

Laporan Akhir Proyek Basis Data

Sistem Manajemen Inventori dan Penjualan untuk Usaha Kecil



Kelompok 4:

Thufail Bahir Al Bariq (24/537843/PA/22801)

Dhimas Putra Sulistio (24/537952/PA/22811)

Maulana Faris Al Ghifari (24/544029/PA/23119)

BAB I PENDAHULUAN

1.1 Latar Belakang

Pengelolaan inventori merupakan elemen vital dalam menjaga kelancaran operasional berbagai organisasi. Menurut Jaya dkk. (2022), sistem manajemen yang tidak terstruktur berisiko menyebabkan masalah serius seperti ketidakakuratan data stok, pemborosan anggaran, hingga hilangnya aset organisasi. Hal ini sering terjadi karena masih banyak pelaku usaha yang mengandalkan proses pencatatan konvensional yang rentan terhadap kesalahan manusia (*human error*).

Sejalan dengan tantangan tersebut, transformasi digital melalui pengembangan sistem informasi berbasis web menjadi kebutuhan mendesak bagi usaha mikro, kecil, dan menengah (UMKM). Setiawan (2020) dalam penelitiannya menekankan bahwa implementasi sistem informasi yang terintegrasi mampu meningkatkan akurasi data transaksi dan mempermudah akses informasi secara *real-time*, yang pada akhirnya memperkuat daya saing bisnis di pasar digital. Namun, kendala yang sering muncul pada platform digital saat ini adalah alur transaksi yang masih terfragmentasi, di mana data pelanggan dan data inventori belum terhubung secara efisien.

Oleh karena itu, proyek A*Star dikembangkan sebagai platform manajemen inventori dan penjualan yang mengedepankan efisiensi melalui integrasi data yang mendalam. Dengan memanfaatkan teknologi Next.js dan Supabase, A*Star tidak hanya berfungsi sebagai alat pencatatan stok, tetapi juga sebagai ekosistem penjualan yang memproteksi setiap transaksi pelanggan. Inovasi utama dalam sistem ini adalah penggunaan data profil pengguna secara otomatis untuk fitur pemesanan. Melalui integrasi Order to WhatsApp, informasi produk dan detail pengiriman pelanggan disinkronisasikan secara otomatis sehingga meminimalisir input manual yang berulang dan mempercepat proses konfirmasi pesanan oleh admin.

1.2 Rumusan Masalah

Usaha kecil dan menengah (UKM) sering menghadapi tantangan dalam mengelola operasional bisnis mereka, terutama dalam hal:

1. Pengelolaan Inventori yang Tidak Efisien
 - Kesulitan dalam melacak stok produk secara real-time
 - Tidak adanya sistem terintegrasi untuk memantau ketersediaan barang
 - Risiko terjadinya kelebihan atau kekurangan stok yang dapat merugikan bisnis
2. Pencatatan Transaksi yang Manual
 - Proses pencatatan penjualan masih dilakukan secara manual atau menggunakan spreadsheet
 - Rentan terhadap kesalahan input data (*human error*)
 - Sulit untuk melakukan analisis riwayat transaksi dan pola penjualan

3. Keterbatasan Akses Informasi Produk
 - Pelanggan tidak memiliki akses mudah untuk melihat katalog produk
 - Informasi produk yang tidak up-to-date
 - Tidak ada platform yang memudahkan pelanggan untuk melakukan pemesanan
4. Kurangnya Sistem Kontrol yang Terstruktur
 - Tidak ada perbedaan akses antara admin dan customer
 - Tidak ada mekanisme validasi dan verifikasi transaksi
 - Kesulitan dalam melakukan operasi CRUD (Create, Read, Update, Delete) terhadap data produk

Berdasarkan permasalahan tersebut, diperlukan sebuah sistem basis data yang terintegrasi untuk mengelola inventori dan transaksi penjualan secara efisien, akurat, dan terstruktur.

1.3 Tujuan Pengembangan Sistem

Pengembangan sistem A* (A Star) memiliki beberapa tujuan utama:

1.3.1 Tujuan Akademis

1. Implementasi Konsep Basis Data Relasional
 - Merancang dan mengimplementasikan basis data yang ter-normalisasi hingga Third Normal Form (3NF)
 - Menerapkan konsep Entity-Relationship Diagram (ERD) dalam desain database
 - Menggunakan constraints, foreign keys, dan indexing untuk menjaga integritas data
2. Penerapan Operasi CRUD
 - Mengimplementasikan operasi Create, Read, Update, dan Delete pada berbagai entitas
 - Memahami interaksi antara aplikasi dan database melalui query SQL
 - Menggunakan prepared statements dan parameterized queries untuk keamanan
3. Pembelajaran Teknologi Terkini
 - Menerapkan prinsip *learning by doing* dalam seluruh tahapan pengembangan
 - Mengintegrasikan berbagai teknologi (Go, Flutter, Next.js, PostgreSQL, Supabase)
 - Memahami arsitektur full-stack development (backend, mobile, web)

1.3.2 Tujuan Fungsional

1. Sistem Manajemen Inventori
 - Menyediakan platform untuk monitoring stok produk secara real-time
 - Memudahkan admin dalam mengelola data produk (create, update, delete)
 - Otomatis mengurangi stok ketika transaksi berhasil diproses

2. Sistem Penjualan Terintegrasi
 - Menyediakan katalog produk yang dapat diakses oleh pelanggan melalui web
 - Memfasilitasi proses pemesanan dengan status tracking (pending, process, done, cancelled)
 - Mengelola data transaksi dengan informasi lengkap (customer, produk, jumlah, total)
3. Multi-Platform Access
 - Mobile App (Flutter): Untuk admin mengelola inventori dan memproses pesanan
 - Web App (Next.js): Untuk customer melihat katalog dan melakukan pemesanan
 - Backend API (Go): Sebagai penghubung antara aplikasi dan database
4. Keamanan dan Kontrol Akses
 - Implementasi autentikasi menggunakan Supabase Auth
 - Row Level Security (RLS) untuk membatasi akses data
 - Implementasi middleware pada Go backend API untuk memperkuat otorisasi dilengkapi dengan JSON Web Token (JWT) secret key dari Supabase

1.4 Batasan Masalah

1.4.1 Batasan Ruang Lingkup Fungsional

1. Manajemen Produk dan Inventori
 - Sistem hanya menangani produk fisik tunggal (tidak ada variasi produk seperti ukuran/warna)
 - Tidak mencakup sistem supplier
 - Tidak ada fitur forecasting atau prediksi stok otomatis
2. Sistem Transaksi dan Pemesanan
 - Setiap pesanan (order) hanya dapat berisi satu jenis produk dengan kuantitas tertentu
 - Tidak ada implementasi shopping cart untuk pembelian multiple produk sekaligus
 - Payment gateway tidak terintegrasi (status pembayaran dikelola manual oleh admin)
 - Tidak tersedia fitur retur, refund, atau komplain (kecuali melalui WA admin)
 - Tidak ada sistem tracking pengiriman atau logistics
3. Manajemen Pengguna
 - Hanya terdapat dua role: admin (pengelola sistem) dan user (customer)
 - Tidak ada fitur multi-level authorization atau custom permissions

1.4.2 Batasan Kinerja dan Skalabilitas

1. Kapasitas Sistem
 - Dirancang untuk skala usaha kecil hingga menengah (bukan enterprise)
 - Estimasi beban: < 1000 transaksi per hari
 - Concurrent users: maksimal ~100 user simultan
2. Performance
 - Tidak ada optimasi khusus untuk high-traffic scenarios
 - Response time tidak dijamin di bawah threshold tertentu
 - Tidak ada load balancing atau horizontal scaling implementation
3. Security
 - Mengandalkan Supabase Auth untuk autentikasi
 - JWT-based authorization untuk admin endpoints
 - Row Level Security (RLS) untuk data isolation
 - Tidak ada penetration testing atau security audit formal
 - HTTPS/TLS menggunakan konfigurasi default Vercel dan Supabase

1.5 Metodologi Pengembangan

Pengembangan sistem A* menggunakan pendekatan Agile Development dengan model Iterative and Incremental, di mana setiap minggu menghasilkan increment yang dapat diuji dan dievaluasi. Metodologi ini dipilih karena sesuai dengan karakteristik proyek akademik yang memiliki timeline terbatas namun memerlukan pembelajaran mendalam.

1.5.1 Pendekatan Metodologi

A. Learning by Doing

Prinsip pembelajaran langsung diterapkan di setiap tahap:

- Eksplorasi teknologi baru (Go, Flutter, Next.js, Supabase)
- Trial and error dalam implementasi
- Dokumentasi proses pembelajaran di setiap sprint
- Refleksi dan improvement berkelanjutan

1.5.2 Tools dan Teknologi

Bagian ini menjelaskan tools pengembangan serta teknologi yang digunakan dalam pembangunan sistem, mencakup proses pengembangan, pengujian, hingga deployment aplikasi.

Development Tools

Dalam proses pengembangan sistem, digunakan beberapa tools pendukung untuk memastikan efisiensi, kolaborasi tim, serta kualitas perangkat lunak, yaitu:

- Version Control
Git dan GitHub digunakan untuk mengelola versi kode sumber, mendukung kolaborasi tim, serta mempermudah proses review dan integrasi kode.
- Code Editor dan IDE

Visual Studio Code dan Android Studio digunakan sebagai lingkungan pengembangan utama untuk penulisan, debugging, dan pengelolaan kode aplikasi.

- API Testing
Postman dan Thunder Client dimanfaatkan untuk melakukan pengujian endpoint API, memastikan validitas request dan response sesuai dengan spesifikasi sistem.
- Database Client
Supabase Dashboard digunakan untuk mengelola basis data PostgreSQL, melakukan query, monitoring data, dan administrasi skema database.
- Design Tools
Figma digunakan untuk perancangan antarmuka pengguna, sedangkan draw.io digunakan untuk membuat diagram sistem, arsitektur, dan alur proses.

