmuka *keypad* untuk sementara waktu guna meminimalisir upaya intrusi paksa (*brute force*) sebelum akhirnya kembali ke titik awal pemindaian sinyal RF.

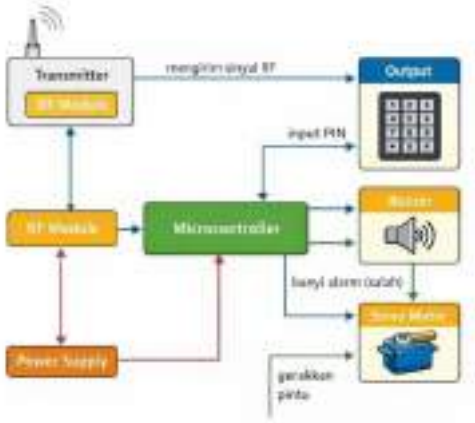
JURNAL ELEKTRO & INFORMATIKA
SWADHARMA (JEIS)
VOLUME 06 NOMOR 02 JULI 2026
ISSN 2774 – 5775 | eISSN 2774-5767



Gambar 2. Skematik *Transmitter*

Gambar 3. Skematik *Transmitter*

Sistem pengaman pintu ini dirancang menggunakan dua sirkuit terpisah, yaitu sirkuit pengirim (*transmitter*) dan penerima (*transmitter*). Skematik rangkaian *transmitter* pada Gambar 2 memperlihatkan modul pengirim RF 433 MHz yang ditenagai oleh sirkuit manajemen baterai lithium berbasis TP4056, lengkap dengan indikator kapasitas baterai dan sakelar daya. Sementara itu, skematik rangkaian *transmitter* pada Gambar 3 menunjukkan koneksi utama pada pusat kendali Esp, di mana pin input terhubung ke modul penerima RF 433 MHz dan *keypad* matriks 4x4, sedangkan pin output mengendalikan motor servo sebagai aktuator pengunci, *push button* sebagai interupsi manual, serta *buzzer* sebagai indikator alarm audio.



Gambar 4. Diagram blok

Berdasarkan Gambar 4 di atas, sistem ini bekerja dengan mengandalkan mikrokontroler sebagai pusat kendali (brain) untuk mengolah data dari dua sumber input utama:

1. RF Module (*Transmitter* & *Transmitter*): Berfungsi sebagai pemicu awal nirkabel. Sinyal dikirimkan oleh *transmitter* dan diterima oleh *transmitter* untuk kemudian diteruskan ke mikrokontroler. Berdasarkan pengujian, komunikasi RF ini sangat efektif untuk akses jarak jauh (*Line of Sight*).
2. *Keypad* (Input PIN): Berfungsi sebagai autentikasi lapis kedua. Mikrokontroler akan mencocokkan kode yang ditekan pada *keypad* dengan data yang tersimpan.

Sesuai dengan alur pada diagram blok, jika kedua input tersebut dinyatakan valid, mikrokontroler akan memberikan perintah kepada Servo Motor untuk menggerakkan mekanisme pengunci pintu secara presisi. Sebaliknya, jika input PIN tidak sesuai, mikrokontroler akan mengaktifkan *Buzzer* sebagai alarm peringatan audio. Keamanan dua faktor (*Two-Factor Authentication*) ini terbukti menurunkan risiko pembobolan dibandingkan metode tunggal. Penggunaan Arduino Nano sebagai pengendali utama dipilih karena efisiensi daya dan ukuran yang praktis untuk prototipe.

Pada tahap ini, dilakukan pengujian laboratorium secara menyeluruh terhadap

seluruh komponen keras dan lunak yang telah diintegrasikan pada prototipe. Pengujian ini bertujuan untuk memvalidasi performa, akurasi logika, dan keandalan sistem pengaman pintu otomatis yang menggunakan kombinasi *Radio Frequency* (RF) dan verifikasi *Personal Identification Number* (PIN) berbasis ESP32.

Data hasil skenario pengujian autentikasi hibrida tersebut dirangkum dalam Tabel 1.

Tabel 1. Hasil Pengujian

No	Kondisi Pengujian	RF	PIN	Hasil
1	Valid	✓	✓	Servo membuka
2	Pin Salah	✓	✗	Servo tidak membuka
3	RF tidak valid	✗	-	Servo tidak membuka
4	Pin tidak dimasukkan	✓	-	Sistem menunggu pin

Berdasarkan data hasil pengujian fungsional pada Tabel 1, sistem pengaman hibrida ini menunjukkan tingkat keberhasilan rata-rata sebesar 85% dalam merespons seluruh variasi input valid yang diberikan. Angka persentase keberhasilan ini diperoleh dari akumulasi pengujian berulang sebanyak 20 kali percobaan untuk setiap skenario kondisi.

Munculnya kegagalan sistem operasional sebesar 15% dikategorikan sebagai galat minor (*minor error*). Hambatan fungsional tersebut umumnya disebabkan oleh faktor eksternal lingkungan, seperti adanya intervensi *noise* gelombang elektromagnetik sekitar yang mengganggu stabilitas pita frekuensi modul *transmitter* 433 MHz, atau jarak fisik transmisi antara remote pengirim (*transmitter*) dan antena penerima yang melebihi batas batas operasional maksimum yaitu 10 meter (*Line of Sight*).

D. PENUTUP

Berdasarkan hasil perancangan, implementasi, dan pengujian laboratorium yang telah dilakukan terhadap sistem pengaman pintu, Sistem pengaman pintu hibrida berbasis hibrida *Two-Factor Authentication* (2FA) berhasil dibangun dan diintegrasikan dengan memanfaatkan mikrokontroler ESP32 sebagai pusat kendali utama.

