

OPTIMALISASI BASIS DATA SISTEM KEUANGAN UMKM PIZZAUPPS UNTUK MENGATASI REDUDANSI DAN ANOMALI DATA

**Muhammad Faiz¹⁾, Karunia Raharjo²⁾, Muhammad Agus³⁾, Muhammad Zidni Nur⁴⁾,
Dicky Anggriawan Nugroho⁵⁾, Imam Prayogo Pujiono⁶⁾**

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

Correspondence author: M.Faiz, F4078755@gmail.com, Kota Pekalongan, Indonesia

Abstract

Digital financial systems are crucial for MSMEs to improve the efficiency and accuracy of record-keeping. Unfortunately, many MSMEs, including PizzaUpps, still rely on poorly structured flat-file record-keeping systems. This condition causes serious problems, such as duplication of customer and product data, and inconsistencies in transaction information, due to the lack of integrity controls in their databases. To address this, this study aims to redesign and optimize the database architecture of PizzaUpps' financial system. Using the Waterfall model, the main methodology is database normalization, which transforms the data structure from unnormalized form (UNF) to Third Normal Form (3NF) and Boyce-Codd Normal Form (BCNF). The results show that the normalization process successfully produces a new, structured relational database design. This design consists of interconnected main tables, including Customer, Product, Transaction, and Transaction_Detail, and is supported by an Entity Relationship Diagram (ERD) and a validated physical relationship schema. Foreign keys and referential integrity constraints ensure data integrity across tables. The proposed database design proved effective in eliminating data redundancy and anomalies, significantly improving the accuracy, consistency, and integrity of PizzaUpps' financial data. This design is highly recommended as a basis for developing a more comprehensive digital financial or accounting information system in the future.

Keywords: *database, financial systems, msme, database normalization, waterfall*

Abstrak

Peran sistem keuangan digital sangat krusial bagi UMKM untuk meningkatkan efisiensi dan keakuratan pencatatan. Sayangnya, banyak UMKM, termasuk PizzaUpps, masih mengandalkan sistem pencatatan flat-file yang tidak terstruktur dengan baik. Kondisi ini menyebabkan masalah serius seperti duplikasi data pelanggan dan produk, serta inkonsistensi informasi transaksi, karena tidak adanya kontrol integritas pada basis data mereka. Untuk mengatasi hal ini, penelitian ini bertujuan untuk merancang ulang dan optimalisasi arsitektur basis data sistem keuangan PizzaUpps. Menggunakan model Waterfall, metodologi utamanya adalah normalisasi basis data, yaitu mentransformasikan struktur data dari bentuk tidak ternormalisasi (UNF) menjadi Third Normal Form (3NF) dan Boyce-Codd Normal Form (BCNF). Hasil penelitian menunjukkan bahwa proses normalisasi berhasil menciptakan desain basis data relasional yang baru dan terstruktur. Desain

ini terdiri dari tabel-tabel utama yang saling terhubung, seperti Pelanggan, Produk, Transaksi, dan Detail_Transaksi, dan dilengkapi dengan Entity Relationship Diagram (ERD) serta skema relasi fisik yang tervalidasi. Integritas data antar tabel dijamin melalui penggunaan foreign key dan batasan integritas referensial. Desain basis data yang diusulkan terbukti efektif dalam menghilangkan redundansi dan anomali data, secara signifikan meningkatkan akurasi, konsistensi, dan integritas data keuangan PizzaUpps. Desain ini sangat direkomendasikan sebagai dasar untuk pengembangan sistem informasi keuangan atau akuntansi digital yang lebih lengkap di masa depan.

Kata Kunci: basis data, sistem keuangan, umkm, normalisasi, *waterfall*

A. PENDAHULUAN

Usaha Mikro, Kecil, dan Menengah (UMKM) memainkan peranan kunci dalam memperkuat perekonomian nasional Indonesia dengan kontribusi lebih dari 60 persen terhadap Produk Domestik Bruto (PDB) serta penyerapan sekitar 97 persen tenaga kerja nasional (Pane et al., 2025). Fenomena globalisasi dan percepatan transformasi digital, terutama dalam era Revolusi Industri 4.0, mendorong sektor UMKM untuk beradaptasi secara progresif dengan teknologi digital, sehingga dapat mempertahankan daya saing dalam lingkungan bisnis yang (Zaini, 2024). Digitalisasi, khususnya pada sistem manajemen keuangan, semakin diperlukan karena pencatatan keuangan berbasis elektronik memberi peluang bagi UMKM untuk mengambil keputusan bisnis secara cepat dan tepat (Chaidir et al., 2025). Namun, proses adopsi sistem digital tersebut kerap tidak dibarengi dengan penyusunan struktur basis data yang memadai, terutama di kalangan pelaku UMKM yang baru mengimplementasikan teknologi informasi (Asmarawati, 2025).

Pada praktiknya, transformasi digital dalam UMKM biasanya diawali dengan penggunaan perangkat lunak sederhana seperti *spreadsheet* atau aplikasi keuangan digital yang belum didesain optimal dari aspek struktur data. Hal ini menyebabkan timbulnya persoalan penting berupa

redundansi maupun inkonsistensi data (Wati et al., 2025). Redundansi merupakan keadaan di mana data yang sama tersimpan berkali-kali pada lokasi atau tabel yang berbeda di dalam basis data, misalnya data pelanggan yang dicantumkan berulang pada setiap transaksi penjualan (Connolly & Begg, 2019). Kondisi demikian merupakan sumber utama anomali data, yaitu ketidak konsistenan atau kejanggalan data dalam sistem basis data yang dapat dikategorikan menjadi tiga, yaitu anomali penyisipan (*insertion anomaly*), anomali pembaruan (*update anomaly*), dan anomali penghapusan (*deletion anomaly*). Ketiga bentuk anomali ini berdampak pada kualitas dan akurasi data keuangan, sehingga menimbulkan risiko kerugian pada pelaku usaha (Romney & Steinbart, 2021).

Permasalahan serupa ditemukan dalam studi kasus di UMKM PizzaUpps. Sistem pencatatan keuangan di entitas tersebut masih mengandalkan satu file *spreadsheet* terpusat (*flat file*) untuk menyimpan seluruh catatan transaksi, data pelanggan, hingga pembelian. Akibatnya, terjadi pengulangan data identitas pelanggan pada tiap baris transaksi, sehingga pembaruan informasi seperti perubahan nomor telepon harus dilakukan secara manual di seluruh entri terkait. Keadaan ini rentan memicu inkonsistensi data ketika ada baris yang terlewat diperbarui, memperbesar potensi kesalahan pada laporan keuangan, serta

memengaruhi kualitas pengambilan keputusan bisnis (Asmarawati, 2025).