Tech Stack Summary

Teknologi yang digunakan dalam sistem ini disusun berdasarkan lapisan arsitektur sebagai berikut:

- Backend
Backend dikembangkan menggunakan Go versi 1.21 atau lebih tinggi, dengan Chi Router sebagai HTTP router, SQLC untuk manajemen query berbasis tipe aman, serta pgx/v5 sebagai PostgreSQL driver. Backend berfungsi sebagai RESTful API server.
- Database
Sistem menggunakan PostgreSQL versi 15 atau lebih tinggi yang disediakan melalui Supabase sebagai media penyimpanan data utama.
- Authentication
Proses autentikasi dan otorisasi pengguna menggunakan Supabase Auth yang dikombinasikan dengan JSON Web Token (JWT).
- Mobile Application
Aplikasi mobile untuk admin dikembangkan menggunakan Flutter versi 3.x dengan bahasa pemrograman Dart.
- Web Application
Aplikasi web untuk pengguna dikembangkan menggunakan Next.js versi 14, TypeScript, dan Tailwind CSS untuk antarmuka pengguna.
- Deployment
Aplikasi backend dan web dideploy menggunakan Vercel, sedangkan aplikasi mobile didistribusikan dalam bentuk file APK.

BAB II ANALISIS DAN PERANCANGAN SISTEM

2.1 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem dilakukan untuk memahami secara detail apa saja yang dibutuhkan oleh sistem A* agar dapat berfungsi dengan baik sesuai dengan tujuan pengembangan. Kebutuhan sistem dibagi menjadi dua kategori utama: **Kebutuhan Fungsional** dan **Kebutuhan Non-Fungsional**.

2.1.1 Kebutuhan Fungsional

Kebutuhan fungsional menjelaskan fungsi utama yang harus disediakan oleh sistem agar dapat beroperasi sesuai tujuan. Kebutuhan ini diklasifikasikan berdasarkan aktor pengguna sistem.

A. Kebutuhan Fungsional Admin

Autentikasi Admin

Elemen	Deskripsi
Deskripsi	Sistem menyediakan mekanisme login khusus untuk admin
Input	Email dan password
Proses	Validasi kredensial melalui Supabase Auth dan verifikasi role admin
Output	Access token (JWT) untuk sesi terautentikasi
Acceptance Criteria	- Login berhasil dengan kredensial valid- Login ditolak jika kredensial salah- Sesi aktif hingga logout atau token kedaluwarsa

Manajemen Produk: Melihat Daftar Produk

Elemen	Deskripsi
Deskripsi	Admin dapat melihat seluruh produk dalam sistem
Input	Request ke endpoint GET /api/admin/products
Proses	Query database untuk mengambil seluruh data produk dan stok
Output	Daftar produk berisi ID, nama, kategori, deskripsi, harga, stok, dan gambar
Acceptance Criteria	- Semua produk ditampilkan tanpa filter stok- Data ditampilkan secara real-time- Tersedia fitur real-time data (tidak perlu refresh manual)

Manajemen Produk: Menambah Produk

Elemen	Deskripsi
Deskripsi	Admin dapat menambahkan produk baru
Input	Nama produk, kategori, deskripsi, harga, gambar, dan stok. Request ke endpoint POST /api/admin/products
Proses	Validasi input dan insert data ke tabel products
Output	Produk baru dan tampilan real-time pada dashboard manage admin
Acceptance Criteria	- Seluruh field wajib diisi- Harga dan stok bernilai ≥ 0 - Tipe gambar harus .webp- Sistem menampilkan status sukses atau gagal

Manajemen Produk: Memperbarui Produk

Elemen	Deskripsi
Deskripsi	Admin dapat memperbarui data produk
Input	Product ID dan data yang diperbarui. Request ke endpoint PUT /api/admin/products/{id}
Proses	Validasi ID produk dan update data di database
Output	Konfirmasi keberhasilan update
Acceptance Criteria	- Produk harus terdaftar- Seluruh field dapat diperbarui- Timestamp updated_at terbarui otomatis

Manajemen Produk: Menghapus Produk

Elemen	Deskripsi
Deskripsi	Admin dapat menghapus produk dari sistem
Input	Product ID. Request ke endpoint DELETE /api/admin/products/{id}
Proses	Pemeriksaan relasi data dan proses penghapusan
Output	Konfirmasi berhasil atau pesan kesalahan
Acceptance Criteria	- Sistem meminta konfirmasi sebelum penghapusan- Penghapusan bersifat permanen

Manajemen Pesanan: Melihat Pesanan

Elemen	Deskripsi
Deskripsi	Admin dapat melihat daftar pesanan berstatus pending dan process

Input	Request ke endpoint GET /api/admin/orders dengan status “pending” dan “process”
Proses	Query pesanan dengan relasi user dan produk
Output	Daftar pesanan pending dan process
Acceptance Criteria	- Hanya pesanan pending dan process yang ditampilkan- Urutan berdasarkan tanggal terlama- Informasi customer dan produk lengkap

Manajemen Pesanan: Membuat Pesanan

Elemen	Deskripsi
Deskripsi	Admin dapat membuat pesanan yang didapat dari WA customer
Input	Request ke endpoint POST /api/admin/orders dengan status “pending” dan “process”
Proses	Query pesanan dengan relasi user dan produk
Output	Daftar pesanan pending dan process
Acceptance Criteria	- Hanya pesanan pending dan process yang ditampilkan- Urutan berdasarkan tanggal terlama- Informasi customer dan produk lengkap

Manajemen Pesanan: Memperbarui Status Pesanan

Elemen	Deskripsi
Deskripsi	Admin dapat mengubah status pesanan
Input	Order ID dan status baru. Request ke endpoint PUT /api/admin/orders/{id}/status
Proses	Update status dan penyesuaian stok jika diperlukan
Output	Konfirmasi perubahan status
Acceptance Criteria	- Status hanya dapat diubah dari “pending” ke “process” ke “done”/“cancelled”- Stok berkurang saat status “done”- Proses berjalan dalam satu transaksi atomik

B. Kebutuhan Fungsional Customer

Registrasi Customer

Elemen	Deskripsi
Deskripsi	Customer dapat membuat akun baru

Input	Email, password, username, nama lengkap, nomor telepon, alamat. Request ke endpoint POST /api/auth/register
Proses	Validasi data dan pembuatan akun melalui Supabase Auth
Output	Akun customer berhasil dibuat
Acceptance Criteria	- Email dan username unik- Password minimal 8 karakter- Seluruh field wajib diisi

Melihat Katalog Produk

Elemen	Deskripsi
Deskripsi	Customer dapat melihat produk yang tersedia
Input	Request ke endpoint GET /api/products
Proses	Query produk dengan stok tersedia
Output	Daftar produk dalam bentuk grid
Acceptance Criteria	- Hanya produk dengan stok > 0- Data diperbarui real-time- Urutan produk terbaru

Melihat Detail Produk

Elemen	Deskripsi
Deskripsi	Customer dapat melihat detail produk
Input	Product ID. Request ke endpoint GET /api/products/{id}
Proses	Query detail produk
Output	Informasi lengkap produk
Acceptance Criteria	- Informasi produk lengkap- Status stok ditampilkan- Gambar memiliki loading state

Filter Produk Berdasarkan Kategori

Elemen	Deskripsi
Deskripsi	Customer dapat memfilter produk berdasarkan kategori
Input	Nama kategori
Proses	Filtering data di sisi klien
Output	Produk sesuai kategori

Acceptance Criteria	- Filter tanpa reload halaman- Tersedia opsi semua kategori- Jumlah produk per kategori terlihat
---------------------	--

C. Kebutuhan Fungsional Umum

Otorisasi dan Kontrol Akses

Sistem membedakan hak akses antara admin dan customer menggunakan JWT dan kebijakan keamanan database.

Validasi Data

Seluruh input divalidasi di sisi klien dan server untuk menjaga konsistensi dan keamanan data.

Penanganan Error

Sistem menangani error secara terstruktur dengan kode status HTTP dan pesan yang informatif tanpa mengekspos data sensitif.

2.1.2 Kebutuhan Non-Fungsional

A. Kinerja (Performance)

Sistem harus memiliki waktu respons cepat, mampu menangani pengguna bersamaan, dan menjalankan query database secara efisien.

B. Keamanan (Security)

Sistem menerapkan autentikasi aman, kontrol akses berbasis peran, serta perlindungan terhadap kebocoran data dan serangan injeksi.

C. Kemudahan Penggunaan (Usability)

Antarmuka dirancang intuitif, responsif, konsisten antar platform, serta memenuhi standar aksesibilitas.

D. Keandalan (Reliability)

Sistem menjamin ketersediaan tinggi, integritas data, serta kemampuan pemulihan dari kegagalan.