Logika pengamanan berlapis yang mengombinasikan modul transmisi nirkabel *Radio Frequency* (RF) 433 MHz (faktor pertama) dan verifikasi kode *Personal Identification Number* (PIN) unik melalui *keypad* matriks (faktor kedua) terbukti mampu meningkatkan kekokohan proteksi ruang dibandingkan sistem autentikasi tunggal konvensional.

Hasil pengujian fungsional secara keseluruhan menunjukkan bahwa sistem memiliki tingkat keberhasilan rata-rata sebesar 85% dalam memproses input valid dan menggerakkan motor servo sebagai aktuator pengunci pintu. Adanya margin kegagalan minor sebesar 15% murni disebabkan oleh faktor intervensi eksternal berupa *noise* gelombang elektromagnetik sekitar serta jarak operasional remote yang melebihi batas optimum (Line of Sight) sebesar 10 meter.

Untuk pengembangan dan penyempurnaan sistem pengaman pintu otomatis ini pada penelitian selanjutnya, disarankan untuk menambahkan metode enkripsi data (seperti algoritma *Rolling Code* atau AES) pada jalur komunikasi nirkabel transmiter RF 433 MHz untuk mengantisipasi risiko penyadapan sinyal frekuensi (signal sniffing) oleh pihak luar.

Memanfaatkan fitur konektivitas Wi-Fi terintegrasi (on-board *wireless*) yang dimiliki oleh chip ESP32 untuk menghubungkan perangkat dengan platform IoT (*Internet of Things*), sehingga pemilik bangunan dapat memonitor log riwayat akses pintu serta menerima notifikasi peringatan dini secara

real-time melalui aplikasi ponsel pintar (*smartphone*).

Mengembangkan sistem kendali daya cadangan menggunakan baterai manajemen sekunder yang lebih stabil guna memastikan perangkat pengaman tetap beroperasi secara optimal ketika terjadi pemutusan arus listrik utama.

E. DAFTAR PUSTAKA

Alfonsius, E., Ruitan, A. S., & Liuw, D. (2024). Pengembangan Sistem Keamanan Pintu Menggunakan Metode Prototype Berbasis RFID dan Keypad 4x4 dengan Arduino Nano. *JIMA-ILKOM : Jurnal Ilmiah Informatika Dan Ilmu Komputer*, *3*(2), 110–123. https://doi.org/10.58602/jima-ilkom.v3i2.33

Arifin, W., Fitriansyah, A., & Setiadi, D. (2022). Pembatasan Akses Secara Fisik Dengan Sistem Fingerprint Doorlock Menggunakan Microcontroller Arduino Uno R3. *JEIS: Jurnal Elektro Dan Informatika Swadharma*, *2*(2), 81–88. https://doi.org/10.56486/jeis.vol2no2.234

Fauza, M., & Muthalib, M. A. (2022). Sistem Pengamanan Pintu Otomatis Menggunakan Sensor Radio Frequency Identification (RFID) Berbasis Arduino Uno. *Jurnal Energi Elektrik*, *11*(1), 30–37. https://doi.org/10.29103/jee.v11i1.8185

Fitriansyah, A., & Suryanto, M. R. (2021). Teknologi Kontrol Lampu dan Kunci Rumah Berbasis IoT. *Jurnal Teknologi Informatika Dan Komputer*, *7*(1), 88–96. https://doi.org/10.37012/jtik.v7i1.505

Hastuti, P. T., Fitriandra, B., Dewi, N. S. S., & Pramono. (2025). Dual Security System: Akses Pintu Berbasis Rfid Dan Kode Rahasia. *Prosiding Seminar Nasional Teknologi Informasi Dan Bisnis (SENATIB)*, 1296–1301. https://doi.org/10.47701/32w10f27

Leander, D. E., Ashidiqie, M. F., & Udoyono, K. (2024). Perancangan Sistem Monitoring Jarak Jauh Pintu Pintar Rumah Indekos Berbasis IoT (Internet of Things) Menggunakan Platform Blynk. *JTIK: Jurnal Teknologi Informasi Dan Komunikasi*, *17*(2), 66–79. https://doi.org/10.47561/a.v17i2.265

Lubis, R. T., & Susilawati. (2024). Sistem Kunci Pintu Berbasis PIN Menggunakan Arduino dan Keypad. *JATI : Jurnal Mahasiswa Teknik Informatika*, *8*(3), 3830–3835. https://doi.org/10.36040/jati.v8i3.9820

Manurung, R. A. P., Erwansyah, K., & Azlan, A. (2023). Sistem Keamanan Gudang Penyimpanan Barang Handphone Berbasis RFID Dan Fingerprint. *Jurnal Sistem Komputer Triguna Dharma (JURSIK TGD)*, *2*(4), 215–220. https://doi.org/10.53513/jursik.v2i4.6713

Mardiono, D. A., Nanra, S., & Rican, D. (2023). Rancang Bangun Pengaman Pintu Menggunakan RFID Dengan Mikrokontroler Atmega 328. *Jurnal Teknologi Riset Terapan*, *1*(1), 11–18. https://doi.org/10.35912/jatra.v1i1.1872

Nazuarsyah, N., Muzakir, U., Mukhroji, M., Ginting, R., & Munadi, R. (2023). Sistem Identifikasi Menggunakan Rfid Dan Sensor Infrared Berbasis Iot Terhadap Pengembangan Kampus Pintar. *Cyberspace: Jurnal Pendidikan Teknologi Informasi*, *7*(2), 109. https://doi.org/10.22373/cj.v7i2.21632

Nizam, M., Yuana, H., & Wulansari, Z. (2022). Mikrokontroller ESP 32 Sebagai Alat Monitoring Pintu Berbasis Web. *JATI : Jurnal Mahasiswa Teknik Informatika*, *6*(2), 767–772. https://doi.org/10.36040/jati.v6i2.5713

Pasmah, R., Lubis, A. J., & Usman, A.

