

## PERANCANGAN INFRASTRUKTUR REDUNDANSI DENGAN MENERAPKAN *HIGH AVAILABILITY* UNTUK MENJAMIN KEANDALAN OPERASIONAL BISNIS

Sapriila Wijaya<sup>1)</sup>, Harjono Padmono Putro<sup>2)</sup>

<sup>1</sup>Prodi Teknik Informatika, Sekolah Tinggi Teknologi Informasi NIIT I-Tech

<sup>2</sup>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

*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 redundansi, *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 *clustering* 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 *downTime*, 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.

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             |
|-----------------------|-------------------------|
| <i>Shared storage</i> | <i>NetApp ASA R2</i>    |
| 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

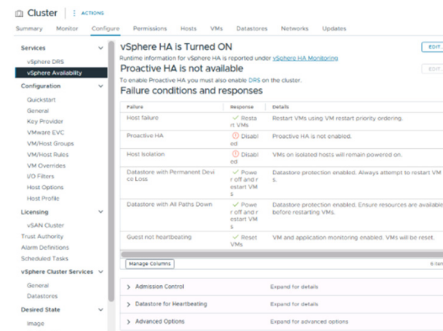
1 sebagai alur penerapan infrastruktur *High Availability*.



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 *vSwitch* 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 direstart 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.



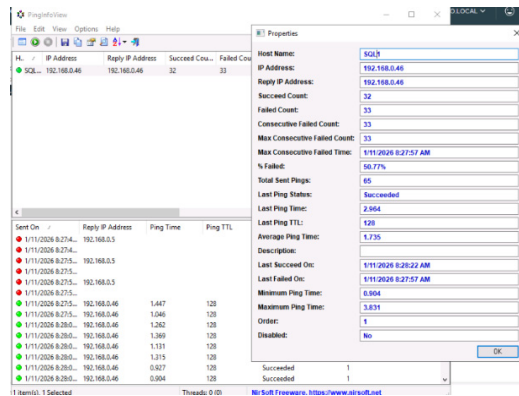
Gambar 2. *vSphere* HA is Turned ON

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 virtualisasi   | 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.



**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                                                                               | DownTime                 | Evaluasi                                                             | Memenuhi SLA 99,9% |
|-------------------------|-------------------------------------------------------------------------------------------|--------------------------|----------------------------------------------------------------------|--------------------|
| <i>Failover Server</i>  | VM sql1 berhasil dipulihkan oleh vSphere HA ( <i>restart</i> /pindah ke <i>host</i> lain) | 33 detik                 | HA berjalan sesuai fungsi pemulihan layanan saat <i>host</i> gagal   | Memenuhi           |
| <i>Failover Network</i> | Saat 1 uplink (vmnic) dimatikan, layanan VM tetap dapat diakses                           | 0 detik (tidak teramati) | Redundansi uplink efektif, gangguan 1 jalur tidak memutuskan layanan | Memenuhi           |
| Failback                | <i>Host</i> kembali Connected, VM sql1 tetap Powered On/Normal                            | 0 detik (tidak teramati) | <i>Cluster</i> kembali stabil setelah <i>host</i> 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                                                         |
|---------------------------|----------|---------|--------------------------------------------------------------------|
| <i>Failover Host ESXi</i> | 33 detik | 0 detik | <i>Virtual machine</i> berhasil <i>directstart</i> secara otomatis |
| <i>Failover Network</i>   | 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>