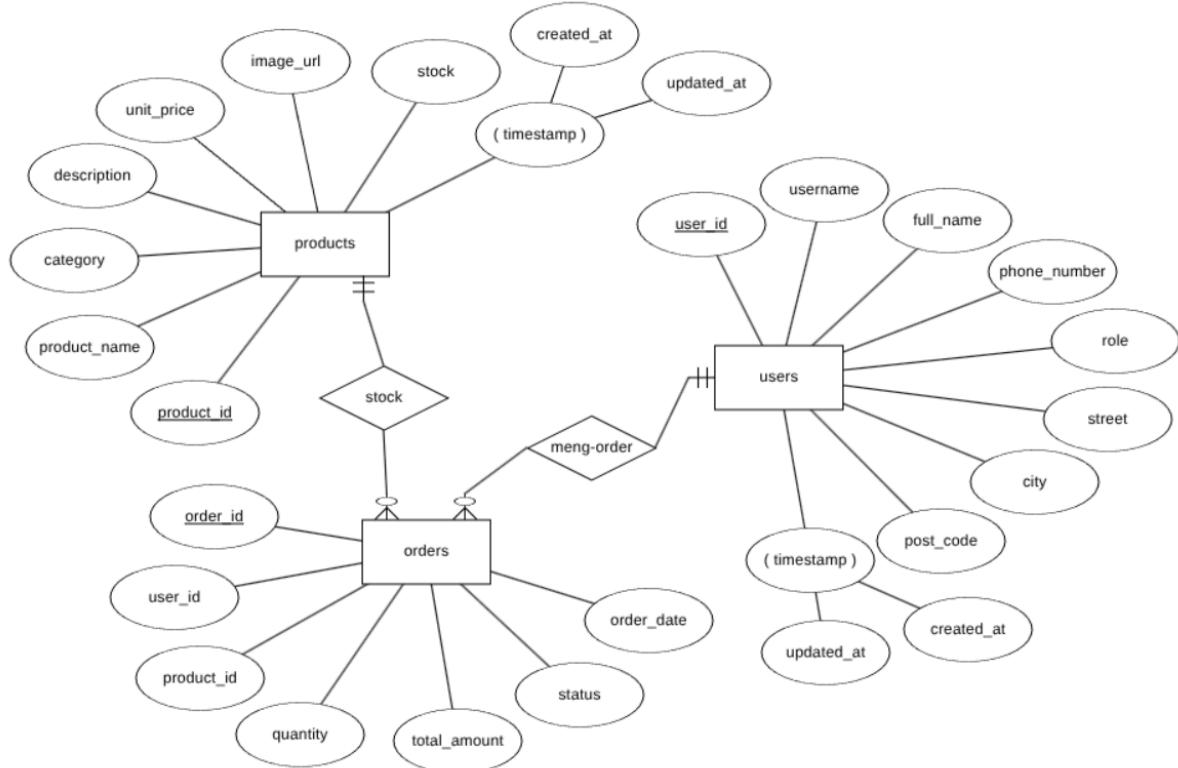
E. Kemudahan Pemeliharaan (Maintainability)

Kode sumber terstruktur, terdokumentasi, menggunakan version control, serta mudah dikembangkan di masa depan.

2.2 Perancangan Basis Data

2.2.1 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) digunakan untuk menggambarkan struktur data serta relasi antar entitas yang terdapat dalam sistem A*. Perancangan ERD pada sistem ini mengalami penyesuaian dari desain konseptual awal hingga mencapai bentuk implementasi final yang sesuai dengan kebutuhan fungsional dan teknis sistem.



A. Identifikasi Entitas

Berdasarkan analisis kebutuhan sistem, terdapat tiga entitas utama yang membentuk basis data sistem A*, yaitu Users, Products, dan Orders.

1. Entitas Users

Entitas Users merepresentasikan seluruh pengguna yang terdaftar dalam sistem, baik sebagai admin maupun customer.

- Atribut:
 - `user_id` (Primary Key, UUID): Identitas unik pengguna yang terintegrasi dengan Supabase Auth
 - `username` (VARCHAR 50, UNIQUE): Nama pengguna untuk keperluan login dan tampilan
 - `full_name` (VARCHAR 100): Nama lengkap pengguna
 - `phone_number` (VARCHAR 20): Nomor telepon pengguna

- role (VARCHAR 20): Peran pengguna dalam sistem, yaitu admin atau user
 - street (VARCHAR 100): Alamat jalan
 - city (VARCHAR 50): Kota
 - post_code (VARCHAR 10): Kode pos
 - created_at (TIMESTAMP): Waktu registrasi pengguna
 - updated_at (TIMESTAMP): Waktu pembaruan terakhir data pengguna
- Aturan Bisnis:
 - Username bersifat unik dan tidak boleh duplikat
 - Role default pengguna adalah user
 - Akun admin ditambahkan secara manual melalui database

2. Entitas Products

Entitas Products merepresentasikan seluruh produk yang dijual dalam sistem.

- Atribut:
 - product_id (Primary Key, SERIAL): Identitas unik produk
 - product_name (VARCHAR 100): Nama produk
 - category (VARCHAR 50): Kategori produk
 - description (TEXT): Deskripsi lengkap produk
 - unit_price (DECIMAL 10,2): Harga satuan produk
 - image_url (TEXT): URL gambar produk
 - stock (INT): Jumlah stok produk yang tersedia
 - created_at (TIMESTAMP): Waktu produk ditambahkan
 - updated_at (TIMESTAMP): Waktu pembaruan terakhir data produk
- Aturan Bisnis:
 - Harga produk harus bernilai nol atau lebih
 - Stok produk tidak boleh bernilai negatif
 - Nama produk wajib diisi

3. Entitas Orders

Entitas Orders merepresentasikan transaksi pemesanan produk yang dilakukan oleh customer.

- Atribut:
 - order_id (Primary Key, SERIAL): Identitas unik pesanan
 - user_id (Foreign Key, UUID): Referensi ke entitas Users
 - product_id (Foreign Key, INT): Referensi ke entitas Products
 - quantity (INT): Jumlah produk yang dipesan
 - total_amount (DECIMAL 12,2): Total harga pesanan
 - status (VARCHAR 20): Status pesanan, meliputi pending, paid, dan cancelled
 - order_date (TIMESTAMP): Waktu pesanan dibuat
- Aturan Bisnis:
 - Quantity harus lebih besar dari nol
 - Total amount tidak boleh bernilai negatif

- Status default pesanan adalah pending
- Setiap pesanan hanya merepresentasikan satu jenis produk

2.2.2 Logical Schema

Logical schema menjelaskan struktur basis data sistem A* dalam bentuk tabel, atribut, tipe data, serta constraint yang diterapkan.

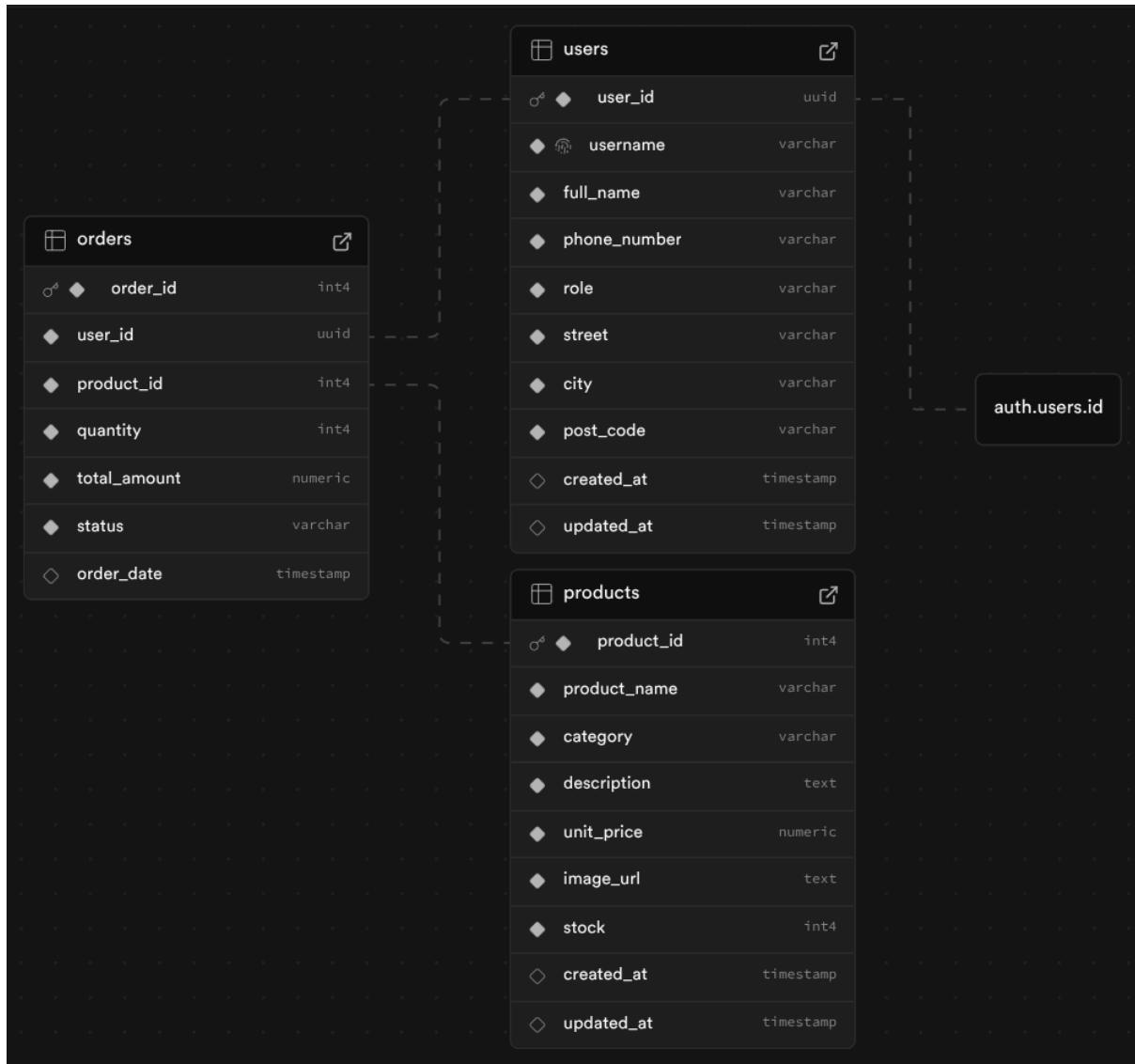


Table Users

Tabel users menyimpan data seluruh pengguna sistem, baik admin maupun customer, dan terintegrasi dengan Supabase Auth.