Redundansi dan anomali data pada sistem dapat mengancam integritas, reliabilitas, dan validitas data keuangan, serta berpotensi menghasilkan analisis usaha yang menyesatkan, seperti perhitungan harga pokok penjualan atau evaluasi loyalitas konsumen (Kieso et al., 2019). Dengan demikian, optimalisasi desain dan struktur basis data menjadi kebutuhan mendesak agar sistem keuangan UMKM PizzaUpps memperoleh data yang akurat, konsisten, dan kredibel untuk keperluan perencanaan maupun evaluasi bisnis.

Salah satu pendekatan efektif yang diusulkan adalah penerapan teknik normalisasi basis data. Normalisasi merupakan metode sistematis dalam perancangan basis data relasional, yang memecah atribut menjadi tabel-tabel yang lebih kecil, efisien, dan bebas dari duplikasi (Renanti et al., 2025). Langkah ini bertujuan mengeliminasi redundansi data dan memastikan semua relasi antar data terekam secara logis agar terhindar dari anomali. Penelitian ini secara khusus bertujuan merancang arsitektur basis data sistem keuangan UMKM PizzaUpps melalui tahapan normalisasi hingga mencapai *Third Normal Form* (3NF) atau *Boyce-Codd Normal Form* (BCNF), demi meminimalkan masalah redundansi dan anomali serta meningkatkan efektivitas sistem informasi keuangan (Rafianto & Voutama, 2025).

B. METODE PENELITIAN

Penelitian ini didasarkan pada metodologi pengembangan sistem Model Air Terjun (Waterfall). Pemilihan model ini didasarkan pada sifatnya yang sekuensial dan terorganisir, menjadikannya sangat cocok untuk proyek pengoptimalan basis data yang memiliki ruang lingkup dan target yang pasti (Pressman & Maxim, 2020). Penelitian ini secara spesifik

berfokus pada rekayasa ulang (*re-engineering*) arsitektur basis data. Pekerjaan ini meliputi analisis menyeluruh terhadap sistem yang ada, hingga akhirnya mencapai validasi desain basis data yang optimal.

Seluruh langkah penelitian akan diuraikan sesuai dengan fase-fase standar dari model *Waterfall*.

1. Tahap Analisis Kebutuhan (Requirements Analysis)

Tahap ini menjadi landasan utama penelitian, di mana fokusnya adalah memahami secara mendalam sistem pencatatan keuangan yang saat ini berjalan (As-Is) di PizzaUpps. Tujuan krusial dari tahap ini adalah mengungkap kelemahan struktural, mencatat semua masalah yang muncul, serta menentukan secara pasti kebutuhan fungsional yang harus dipenuhi oleh sistem basis data yang akan dirancang.

Aktivitas sentral pada fase ini adalah pengumpulan dan validasi data. Teknik yang kami gunakan untuk mengumpulkan informasi adalah sebagai berikut

- a. Wawancara (*Interview*): Kami melakukan wawancara semi-terstruktur dengan pemilik dan staf administrasi PizzaUpps. Tujuannya adalah untuk mendapatkan pemahaman komprehensif mengenai alur proses bisnis keuangan (mulai dari pemesanan hingga pembuatan laporan), mengidentifikasi kendala atau *pain points* yang dihadapi pengguna, dan menetapkan kebutuhan informasi dan pelaporan fungsional untuk sistem usulan.
- b. Observasi (*Observation*): Dilakukan pengamatan langsung terhadap kegiatan kerja harian staf untuk memverifikasi data aktual dari sistem yang sedang berjalan. Melalui pengamatan pada *spreadsheet* dan catatan internal, kami dapat mengidentifikasi secara empiris titik-titik di mana terjadi redundansi data (misalnya, entri data pelanggan yang berulang) dan anomali data (masalah saat melakukan penyisipan, pembaruan,

atau penghapusan data) (Creswell & Creswell, 2018)

- c. Studi Literatur (*Literature Review*): Literatur yang digunakan termasuk buku teks, jurnal, dan artikel ilmiah, untuk membangun fondasi teoretis yang kuat. Fokus utama kajian literatur adalah pada Teori Perancangan *Database* Relasional, Analisis Ketergantungan Fungsional (*Functional Dependency*), dan metode Normalisasi (1NF, 2NF, 3NF, BCNF), yang dijadikan sebagai alat analisis utama untuk mengatasi masalah redundansi dan anomali data (Yusril et al., 2025).

Hasil dari tahap ini adalah dokumentasi formal mengenai ketergantungan Fungsional (FD) pada sistem lama, daftar anomaly dan redudansi yang teridentifikasi serta seperangkat kebutuhan fungsional untuk perancangan basis data yang baru.

2. Tahap Perancangan (*Design*)

Setelah semua temuan dari tahap analisis terkumpul, fase perancangan (*Desain To-Be*) dimulai. Fase ini didedikasikan untuk merumuskan solusi arsitektur basis data baru yang dirancang khusus untuk menghilangkan semua masalah yang telah diidentifikasi, terutama melalui penerapan teknik normalisasi. Proses ini dibagi menjadi tiga sub-tahap standar (Connolly & Begg, 2019):

- a. Perancangan Konseptual: Langkah awal ini melibatkan identifikasi entitas-entitas kunci (seperti Pelanggan, Produk, dan Penjualan) serta hubungan (relasi) di antara mereka. Hasilnya kemudian divisualisasikan menggunakan Diagram Hubungan Entitas (*Entity Relationship Diagram - ERD*).
- b. Perancangan Logis: Ini adalah tahapan di mana *ERD* ditransformasikan menjadi skema relasi atau struktur tabel. Sub-tahap ini merupakan inti dari penelitian, yaitu normalisasi basis data secara bertahap. Proses ini dilakukan mulai dari bentuk yang belum

dinormalisasi *Unnormalized Form* – *UNF* hingga mencapai Bentuk Normal Ketiga *Third Normal Form - 3NF* atau Bentuk *Normal Boyce-Codd (BCNF)* (Elmasri & Navathe, 2011). Tujuannya yang paling utama adalah menghilangkan ketergantungan parsial dan transitif, sehingga masalah anomali data dapat teratasi sepenuhnya.

- c. Perancangan Fisik: Pada langkah terakhir ini, skema relasi yang sudah logis diterjemahkan ke dalam spesifikasi teknis dan fisik. Ini mencakup penetapan tipe data yang paling efisien, penentuan kunci utama (*primary key*) dan kunci asing (*foreign key*), serta pembuatan indeks untuk setiap atribut yang ada di dalam tabel.

3. Tahap Implementasi(*implementation*)

Pada tahap implementasi, desain fisik yang sudah rampung diwujudkan menjadi basis data yang siap berfungsi. Perlu dicatat, dalam lingkup penelitian ini, implementasi tidak mencakup pembangunan aplikasi secara utuh, melainkan difokuskan sepenuhnya pada penciptaan struktur basis data itu sendiri.