(2021). Prototipe Sistem Keamanan Ruangan Menggunakan Finger Print dan Keypad Matrix dengan One Time Pad. *Explorer: Journal of Computer Science and Information Technology*, *1*(2), 53–62. https://doi.org/10.47065/explorer.v1i2.89

Putri, W. G. R., & Agung, I. W. P. (2025). IoT Smart Door Lock System Menggunakan Double Sensor Berbasis Mikrokontroler ESP32. *Mnemonic : Jurnal Ilmu Komputer Dan Teknik Informatika*, *8*(1), 74–82. https://doi.org/10.36040/mnemonic.v8i1.12420

JURNAL ELEKTRO & INFORMATIKA
SWADHARMA (JEIS)
VOLUME 06 NOMOR 02 JULI 2026
ISSN 2774 – 5775 | eISSN 2774-5767

PURWARUPA SISTEM PERINGATAN DINI KEBAKARAN BERBASIS *INTERNET OF THINGS* MENGGUNAKAN MIKROKONTROLER ESP8266

Ahmad Fitriansyah[1)], Izzul Islam[2)], Christine Sientta Dewi[3)]

[1,2]Prodi Teknik Informatika, Fakultas Teknologi, ITB Swadharma
[3]Prodi Sistem Informasi, Fakultas Teknologi, ITB Swadharma

Correspondence author: A Fitriansyah, ahmad.fitriansyah@swadharma.ac.id, Jakarta, Indonesia

Abstract

Fire is a common disaster that can significantly impact human safety, damage infrastructure, and disrupt organizational activities. Delays in responding to fires can lead to greater losses. This research aims to design and build a Prototipe Internet of Things (IoT)-based fire early detection system using the ESP8266 microcontroller. The research method used in this study is the Research and Development (R&D) approach, comprising research, development, testing, and evaluation stages. The research resulted in a system designed using a DHT22 temperature and humidity sensor and a flame sensor to detect potential fires in real time. When the sensor detects extreme temperatures or a fire, the system automatically activates a buzzer as a local alert and sends a notification to the user via a Telegram bot. Testing was conducted at SMK YP IPPI Petojo. Test results showed that the system can detect potential fires with high accuracy, respond quickly, and provide warnings promptly. This Prototipe is expected to be a practical, affordable, and efficient solution for early fire hazard mitigation efforts, especially in schools and other public spaces that have limited fire safety systems.

Keyword *: Prototipe, early warning, fire, Internet of Things*

Abstrak

Kebakaran merupakan salah satu bencana yang sering terjadi dan dapat menimbulkan dampak yang signifikan terhadap keselamatan jiwa, kerusakan infrastruktur, serta gangguan terhadap aktivitas organisasi. Keterlambatan dalam merespons kebakaran dapat menyebabkan kerugian yang lebih besar. Penelitian ini bertujuan untuk merancang dan membangun sebuah Prototipe sistem deteksi dini kebakaran berbasis *Internet of Things* (IoT) yang menggunakan mikrokontroler ESP8266. Metode yang digunakan pada penelitian ini yaitu metode Research and Development (R&D) yang terdiri dari tahap research, development, testing & evaluation. Penelitian menghasilkan sistem yang dirancang menggunakan sensor suhu dan kelembaban DHT22 serta *flame sensor* untuk mendeteksi potensi kebakaran secara *real-time*. Ketika sensor mendeteksi suhu ekstrem atau keberadaan api, sistem secara otomatis akan mengaktifkan *buzzer* sebagai peringatan lokal dan mengirimkan notifikasi ke pengguna melalui bot Telegram. Pengujian dilakukan di lingkungan SMK YP IPPI Petojo. Hasil pengujian menunjukkan bahwa sistem mampu mendeteksi potensi kebakaran dengan akurasi yang baik, merespons dalam waktu singkat, dan memberikan

informasi peringatan secara cepat. Purwarupa ini diharapkan dapat menjadi solusi praktis, terjangkau, dan efisien dalam upaya mitigasi dini terhadap bahaya kebakaran, khususnya di lingkungan sekolah dan ruang publik lainnya yang memiliki keterbatasan dalam sistem keamanan kebakaran.

Kata Kunci : purwarupa, peringatan dini, kebakaran, *Internet of Things*

A. PENDAHULUAN

Kebakaran merupakan salah satu bencana yang dapat terjadi kapan saja dan di mana saja, termasuk di lingkungan pendidikan seperti sekolah. Dampak dari kebakaran tidak hanya mencakup kerusakan fisik terhadap bangunan dan aset, tetapi juga dapat mengancam keselamatan jiwa manusia, khususnya siswa, guru, dan staf sekolah (Ahmed & Saad, 2024). Dalam banyak kasus, keterlambatan dalam mendeteksi dan merespons kebakaran menjadi penyebab utama tingginya tingkat kerugian yang ditimbulkan (Elhanashi et al., 2025).