Atribut	Tipe Data	Constraints	Deskripsi
---------	-----------	-------------	-----------

user_id	UUID	Primary Key, Foreign Key ke auth.users(id), ON DELETE CASCADE	Identitas unik pengguna dari Supabase Auth
username	VARCHAR(50)	NOT NULL, UNIQUE	Username untuk identifikasi pengguna
full_name	VARCHAR(100)	NOT NULL	Nama lengkap pengguna
phone_number	VARCHAR(20)	NOT NULL	Nomor telepon pengguna
role	VARCHAR(20)	NOT NULL, DEFAULT 'user', CHECK (role IN ('user', 'admin'))	Peran akses pengguna
street	VARCHAR(100)	NOT NULL	Alamat jalan pengguna
city	VARCHAR(50)	NOT NULL	Kota tempat tinggal
post_code	VARCHAR(10)	NOT NULL	Kode pos
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Waktu pendaftaran akun
updated_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Waktu pembaruan terakhir data

Index yang diterapkan:

- Primary key index pada kolom user_id
- Unique index pada kolom username
- Index tambahan pada username untuk mempercepat pencarian berdasarkan username

Row Level Security (RLS):

- Pengguna hanya dapat melihat data miliknya sendiri
- Admin memiliki akses untuk melihat seluruh data pengguna

Tabel Products

Tabel products menyimpan informasi seluruh produk yang tersedia dalam sistem.

Atribut	Tipe Data	Constraints	Deskripsi
product_id	SERIAL	Primary Key	Identitas unik produk

product_name	VARCHAR(100)	NOT NULL	Nama produk
category	VARCHAR(50)	NOT NULL	Kategori produk
description	TEXT	NOT NULL	Deskripsi lengkap produk
unit_price	DECIMAL(10, 2)	NOT NULL, CHECK (unit_price >= 0)	Harga satuan produk
image_url	TEXT	NOT NULL	URL gambar produk
stock	INTEGER	NOT NULL, DEFAULT 0, CHECK (stock >= 0)	Jumlah stok tersedia
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Waktu produk ditambahkan
updated_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Waktu pembaruan terakhir data produk

Index yang diterapkan:

- Primary key index pada kolom product_id
- Index pada kolom category untuk mendukung filter produk berdasarkan kategori
- Index pada kolom stock untuk mendukung query produk yang tersedia

Row Level Security (RLS):

- Seluruh pengguna dapat melihat data produk
- Hanya admin yang dapat melakukan operasi insert, update, dan delete

Tabel Orders

Tabel orders menyimpan data transaksi pemesanan produk yang dilakukan oleh pengguna.

Atribut	Tipe Data	Constraints	Deskripsi
order_id	SERIAL	Primary Key	Identitas unik pesanan
user_id	UUID	NOT NULL, Foreign Key ke users(user_id), ON DELETE CASCADE	Identitas pengguna pemesan
product_id	INTEGER	NOT NULL, Foreign Key ke products(product_id), ON DELETE RESTRICT	Identitas produk yang dipesan
quantity	INTEGER	NOT NULL, CHECK (quantity > 0)	Jumlah produk yang dipesan

total_amount	DECIMAL(12, 2)	NOT NULL, CHECK (total_amount >= 0)	Total nilai transaksi
status	VARCHAR(20)	NOT NULL, DEFAULT 'pending'	Status pesanan
order_date	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Waktu pemesanan dibuat

Index yang diterapkan:

- Primary key index pada kolom order_id
- Index pada kolom user_id untuk pencarian pesanan berdasarkan pengguna
- Index pada kolom status untuk filter pesanan berdasarkan status
- Composite index pada (user_id, status) untuk meningkatkan efisiensi query kombinasi

Row Level Security (RLS):

- Pengguna hanya dapat melihat pesanan miliknya sendiri
- Admin memiliki akses untuk melihat seluruh data pesanan

BAB III IMPLEMENTASI SISTEM

3.1 Implementasi Basis Data

Implementasi basis data merupakan tahapan realisasi dari desain yang telah dirancang pada BAB 2. Pada bagian ini dijelaskan langkah-langkah implementasi database sistem A* menggunakan PostgreSQL yang di-hosting pada platform Supabase.

3.1.1 Setup Lingkungan Database

Basis data sistem dibangun menggunakan PostgreSQL versi 15 yang di-host pada Supabase sebagai layanan Backend-as-a-Service. Supabase dipilih karena menyediakan integrasi autentikasi, keamanan, serta manajemen database yang terkelola dengan baik.

Konfigurasi utama database adalah sebagai berikut:

- DBMS: PostgreSQL 15.x
- Hosting: Supabase
- Region: Singapore
- Connection pooling: PgBouncer
- Keamanan koneksi: SSL/TLS aktif secara default

Database menggunakan koneksi pooled untuk kebutuhan aplikasi dan koneksi langsung untuk kebutuhan administrasi dan definisi skema.

3.1.2 Pembuatan Struktur Tabel

Struktur tabel diimplementasikan berdasarkan ERD dan logical schema yang telah dirancang sebelumnya. Terdapat tiga tabel utama, yaitu users, products, dan orders.

Tabel users berfungsi untuk menyimpan data pengguna sistem, baik customer maupun admin. Tabel ini terintegrasi langsung dengan Supabase Auth menggunakan UUID sebagai primary key. Atribut utama meliputi:

- Identitas pengguna
- Informasi kontak
- Role pengguna
- Informasi alamat
- Timestamp pembuatan dan pembaruan data

Tabel ini menerapkan constraint untuk memastikan keunikan username serta validasi role pengguna.

Tabel products digunakan untuk menyimpan data produk yang tersedia dalam sistem. Informasi yang disimpan mencakup nama produk, kategori, deskripsi, harga satuan, stok, dan gambar produk. Validasi harga dan stok diterapkan untuk mencegah nilai negatif, serta indeks dibuat untuk mendukung proses pencarian dan filtering produk.

Tabel orders menyimpan data transaksi pemesanan produk oleh customer. Tabel ini memiliki relasi ke tabel users dan products melalui foreign key. Setiap pesanan menyimpan informasi jumlah pembelian, total harga, status pesanan, dan waktu pemesanan. Relasi foreign key dikonfigurasi untuk menjaga integritas data antar tabel.

3.1.3 Penerapan Constraint dan Integritas Data

Untuk menjaga konsistensi dan validitas data, sistem menerapkan beberapa jenis constraint, antara lain:

- CHECK constraint untuk validasi nilai numerik seperti harga, stok, dan kuantitas
- FOREIGN KEY constraint untuk menjaga relasi antar tabel
- DEFAULT value untuk atribut tertentu seperti role dan status pesanan

Penerapan constraint ini memastikan bahwa data yang disimpan selalu sesuai dengan aturan bisnis sistem.

3.1.4 Implementasi Indexing

Index dibuat untuk meningkatkan performa query yang sering dijalankan oleh sistem. Selain primary key index yang dibuat secara otomatis oleh PostgreSQL, beberapa index tambahan diterapkan pada kolom:

- Username pengguna
- Kategori dan stok produk
- Relasi pesanan dengan pengguna dan status pesanan

Index komposit digunakan untuk mendukung query spesifik seperti pencarian pesanan berdasarkan pengguna dan status.

3.1.5 Implementasi Trigger dan Function

Trigger digunakan untuk mendukung otomasi proses tertentu di dalam database. Salah satu trigger yang diterapkan adalah penambahan otomatis user baru jika ada yang mendaftar:

```
SQL
BEGIN
  INSERT INTO public.users (
    user_id,
    username,
    full_name,
    phone_number,
    role,
    street,
    city,
    post_code
  )
  VALUES (
    new.id,
    COALESCE(new.raw_user_meta_data->>'username', ''),
    COALESCE(new.raw_user_meta_data->>'full_name', ''),
    COALESCE(new.raw_user_meta_data->>'phone_number', ''),
    'user', -- default role
    COALESCE(new.raw_user_meta_data->>'street', ''),
    COALESCE(new.raw_user_meta_data->>'city', ''),
    COALESCE(new.raw_user_meta_data->>'post_code', '')
  );
  RETURN new;
END;
```

Ada juga function untuk cek role admin:

```
SQL
BEGIN
  -- Cek apakah user yang request punya role admin
  -- Karena jalan sebagai superuser, query ini nembus RLS apapun.
  RETURN EXISTS (
    SELECT 1
    FROM public.users
    WHERE user_id = auth.uid()
    AND role = 'admin'
```

```
);  
END;
```

Kami juga menambahkan function yang fungsinya untuk cek role kemudian masukkan ke app_metadata:

```
SQL  
DECLARE  
claims jsonb;  
user_role text;  
BEGIN  
-- Ambil role dari tabel public.users  
SELECT role INTO user_role FROM public.users WHERE user_id =  
(event->>'user_id')::uuid;  
claims := event->'claims';  
-- Jika role ditemukan, masukkan ke app_metadata  
IF user_role IS NOT NULL THEN  
claims := jsonb_set(claims, '{app_metadata, role}', to_jsonb(user_role));  
ELSE  
claims := jsonb_set(claims, '{app_metadata, role}', '"user"');  
END IF;  
event := jsonb_set(event, '{claims}', claims);  
return event;  
END;
```

Selain itu, di Go backend juga ada function untuk mengurangi jumlah stok jika ada order yang statusnya “done”.

3.1.6 Penerapan Row Level Security (RLS)

Untuk menjaga keamanan data, sistem menerapkan Row Level Security pada seluruh tabel utama. Kebijakan RLS memastikan bahwa:

- Customer hanya dapat mengakses data miliknya sendiri
- Admin memiliki akses penuh terhadap data sistem
- Data produk dapat diakses secara publik untuk keperluan katalog

Penerapan RLS ini terintegrasi dengan sistem autentikasi Supabase.

