

# RANCANG BANGUN APLIKASI MANAJEMEN WARUNG DIGITAL MENGGUNAKAN FLUTTER, GOLANG, DAN POSTGRESQL

<sup>1</sup>Dimas Febriyanto

<sup>2</sup>M. Achsan Isa Al Anshori. S.Kom.,MMSI

Jurusan Informatika  
Fakultas Teknologi Industri  
Universitas Gunadarma  
Jl. Margonda Raya No. 100, Pondok Cina, Depok, 15424  
dimassfeb@gmail.com.

## ABSTRAK

*Warung sembako memiliki peran penting dalam menunjang ekonomi rumah tangga, terutama di lingkungan permukiman padat. Namun, pengelolaan operasional warung sebagian besar masih dilakukan secara manual, termasuk pencatatan stok, transaksi, dan harga barang. Hal ini seringkali menyebabkan ketidakteraturan data, kesalahan transaksi, hingga potensi kerugian. Penelitian ini bertujuan untuk merancang dan mengembangkan aplikasi manajemen warung digital berbasis mobile yang dapat membantu pemilik warung dalam mengelola operasional sehari-hari secara efisien dan akurat. Aplikasi dibangun menggunakan Flutter sebagai antarmuka pengguna, Golang untuk layanan backend, serta PostgreSQL sebagai sistem manajemen basis data. Metode pengembangan menggunakan pendekatan System Development Life Cycle (SDLC) model waterfall yang mencakup tahap perencanaan, analisis kebutuhan, perancangan sistem, implementasi, dan pengujian. Pengujian dilakukan menggunakan metode black-box untuk memverifikasi fungsionalitas setiap fitur utama, serta evaluasi usability dengan skala Likert yang melibatkan pengguna dari kalangan non-teknis. Hasil pengujian menunjukkan bahwa aplikasi mampu mencatat dan mengelola produk, transaksi, serta menyajikan laporan penjualan dalam bentuk ringkas dan grafik sederhana. Aplikasi ini juga telah dipublikasikan melalui Google Play Store dan dapat digunakan secara umum. Temuan ini menunjukkan bahwa sistem yang dikembangkan efektif dalam meningkatkan efisiensi pengelolaan warung serta mendukung digitalisasi UMKM.*

**Kata Kunci:** warung, manajemen stok, flutter, golang, postgresql.

## ABSTRACT

*Grocery stalls (warung) play a vital role in supporting household economies, especially in densely populated residential areas. However, many still rely on manual operations for inventory, pricing, and sales management, often leading to data inaccuracy, transaction errors, and financial losses. This research aims to design and develop a mobile-based digital stall management application to improve daily operations' accuracy and efficiency. The application is built using Flutter for the user interface, Golang as the backend service, and PostgreSQL as the database system. The development follows the System Development Life Cycle (SDLC) using the waterfall model, consisting of planning, requirements analysis, system design, implementation, and testing stages. Black-box testing was employed to ensure*

*feature functionality, while a Likert scale-based usability evaluation was conducted with non-technical users. Test results indicate the application successfully manages product data, processes transactions, and generates periodic sales reports with basic visualizations. The application has been released publicly on the Google Play Store and is ready for real-world use. These findings demonstrate that the developed system is effective in enhancing stall management operations and contributes to the broader digital transformation of micro-enterprises.*

**Keywords:** stall, inventory management, flutter, golang, postgresql

## PENDAHULUAN

Warung sembako merupakan usaha mikro yang umum ditemukan di lingkungan permukiman dan memiliki peran penting dalam perekonomian lokal. Namun, sebagian besar pengelolaan warung masih menggunakan metode manual. Berdasarkan data Kementerian Koperasi dan UKM (2023), hanya 22% UMKM di Indonesia yang sudah menerapkan teknologi digital. Oleh karena itu, diperlukan solusi digital yang mudah diakses dan digunakan oleh pelaku usaha mikro. Tujuan penelitian ini adalah merancang dan mengembangkan aplikasi manajemen warung berbasis mobile yang sederhana dan efisien, menggunakan teknologi Flutter, Golang, dan PostgreSQL.

## METODE PENELITIAN

Metode pengembangan aplikasi mengikuti model *System Development Life Cycle (SDLC)* dengan pendekatan *waterfall*, terdiri dari:

- **Perencanaan:** Menentukan ruang lingkup aplikasi dan teknologi yang digunakan (Flutter, Golang, PostgreSQL, BloC).
- **Analisis:** Identifikasi kebutuhan pengguna berdasarkan observasi dan wawancara langsung.
- **Perancangan:** Membuat struktur navigasi, UI mockup, ERD, dan diagram UML.
- **Implementasi:** Frontend dikembangkan dengan Flutter, backend dengan Golang, dan integrasi dilakukan melalui REST API.
- **Pengujian:** Menggunakan metode *black-box* dan evaluasi usability dengan skala Likert.