Berdasarkan observasi yang dilakukan di SMK YP IPPI PETOJO, diketahui bahwa sekolah tersebut belum memiliki sistem deteksi kebakaran dini yang terpasang di dalam gedung. Hal ini tentu menjadi perhatian serius, mengingat aktivitas pembelajaran di sekolah melibatkan banyak orang dan perangkat elektronik yang berpotensi menimbulkan kebakaran. Ketiadaan sistem deteksi kebakaran dini dapat memperbesar risiko keterlambatan penanganan apabila terjadi insiden, yang pada akhirnya dapat berdampak fatal (Khan et al., 2022).

Seiring berkembangnya teknologi, khususnya dalam bidang *Internet of Things* (IoT), kini dimungkinkan untuk membangun sistem deteksi kebakaran yang lebih efektif, efisien, dan dapat dipantau secara jarak jauh (Li et al., 2022). *Internet of Things* adalah konsep di mana perangkat-perangkat fisik saling terhubung melalui jaringan internet untuk saling bertukar data secara *real-time*. Dengan teknologi ini, sistem deteksi kebakaran tidak hanya memberikan peringatan lokal menggunakan *buzzer*, tetapi juga dapat mengirimkan notifikasi secara langsung ke perangkat *mobile* melalui aplikasi media sosial seperti Telegram, sehingga pihak terkait dapat segera mengetahui dan merespons kondisi darurat meskipun sedang tidak berada di lokasi (Swetha et al., 2026).

Penelitian terdahulu berperan penting dalam merancang sistem yang diusulkan. Studi sebelumnya memberikan dasar pemikiran serta pembanding terhadap teknologi dan metode yang diterapkan dalam pengembangan sistem peringatan dini kebakaran yang akan dibuat. Salah satu penelitian yang relevan dilakukan oleh (Salindeho & Wellem, 2023) yang merancang sistem pendeteksi api berbasis IoT menggunakan NodeMCU ESP8266 sebagai pusat kendali dan sensor api berbasis inframerah (phototransistor YG1006) sebagai alat pendeteksi nyala api. Sistem yang dirancang mampu mendeteksi radiasi cahaya dari nyala api hingga jarak maksimal 50 cm. Ketika nyala api terdeteksi, sistem secara otomatis mengaktifkan *buzzer* sebagai alarm lokal dan mengirimkan notifikasi peringatan melalui dua media, yaitu Telegram dan email. Selain itu, sistem ini terhubung ke platform Thinger.io, yang memungkinkan pengguna untuk memantau status lingkungan secara *real-time* dari jarak jauh. Penelitian ini membuktikan bahwa kombinasi antara sensor api dan teknologi IoT dapat membentuk sistem peringatan dini yang efektif, meskipun masih terbatas pada jangkauan deteksi dan belum mencakup parameter lain seperti asap atau suhu.

Penelitian yang dilakukan oleh (Asronika et al., 2024) yang merancang dan

mengimplementasikan sistem pendeteksi kebakaran ruangan dengan memanfaatkan teknologi *Internet of Things* (IoT) yang terintegrasi dengan aplikasi Blynk App pada *smartphone* Android. Sistem menggunakan NodeMCU ESP8266 sebagai mikrokontroler utama serta dilengkapi dengan beberapa sensor, seperti sensor api KY-026, sensor asap MQ-2, dan sensor suhu DHT11, yang bekerja secara bersamaan untuk memantau kondisi lingkungan secara *real-time*. Ketika terdeteksi suhu tinggi, api, atau asap, sistem akan mengaktifkan *buzzer* sebagai alarm lokal, serta mengirimkan notifikasi langsung ke pengguna melalui aplikasi Blynk dan email. Penelitian ini menunjukkan bahwa sistem mampu memberikan respons cepat terhadap kondisi bahaya dan diharapkan dapat menjadi solusi efektif dalam pencegahan dini kebakaran di lingkungan dalam ruangan. Sistem yang dikembangkan juga dinilai cukup efisien dan mudah diakses oleh pengguna karena hanya memerlukan koneksi internet dan *smartphone*. (Asronika et al., 2024).

Penelitian yang dilakukan oleh (Mulyono et al., 2021) yang merancang dan mensimulasikan sistem alarm kebakaran otomatis dengan memanfaatkan dua jenis sensor, yaitu sensor api (*flame sensor*) dan sensor asap (MQ-2), yang dikendalikan oleh mikrokontroler Arduino Uno. Sistem ini dirancang untuk memberikan output berupa tampilan pada LCD (LM016L) dan bunyi peringatan dari *buzzer* ketika terdeteksi adanya kebakaran. Pengujian dilakukan melalui software Proteus 8 Professional dengan skenario simulasi yang melibatkan tiga kondisi logika dari masing-masing sensor. Hasil simulasi menunjukkan bahwa ketika hanya satu sensor aktif (baik api maupun asap), sistem akan menampilkan status "*smart home standby*". Namun, jika kedua sensor mendeteksi secara bersamaan, maka *buzzer* berbunyi dan LCD menampilkan teks "kebakaran". Sistem ini dinilai efektif sebagai alat edukasi sekaligus representasi kerja sistem deteksi kebakaran otomatis dengan tingkat keakuratan yang tinggi serta respon cepat terhadap keberadaan asap dan nyala api di suatu ruangan.