3.1.7 Penyediaan Data Awal (Seeding)

Database diisi dengan data awal untuk mendukung proses pengembangan dan pengujian sistem. Data awal meliputi:

- Satu akun admin
- Beberapa data produk contoh

- Satu akun customer
- Contoh data pesanan

Seeding ini bertujuan untuk memastikan seluruh fitur sistem dapat diuji dengan kondisi data yang realistik.

3.1.8 Backup dan Pemeliharaan Database

Supabase menyediakan mekanisme backup otomatis harian untuk menjaga keamanan data. Backup disimpan sesuai kebijakan retensi layanan yang digunakan dan dapat dipulihkan melalui dashboard Supabase apabila diperlukan.

3.2 Implementasi SQL

Bagian ini menjelaskan implementasi query SQL yang digunakan dalam sistem A* untuk mendukung operasi Create, Read, Update, dan Delete pada data. Seluruh query disusun secara terstruktur berdasarkan kebutuhan endpoint dan dikelola menggunakan **SQLC** untuk menghasilkan kode Go yang bersifat type-safe, sehingga meningkatkan keamanan, konsistensi, dan maintainability sistem.

3.2.1 Query Publik (Storefront)

Query publik digunakan untuk kebutuhan aplikasi web pelanggan. Query publik mencakup:

- **List Available Products** digunakan untuk menampilkan daftar produk yang tersedia, yaitu produk dengan stok lebih dari nol. Data diurutkan berdasarkan waktu pembuatan secara menurun sehingga produk terbaru muncul di bagian atas. Query ini dioptimalkan dengan index pada kolom stok dan memiliki waktu eksekusi rata-rata di bawah 50 ms untuk skala 1.000 data produk.
- **Get Product Detail** digunakan untuk menampilkan detail lengkap suatu produk berdasarkan product_id. Selain informasi dasar, query ini juga mengambil deskripsi produk. Query memanfaatkan primary key index sehingga waktu eksekusi sangat cepat, umumnya di bawah 10 ms.

3.2.2 Query Admin untuk Manajemen Katalog

Query pada bagian ini hanya dapat diakses oleh admin dan digunakan untuk pengelolaan produk. Query admin mencakup:

- **List All Products (Admin)**. Berbeda dengan query publik, query ini menampilkan seluruh produk tanpa memandang ketersediaan stok. Pengurutan dilakukan berdasarkan product_id secara menaik untuk memudahkan pengelolaan inventaris.
- **Create Product**. Query ini digunakan untuk menambahkan produk baru ke dalam sistem. Proses insert memanfaatkan parameterized query dan mengembalikan product_id dari produk yang baru dibuat. Validasi dilakukan baik di sisi aplikasi maupun di tingkat database melalui constraint seperti NOT NULL dan CHECK.

- **Update Product.** Query ini digunakan untuk memperbarui informasi produk yang sudah ada, termasuk pembaruan stok dan harga. Kolom `updated_at` diperbarui secara otomatis untuk mencatat waktu perubahan terakhir.
- **Delete Product.** Query ini digunakan untuk menghapus produk dari katalog. Database menerapkan foreign key constraint dengan ON DELETE RESTRICT sehingga produk tidak dapat dihapus apabila masih direferensikan oleh data pesanan aktif.

3.2.3 Query Admin untuk Manajemen Pesanan

Query pada bagian ini digunakan untuk mengelola alur pemesanan dari sisi admin. Query admin yang digunakan untuk manajemen pesanan mencakup:

- **List Orders.** Query ini menampilkan daftar pesanan dengan status pending dan process. Data pesanan digabungkan dengan data pengguna dan produk untuk memberikan informasi yang lengkap kepada admin. Pengurutan dilakukan berdasarkan tanggal pesanan secara menaik untuk menerapkan prinsip FIFO.
- **Update Order Status.** Query ini digunakan untuk mengubah status pesanan, misalnya menjadi paid atau cancelled. Perubahan status ini menjadi pemicu logika bisnis lanjutan di tingkat aplikasi.
- **Get Order Quantity and Product.** Query pendukung ini digunakan untuk mengambil informasi `product_id` dan jumlah pesanan dari suatu order, yang kemudian digunakan dalam proses pengurangan stok.
- **Decrease Product Stock.** Query ini digunakan untuk mengurangi stok produk secara atomik setelah pesanan disetujui. PostgreSQL menjamin atomicity sehingga tidak terjadi race condition pada pembaruan stok.

Seluruh proses persetujuan pesanan dijalankan dalam satu transaksi database untuk menjamin sifat ACID, yaitu atomicity, consistency, isolation, dan durability.

3.2.4 Ringkasan Query SQL

Secara keseluruhan, sistem mengimplementasikan query publik untuk penelusuran katalog dan query admin untuk manajemen produk serta pesanan. Seluruh endpoint admin dilindungi oleh autentikasi dan otorisasi, sementara endpoint publik dapat diakses tanpa autentikasi.

3.3 Implementasi CRUD

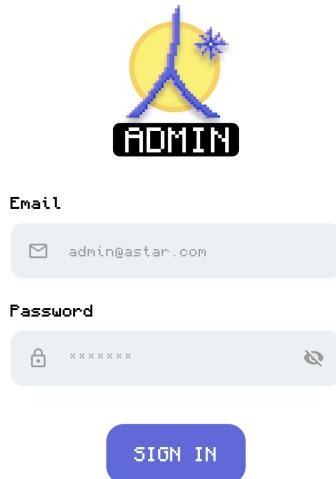
3.3.1 CRUD pada Web

3.3.2 CRUD pada Mobile App (Admin)

A. Mekanisme Masuk (Sign In / Login)

Admin akan memasukkan email dan password yang valid. Proses ini akan divalidasi oleh supabase.auth dengan JWT token serta diperkuat dengan JWT secret key (untuk

benar-benar memastikan bahwa anda adalah admin). Lalu anda akan dibawa ke loading screen yang berguna untuk memberi spare waktu agar fetching data selesai lebih dahulu sebelum memasuki manage screen.



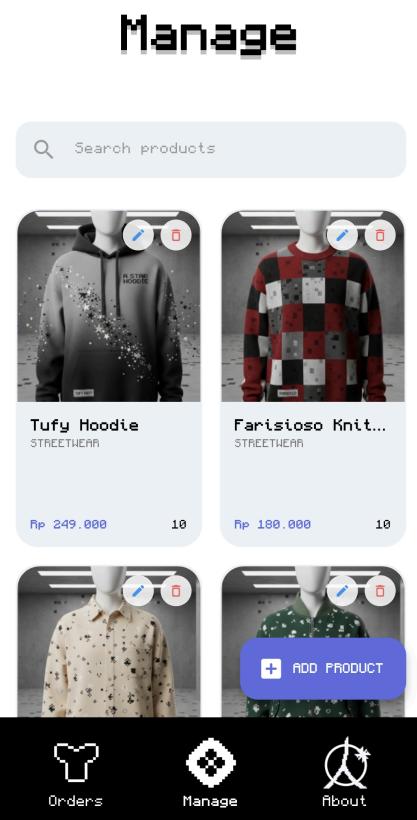
Gambar 3.3.2.1. Antarmuka Halaman Masuk (Login Screen)



Gambar 3.3.2.2 Antarmuka Halaman Loading (Loading Screen)

B. Mekanisme CRUD

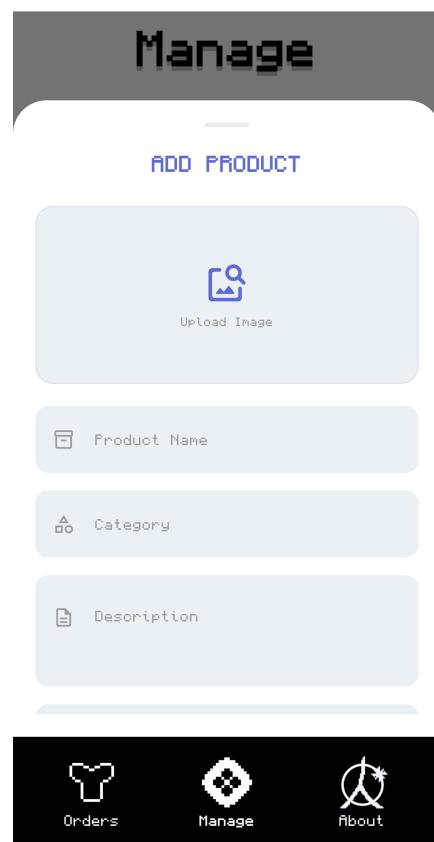
Admin dapat menambah, mengurangi, dan memperbarui stock. Admin juga dapat membuat order berdasarkan order customer melalui WA. Order yang statusnya pending dan process juga dapat terlihat langsung di aplikasi. Semua itu dilakukan melalui endpoint yang sudah disebutkan di atas. Akses endpoint /api/admin ini hanya dapat diakses oleh user yang memiliki role admin, dengan begitu keamanan lebih terjaga untuk perubahan data yang sensitif. (alur lebih jelasnya ada pada video demo)



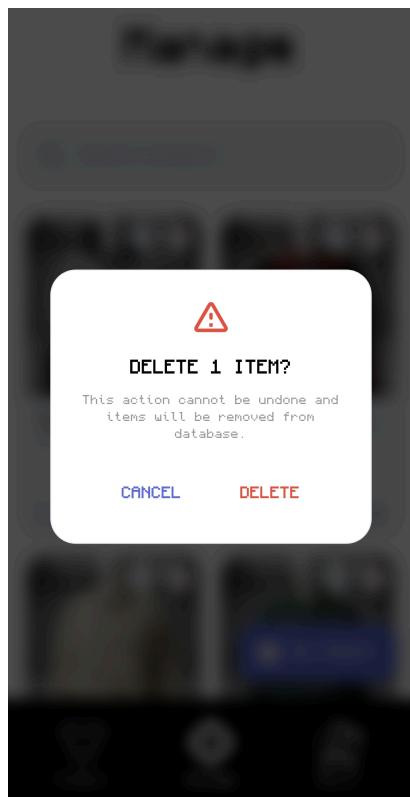
Gambar 3.3.2.3 Antarmuka Awal Manage Screen