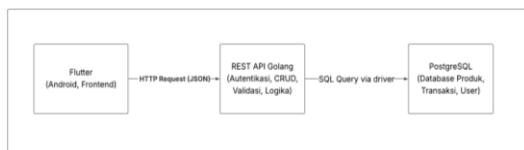
## PEMBAHASAN

Aplikasi manajemen warung digital dikembangkan untuk membantu pemilik warung sembako dalam mengelola produk, mencatat transaksi penjualan, dan melihat laporan keuangan secara efisien. Aplikasi ini dirancang menggunakan arsitektur

*client-server* dengan pemisahan antara sisi pengguna (frontend), logika bisnis (backend), dan penyimpanan data (basis data).

Aplikasi dibangun menggunakan Flutter sebagai teknologi frontend yang memungkinkan pengembangan antarmuka pengguna mobile secara lintas platform. Backend dikembangkan dengan Golang karena kemampuannya dalam menangani permintaan REST API secara ringan dan cepat. PostgreSQL dipilih sebagai sistem manajemen basis data karena mendukung relasi antar tabel, integritas data, serta skalabilitas yang baik.

Untuk memberikan gambaran menyeluruh tentang aliran data dalam sistem, berikut disajikan diagram arsitektur sistem:



Gambar 1 Arsitektur Sistem

Diagram ini menunjukkan interaksi antara komponen utama aplikasi. Pengguna mengakses aplikasi melalui antarmuka Flutter pada perangkat mobile. Seluruh permintaan pengguna, seperti login, pencatatan produk, dan

transaksi, dikirimkan ke backend melalui REST API. Backend yang dibangun dengan Golang akan memproses permintaan tersebut, melakukan validasi, dan berinteraksi dengan basis data PostgreSQL untuk menyimpan atau mengambil data yang dibutuhkan. Backend kemudian mengirimkan respons kembali ke frontend dalam format JSON. Komunikasi antara frontend dan backend diamankan menggunakan protokol HTTPS, sedangkan sistem autentikasi menggunakan token JWT.

## Perencanaan

Tahap perencanaan dilakukan untuk menetapkan arah dan ruang lingkup pengembangan sistem. Proyek ini bertujuan membangun aplikasi manajemen warung yang mampu membantu pelaku usaha mikro dalam mengelola aktivitas operasional seperti pencatatan produk, transaksi penjualan, dan penyusunan laporan keuangan. Sistem ditargetkan berjalan pada perangkat Android agar mudah diakses oleh pengguna dengan keterbatasan perangkat dan literasi digital.

Berdasarkan kebutuhan pengguna dan tujuan sistem, dipilihlah beberapa

teknologi inti:

- **Flutter:** digunakan sebagai framework frontend karena mendukung pengembangan lintas platform dengan satu basis kode dan menghasilkan antarmuka pengguna yang responsif dan modern.
- **Golang:** dipilih untuk backend karena ringan, cepat, dan andal dalam menangani REST API serta memiliki dukungan pustaka untuk keamanan dan middleware.
- **PostgreSQL:** dipilih sebagai sistem basis data relasional karena mendukung struktur tabel kompleks, integritas data, serta dapat dijalankan di berbagai lingkungan produksi.
- **BLoC (Business Logic Component):** digunakan sebagai pendekatan *state management* pada Flutter untuk memisahkan antara logika bisnis dan tampilan, serta memudahkan pengujian dan pengelolaan alur data.

Aplikasi dirancang menggunakan arsitektur *client-server*, dengan komunikasi antara frontend dan backend dilakukan melalui protokol REST API berbasis JSON. Sistem

autentikasi akan dibangun menggunakan skema JWT (*JSON Web Token*) untuk menjamin keamanan dan identitas pengguna.

Lingkup pengembangan pada tahap ini meliputi fitur-fitur berikut:

- Registrasi dan login pengguna
- Manajemen produk: tambah, edit, hapus, dan pencarian
- Pencatatan transaksi penjualan
- Laporan pendapatan harian, mingguan, dan bulanan
- Ringkasan statistik dalam bentuk teks dan grafik

Untuk menjamin pengembangan berjalan terstruktur, proses pembangunan aplikasi akan mengikuti metode *System Development Life Cycle (SDLC)* model *waterfall*, yang terdiri dari tahapan: Perencanaan, Analisis, Perancangan, Implementasi, dan Pengujian. Setiap tahapan dilakukan secara sistematis dan terurut agar hasil akhir memenuhi kebutuhan fungsional dan non-fungsional dari sistem.