Penelitian yang dilakukan oleh (Abrar et al., 2020) yang merancang dan membangun prototipe sistem pendeteksi kebakaran berbasis IoT yang mampu mendeteksi keberadaan api dan asap secara otomatis dengan notifikasi berupa panggilan darurat (emergency call) dan SMS. Sistem ini menggabungkan penggunaan sensor *flame*, sensor asap MQ-2, dan modul GSM SIM800L, serta dikendalikan oleh mikrokontroler Arduino Uno. Hasil pengujian menunjukkan bahwa *flame sensor* mampu mendeteksi api hingga jarak 25 cm, dan MQ-2 mendeteksi asap dengan waktu respon rata-rata mulai dari 1,4 detik (jarak 5 cm) hingga 10,2 detik (jarak 25 cm). Ketika sensor mendeteksi kebakaran, sistem akan mengaktifkan *buzzer* dan mengirim peringatan melalui SIM800L berupa miscall dan pesan teks "Kebakaran Terjadi!" ke nomor yang telah diprogram. Penelitian ini menyimpulkan bahwa sistem mampu bekerja secara efektif dan cepat dalam mendeteksi potensi kebakaran serta memberikan peringatan kepada pengguna secara *real-time*. Meskipun begitu, penelitian juga mencatat kelemahan pada kestabilan tegangan SIM800L dan sensitivitas terhadap kualitas sinyal. Sistem ini diusulkan untuk ditingkatkan dengan penambahan sensor suhu dan IC regulator untuk efisiensi yang lebih baik.

Penelitian yang dilakukan oleh (Nugraha & Satria, 2022) yang merancang sebuah prototipe alat pendeteksi kebakaran berbasis mikrokontroler Arduino Uno dengan menggunakan sensor *flame* untuk mendeteksi keberadaan api dan sensor MQ-2 untuk mendeteksi asap serta gas. Ketika sensor-sensor ini mendeteksi tanda-tanda kebakaran, sistem secara otomatis akan mengaktifkan *buzzer* dan LED sebagai indikator peringatan dini. Pengujian menunjukkan bahwa *flame sensor* dapat mendeteksi api pada jarak 5–25 cm, sementara sensor MQ-2 berhasil

mendeteksi kepadatan asap dan gas pada rentang nilai 760–775 ppm. Sistem yang dikembangkan terbukti mampu memberikan notifikasi dini secara lokal melalui suara dan cahaya, serta menunjukkan performa yang stabil selama pengujian. Penelitian ini menyimpulkan bahwa perangkat yang dibuat layak digunakan sebagai alat bantu untuk meningkatkan kewaspadaan terhadap potensi kebakaran di lingkungan terbatas seperti ruangan atau bangunan kecil.

Perancangan sistem peringatan dini kebakaran yang akan dibuat merupakan penyempurnaan dari studi terdahulu. Persyaratan sistem konseptual yang akan dibuat bertujuan untuk membantu mendeteksi potensi kebakaran secara dini dengan memberikan notifikasi melalui jaringan *Internet of Things* (IoT). Sistem ini akan dirancang dalam bentuk purwarupa menggunakan mikrokontroler ESP8266

Penelitian ini bertujuan untuk merancang dan membangun sebuah Prototipe sistem deteksi kebakaran dini berbasis *Internet of Things* menggunakan mikrokontroler ESP8266 yang akan diuji coba pada studi kasus di SMK YP IPPI PETOJO. Prototipe ini akan menggunakan sensor api dan suhu untuk mendeteksi potensi kebakaran, dan memberikan peringatan baik secara lokal maupun jarak jauh. Dengan sistem ini, diharapkan potensi keterlambatan dalam penanganan awal kebakaran dapat diminimalisir.

Sistem peringatan dini kebakaran yang dirancang memiiliki keterbatasan/limitasi sebagai berikut : (1) Sistem hanya mendeteksi indikator kebakaran dari suhu tinggi dan keberadaan api; (2) Sistem bersifat Prototipe dan diuji dalam ruang simulasi, belum digunakan untuk implementasi skala besar. (3) Sistem hanya dapat memberikan peringatan, tidak memiliki fitur pemadaman otomatis.

Melalui pengembangan sistem ini, tidak hanya diharapkan dapat memberikan solusi terhadap permasalahan yang ada di SMK YP IPPI PETOJO, tetapi juga menjadi kontribusi nyata dalam memperluas implementasi teknologi IoT di bidang keselamatan dan keamanan.

B. METODE PENELITIAN

Metode yang digunakan pada penelitian ini yaitu metode *Research and Development* (*R&D*). Metode R&D adalah suatu proses atau langkah-langkah untuk mengembangkan suatu produk baru, atau menyempurnakan produk yang telah ada, dan menguji keefektifan produk tersebut. Tahapan-Tahapan R&D untuk Alat Deteksi Kebakaran IoT adalah sebagai berikut :

Fase 1: *Research* (Penelitian dan Analisis Kebutuhan)

Tujuan: Memahami masalah secara mendalam dan merumuskan spesifikasi yang jelas.

1. *Research and Information Collecting* (Pengumpulan Data dan Informasi)

Studi Literatur: Mempelajari teori dasar tentang kebakaran, karakteristik asap/api, jenis sensor (seperti Sensor MQ-2 untuk asap/gas, sensor suhu DHT22, sensor flame), mikrokontroler (ESP32 atau ESP8266 karena sudah memiliki WiFi internal), dan platform IoT (seperti Blynk, Firebase, atau Node-RED).

2. Analisis Kebutuhan (*Need Assessment)*

Mengidentifikasi masalah yang dihadapi. Misalnya: respons sistem pemadam konvensional yang lambat, tidak adanya notifikasi jarak jauh, atau kurangnya pemantauan real-time.

3. Studi *Existing Product*

Menganalisis kelebihan dan kekurangan produk sejenis yang sudah ada di pasaran untuk menemukan celah inovasi.