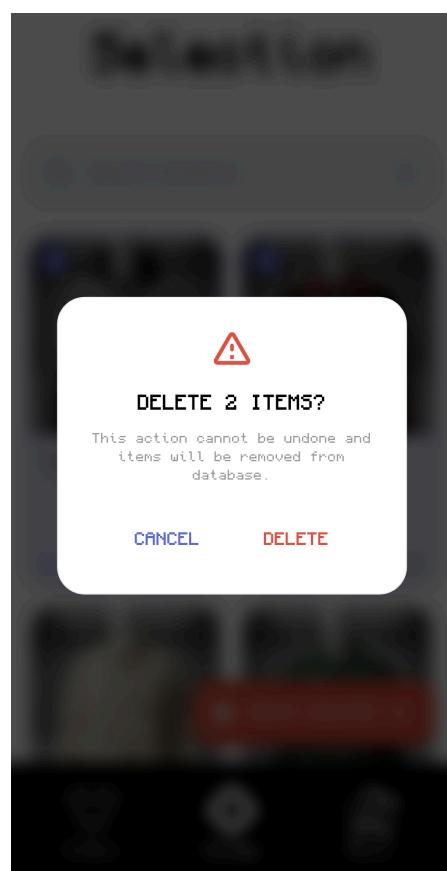
Gambar 3.3.2.4 Antarmuka Edit/Update Product



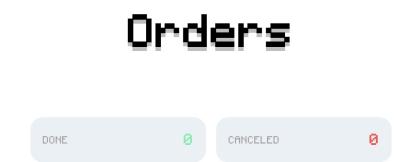
Gambar 3.3.2.5 Antarmuka Add/Create Product



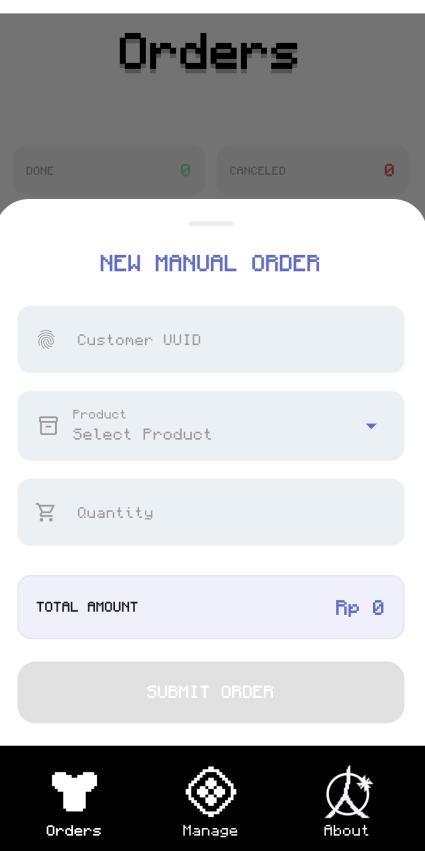
Gambar 3.3.2.6 Antarmuka Delete 1 Item/Product



Gambar 3.3.2.7 Antarmuka Delete Multiple/Bulk Items/Products

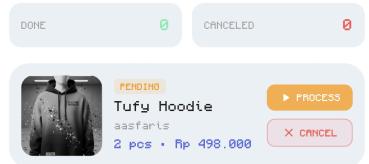


Gambar 3.3.2.8 Antarmuka Awal Orders Screen



Gambar 3.3.2.9 Antarmuka Add/Create Order

Orders

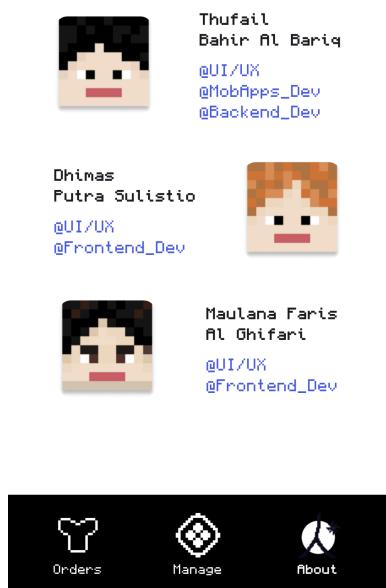


Gambar 3.3.2.10 Antarmuka Orders Screen Setelah Add/Create Order

C. Miscellaneous

Kami juga membuat about screen untuk memperlihatkan kami sebagai developer di dalam aplikasi. Klik 1 kali untuk menampilkan sound effect minecraft hurt. Klik Tahan salah satu karakter kami untuk menuju ke profile github.

About Us



Gambar 3.3.2.11 Antarmuka About Us Screen

3.4 Integrasi Website Menggunakan Next.js

3.4.1 Autentikasi (Sign In dan Sign Up)

A. Mekanisme Pendaftaran (Sign Up)

Proses pendaftaran dirancang untuk melakukan dua tugas utama secara atomik (berurutan) guna memastikan integritas data:

1. Autentikasi Kredensial: Sistem menggunakan fungsi `supabase.auth.signUp` untuk mendaftarkan email dan kata sandi pengguna ke dalam skema internal Supabase (`auth.users`). Pada tahap ini, Supabase secara otomatis menghasilkan UUID (Universally Unique Identifier) yang unik untuk setiap pengguna.
2. Penyimpanan Profil Pengguna: Setelah akun autentikasi berhasil dibuat, sistem secara otomatis mengambil `user_id` (UUID) tersebut dan menyimpannya ke dalam tabel `public.users`. Tabel ini menyimpan informasi tambahan yang diperlukan untuk pengiriman, yaitu:
 - `username` (sebagai identitas unik publik).
 - `full_name` (nama lengkap pelanggan).
 - `phone_number` (nomor kontak untuk integrasi WhatsApp).
 - `street, city, post_code` (detail alamat pengiriman).



Start Shopping With Us!

Email Address: email@example.com

Username: dhimas_ghifari

Full Name: Dhimas Bahir Al Ghifari

Phone Number: 08123456789

Street Address: Jl. Geografi No. 5

City: Sleman

Post Code: 55281

Password: *****

Already have an account? [Sign In](#)

REGISTER

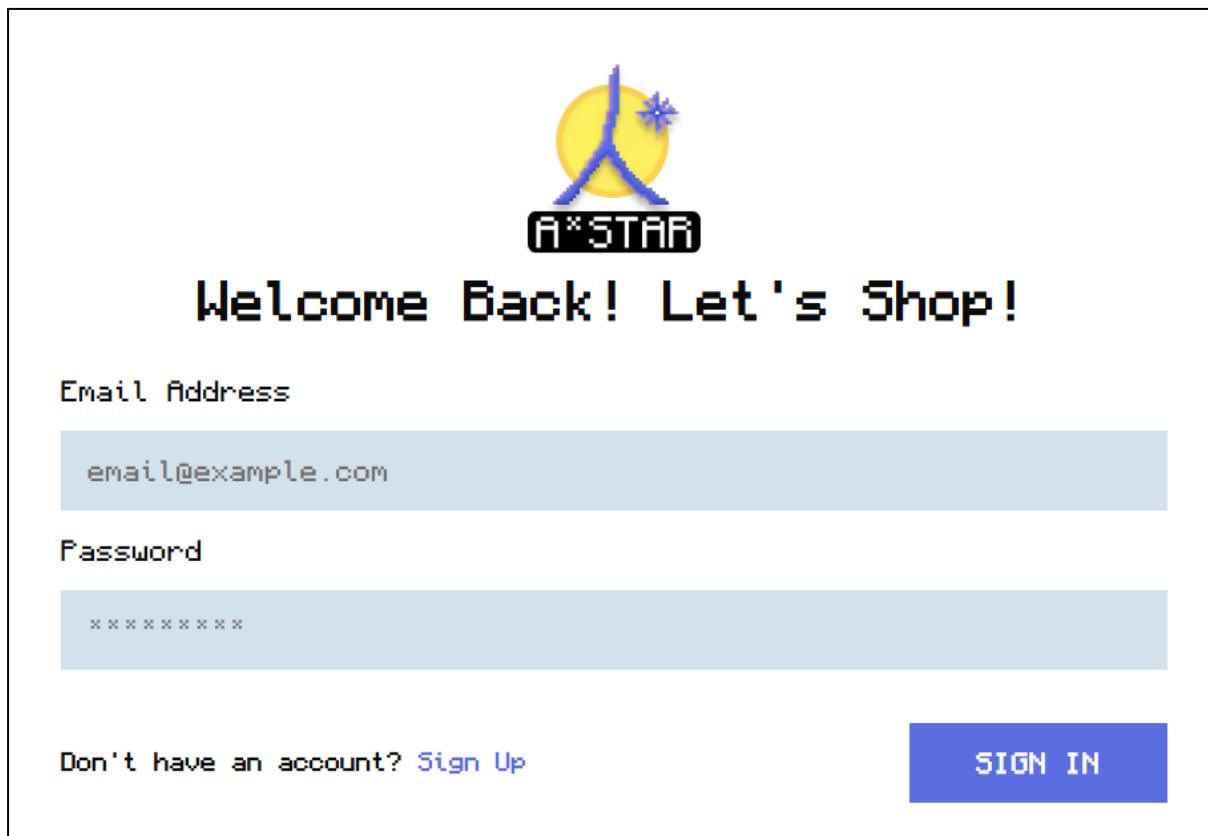
Gambar 3.4.1.1. Antarmuka Halaman Pendaftaran (Register Page)

users					
	user_id	username	full_name	phone_number	role
	87b1d3e8-cc3f-4522-a79a-b1a581...	dmin1	admin	-	admin
	c4f52e1d-03d3-401e-800c-6f47a...	baafaris	aasfaris	087714897812	user

Gambar 3.4.1.2 Hasil Data pada Supabase Table Editor

B. Mekanisme Masuk (Sign In)

Proses masuk menggunakan fungsi `supabase.auth.signInWithEmailAndPassword`. Sistem akan memvalidasi kredensial pengguna dan mengelola session menggunakan JWT (JSON Web Token).