Bentuk konkret dari implementasi ini adalah penulisan skrip *Data Definition Language* (DDL) menggunakan SQL (*Structured Query Language*). Skrip inilah yang bertugas untuk menciptakan semua tabel, mendefinisikan relasi, dan menetapkan batasan *constraint* yang sesuai dengan skema relasi logis yang telah mencapai bentuk normal *3NF/BCNF*.

4. Tahap Pengujian dan Validasi (*Testing and Validation*)

Tahap terakhir ini berfungsi sebagai kontrol kualitas yang krusial. Tujuannya adalah memastikan bahwa arsitektur basis data yang baru *To-Be* benar-benar berhasil menyelesaikan semua masalah yang ada pada sistem lama *As-Is* (Pressman & Maxim, 2020).

Pengujian ini dilakukan dengan menjalankan berbagai skenario untuk memverifikasi hal-hal berikut:

- a. Redundansi Data Telah Diatasi: memastikan tidak ada lagi pengulangan data yang tidak perlu. Contohnya, data pelanggan kini hanya tersimpan satu kali di tabel Pelanggan dan tidak diulang pada setiap catatan transaksi.
- b. Anomali Data Telah Teratasi: Dilakukan pengujian skenario untuk:
 - 1) Anomali Penyisipan (*Insertion Anomaly*): Memastikan data baru (misal: pelanggan baru) dapat dimasukkan tanpa harus melakukan transaksi.
 - 2) Anomali Pembaruan (*Update Anomaly*): Memastikan bahwa jika ada pembaruan data (misalnya, penggantian nomor telepon pelanggan), perubahan tersebut cukup dilakukan di satu tempat dan akan secara otomatis konsisten di seluruh sistem (Connolly & Begg, 2019).
 - 3) Anomali Penghapusan (*Deletion Anomaly*): Memastikan penghapusan satu data transaksi tidak menyebabkan hilangnya data master (misal: data pelanggan).
- c. Kebutuhan Fungsional Terpenuhi: memvalidasi bahwa semua data yang diperlukan oleh pemilik untuk tujuan pelaporan keuangan dan analisis bisnis dapat diakses dengan mudah dari struktur basis data yang baru.

C. HASIL DAN PEMBAHASAN

Tahap Analisis kebutuhan (*Requirements Analysis*)

Tahap analisis kebutuhan merupakan fondasi dari proses optimalisasi sistem keuangan umkm pizzaupps, yang bertujuan memahami kondisi sistem berjalan (*as-is*), mengidentifikasi permasalahan struktural, dan mendefinisikan kebutuhan fungsional sistem basis data baru. proses ini

dilaksanakan melalui wawancara dengan pemilik usaha dan staf administrasi, observasi langsung terhadap kegiatan operasional, serta analisis dokumen pencatatan keuangan yang digunakan saat ini.

Hasil Identifikasi Sistem Berjalan (*As-Is*)

Wawancara semi-terstruktur dilakukan dengan pemilik usaha (*owner*) dan staf administrasi PizzaUpps. Berdasarkan hasil wawancara, diperoleh informasi bahwa pencatatan transaksi keuangan masih dilakukan menggunakan satu berkas *spreadsheet* yang berfungsi ganda untuk transaksi dan pelanggan.

Pemilik usaha mengakui bahwa sistem tersebut “cukup praktis untuk penggunaan awal,” namun mulai menimbulkan kesulitan saat data transaksi meningkat. Beberapa pernyataan penting hasil wawancara adalah:

1. Proses pembaruan data pelanggan dilakukan manual di setiap baris transaksi.
2. Tidak ada fitur pencarian pelanggan atau laporan otomatis.
3. Kesalahan input sering terjadi karena tidak ada validasi data.

Observasi langsung terhadap aktivitas administrasi keuangan menunjukkan pola kerja yang tidak efisien. Semua catatan keuangan disimpan dalam satu *flat-file* (lembar *spreadsheet* tunggal), sehingga pengulangan data terjadi di hampir semua transaksi. Gambar berikut menunjukkan representasi struktur data *flat-file* yang digunakan dalam sistem keuangan PizzaUpps.

No. Tr ansaksi	Tgl. Tr ansaksi	ID_Pel anggan	Nama_P elanggan	No. Tel epon	ID_P roduk	Nama Produk	Harga Produk	Jumla h Beli
TR001	01/11/2 025	P001	Budi Santoso	081234 567890	PR00 1, PR00 3	Pizza Meat, Kentang Goreng	50000, 15000	1, 2
TR002	01/11/2 025	P002	Citra Lestari	085678 901234	PR00 2	Pizza Keju	45000	1
TR003	02/11/2 025	P001	Budi Santoso	081234 567890	PR00 1, PR00 2	Pizza Meat, Pizza Keju	50000, 45000	1, 1
TR004	03/11/2 025	P003	Andi Rahman	082345 678901	PR00 4, PR00 5, PR00 6	Spaghet ti, Salad, Air Mineral	30000, 20000, 5000	1, 1, 3

Gambar 1. Representasi Struktur Data *Flat-File*

Struktur di atas menunjukkan bahwa setiap transaksi dan produk dicatat berulang kali, termasuk informasi pelanggan dan produk yang sama pada setiap baris transaksi. Hal ini mengindikasikan adanya desain basis data yang belum terstruktur secara relasional, di mana setiap atribut tidak dikelompokkan berdasarkan entitas fungsional (pelanggan, produk, transaksi).

Kajian literatur dilakukan terhadap beberapa sumber pustaka tentang perancangan basis data relasional, konsep normalisasi (1NF sampai 3NF/BCNF), serta keteergantungan fungsional (*Functional Depedency*). Refrensi utama mencakup panduan dari (Elmasri & Navathe, 2011) tentang pentingnya normalisasi untuk menghilangkan redundansi, dan penelitian (Yusril et al., 2025) mengenai penerapan model *Database Life Cycle* untuk sistem pendidikan. Hasil kajian literatur ini memperkuat pemahaman bahwa sistem *flat-file* PizzaUpps masih berada di tahap awal bentuk *Unnomalized Form* (UNF) dan perlu di normalisasi hingga minimal *Third Normal Form* (3NF) untuk menjamin integritas data.

Analisis Permasalahan

Dari hasil observasi dan analisis struktur data pada Gambar 1, ditemukan dua masalah utama, yakni redundansi data dan anomali data (*insertion, update, dan delete*).

1. Redudansi Data

Terjadi pengulangan data pelanggan dan produk di setiap transaksi. Misalnya,

data pelanggan Budi Santoso (P001) dan nomor teleponnya muncul berulang di setiap baris transaksi yang dilakukannya. Hal ini menyebabkan pemborosan ruang penyimpanan dan potensi inkonsistensi data.