4. *Planning* (Perencanaan)

Perumusan Spesifikasi Produk: Berdasarkan analisis kebutuhan, tentukan spesifikasi alat yang akan dibuat yaitu dapat mendeteksi api dan kenaikan suhu, dapat

JURNAL ELEKTRO & INFORMATIKA
SWADHARMA (JEIS)
VOLUME 06 NOMOR 02 JULI 2026
ISSN 2774 – 5775 | eISSN 2774-5767

mengirimkan notifikasi push ke *smartphone*, memiliki sirene atau *buzzer* lokal.

Fase 2: *Development* (Pengembangan Produk Awal)

Tujuan: Membuat purwarupa (prototype) pertama berdasarkan hasil penelitian.

1. *Development of Preliminary Form of Product* (Pengembangan Desain Awal)

Desain Sistem & Skematik: Merancang diagram blok sistem dan skematik rangkaian elektronik. Blok Input: Sensor Suhu, Sensor Flame. Blok Proses: Mikrokontroler ESP8266. Blok Output: *Buzzer*. Blok IoT Platform: Server untuk menerima dan mengolah notifikasi

2. Pengembangan Perangkat Keras (*Hardware*):

Membeli komponen sesuai spesifikasi, Merakit komponen pada papan prototyping (breadboard) untuk pengujian fungsionalitas dasar.

3. Pengembangan Perangkat Lunak (*Software*):

Firmware: Menulis kode program untuk mikrokontroler (biasanya menggunakan Arduino IDE) untuk membaca sensor, mengolah data, dan mengirimkannya ke platform IoT via WiFi.

Fase 3: *Testing & Evaluation* (Pengujian dan Evaluasi)

Tujuan: Menguji purwarupa dalam kondisi nyata dan mengevaluasi keefektifannya.

1. *Product Field Testing* (Pengujian Lapangan Terbatas)

Simulasi Kebakaran: Purwarupa diuji dengan sumber asap (kertas dibakar), panas (solder), atau api langsung di lingkungan yang terkendali. Pengujian Performa: Dilakukan pengujian berulang untuk mengukur: Akurasi Sensor: Seberapa baik sensor mendeteksi pemicu kebakaran. Kecepatan Respon: Delay dari deteksi hingga notifikasi dikirim. Dan Kestabilan Jaringan: Keandalan koneksi WiFi dan pengiriman data.

2. *Revision of Main Product* (Revisi Produk Utama)

Berdasarkan hasil pengujian lapangan, dilakukan penyempurnaan pada purwarupa.

3. *Final Product Revision* (Revisi Produk Akhir)

Melakukan penyempurnaan akhir berdasarkan semua hasil pengujian. Pada tahap ini, desain casing atau enclosure dirancang untuk melindungi komponen.

C. HASIL DAN PEMBAHASAN

Penelitian diawali dengan melakukan identifikasi terhadap lokasi (ruangan) yang memiliki tingkat kerawanan tinggi terhadap bahaya kebakaran. Hasil identiifkasi dapat dilihat pada gambar 1.

Gambar 1. Identifikasi ruangan rawan risiko bahaya kebakaran

Berdasarkan denah lokasi SMK YP IPPI Petojo Jakarta, area yang diberi warna hijau menunjukkan lokasi-lokasi yang memiliki risiko tinggi terhadap bencana kebakaran karena berkaitan dengan pengelolaan sekolah, laboratorium, serta area kerja staf dan kepala sekolah. Pada lokasi yang berwarna hijau tersebut purwarupa sistem peringatan dini kebakaran yang dibuat akan dipasang.

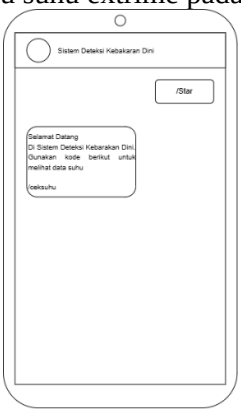
Rancangan cara kerja sistem yang dibuat dapat dilihat pada gambar 2 berikut:

Gambar 2. Flowchart Purwarupa Sistem Peringatan Dini Kebakaran

Proses dimulai dengan pembacaan data dari dua sensor, yaitu sensor suhu dan kelembaban *DHT22* serta sensor api. Setelah data dibaca, sistem akan mengevaluasi apakah terdapat kondisi bahaya, yaitu terdeteksinya api atau suhu tinggi yang melebihi batas aman. Jika tidak ditemukan indikasi tersebut, sistem akan kembali melakukan pembacaan sensor secara berkala. Namun, jika terdapat api atau suhu tinggi, maka sistem secara otomatis akan mengaktifkan *buzzer* sebagai peringatan lokal dan mengirimkan notifikasi ke bot Telegram agar pengguna dapat segera mengetahui adanya potensi kebakaran. Setelah notifikasi dikirim, proses dianggap selesai. *Flowchart* ini menggambarkan efisiensi dan kecepatan sistem dalam mendeteksi serta merespons potensi kebakaran tanpa perlu keterlibatan manual dari pengguna.

Perancangan Perangkat Lunak

Perangkat lunak yang pada sistem ini adalah aplikasi Telegram yang nantinya digunakan untuk monitoring jika terdeteksi adanya api atau suhu extrime pada sekolah.



Gambar 3. Tampilan Awal Bot Telegram

Pada gambar 3 menunjukkan tampilan awal perancangan antarmuka perangkat lunak pada bot Telegram. Saat pengguna pertama kali memulai interaksi dengan bot, mereka akan melihat pesan sambutan bertuliskan: "Selamat Datang di Sistem Deteksi Kebakaran Dini. Gunakan kode berikut untuk melihat data suhu /ceksuhu"

Gambar 4. Tampilan Output Bot Telegram

JURNAL ELEKTRO & INFORMATIKA
SWADHARMA (JEIS)
VOLUME 06 NOMOR 02 JULI 2026
ISSN 2774 – 5775 | eISSN 2774-5767

Halaman ini juga menampilkan tombol perintah awal /start, yang berfungsi untuk memulai layanan bot. Setelah pengguna menekan tombol tersebut, bot akan mengirimkan pesan informasi berupa petunjuk penggunaan, seperti perintah /ceksuhu untuk memeriksa data suhu dari sistem.