Tabel 1 Teknologi Aplikasi Warung Digital

Komponen	Teknologi	Alasan Pemilihan
Frontend	Flutter	Cepat, lintas platform, UI responsif
Backend	Golang	Ringan, cepat, cocok untuk REST API
Basis Data	PostgreSQL	Stabil, mendukung relasi dan transaksi

Autentikasi	JWT	Aman, stateless, cocok untuk REST API
State Mgmt	BloC	Pisahkan logic dan UI, scalable

## Analisis

Tahap analisis dilakukan untuk mengidentifikasi kebutuhan sistem berdasarkan hasil observasi langsung dan wawancara dengan pemilik warung sembako yang menjadi target pengguna aplikasi. Ditemukan bahwa sebagian besar proses operasional warung, termasuk pencatatan produk, stok barang, dan transaksi penjualan, masih dilakukan secara manual menggunakan buku catatan. Hal ini menimbulkan berbagai masalah seperti:

- Kesalahan dalam penghitungan stok dan harga barang
- Ketidakteraturan pencatatan transaksi harian
- Tidak adanya sistem laporan pendapatan harian, mingguan, atau bulanan
- Sulitnya pemilik warung melacak performa usaha secara keseluruhan

Untuk menjawab permasalahan tersebut, dilakukan analisis kebutuhan fungsional dan non-fungsional yang dirumuskan sebagai berikut:

## Kebutuhan Fungsional

1. **Autentikasi pengguna:** Fitur login dan registrasi untuk mengakses aplikasi secara personal.
2. **Manajemen produk:** Pengguna dapat menambahkan, mengubah, menghapus, dan mencari data produk.
3. **Pencatatan transaksi:** Transaksi penjualan dicatat secara otomatis dan disimpan ke basis data.
4. **Laporan keuangan:** Menampilkan pendapatan berdasarkan rentang waktu tertentu (harian, mingguan, bulanan).
5. **Ringkasan dalam bentuk grafik:** Menyajikan data penjualan dan pendapatan dalam bentuk visual.

## Kebutuhan Non-Fungsional

1. **Kemudahan penggunaan:** Antarmuka pengguna yang sederhana dan mudah dipahami oleh pengguna non-teknis.
2. **Responsivitas:** Performa aplikasi ringan saat dijalankan di perangkat Android kelas menengah ke bawah.
3. **Keamanan:** Menggunakan token

JWT untuk menjaga keamanan akses pengguna dan transaksi.

4. **Skalabilitas:** Struktur sistem mendukung pengembangan ke fitur tambahan seperti QRIS dan multi warung.

Analisis kebutuhan ini menjadi dasar utama dalam perancangan struktur sistem, antarmuka pengguna, serta alur interaksi yang akan diimplementasikan pada tahap selanjutnya.

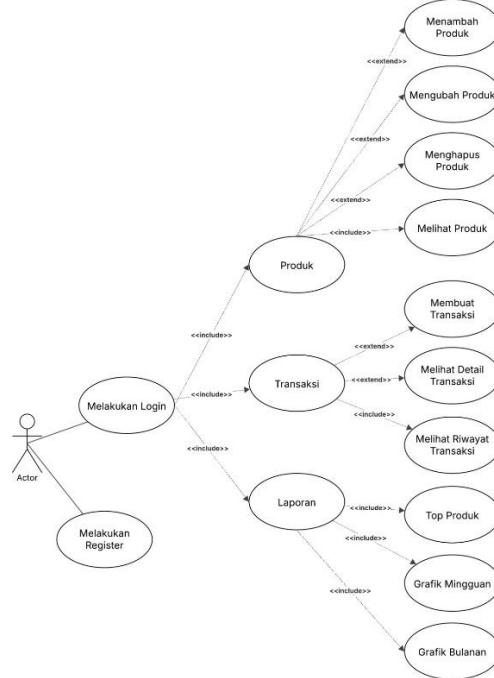
## Perancangan

Tahap perancangan merupakan proses menyusun blueprint sistem berdasarkan kebutuhan yang telah dianalisis sebelumnya. Perancangan dilakukan secara menyeluruh, meliputi: use case diagram, struktur basis data, arsitektur frontend-backend, dan desain antarmuka pengguna. Tujuannya adalah memastikan bahwa sistem yang dibangun nantinya sesuai dengan kebutuhan pengguna akhir dan mudah diimplementasikan secara modular.

### 1. Usecase Diagram

Use Case Diagram digunakan untuk menggambarkan hubungan antara pengguna (aktor) dengan fungsionalitas sistem. Dalam konteks aplikasi ini, aktor utama

adalah **pemilik warung**, yang berinteraksi langsung dengan aplikasi untuk mengelola produk, mencatat transaksi, dan melihat laporan.