2. Anomali Data

Redudansi tersebut menimbulkan tiga bentuk anomali:

a. Anomali Penyisipan (*Insertion Anomaly*)

Sistem tidak memungkinkan penambahan pelanggan baru tanpa transaksi karena N0 Transaksi digunakan sebagai kunci utama (*primary key*).

b. Anomali Pembaruan (*Update Anomaly*)

Perubahan informasi pelanggan harus dilakukan di seluruh baris terkait. Contohnya, jika Budi Santoso mengganti nomor telepon, staff harus memperbarui setiap transaksi yang memuat data pelanggan tersebut.

c. Anomali Penghapusan (*Delete Anomaly*)

Penghapusan satu transaksi dapat menyebabkan hilangnya data pelanggan atau produk yang masih diperlukan sebagai referensi.

Masalah ini memperlihatkan bahwa struktur *flat-file* tidak memenuhi prinsip dasar efisiensi dan integritas data dalam sistem basis data relasional (Elmasri & Navathe, 2011; Yusril et al., 2025).

Tahap Perancangan (*Design*)

Tahap perancangan (*Design Phase*) merupakan langkah lanjutan dari analisis kebutuhan, yang bertujuan untuk mengubah hasil temuan pada sistem berjalan (*As-Is*) menjadi rancangan sistem basis data baru (*To-Be*). Dalam penelitian ini, proses perancangan dilakukan secara bertahap melalui tiga pendekatan sistematis, yaitu Perancangan Konseptual, Perancangan Logis, dan Perancangan Fisik, sebagaimana prinsip *Waterfall* pada tahap *System Design*

(Pressman & Maxim, 2020; Yusril et al., 2025).

1. Perancangan Konseptual (*Entity Relationship Diagram – ERD*)

Tahap perancangan konseptual bertujuan untuk memodelkan struktur logis sistem keuangan UMKM PizzaUpps dalam bentuk Entity Relationship Diagram (ERD). Model ini merepresentasikan hubungan antar entitas utama dalam sistem serta atribut-atribut yang membentuknya.

Perancangan konseptual ini didasarkan pada hasil analisis kebutuhan dan ketergantungan fungsional yang telah dijabarkan pada tahap sebelumnya.

a. Identifikasi Entitas dan Atribut

Berdasarkan hasil analisis proses bisnis, terdapat empat entitas utama yang mempresentasikan aliran data dalam sistem keuangan UMKM PizzaUpps, yaitu:

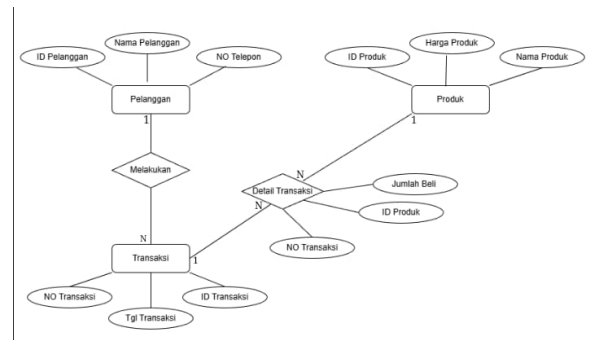
Entitas	Atribut	Keterangan
Pelanggan	ID_Pelanggan(PK), Nama_Pelanggan, No_Telepon	Menyimpan informasi identitas pelanggan yang melakukan transaksi.
Produk	ID_Produk(PK), Nama_Produk, Harga_Produk	Menyimpan data produk yang dijual oleh PizzaUpps.
Transaksi	No_Transaksi(PK), Tgl_Transaksi, ID_Pelanggan (FK)	Menyimpan data transaksi penjualan yang dilakukan oleh pelanggan.
Detail_Transaksi	No_Transaksi(FK), ID_Produk(FK), Jumlah_Beli	Menyimpan rincian produk yang dibeli pada setiap transaksi. Merupakan entitas hubungan antara Transaksi dan Produk.

Gambar 2. Empat entitas utama aliran data

Penentuan atribut didasarkan pada hasil observasi data aktual di sistem berjalan (*flat-file*) serta wawancara dengan staf administrasi. Entitas Detail Transaksi merupakan entitas asosiatif (*associative entity*) yang muncul karena adanya relasi *many-to-many* antara entitas transaksi dan produk.

b. Kardinalitas dan Relasi Antar Entitas

Relasi anatar entitas dalam ERD ini diuraikan sebagai berikut:



Gambar 3. Entity Relational Diagram

Pelanggan – Transaksi

Kardinalitas: 1:N dimana satu pelanggan dapat melakukan banyak transaksi, tetapi satu transaksi hanya dapat dilakukan oleh satu pelanggan. Dengan relasi *Pelanggan* (*ID_Pelanggan*) ke *Transaksi* (*ID_Pelanggan FK*).

Transaksi – Detail_Transaksi

Kardinalitas: 1:N dimana satu transaksi dapat memiliki lebih dari satu detail pembelian produk. Relasinya *Transaksi* (*No_Transaksi*) ke *Detail_Transaksi* (*No_Transaksi FK*).

Produk – Detail_Transaksi

Kardinalitas 1:N satu produk dapat muncul di banyak detail transaksi. Relasinya *Produk* (*ID_Produk*) ke *Detail_Transaksi* (*ID_Produk FK*).

Dengan demikian, entitas Detail_Transaksi menjadi jembatan penghubung antara Transaksi dan Produk, sehingga hubungan *many-to-many* (M:N) antara keduanya diubah menjadi dua hubungan *one-to-many* (1:N).

c. Deskripsi ERD

Deskripsi tekstual dari *ERD* adalah sebagai berikut:

- 1) Entitas pelanggan terhubung ke entitas Transaksi melalui *ID_Pelanggan*, yang berfungsi sebagai *Foreign Key* (FK) di tabel Transaksi.
- 2) Entitas Transaksi terhubung ke entitas Detail_Transaksi melalui atribut *No_Transaksi* sebagai FK.

- 3) Entitas Produk juga terhubung ke Detail_Transaksi melalui *ID_Produk* sebagai FK.
- 4) Hubungan antar entitas membentuk pola hirarki logis yang efisien, dengan entitas Detail_Transaksi berperan sebagai penghubung antar entitas utama tanpa menimbulkan duplikasi data.

Secara struktural, *ERD* ini menunjukkan bahwa sistem PizzaUpps bersifat relasional murni, di mana setiap entitas hanya menyimpan data yang relevan dengan fungsinya, dan hubungan antar entitas dijaga melalui kunci relasional (primary key–foreign key). Model konseptual ini akan menjadi dasar utama bagi proses perancangan logis dan fisik yang akan diterjemahkan ke dalam skema tabel *SQL* pada tahap selanjutnya.