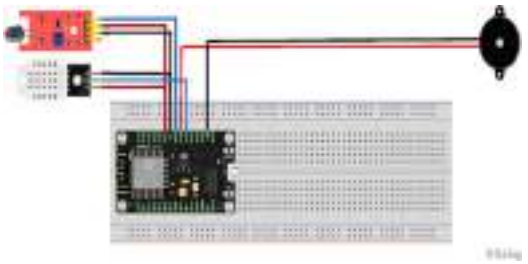
Pada gambar 4 merupakan tampilan output dari bot Telegram setelah pengguna mengirimkan perintah /ceksuhu. Output yang ditampilkan mencerminkan data yang diperoleh dari sistem deteksi kebakaran dini, dengan beberapa informasi penting sebagai berikut:

1. Data Suhu dan Kelembaban
   Bot memberikan respon berupa hasil pembacaan sensor. Informasi ini memberikan gambaran kondisi lingkungan secara real-time.
2. Notifikasi Deteksi Api
   Muncul pesan "Api Terdeteksi!!!!" yang menandakan bahwa sensor api mendeteksi adanya keberadaan api di lingkungan.
3. Peringatan Suhu Ekstrem
   Disertakan pula pesan "Suhu extrimee!!!!" sebagai tanda bahwa suhu telah melewati ambang batas normal dan berpotensi berbahaya.

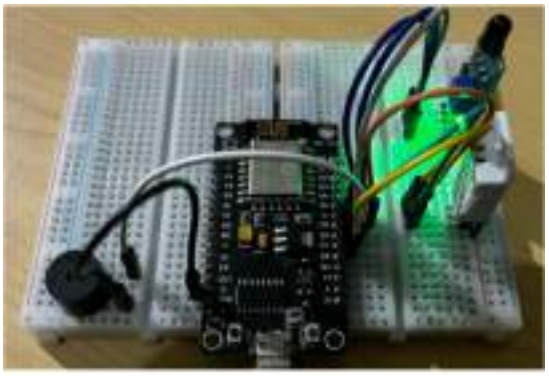
Tampilan ini menunjukkan bahwa bot Telegram berfungsi secara otomatis dan responsif dalam menyampaikan peringatan dini kepada pengguna berdasarkan data dari sensor, sehingga mendukung pengambilan tindakan cepat untuk mencegah kebakaran.

Perancangan Perangkat Keras

Rangkaian ini merupakan bentuk dari keseluruhan perangkat yang digunakan pada Sistem Deteksi Kebakaran Dini yang terdiri dari Beardboard, NodeMcu/*ESP8266*, Sensor *DHT22*, *Flame Sensor*, *Buzzer*, dan Kabel Jumper.



Gambar 5. Perancangan perangkat keras



Gambar 6. Hasil Perakitan

Pada rangkaian ini ditampilkan komponen-komponen yang digunakan dalam perancangan prototipe sistem deteksi kebakaran dini. Rangkaian tersebut terdiri atas mikrokontroler *ESP8266* yang berfungsi sebagai pusat pemrosesan data dengan mengonversi sinyal input elektronik menjadi sinyal output elektronik. Sensor api (flame sensor) digunakan untuk mendeteksi keberadaan api, sedangkan sensor *DHT22* berperan dalam memantau suhu dan kelembaban udara. *Buzzer* difungsikan sebagai alarm yang akan diaktifkan apabila flame sensor mendeteksi adanya api atau ketika sensor *DHT22* mengirimkan data suhu ekstrem. Seluruh komponen tersebut dirangkai menggunakan breadboard dan kabel jumper sebagai media penghubung.

Hasil Pengujian Sistem

Pengujian sistem bertujuan untuk mengevaluasi kinerja logika yang ada pada program. Pengujian ini dilakukan secara langsung pada perangkat yang beroperasi untuk memastikan bahwa semua fitur utama sistem berfungsi dengan optimal.

Tabel 1. Hasil Pengujian Sistem

No	Pengujian	Test Case	Hasil Yang Diharapkan	Hasill Pengujian	Ket
1	Test *Flame Sensor*	Memberikan sumber api (Menyalakan korek api)	Flame Sensor mendeteksi adanya api, kemudian mengirim notifikasi kepada bot telegram dan membunyikan *Buzzer*	Sesuai harapan	Valid
2	Test Sensor *DHT22*	Membuat suhu menjadi > 40c	Jika suhu >40c maka akan mengirim notifikasi kepada bot telegram dan membunyikan *buzzer*	Sesuai harapan	Valid

Gambar 7. Simulasi Pengujian Alat

Setelah dilakukan pengujian pada sensor dan komponen lainya maka didapat hasil bahwa ketika *Flame Sensor* (sensor api) mendeteksi ada nya api atau ketika Sensor *DHT22* mendeteksi adanya suhu ekstrim maka *buzzer* akan berbunyi dan *ESP8266* akan mengirimkan notifikasi kepada Bot Telegram.