Gambar 2 Usecase Diagram Aplikasi

### 2. Entity Relationship Diagram (ERD)

Untuk menyusun basis data yang efisien dan terstruktur, digunakan Entity Relationship Diagram (ERD) yang menggambarkan entitas dan relasi antar tabel. Beberapa entitas utama dalam sistem ini meliputi:

- users: menyimpan data pengguna aplikasi
- products: menyimpan data

- barang dagangan
- transactions: menyimpan catatan transaksi
  - transaction\_items: menyimpan rincian barang dalam transaksi

Relasi antar entitas dirancang sebagai berikut:

- Satu user dapat memiliki banyak transactions
- Satu transaction memiliki banyak transaction\_items
- Satu product dapat muncul pada banyak transaction\_items



Gambar 3 ERD Aplikasi

3. Navigasi dan Struktur Antarmuka
- Aplikasi frontend dirancang agar dapat dioperasikan dengan mudah oleh pengguna non-teknis. Navigasi utama menggunakan **Bottom Navigation Bar** dengan beberapa menu utama seperti:

- Dashboard
- Produk
- Transaksi
- Laporan
- Profil

Desain antarmuka mengikuti prinsip **mobile-first** dan **minimalist UI**, dengan mempertimbangkan keterbatasan literasi digital pengguna.

#### 4. Arsitektur Backend

Struktur backend menggunakan pendekatan **Modular Service Architecture**, dengan pemisahan antara komponen berikut:

- **Controller**: menangani permintaan dari frontend
- **Service**: menangani logika bisnis
- **Repository**: berinteraksi langsung dengan basis data
- **Middleware**: validasi, autentikasi, dan logging

Setiap endpoint API dikembangkan mengikuti standar REST dan merespons dalam format JSON. Autentikasi pengguna menggunakan skema JWT (JSON Web Token) untuk menjaga keamanan dan otorisasi setiap permintaan.

### Implementasi

Tahap implementasi merupakan proses penerjemahan rancangan sistem ke dalam bentuk kode program yang dapat

dijalankan dan digunakan oleh pengguna. Implementasi dilakukan secara modular dan terpisah antara frontend (Flutter) dan backend (Golang), yang saling berkomunikasi melalui REST API dengan pertukaran data dalam format JSON.

### 1. Implementasi Frontend

Frontend dibangun menggunakan framework **Flutter** yang mendukung pengembangan aplikasi mobile lintas platform. Aplikasi ini dirancang untuk dijalankan pada sistem operasi Android, dengan desain antarmuka yang sederhana dan mudah digunakan oleh pengguna non-teknis.

Untuk pengelolaan alur data dan logika bisnis, digunakan pendekatan **BLoC (Business Logic Component)**. Setiap fitur utama memiliki BloC tersendiri, antara lain:

- AuthBloc untuk proses login dan registrasi pengguna
- ProductBloc untuk pengelolaan data produk
- TransactionBloc untuk pencatatan dan histori transaksi

- ReportBloc untuk menampilkan laporan penjualan

Setiap Bloc bekerja berdasarkan event dan menghasilkan state, sehingga antarmuka dapat berubah secara reaktif sesuai kondisi logika.

### 2. Implementasi Backend

Backend dikembangkan menggunakan bahasa pemrograman **Golang** dan framework **Gin**. Backend bertanggung jawab untuk:

- Autentikasi pengguna menggunakan JWT
- Validasi dan pemrosesan data dari frontend
- Mengelola relasi dan query data pada PostgreSQL
- Menyediakan REST API untuk setiap endpoint yang dibutuhkan frontend

Struktur backend disusun dalam arsitektur modular yang terdiri dari:

- **Controller:** menerima dan merespons permintaan HTTP
- **Service:** menangani logika bisnis utama
- **Repository:** terhubung

langsung ke basis data PostgreSQL

- **Middleware:** memfilter akses, autentikasi JWT, dan logging

Beberapa endpoint yang dikembangkan antara lain:

- /auth/login
- /auth/register
- /products
- /transactions
- /reports/daily,  
/reports/monthly

### 3. Implementasi Basis Data

Sistem manajemen basis data yang digunakan adalah **PostgreSQL**. Struktur tabel dibuat berdasarkan rancangan ERD dengan relasi sebagai berikut:

```
users → transactions →  
transaction_items → products
```

Beberapa teknik optimasi yang digunakan:

- Indeks pada kolom pencarian seperti product\_name
- Validasi tipe data dan batasan nilai (constraint)
- UUID untuk primary key agar lebih aman dan unik

### 4. Integrasi dan Komunikasi

Frontend–Backend

Integrasi dilakukan menggunakan **HTTP request** pada Flutter dengan memanfaatkan pustaka http. Setiap permintaan mengandung token JWT di bagian header untuk mengakses data pengguna secara aman.

Contoh Implementasi:

```
final response = await  
http.get(  
Uri.parse('https://api.warung  
digital.com/products'),  
headers: {'Authorization':  
'Bearer $token'},  
)
```

Backend kemudian memvalidasi token dan merespons permintaan sesuai data yang diminta.

### 5. Deployment

Pada tahap akhir, sistem di-deploy dalam lingkungan pengujian untuk memastikan integrasi frontend dan backend berjalan dengan baik. Karena proyek ini masih berada pada tahap awal dan bersifat pengujian (staging), digunakan **Ngrok** sebagai solusi untuk membuat backend lokal dapat diakses secara publik melalui internet.