2. Perancangan Logis (Normalisasi UNF KE 3NF/BCNF)

Tahap perancangan logis bertujuan untuk mentransformasikan model konseptual (*ERD*) menjadi struktur relasional yang efisien dan bebas dari redundansi. Proses ini dilakukan melalui normalisasi teknik sistematis dalam perancangan basis data yang memecah tabel besar menjadi tabel-tabel kecil tanpa kehilangan informasi, dengan memastikan setiap tabel hanya menyimpan data yang relevan terhadap kunci utamanya (Connolly & Begg, 2019). Normalisasi dilakukan secara bertahap mulai dari bentuk *Unnormalized Form* (UNF) hingga *Third Normal Form* (3NF). Berikut penjelasan dan hasil setiap tahapannya.

a. Bentuk Tidak Normal (Unnormalized Form – UNF)

Pada sistem lama (*As-Is*), semua data keuangan PizzaUpps dicatat dalam satu lembar kerja spreadsheet tanpa struktur relasional. Data pelanggan, produk, dan transaksi digabungkan dalam satu tabel yang sama. Contoh struktur data tersebut disajikan pada Gambar 4. berikut.

No_Transaksi	Tgl_Transaksi	ID_Pelanggan	Nama_Pelanggan	No_Telepon	ID_Produk	Nama_Produk	Harga_Produk	Jumlah_Beli
TR001	01/11/2025	P001	Budi Santoso	081234567890	PR001, PR003	Pizza Meat, Kentang Goreng	50000, 15000	1, 2
TR002	01/11/2025	P002	Citra Lestari	085678901234	PR002	Pizza Keju	45000	1
TR003	02/11/2025	P001	Budi Santoso	081234567890	PR001, PR002	Pizza Meat, Pizza Keju	50000, 45000	1, 1
TR004	03/11/2025	P003	Andi Rahman	082345678901	PR004, PR005, PR006	Spaghetti, Salad, Air Mineral	30000, 20000, 5000	1, 1, 3

Gambar 4. Struktur Data dalam Bentuk *UNF*

Gambar 4 diatas menggambarkan kondisi *flat-file* pada sistem PizzaUpps yang masih berada dalam bentuk *UNF* (Unnormalized Form), dengan ciri-ciri sebagai berikut:

- 1) Atribut multi-valued (nilai jamak): Kolom *ID_Produk*, *Nama_Produk*, *Harga_Produk*, dan *Jumlah_Beli* mengandung beberapa nilai dalam satu sel, misalnya transaksi TR001 yang berisi dua produk (Pizza Meat dan Kentang Goreng).
- 2) Tidak ada kunci unik yang valid: Karena satu baris dapat memuat lebih dari satu produk, maka *No_Transaksi* tidak bisa dijadikan kunci utama yang unik.
- 3) Redundansi dan duplikasi: Data pelanggan seperti *Budi Santoso* muncul berulang kali di beberapa transaksi, menyebabkan duplikasi data.
- 4) Potensi anomali: *Update anomaly*: perubahan harga produk harus dilakukan di banyak baris. *Insertion anomaly*: produk baru tidak dapat ditambahkan tanpa transaksi. *Deletion anomaly*: menghapus satu transaksi dapat menghapus seluruh data pelanggan terkait.

b. First Normal Form (1NF)

Sebuah tabel dikatakan memenuhi *1NF* apabila semua nilai bersifat *atomik* (tidak mengandung set atau daftar nilai) dan setiap baris dapat diidentifikasi secara unik menggunakan kunci primer (Elmasri & Navathe, 2011).

Untuk mencapai 1NF, setiap nilai multi-valued diuraikan menjadi baris terpisah, sehingga setiap sel berisi hanya satu nilai. Kombinasi $\{No_Transaksi, ID_Produk\}$ ditetapkan sebagai *primary key*.

No. Transaksi	Tgl. Transaksi	ID_Pelanggan	Nama_Pelanggan	No. Telepon	ID_Produk	Nama_Produk	Harga_Produk	Jumlah_Beli
TR001	01/11/2025	P001	Budi Santoso	081234567890	PR001	Pizza Meat	50000	1
TR001	01/11/2025	P001	Budi Santoso	081234567890	PR003	Kentang Goreng	15000	2
TR002	01/11/2025	P002	Citra Lestari	085678901234	PR002	Pizza Keju	45000	1
TR003	02/11/2025	P001	Budi Santoso	081234567890	PR001	Pizza Meat	50000	1
TR003	02/11/2025	P001	Budi Santoso	081234567890	PR002	Pizza Keju	45000	1
TR004	03/11/2025	P003	Andi Rahman	082345678901	PR004	Spaghetti	30000	1
TR004	03/11/2025	P003	Andi Rahman	082345678901	PR005	Salad	20000	1
TR004	03/11/2025	P003	Andi Rahman	082345678901	PR006	Air Mineral	5000	3

Gambar 5. Struktur Data Normalisasi 1NF

c. Second Normal Form (2NF)

Sebuah tabel memenuhi 2NF apabila telah memenuhi 1NF dan tidak ada ketergantungan parsial, yakni setiap atribut non-kunci harus untuk mencapai 2NF, tabel 1NF dipecah menjadi empat tabel berdasarkan ketergantungan penuh terhadap kunci utama (Elmasri & Navathe, 2011).

Tabel 1. Normalisasi 2NF Tabel Pelanggan

Tabel Pelanggan		
ID_Pelanggan (PK)	Nama_Pelanggan	No_Telepon
P001	Budi Santoso	081234567890
P002	Citra Lestari	085678901234
P003	Andi Rahman	082345678901

Tabel 2. Normalisasi 2NF Tabel Produk

Tabel Produk		
ID_Produk (PK)	Nama_Produk	Harga_Produk
PR001	Pizza Meat	50000
PR002	Pizza Keju	45000
PR003	Kentang Goreng	15000
PR004	Spaghetti	30000
PR005	Salad	20000
PR006	Air Mineral	5000

Tabel 3. Normalisasi 2NF Tabel Transaksi

Tabel Transaksi		
No_Transaksi (PK)	Tgl_Transaksi	ID_Pelanggan (FK)
TR001	01/11/2025	P001
TR002	01/11/2025	P002
TR003	02/11/2025	P001
TR004	03/11/2025	P003

Tabel 4. Normalisasi 2NF Tabel Detail Transaksi

Tabel Detail Transaksi		
No_Transaksi (FK)	ID_Produk (FK)	Jumlah_Beli
TR001	PR001	1
TR001	PR003	2
TR002	PR002	1
TR003	PR001	1
TR003	PR002	1
TR004	PR004	1
TR004	PR005	1
TR004	PR006	3

d. Third Normal Form (3NF)

Tabel memenuhi 3NF apabila telah memenuhi 2NF dan tidak terdapat ketergantungan *transitif*, yaitu atribut non-kunci bergantung pada atribut non-kunci lainnya (Elmasri & Navathe, 2011).