D. PENUTUP

Dari hasil perancangan dan pengujian purwarupa, maka didapatkan bahwa Flame Sensor dapat bekerja dengan baik ketika terdeteksi adanya api yang terpantau oleh sensor. Sensor DHT22 dapat bekerja dengan baik ketika terdeteksi suhu extrime yang terpantau oleh sensor. Pengguna dapat memantau data suhu dan kelembaban melalui sistem yang dapat di akses dan dimonitoring melalui aplikasi Telegram. Perancangan Internet Of Things untuk sistem peringatan dini kebakaran dengan mikrokontroler ESP8266 akan menerima input data dari flame sensor dan sensor DHT22 lalu mengirimnya ke aplikasi Telegram untuk pengiriman notifikasi kepada pihak-pihak terkait.

E. DAFTAR PUSTAKA

Abrar, A. R., Kaharmen, H. M., & Hakim, I. N. (2020). Prototipe Alat Pendeteksi Kebakaran Berbasis Internet Of Things Dengan Aktifasi *Flame* Sensor Menggunakan Arduino. *Jurnal Keselamatan Transportasi Jalan*, *7*(2), 83–93. https://doi.org/10.46447/ktj.v7i2.156

Ahmed, F. M., & Saad, A. M. (2024). Fire Disaster Preparedness and Life Safety Program for Preparatory School Students Mentored by Community Health Nursing Students. *Tanta Scientific Nursing Journal*, *32*(1), 236–256. https://doi.org/10.21608/tsnj.2024.341025

Asronika, N., Akyun, A., Mahmudah, N. L., Wahyuningtyas, R., & Pramudhita, A. N. (2024). Implementasi NodeMCU dan Blynk Dalam Sistem Pendeteksi Kebakaran Berbasis IoT. *SHIFT : Journal Software, Hardware and Information Technology*, *4*(1), 11–18. https://doi.org/10.24252/shift.v4i1.100

Elhanashi, A., Essahraui, S., Dini, P., & Saponara, S. (2025). Early Fire and Smoke Detection Using Deep Learning:

A Comprehensive Review of Models, Datasets, and Challenges. *Applied Sciences*, *15*(18), 10255. https://doi.org/10.3390/app151810255

Khan, F., Xu, Z., Sun, J., Khan, F. M., Ahmed, A., & Zhao, Y. (2022). Recent Advances in Sensors for Fire Detection. *Sensors*, *22*(9), 3310. https://doi.org/10.3390/s22093310

Li, X., Vázquez-López, A., Sáez, J. S. del R., & Wang, D.-Y. (2022). Recent Advances on Early-Stage Fire-Warning Systems: Mechanism, Performance, and Perspective. *Nano-Micro Letters*, *14*, 197. https://doi.org/10.1007/s40820-022-00938-x

Lubis, R. T., & Susilawati. (2024). Sistem Kunci Pintu Berbasis PIN Menggunakan Arduino dan Keypad. *JATI : Jurnal Mahasiswa Teknik Informatika*, *8*(3), 3830–3835. https://doi.org/10.36040/jati.v8i3.9820

Mulyono, J., Djuniadi, & Apriaskar, E. (2021). Simulasi Alarm Kebakaran Menggunakan Sensor Mq-2, Falme Sensor Berbasis Mikrokontroler Arduino. *ELKOM : Jurnal Elektronika Dan Komputer*, *14*(1), 16–25. https://doi.org/10.51903/elkom.v14i1.305

Nazuarsyah, N., Muzakir, U., Mukhroji, M., Ginting, R., & Munadi, R. (2023). Sistem Identifikasi Menggunakan Rfid Dan Sensor Infrared Berbasis Iot Terhadap Pengembangan Kampus Pintar. *Cyberspace: Jurnal Pendidikan Teknologi Informasi*, *7*(2), 109. https://doi.org/10.22373/cj.v7i2.21632

Nizam, M., Yuana, H., & Wulansari, Z. (2022). Mikrokontroller ESP 32 Sebagai Alat Monitoring Pintu Berbasis Web. *JATI : Jurnal Mahasiswa Teknik Informatika*, *6*(2), 767–772. https://doi.org/10.36040/jati.v6i2.5713

Nugraha, D. A., & Satria, B. (2022). Prototipe Alat Pendeteksi Kebakaran Menggunakan Sensor *Flame* dan MQ-2 Berbasis Arduino Uno. *The Indonesian Journal of Computer Science*, *11*(3), 936–944. https://doi.org/10.33022/ijcs.v11i3.3102

Pasmah, R., Lubis, A. J., & Usman, A. (2021). Prototipe Sistem Keamanan Ruangan Menggunakan Finger Print dan Keypad Matrix dengan One Time Pad. *Explorer: Journal of Computer Science and Information Technology*, *1*(2), 53–62. https://doi.org/10.47065/explorer.v1i2.89

Salindeho, G. G., & Wellem, T. (2023). Perancangan dan Implementasi Sistem Pendeteksi dan Peringatan Kebakaran Berbasis IoT Menggunakan NodeMCU ESP8266 dan Sensor Api. *IT-Explore: Jurnal Penerapan Teknologi Informasi Dan Komunikasi*, *2*(3), 179–191. https://doi.org/10.24246/itexplore.v2i03.2023.pp179-191

Swetha, P. B., Vennakandla, C. S. R., Sonti, S. R., Periasamy, P. S., Naresh, A., & Gali, S. (2026). Real-Time Smart Fire Detection and Prevention System Using IoT Sensors and ML Algorithms. *International Conference on Visual Analytics and Data Visualization (ICVADV)*, 1330–1336. https://doi.org/10.1109/ICVADV67766.2026.11470528

Alamat Redaksi
Kampus 1 Institut Teknologi dan Bisnis Swadharma
Jl. Malaka No.3, Tambora, Jakarta Barat
emal : jurnal.jeis@swadharma.ac.id