Langkah-langkah deployment yang dilakukan meliputi:

1. **Menyiapkan lingkungan**

**backend** di mesin lokal (Windows/Linux) yang telah menginstal Golang dan PostgreSQL.

2. **Menjalankan server backend Golang** secara lokal pada port 8080.
3. **Menggunakan Ngrok** untuk membuat *tunnel* dari port lokal ke domain publik dengan perintah seperti:

```
ngrok http 8080
```

Ngrok akan memberikan URL seperti <https://dimas-api.ngrok.io>, yang kemudian digunakan oleh frontend untuk mengakses backend dari perangkat Android.

4. **Mengonfigurasi frontend Flutter** agar menggunakan domain publik dari Ngrok, misalnya pada file konfigurasi `api_base_url.dart`.
5. **Build dan install APK Flutter** ke perangkat Android untuk pengujian fungsionalitas secara langsung.
6. **Distribusi aplikasi** dilakukan secara terbatas melalui

sideloading .abb Google Play Console internal testing.

## Pengujian

Pengujian dilakukan dalam dua pendekatan:

### Pengujian Black-Box

Pengujian dilakukan untuk memastikan bahwa setiap fitur dalam aplikasi Warung Digital berjalan sesuai dengan kebutuhan fungsional yang telah dirancang. Metode yang digunakan adalah **Black-Box Testing**, yang berfokus pada pengujian input dan output tanpa memperhatikan struktur internal program. Dalam pengujian ini digunakan pendekatan **Equivalence Partitioning**, yaitu dengan mengelompokkan input ke dalam kelas-kelas ekuivalen (valid dan tidak valid), lalu menguji representasi dari masing-masing kelas tersebut. Tujuannya adalah untuk memastikan bahwa sistem dapat menangani input dari berbagai kategori secara benar dan konsisten.

Berikut adalah hasil pengujian untuk beberapa fitur utama:

Tabel 2 Hasil Pengujian Blackbox Testing

No	Fitur	Input (Kelas Ekuivalen)	Output yang Diharapkan	Hasil	Status
1	Tambah Produk	Nama = ""	Error: Nama produk wajib diisi	Error	Sesuai
2	Tambah Produk	Harga = 50	Error: Harga tidak valid	Error	Sesuai
3	Tambah Produk	Harga = 100000	Produk berhasi l disimpan	OK	Sesuai
4	Tambah Produk	Harga = "abc" (non-numerik)	Error: Input tidak valid	Error	Sesuai
5	Tambah Produk	Stok = -10	Error: Stok tidak boleh negatif	Error	Sesuai
6	Tambah Produk	Stok = 50	Produk berhasi l disimpan	OK	Sesuai
7	Registrasi	Email = user@domain	Error: Email tidak valid	Error	Sesuai
8	Registrasi	Email = user@email.com	Pengguna berhasi l terdaftar	OK	Sesuai
9	Login	Password = ""	Error: Password wajib diisi	Error	Sesuai
10	Login	Password valid	Masuk dashboard	OK	Sesuai

Berdasarkan hasil pengujian menggunakan teknik Equivalence Partitioning, dapat disimpulkan bahwa sistem mampu merespons secara tepat terhadap berbagai kategori input, baik valid maupun tidak valid. Semua pengujian menunjukkan bahwa sistem

memberikan output yang sesuai dengan spesifikasi fungsional yang telah ditentukan.

Hasil ini menunjukkan bahwa **mekanisme validasi input** pada aplikasi Warung Digital telah diimplementasikan dengan baik dan berjalan sebagaimana mestinya. Dengan pendekatan ini, pengujian menjadi lebih efisien, terarah, dan tetap representatif terhadap perilaku sistem secara keseluruhan.

## Evaluasi Usability

Pengujian ini dilakukan untuk mengukur tingkat *usability* dari aplikasi **Warung Digital** berdasarkan perspektif pengguna langsung, yaitu penjaga atau pemilik warung sembako. Tujuan dari pengujian ini adalah untuk mengevaluasi sejauh mana aplikasi dapat digunakan secara efisien, efektif, dan memuaskan oleh pengguna non-teknis.

Aspek yang dinilai mencakup:

1. Kemudahan penggunaan aplikasi
2. Kejelasan navigasi
3. Kenyamanan tampilan antarmuka
4. Kesesuaian fitur dengan kebutuhan operasional

## 5. Kepuasan umum terhadap aplikasi

### Latar Belakang Responden

Responden pengujian terdiri dari tiga orang yang memiliki latar belakang usia dan pengalaman usaha yang berbeda. Hal ini bertujuan untuk mengukur *usability* dari berbagai sudut pandang pengguna:

Tabel 3 Latar Belakang Responden

No	Nama	Usia	Status	Nama Warung	Lama Usaha
P1	Jai	20	Penjaga (karyawannya)	Toko Jamil	15 tahun
P2	Sutarti	48	Pemilik dan penjaga	Java Hills Honey	3 tahun
P3	Novi	52	Pemilik dan penjaga	Dzakiyah Store	5 tahun

### Hasil Pengujian Usability

Masing-masing responden diminta memberikan nilai terhadap 17 pernyataan yang merepresentasikan aspek *usability* aplikasi. Skor diberikan dalam rentang 1 (sangat tidak setuju) hingga 5 (sangat setuju).