Tabel	Kunci Primer (PK)	Kunci Asing (FK)	Ketergantungan Fungsional Utama
Pelanggan	ID_Pelanggan	-	ID_Pelanggan → Nama_Pelanggan, No_Telepon
Produk	ID_Produk	-	ID_Produk → Nama_Produk, Harga_Produk
Transaksi	No_Transaksi	ID_Pelanggan	No_Transaksi → Tgl_Transaksi, ID_Pelanggan
Detail_Transaksi	{No_Transaksi, ID_Produk}	FK ke Transaksi, FK ke Produk	{No_Transaksi, ID_Produk} → Jumlah_Beli

Gambar 6. Normalisasi 3NF

Kesimpulan Tahap Normalisasi: Hasil transformasi dari UNF → 3NF berhasil mengubah atribut multi-valued menjadi atomik, menghilangkan ketergantungan parsial dan transitif, dan menurunkan tingkat redundansi data secara signifikan.

Struktur ini kini memenuhi kriteria BCNF (Boyce-Codd Normal Form) karena setiap determinan merupakan kandidat kunci yang unik. Struktur ini menjadi dasar bagi tahap Perancangan Fisik.

3. Perancangan Fisik (Skema Relasi dan Struktur Database)

Tahap perancangan fisik menerjemahkan model logis menjadi struktur implementatif dalam bahasa SQL. Setiap tabel didefinisikan dengan tipe data, *primary key* (PK), dan *foreign key* (FK)

untuk memastikan integritas referensial. Berikut implementasi skema relasi:

```
CREATE TABLE Pelanggan (  
  ID_Pelanggan CHAR(5) PRIMARY  
  KEY,  
  Nama_Pelanggan VARCHAR(50),  
  No_Telepon VARCHAR(15)  
);
```

```
CREATE TABLE Produk (  
  ID_Produk CHAR(5) PRIMARY KEY,  
  Nama_Produk VARCHAR(50),  
  Harga_Produk DECIMAL(10,2)  
);
```

```
CREATE TABLE Transaksi (  
  No_Transaksi CHAR(5) PRIMARY  
  KEY,  
  Tgl_Transaksi DATE,  
  ID_Pelanggan CHAR(5),  
  FOREIGN KEY (ID_Pelanggan)  
  REFERENCES Pelanggan(ID_Pelanggan)  
);
```

```
CREATE TABLE Detail_Transaksi (  
  No_Transaksi CHAR(5),  
  ID_Produk CHAR(5),  
  Jumlah_Beli INT,  
  PRIMARY KEY (No_Transaksi,  
  ID_Produk),  
  FOREIGN KEY (No_Transaksi)  
  REFERENCES Transaksi(No_Transaksi),  
  FOREIGN KEY (ID_Produk)  
  REFERENCES Produk(ID_Produk)  
);
```

Skema fisik ini memastikan bahwa setiap entitas terhubung dengan benar melalui *foreign key* tanpa adanya pengulangan data. Sebagai contoh, data pelanggan hanya tersimpan satu kali dalam tabel Pelanggan, sedangkan tabel Transaksi dan Detail_Transaksi mengacu pada data tersebut menggunakan referensi kunci.

Tahap Implementasi (*Implementation*)

Tahap implementasi merupakan proses realisasi dari rancangan logis dan fisik yang telah dihasilkan pada tahap perancangan sebelumnya menjadi bentuk nyata (*executable database schema*).

Dalam penelitian ini, implementasi dilakukan menggunakan *MySQL DBMS versi 8.0*, yang dipilih karena mendukung standar *SQL:2016*, memiliki integritas relasional yang kuat, serta banyak digunakan oleh pelaku UMKM untuk sistem berbasis web.

1. Pembuatan Database

Langkah pertama dalam implementasi adalah membuat basis data baru dengan perintah *SQL* berikut:

```
CREATE DATABASE db_pizzaapps;  
USE db_pizzaapps;
```

Perintah *CREATE DATABASE* berfungsi untuk membuat wadah logis (logical container) penyimpanan seluruh tabel dan objek basis data PizzaUpps, sedangkan *USE* menetapkan konteks kerja aktif ke dalam database tersebut.

2. Pembuatan Struktur Tabel Utama

Tahap selanjutnya adalah mendefinisikan struktur tabel utama berdasarkan hasil perancangan logis pada tahap normalisasi. Tabel-tabel yang dibuat meliputi: Pelanggan, Produk, Transaksi, dan Detail_Transaksi.

Tabel Pelanggan

Tabel ini menyimpan data identitas pelanggan, yang menjadi sumber relasi dengan tabel *Transaksi*.

```
CREATE TABLE Pelanggan (  
  ID_Pelanggan CHAR(5) PRIMARY  
  KEY,  
  Nama_Pelanggan VARCHAR(50) NOT  
  NULL,  
  No_Telepon VARCHAR(15) NOT  
  NULL  
);
```

Tabel Produk

Menyimpan data produk yang dijual PizzaUpps, seperti nama produk dan harga satuan.

```
CREATE TABLE Produk (  
  ID_Produk CHAR(5) PRIMARY KEY,  
  Nama_Produk VARCHAR(50) NOT  
  NULL,  
  Harga_Produk DECIMAL(10,2) NOT  
  NULL  
);
```

Tabel Transaksi

Menyimpan informasi dasar transaksi penjualan, termasuk tanggal transaksi dan pelanggan yang melakukan pembelian.

```
CREATE TABLE Transaksi (  
  No_Transaksi CHAR(5) PRIMARY  
  KEY,  
  Tgl_Transaksi DATE NOT NULL,  
  ID_Pelanggan CHAR(5),  
  FOREIGN KEY (ID_Pelanggan)  
  REFERENCES Pelanggan(ID_Pelanggan)  
  ON UPDATE CASCADE  
  ON DELETE RESTRICT  
);
```

Dengan cara ini, integritas referensial antara pelanggan dan transaksi tetap terjamin.

Tabel Detail_Transaksi

Tabel ini menyimpan rincian produk yang dibeli pada setiap transaksi dan menjadi penghubung (*bridge table*) antara entitas Transaksi dan Produk.

```
CREATE TABLE Detail_Transaksi (  
  No_Transaksi CHAR(5),  
  ID_Produk CHAR(5),  
  Jumlah_Beli INT NOT NULL,  
  PRIMARY KEY (No_Transaksi,  
  ID_Produk),  
  FOREIGN KEY (No_Transaksi)  
  REFERENCES Transaksi(No_Transaksi)  
  ON UPDATE CASCADE  
  ON DELETE CASCADE,  
  FOREIGN KEY (ID_Produk)  
  REFERENCES Produk(ID_Produk)  
  ON UPDATE CASCADE  
  ON DELETE RESTRICT  
);
```

3. Implementasi Relasi Antar Tabel

Setelah semua tabel didefinisikan, relasi antar tabel dapat diverifikasi menggunakan perintah:

```
SHOW CREATE TABLE Transaksi;  
SHOW CREATE TABLE  
Detail_Transaksi;
```

Perintah ini digunakan untuk memastikan bahwa relasi antar tabel telah terbentuk secara benar (melalui definisi *foreign key*). Hasil eksekusi akan menunjukkan bahwa:

- Tabel Transaksi memiliki hubungan 1:N terhadap tabel Detail_Transaksi.
- Tabel Produk juga memiliki hubungan 1:N terhadap Detail_Transaksi.
- Tabel Pelanggan memiliki hubungan 1:N terhadap Transaksi.

Struktur ini sepenuhnya konsisten dengan *Entity Relationship Diagram* (ERD) pada Gambar 3 (Bagian 2.1).

4. Verifikasi Struktur Database

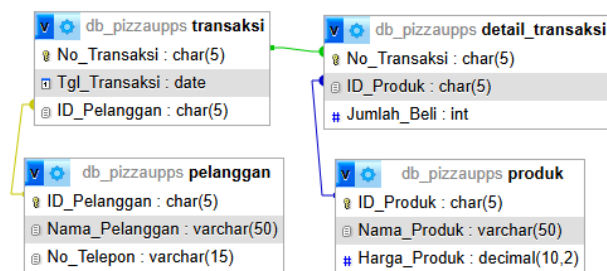
Setelah seluruh perintah DDL berhasil dijalankan, verifikasi dilakukan menggunakan perintah:

```
SHOW TABLES;  
DESCRIBE Pelanggan;  
DESCRIBE Transaksi;  
DESCRIBE Detail_Transaksi;
```

Langkah ini memastikan:

- Semua tabel terbentuk dengan atribut sesuai rancangan.
- Constraint telah diterapkan dengan benar (PK, FK, dan NOT NULL).
- Relasi antar tabel saling terhubung tanpa error dependensi.

Hasil eksekusi perintah DESCRIBE dapat divisualisasikan dalam bentuk *screenshot MySQL Workbench* **Gambar 6.** untuk menunjukkan bahwa setiap kolom, tipe data, dan *constraint* sudah sesuai dengan rancangan.



Gambar 6. MySQL Workbench

Hasil implementasi menunjukkan bahwa rancangan basis data relasional yang dibangun berdasarkan hasil normalisasi hingga *Third Normal Form* (3NF) dapat diwujudkan sepenuhnya dalam lingkungan MySQL. Secara fungsional, hasil pengujian menunjukkan:

1. Redundansi Data Hilang
Data pelanggan, produk, dan transaksi tersimpan dalam tabel terpisah, sesuai prinsip *data independence*.
2. Integritas Referensial Terjamin
Penggunaan *foreign key* dengan aturan *CASCADE* dan *RESTRICT* memastikan konsistensi antar tabel.
3. Kemudahan Pemeliharaan dan Ekspansi Sistem
Struktur tabel modular memungkinkan penambahan fitur baru, seperti sistem laporan keuangan atau inventori bahan baku, tanpa memengaruhi integritas data yang sudah ada.
4. Kesesuaian dengan Prinsip Desain Relasional Modern
Desain dan implementasi ini mengikuti prinsip desain yang direkomendasikan oleh (Elmasri & Navathe, 2011) serta (Connolly & Begg, 2019), di mana setiap tabel merepresentasikan entitas tunggal dan setiap relasi direpresentasikan secara eksplisit melalui constraint antar tabel.

Dengan demikian, tahap implementasi berhasil membangun *database* relasional PizzaUpps yang terstruktur, efisien, dan siap diuji lebih lanjut pada tahap pengujian dan validasi.

Tahap Pengujian dan Validasi (Testing and Validation)

Tahap pengujian dan validasi merupakan fase akhir dalam siklus pengembangan berbasis model *Waterfall*, yang berfungsi memastikan bahwa desain basis data baru (*To-Be Design*) telah mampu mengatasi kelemahan sistem lama (*As-Is*). Pengujian dilakukan dengan metode uji skenario (*scenario testing*) terhadap tiga jenis operasi utama yaitu *update*, *insert*, dan *delete*, yang sebelumnya menjadi sumber utama anomali data.

Selain itu, tahap ini juga mencakup validasi kebutuhan fungsional, yakni penilaian terhadap kemampuan sistem baru untuk menghasilkan data yang konsisten, bebas duplikasi, dan siap digunakan dalam pelaporan keuangan UMKM PizzaUpps.

1. Hasil Uji Skenario Pengelolaan Data

Uji efektivitas dilakukan dengan menjalankan tiga skenario utama pada sistem lama (*As-Is*) dan sistem baru (*To-Be*). Fokus pengujian diarahkan pada tiga bentuk anomali: *insertion anomaly*, *update anomaly*, dan *deletion anomaly*. Hasil pengujian dirangkum dalam **Gambar 7**. berikut:

Skenario Pengujian	Sistem As-Is (Flat-File)	Sistem To-Be (Database Relasional)
1. Insertion (Menambah Pelanggan Baru)	Tidak dapat dilakukan tanpa transaksi baru karena No_Transaksi menjadi kunci utama.	Dapat dilakukan langsung pada tabel <i>Pelanggan</i> tanpa memerlukan transaksi.
2. Update (Perubahan Nomor Telepon Pelanggan)	Harus diperbarui secara manual pada setiap baris transaksi yang memuat data pelanggan tersebut, berpotensi tidak konsisten.	Perubahan cukup dilakukan sekali pada tabel <i>Pelanggan</i> dan otomatis tercermin di seluruh transaksi.
3. Delete (Penghapusan Transaksi Pembelian)	Menghapus satu transaksi dapat menghilangkan seluruh data pelanggan terkait (<i>deletion anomaly</i>).	Data pelanggan tetap tersimpan karena memiliki tabel terpisah yang tidak bergantung pada transaksi.

Gambar 7. Perbandingan Hasil Pengujian Sistem *As-Is* dan *To-Be*

2. Analisis Hasil Pengujian

Hasil uji menunjukkan bahwa sistem basis data baru berhasil mengatasi seluruh bentuk anomali yang sebelumnya ditemukan pada sistem lama.

- a. Eliminasi Redundansi: Data pelanggan dan produk kini hanya tersimpan satu kali dalam tabel masing-masing, menghapus pengulangan data yang

sebelumnya menyebabkan pemborosan ruang penyimpanan.