Tabel 4 Hasil Pengujian Usability

Pengujian	P1	P2	P3
Aplikasi mudah digunakan tanpa pelatihan	5	5	5
Navigasi jelas dan tidak membingungkan	5	5	5
Tampilan nyaman dilihat dan enak	5	5	5

digunakan			
Fitur sesuai kebutuhan pengelolaan warung	5	5	5
Secara umum saya puas menggunakan aplikasi ini	5	5	5
Teks dan ikon pada aplikasi mudah dibaca.	5	4	5
Tata letak (layout) aplikasi memudahkan saya memahami alur penggunaannya.	5	5	5
Warna dan tampilan aplikasi tidak membuat saya bingung atau lelah saat digunakan.	5	5	5
Saya dapat mencatat transaksi pelanggan dengan cepat.	5	5	5
Menambahkan produk baru ke dalam stok terasa mudah dan tidak memakan waktu lama	5	5	5
Saya tidak perlu banyak mencoba-coba untuk menemukan fitur yang saya butuhkan.	5	4	5
Fitur laporan harian membantu saya memantau pendapatan warung.	5	4	5
Aplikasi membantu saya menghindari kesalahan pencatatan stok dan harga.	5	4	5
Saya merasa aplikasi ini mendukung operasional warung secara menyeluruh.	5	4	5
Saya dapat menggunakan aplikasi tanpa bantuan orang lain.	5	4	5
Saya merasa tidak perlu pelatihan khusus untuk	5	5	5

menggunakan aplikasi ini secara penuh.			
Aplikasi tetap mudah digunakan meskipun saya baru pertama kali memakainya.	5	5	5
Skor Rata-Rata	5	4.6	5

Hasil pengujian menunjukkan bahwa seluruh aspek usability memperoleh **skor yang sangat tinggi**, dengan mayoritas responden memberikan nilai **5 (Sangat Setuju)** untuk setiap pernyataan. Hanya pada beberapa aspek seperti *kemudahan membaca ikon, fitur laporan harian, dan dukungan terhadap operasional warung*, satu responden memberikan skor **4**, yang tetap berada dalam kategori *Setuju*.

Skor rata-rata keseluruhan adalah  $\geq 4,9$ , yang menandakan bahwa aplikasi **Warung Digital**:

- Mudah digunakan bahkan oleh pengguna baru tanpa pelatihan teknis.
- Menyediakan antarmuka yang nyaman, intuitif, dan informatif.
- Menyediakan fitur-fitur yang sesuai dan bermanfaat bagi pengelolaan operasional warung.
- Mendukung efisiensi kerja dan mengurangi potensi kesalahan pencatatan.

Dengan demikian, aplikasi dinyatakan **memiliki usability yang sangat baik**, dan

siap digunakan oleh pengguna warung dengan latar belakang teknologi yang beragam.

### Pengujian Lapangan Pengguna Aplikasi

Pengujian lapangan dilakukan untuk membandingkan efektivitas dan akurasi pencatatan transaksi antara metode konvensional (manual) dan metode digital menggunakan aplikasi **Warung Digital**. Pengujian berlangsung selama enam hari berturut-turut di warung sembako milik keluarga penulis, dengan skenario sebagai berikut:

- Hari 1–3: Pencatatan dilakukan secara manual di buku tulis.
- Hari 4–6: Pencatatan menggunakan aplikasi Warung Digital.

Indikator yang diamati meliputi:

1. Jumlah transaksi per hari
2. Rata-rata waktu per transaksi
3. Jumlah produk terjual
4. Total nilai penjualan
5. Tingkat kesalahan input (error rate)

Tabel 5 Perbandingan Efektivitas Metode Konvensional vs Digital

Hari	Metode	Transaksi	Rata-rata Waktu (mnt)	Produk Terjual	Total Penjualan
1–3	Konvensional	24–27	4,47	52–60	Rp 670.000–745.000
4–6	Digital	31–34	1,77	70–75	Rp 920.000–980.000

Penggunaan aplikasi **mempercepat proses**

pencatatan dengan penurunan rata-rata waktu transaksi dari **4,47 menit** menjadi **1,77 menit**, tanpa mengubah kondisi operasional dasar warung.

## PENUTUP

### Kesimpulan

Penelitian ini berhasil menghasilkan sebuah aplikasi **Warung Digital** berbasis mobile yang dirancang untuk mempermudah pencatatan transaksi, pengelolaan stok, dan pelaporan pendapatan pada warung sembako. Pengembangan dilakukan menggunakan Flutter (frontend), Golang (backend), dan PostgreSQL (basis data), dengan pendekatan SDLC model waterfall.

Berdasarkan hasil implementasi dan pengujian, dapat disimpulkan bahwa:

- Aplikasi mampu menyederhanakan proses pencatatan produk dan transaksi secara signifikan dibanding metode konvensional.
- Rata-rata waktu pencatatan transaksi menurun dari 4,47 menit menjadi 1,77 menit setelah menggunakan aplikasi.
- Tingkat kesalahan pencatatan (error rate) juga menurun drastis dari 13,93% menjadi 0,97% melalui validasi otomatis terhadap stok dan harga.

- Hasil pengujian *usability* menunjukkan tingkat kepuasan yang sangat tinggi, dengan skor rata-rata  $\geq 4,9$  dari skala Likert, membuktikan bahwa aplikasi mudah digunakan, sesuai kebutuhan, dan tidak memerlukan pelatihan khusus.

Dengan demikian, aplikasi **Warung Digital** terbukti efektif dalam meningkatkan efisiensi dan akurasi pengelolaan operasional warung sembako, serta layak digunakan oleh pelaku usaha mikro non-teknis.

### Saran

Beberapa pengembangan lanjutan yang direkomendasikan untuk meningkatkan manfaat dan cakupan aplikasi adalah sebagai berikut:

1. **Integrasi Sistem Pembayaran Digital**  
Penambahan fitur integrasi dengan layanan pembayaran seperti QRIS dapat memperluas fungsi aplikasi dan meningkatkan kepraktisan bagi pemilik warung maupun pelanggan.
2. **Manajemen Piutang dan Utang Pelanggan**  
Fitur pencatatan utang dan sistem pengingat pembayaran akan sangat berguna dalam operasional warung yang sering melayani sistem bayar kemudian.

### **3. Multi-Tenant dan Multi-User**

Perlu dikembangkan arsitektur multi-tenant agar satu aplikasi dapat digunakan oleh banyak warung secara terpisah. Fitur multi-user juga memungkinkan pembagian peran antara pemilik dan karyawan.

### **4. Pengolahan Data dan Laporan Bisnis**

#### **Cerdas (Smart Reporting)**

Integrasi fitur analisis penjualan, rekomendasi restock otomatis, atau pengingat stok minimum dapat membantu pengambilan keputusan yang lebih baik oleh pemilik warung.

Penelitian ini merupakan langkah awal dari rangkaian pengembangan sistem digitalisasi UMKM skala mikro. Diperlukan penelitian lanjutan yang lebih luas untuk menguji efektivitas sistem ini dalam konteks pasar yang lebih beragam dan dengan jumlah pengguna yang lebih besar.

## **DAFTAR PUSTAKA**

Abdillah, R. (2021). Pemodelan UML untuk sistem informasi persewaan alat pesta. *Jurnal Fasilkom*, 11(2).  
<https://ejurnal.umri.ac.id/index.php/JIK/article/view/2673>

Binanto, I. (2013). *Perbandingan metode pengembangan perangkat lunak multimedia*. Seminar Nasional RITEKTRA 2013, Universitas Atma Jaya Jakarta. Diakses dari [https://www.researchgate.net/publication/264496799\\_Perbandingan\\_Metode\\_Pengembangan\\_Perangkat\\_Lunak\\_Multimedia](https://www.researchgate.net/publication/264496799_Perbandingan_Metode_Pengembangan_Perangkat_Lunak_Multimedia)

#### **ngan Perangkat Lunak Multimedia**

Arifin, M. N., & Siahaan, D. (2020). Structural and semantic similarity measurement of UML use case diagram. *Lontar Komputer: Jurnal Ilmiah Teknologi Informasi*, 11(2), 88. Diakses dari <https://pdfs.semanticscholar.org/b4d0/43c1f704435b73c3e74ac83077653fabff2a.pdf>

Bhagat, S. A. (2022). Review on mobile application development based on Flutter platform. *International Journal for Research in Applied Science and Engineering Technology (IJRASET)*, 10(1), 1416–1420. Diakses dari <https://www.ijraset.com/best-journal/review-on-mobile-application-development-based-on-flutter-platform>

Chandra, A. K., Rahayudi, B., & Purnomo, W. (2024). Perancangan dan pembuatan aplikasi point of sales UMKM berbasis mobile menggunakan framework Flutter (Studi kasus: PW Café). *J-PTIICK*, 8(2), 1185–1192. <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/13239>

Sumarto, M. A. (2023). Analisis dan perancangan aplikasi point of sale berbasis mobile menggunakan framework Flutter dengan metode Rapid Application Development (RAD). *Jurnal Studi Komunikasi dan Media*, 27(1). Diambil dari <https://jkd.komdigi.go.id/index.php/jskm/article/view/5115>