- b. Pencegahan Anomali: (1) *Insertion anomaly* diatasi dengan pemisahan tabel Pelanggan, sehingga data pelanggan baru dapat ditambahkan tanpa transaksi; (2) *Update anomaly* dihilangkan karena perubahan informasi pelanggan dilakukan pada satu tempat dan berlaku global melalui mekanisme *foreign key*; (c) *Deletion anomaly* tidak terjadi karena penghapusan transaksi tidak menghapus data master pelanggan atau produk.
- c. Peningkatan Integritas dan Konsistensi Data: Struktur relasional memungkinkan sistem menjaga keterkaitan antar tabel melalui *referential integrity constraint*. Uji *foreign key* menunjukkan tidak ada data yatim (*orphan record*) yang muncul selama proses penghapusan atau pembaruan.
3. Validasi Kebutuhan Fungsional
Validasi kebutuhan fungsional dilakukan dengan mengevaluasi sejauh mana sistem baru memenuhi keperluan informasi dan pelaporan keuangan UMKM PizzaUpps. Hasil validasi menunjukkan bahwa:
 - a. Data transaksi dapat ditarik secara cepat dan akurat melalui perintah SQL berbasis relasi antar tabel.
 - b. Laporan keuangan, seperti rekap penjualan per pelanggan atau per produk, dapat dihasilkan tanpa risiko duplikasi data.
 - c. Sistem baru mempermudah analisis performa bisnis, seperti penghitungan laba per produk atau pelanggan loyal, karena data lebih terstruktur dan konsisten.
 - d. Staf administrasi mengonfirmasi penurunan signifikan terhadap kesalahan pencatatan (*human error*) dan waktu yang dibutuhkan untuk

memperbarui data pelanggan serta transaksi.

D. PENUTUP

Berdasarkan hasil penelitian yang telah dilakukan melalui tahapan metode Waterfall meliputi analisis kebutuhan, perancangan, implementasi, serta pengujian dan validasi dapat disimpulkan bahwa penelitian ini berhasil merancang arsitektur basis data yang optimal untuk sistem keuangan UMKM PizzaUpps.

Proses perancangan dilakukan dengan menerapkan teknik normalisasi hingga tingkat Third Normal Form (3NF) bahkan memenuhi kriteria Boyce-Codd Normal Form (BCNF) untuk menjamin efisiensi dan integritas data. Melalui analisis sistem berjalan (*As-Is*), ditemukan bahwa struktur basis data lama masih berbentuk flat-file yang melanggar prinsip Second Normal Form (2NF) dan Third Normal Form (3NF). Kondisi ini mengakibatkan redundansi data (pengulangan data pelanggan dan produk) serta anomali penyisipan, pembaruan, dan penghapusan yang berpotensi menurunkan konsistensi informasi keuangan.

Desain baru (*To-Be*) yang diusulkan terdiri atas tabel Pelanggan, Produk, Transaksi, dan Detail_Transaksi telah teruji dan tervalidasi mampu mengatasi permasalahan tersebut. Struktur basis data hasil normalisasi menghasilkan sistem yang menghilangkan redundansi melalui pemisahan entitas utama dan penerapan *foreign key* antar tabel, menjamin konsistensi data melalui integritas referensial dengan aturan ON UPDATE CASCADE dan ON DELETE RESTRICT, dan mempermudah pengelolaan serta pemrosesan laporan keuangan dengan struktur data yang efisien dan terstandarisasi.

Dengan demikian, tujuan penelitian untuk mengoptimalkan basis data sistem keuangan UMKM PizzaUpps agar terhindar dari redundansi dan anomali data telah

tercapai sepenuhnya. Desain ini terbukti mendukung prinsip *data consistency*, *data independence*, dan *referential integrity* sesuai teori relasional modern.

E. DAFTAR PUSTAKA

- Asmarawati, B. (2025). Dari manual ke digital: Evolusi akuntansi UMKM di era teknologi. *Management Studies and Entrepreneurship Journal (MSEJ)*, 6(3), 3205–3214.
<https://doi.org/10.37385/msej.v6i3.7801>
- Chaidir, M., Ruslaini, R., & Irawan, D. (2025). Transformasi digital dalam manajemen keuangan: Studi kasus pada UMKM Indonesia di era ekonomi digital. *Jurnal Mahasiswa Manajemen Dan Akuntansi*, 4(1), 239–249.
<https://doi.org/10.30640/jumma45.v4i1.4138>
- Connolly, T. M., & Begg, C. E. (2019). *Database systems: A practical Approach to Design, Implementation, and Management, 6th Edition* (6th ed.). Harlow UK: Pearson Education Limited.
- Creswell, J. W., & Creswell, J. D. (2018). *Research design: Qualitative, quantitative, and mixed methods approaches* (5th ed.). SAGE Publications.
- Elmasri, R., & Navathe, S. B. (2011). *Fundamentals of Database Systems* (7th ed.). Boston: Pearson Education.
- Kieso, D. E., Weygandt, J. J., & Warfield, T. D. (2019). *Intermediate accounting* (17th ed.). New Jersey: John Wiley & Sons Inc.
- Pane, A. S. A. S., Putri, H. R., Purba, A. T., & Harahap, L. M. (2025). Peran Usaha Mikro Kecil Menengah (UMKM) terhadap Perekonomian Indonesia. *JPIM: Jurnal Penelitian Ilmiah Multidisipliner*, 1(2), 122–129.
<https://ojs.ruangpublikasi.com/index.php/jpim/article/view/259>
- Pressman, R. S., & Maxim, B. R. (2020). *Software Engineering: A Practitioner's Approach* (9th ed.). New York: McGraw Hill.
- Rafianto, G., & Voutama, A. (2025). Implementasi basis data terstruktur dengan pencegahan SQL injection pada sistem manajemen penjualan. *Jurnal Informatika Dan Teknik Elektro Terapan (JITET)*, 13(2), 895–903.
<https://doi.org/10.23960/jitet.v13i2.6354>
- Renanti, M. D., Pratini, C. N. P., Nisrina, S. N. A., Zaki, F. M., Luthfiana, D. E., & Baruna, M. E. (2025). Optimalisasi Desain Basis Data E-Commerce untuk Menjamin Integritas Data (Studi Kasus: Web E-Commerce Rie.charge). *CoSciTech: Computer Science and Information Technology*, 6(3), 444–452.
<https://doi.org/10.37859/coscitech.v6i3.10643>
- Romney, M. B., & Steinbart, P. J. (2021). *Accounting information systems* (15th ed.). Pearson.
- Wati, K. Iita, Rita, W., & Rahmah, W. (2025). Transformasi Manajemen Keuangan Umkm dalam Era Digital dan Ketidakpastian Ekonomi: Sebuah Eksplorasi Humanistik. *RIGGS: Journal of Artificial Intelligence and Digital Business*, 4(3), 566–575.
<https://doi.org/10.31004/riggs.v4i3.2033>
- Yusril, M., Kawuta, A. J. J., Banni, S. D., & Ratu, Y. S. (2025). Desain Model Relational Database Perpustakaan SMP Negeri 13 Tarakan Menggunakan Metode Database Life Cycle. *Jurnal Terapan Teknologi Informasi (JUTEI)*, 9(1), 1–10.
<https://doi.org/10.21460/jutei.2025.91.398>
- Zaini, R. (2024). Efektivitas digitalisasi UMKM dalam meningkatkan daya saing ekonomi di era industri 4.0. *Jurnal Pro Ekonomi*, 1(1), 26–33.
<https://ekonomi.jurnalpro.com/index.php/ekonomi/article/view/6>