Yunus, C. D. P., & Ulum, M. B. (2023). Pengembangan aplikasi penjadwalan konten Instagram otomatis bagi pelaku UMKM dengan Flutter framework. *Jurnal Ilmiah Informatika*, 11(2), 196–205. [https://forum.upbatam.ac.id/index.php/jif/article/view/8038?utm\\_source=chatgpt.com](https://forum.upbatam.ac.id/index.php/jif/article/view/8038?utm_source=chatgpt.com)

Prasetyo, G. W. T., Pradana, F., & Prakoso, B. S. (2022). Pengembangan aplikasi Point of Sales Warung dan UMKM “WarunkQu” menggunakan framework Flutter. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 6(10), 4724–4730. Diambil

- dari <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/11665>
- Choirudin, R., Adil, A., Studi, P., & Informatika, T. (2019). Implementasi REST API web service dalam membangun aplikasi. *Jurnal Teknologi Informasi*, 18(2), 284–293.  
<https://www.researchgate.net/publication/334541718>
- LEMKE, G., 2018. The software development life cycle and its application.
- Fithry Tahel, & Ginting, E. (2019). Perancangan aplikasi media pembelajaran pengenalan pahlawan nasional untuk meningkatkan rasa nasionalis berbasis Android. *Teknomatika*, 9(2), 113–120.  
<https://ojs.palcomtech.ac.id/index.php/teknomatika/article/view/467>
- Prayoga, R. R., Munawar, G., Jumiyani, R., & Syalsabila, A. (2021). Performance Analysis of BLoC and Provider State Management Library on Flutter. *Jurnal Mantik*, 5(3), 1591–1597.
- Swathiga, U. U. A. S., Vinodhini, P., & Sasikala, V. (2021). An Interpretation of Dart Programming Language. *DRSR Journal*, 11(3), 144–149.
- Kementerian Koordinator Bidang Perekonomian Republik Indonesia. (2025). Pemerintah dorong UMKM naik kelas, tingkatkan kontribusi terhadap ekspor Indonesia.  
<https://www.ekon.go.id/publikasi/detail/6152/pemerintah-dorong-umkm-naik-kelas-tingkatkan-kontribusi-terhadap-ekspor-indonesia>
- Kurniawan, I., & Rozi, F. (2020). REST API menggunakan NodeJS pada aplikasi transaksi jasa elektronik berbasis Android. *JITSI: Jurnal Ilmiah Teknologi Sistem Informasi*, 1(4), 127–132.  
<https://doi.org/10.62527/jitsi.1.4.18>
- Luhur Pambudi, P. D. (2024). Cross-platform development dengan Flutter dan React Native: Solusi efektif untuk pengembangan aplikasi modern. *BINUS Online*.  
<http://online.binus.ac.id/computer-science/2024/11/15/cross-platform-development-dengan-Flutter-dan-react-native-solusi-efektif-untuk-pengembangan-aplikasi-modern/>
- Priyatna, A. I., Arfandy, H., & Surasa, H. (2021). Pengembangan Servio menggunakan full REST API untuk mendukung layanan multiplatform. *KHARISMA Tech*, 2(1), 15–22.  
<https://jurnal.kharisma.ac.id/kharismatech/article/view/108>
- Ranga, V., & Soni, A. (2019). API features individualizing of web services: REST and SOAP. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, August.  
<https://doi.org/10.35940/ijitee.I1107.0789S19>
- Rizki, K., Adil, A., Pendahuluan, I., Barang, M., Motor, S., & Penumpang, M. (2018). Implementasi Google Maps API berbasis Android untuk lokasi fasilitas umum di Kabupaten Sumbawa. *Matrik: Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, 17(2).  
<https://repository.universitasbumigora.ac.id/d/eprint/2259>
- Bourhis, P., Reutter, J. L., Suárez, F., & Vrgoč, D. (2020). JSON: Data model and query languages. *Information Systems*, 89, 101482.  
<https://www.sciencedirect.com/science/article/pii/S0306437919305344?via%3Dhub>
- Nurseitov, N., Paulson, M., Reynolds, R., & Izurieta, C. (2009). Comparison of JSON and XML data interchange formats: A case study. *Caine 2009, International Conference on Computers, Software and Applications*.  
<https://www.cs.montana.edu/izurieta/pubs/IzurietaCAINE2009.pdf>
- Setiawan, R. (2017). *Pengujian perangkat lunak*. FASILKOM Universitas Indonesia.
- Widyastuti, R. (2020). Rancang bangun sistem

informasi inventory boneka berbasis web.

*PROSISKO: Jurnal Pengembangan Riset  
dan Observasi Sistem Komputer*, 7(2).

[https://e-](https://e-jurnal.lppmunsera.org/index.php/PROSISKO/article/view/2316)

[jurnal.lppmunsera.org/index.php/PROSISKO/article/view/2316](https://e-jurnal.lppmunsera.org/index.php/PROSISKO/article/view/2316)