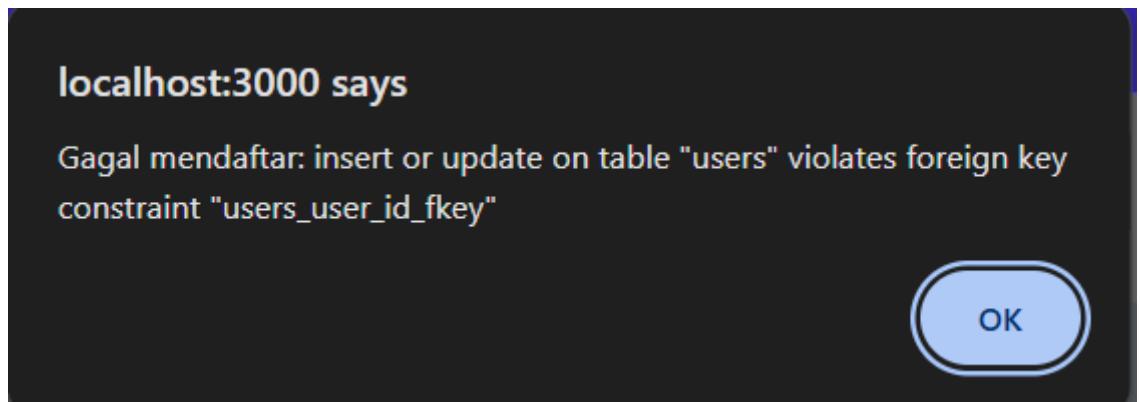


Gambar 3.4.1.3 Antarmuka Halaman Login

C. Keamanan dan Validasi Data Dalam implementasinya

Kelompok 4 menerapkan beberapa lapisan validasi untuk menjaga kualitas basis data:

- Unique Constraint: Kolom `username` dan `email` diatur sebagai *unique* di tingkat database. Hal ini mencegah terjadinya duplikasi identitas pengguna.
- Foreign Key Constraint: Kolom `user_id` pada tabel `public.users` bertindak sebagai *Foreign Key* yang merujuk pada `auth.users(id)`. Dengan pengaturan ON DELETE CASCADE, jika akun autentikasi dihapus, maka data profil di tabel publik akan ikut terhapus secara otomatis.
- Error Handling: Sistem dilengkapi dengan logika penanganan kesalahan untuk mendeteksi konflik data (seperti *duplicate key violation*). Jika pengguna mencoba mendaftar dengan username yang sudah ada, sistem akan memberikan notifikasi yang informatif melalui *user interface*.



Gambar 3.4.1.4: Pesan Validasi Error (Duplicate Username)

D. Hasil Implementasi

Dengan integrasi ini, aplikasi berhasil mencapai sinkronisasi data yang sempurna antara akun login dan profil pengiriman. Data yang dimasukkan saat pendaftaran inilah yang nantinya akan digunakan secara otomatis pada fitur Order to WhatsApp untuk mempermudah proses transaksi bagi admin maupun pengguna.



Toefl T-shirt

Rp 75.000

Available: 10 items

[Order Now](#)

...
...
...

Chat on WhatsApp with +62 813-5713-5429

Hai admin saya ingin order:
Produk: Toefl T-shirt
ID: 6
Harga: Rp 75.000

--- DATA PENGIRIMAN ---
User ID: c4f52e1d-03d3-401e-800c-6f47a3834475
Nama: aasfaris
No. HP: 087714897812
Alamat: Jl. Ciwaru Raya Komp. Citra 1 no. 61
Kota: Kota Serang
Kode Pos: 11111

[Open app](#)

[Continue to WhatsApp Web](#)

 Don't have the app? [Download it now](#)

Hai admin saya ingin order:
Produk: Toefl T-shirt
ID: 6
Harga: Rp 75.000

--- DATA PENGIRIMAN ---
User ID :
c4f52e1d-03d3-401e-800c-6f47a3834475

Nama: aasfaris
No. HP: 087714897812
Alamat: Jl. Ciwaru Raya Komp. Citra 1 no. 61
Kota: Kota Serang
Kode Pos: 11111

Mohon pesanan saya diproses ya, terima kasih! 21.03 ✓✓

Hai admin saya ingin order:
Produk: Toefl T-shirt
ID: 6
Harga: Rp 75.000

--- DATA PENGIRIMAN ---
User ID :
c4f52e1d-03d3-401e-800c-6f47a3834475

Nama: aasfaris
No. HP: 087714897812
Alamat: Jl. Ciwaru Raya Komp. Citra 1 no. 61
Kota: Kota Serang
Kode Pos: 11111

Mohon pesanan saya diproses ya, terima kasih!

Gambar 3.4.1.5: Preview Pesan WhatsApp dengan Data Otomatis

3.4.2 Homepage

Hero Section adalah komponen landing page pertama yang dilihat user saat mengakses website. Komponen ini bertujuan untuk menarik perhatian dan mengarahkan user ke katalog produk.

Tujuan:

- Landing page visual yang menarik
- Call-to-action untuk mengarahkan user ke collections
- Responsive design untuk berbagai ukuran layar

Fitur:

- Image optimization dengan Next.js Image component
- Smooth scroll animation
- Interactive button dengan hover effect
- Responsive layout (mobile dan desktop)



Gambar 3.4.2.1 Antarmuka Halaman Beranda

3.4.3 Collection Page

Collections Component menampilkan katalog produk yang diambil dari backend API. Komponen ini bertanggung jawab untuk data fetching dan rendering product grid.

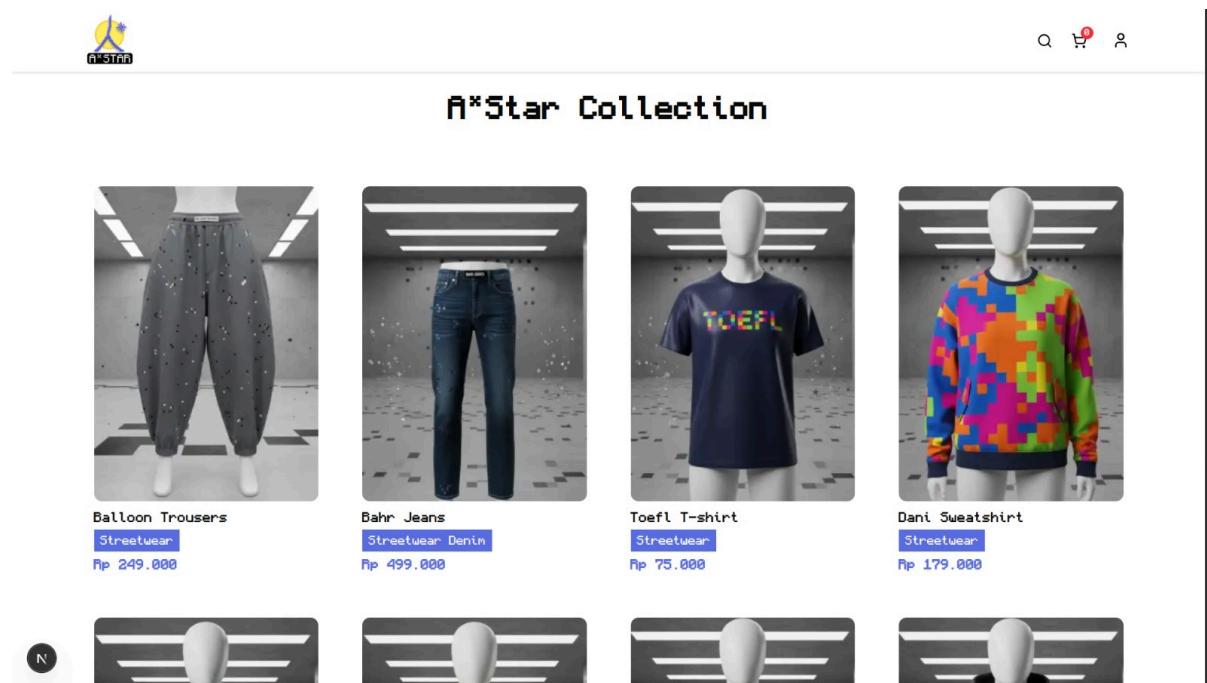
Purpose:

- Menampilkan katalog produk dari database
- Server-side data fetching untuk SEO
- Fallback ke dummy data jika API gagal

Key Features:

- Server Component (SEO-friendly)
- API integration dengan backend

- Sorting produk (in-stock first)
- Error handling dengan graceful degradation



Gambar 3.4.3.1 Antarmuka Halaman Koleksi

3.4.4 Pop Up Product Detail

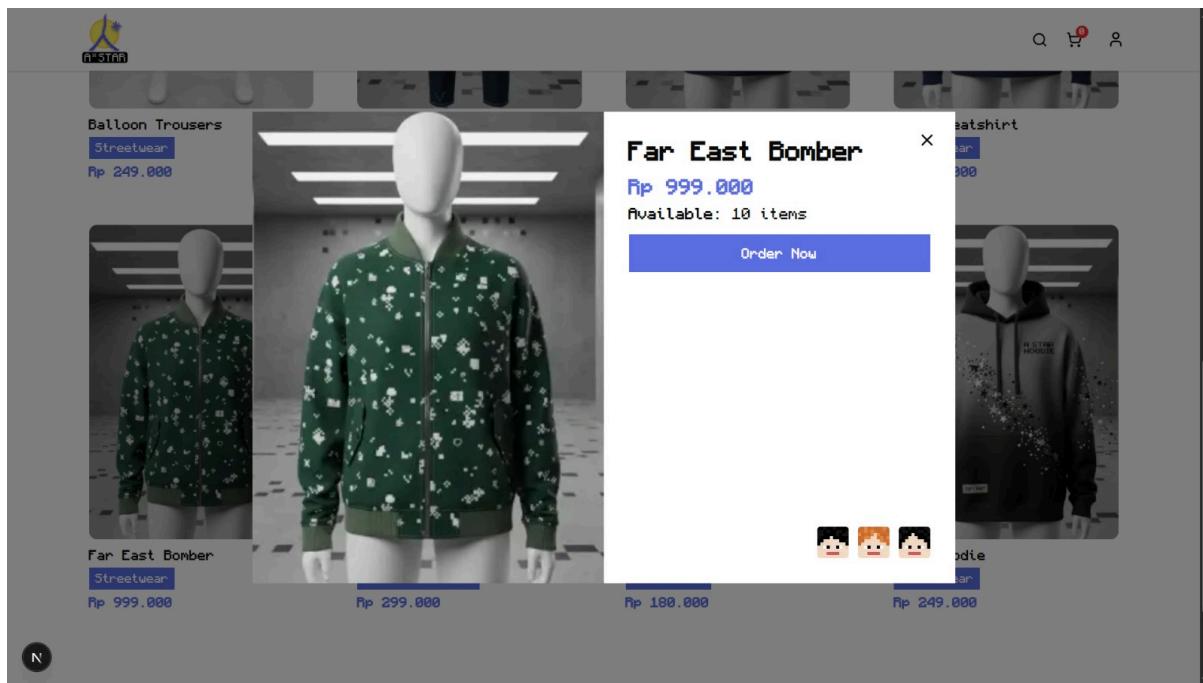
Pop-Up Component menampilkan detail produk dalam modal dan handle order process melalui WhatsApp integration.

Purpose:

- Tampilkan detail produk lengkap
- Handle order placement via WhatsApp
- Authentication check sebelum order
- Auto-populate user address dari database

Key Features:

- Modal overlay dengan backdrop
- WhatsApp deep link integration
- Supabase authentication check
- User data fetching dari database
- Responsive layout (mobile dan desktop)



Gambar 3.4.4.1 Antarmuka Pop Up Product Detail

3.5 Integrasi Fitur Order ke WhatsApp

Bagian ini menguraikan implementasi fitur pemesanan produk yang menghubungkan antarmuka situs web langsung ke layanan pesan instan WhatsApp Admin. Fitur ini dirancang untuk menyederhanakan proses transaksi bagi pelaku usaha kecil dengan mengotomatisasi pengiriman data pesanan dan alamat pengiriman.

A. Alur Logika Pemesanan (Order Workflow)

Integrasi ini tidak hanya sekadar tautan statis, melainkan sebuah proses dinamis yang melibatkan validasi keamanan dan pengambilan data *real-time* dari basis data. Alur kerja fitur ini meliputi:

1. Verifikasi Autentikasi: Sebelum pesan dibuat, sistem melakukan pengecekan status session pengguna menggunakan supabase.auth.getUser(). Secara etis dan fungsional, sistem membatasi fitur pemesanan hanya bagi pengguna yang telah terdaftar dan masuk (*logged-in*).
2. Proteksi Akses: Jika pengguna belum terautentikasi, sistem akan menampilkan notifikasi peringatan dan secara otomatis mengarahkan (*redirect*) pengguna ke halaman /register. Hal ini dilakukan untuk memastikan bahwa setiap pesanan didukung oleh data identitas yang valid di basis data.
3. Ekstraksi Data Profil: Setelah verifikasi berhasil, sistem melakukan kueri ke tabel public.users berdasarkan user_id pengguna. Data yang diambil meliputi nama lengkap, nomor telepon, dan alamat pengiriman lengkap (jalan, kota, dan kode pos).

Gambar 3.5.1: Diagram Alur (Flowchart) Fitur Order ke WhatsApp *Keterangan: Alur logika sistem mulai dari pengecekan session hingga pembuatan URL WhatsApp.*

B. Pengolahan Pesan Dinamis (Dynamic Message Formatting)

Pesan yang dikirimkan ke WhatsApp disusun menggunakan teknik *string template literals* pada JavaScript. Untuk memastikan pesan dapat dikirimkan melalui URL secara aman, sistem menerapkan fungsi encodeURIComponent. Komponen pesan terdiri dari:

- Informasi Produk: Nama produk, ID produk, dan harga yang terformat (IDR).
- Informasi Pengiriman: Data yang ditarik otomatis dari database untuk menghindari kesalahan input manual oleh pelanggan.
- Format Markdown: Penggunaan karakter khusus seperti asterisk (*) untuk menghasilkan teks tebal (*bold*) pada aplikasi WhatsApp guna meningkatkan keterbacaan pesan oleh admin.

Gambar 3.5.2: Implementasi Kode Fungsi handleOrderNow *Keterangan: Potongan kode yang menunjukkan proses pengambilan data dari Supabase dan penyusunan pesan URL WhatsApp.*

C. Aspek Etika dan Privasi Data

Dalam mengintegrasikan fitur ini, Kelompok 4 memperhatikan standar etika perlindungan data pengguna:

- Transparency: Data pribadi pengguna (seperti alamat) hanya akan dikirimkan ke WhatsApp Admin apabila pengguna secara sadar menekan tombol "Order Now".
- Data Minimization: Hanya data yang relevan untuk kebutuhan pengiriman barang yang disertakan dalam pesan otomatis tersebut.
- Security: Penggunaan kunci API Supabase pada sisi klien dibatasi dengan kebijakan *Row Level Security* (RLS) untuk mencegah pengambilan data profil pengguna lain oleh pihak yang tidak bertanggung jawab.

Gambar 3.5.3: Tampilan Pesan yang Diterima oleh WhatsApp Admin *Keterangan: Hasil akhir pesan WhatsApp yang rapi, informatif, dan berisi detail pengiriman lengkap tanpa perlu input ulang.*

D. Hasil Implementasi

Fitur ini berhasil memangkas waktu komunikasi antara penjual dan pembeli. Admin tidak perlu lagi melakukan tanya jawab manual mengenai detail alamat pengiriman karena sistem telah menyediakan informasi tersebut secara otomatis berdasarkan data akun pengguna.

BAB IV PENUTUP

4.1 Kesimpulan

Berdasarkan seluruh tahapan perancangan, implementasi, hingga pengujian sistem manajemen inventori A*Star, dapat disimpulkan bahwa sinergi antara teknologi *framework* Next.js, bahasa pemrograman Go, dan *backend-as-a-service* Supabase telah berhasil menciptakan solusi digital yang komprehensif bagi permasalahan operasional pada sektor usaha kecil. Keberhasilan sistem ini tercermin dari arsitektur basis data PostgreSQL yang telah memenuhi standar normalisasi hingga *Third Normal Form* (3NF) yang menjamin integritas data serta meminimalisir redundansi informasi. Implementasi fitur keamanan melalui *Supabase Auth* yang terintegrasi dengan kebijakan *Row Level Security* (RLS) terbukti mampu menciptakan ekosistem data yang aman dan terisolasi, memastikan bahwa informasi sensitif pengguna tetap terlindungi di tengah arus transaksi digital.

Proyek ini membuktikan bahwa integrasi data yang mendalam antara sistem autentikasi dengan fitur layanan pelanggan mampu meningkatkan efisiensi waktu secara signifikan. Inovasi pada fitur *Order to WhatsApp* yang mengotomatisasi pengiriman profil pelanggan, termasuk identitas unik *User ID* dan detail alamat yang telah berhasil memitigasi risiko kesalahan manusia (*human error*) yang selama ini menjadi kendala utama dalam pencatatan manual. Secara keseluruhan, A*Star bukan sekadar platform pengelolaan stok barang, melainkan sebuah instrumen transformasi digital yang mampu menyederhanakan alur kerja yang kompleks menjadi antarmuka yang intuitif bagi admin maupun pelanggan. Pencapaian ini menegaskan bahwa penggunaan teknologi *full-stack* modern yang tepat guna dapat memberikan dampak nyata dalam memperkuat daya saing dan profesionalisme pelaku usaha di pasar global.

4.2 Saran

Meskipun sistem manajemen inventori A*Star telah mencapai sasaran fungsional utamanya, terdapat beberapa aspek pengembangan yang dapat diimplementasikan di masa depan untuk meningkatkan skalabilitas dan kematangan sistem. Salah satu fokus utama pengembangan selanjutnya adalah integrasi sistem pembayaran otomatis (*payment gateway*) dan fitur keranjang belanja (*shopping cart*) yang lebih kompleks. Penambahan fitur ini akan memungkinkan pelanggan untuk melakukan transaksi multiproduk secara simultan tanpa harus melakukan pemesanan satu per satu sehingga mampu menciptakan pengalaman berbelanja yang lebih efisien dan modern sesuai dengan standar industri *e-commerce*.

Selain pada sisi fungsionalitas transaksi, pengembangan pada sisi analisis data juga menjadi langkah strategis yang sangat disarankan. Implementasi dasbor analitik yang dilengkapi dengan fitur prakiraan stok (*stock forecasting*) berbasis kecerdasan buatan akan memungkinkan pemilik usaha untuk memprediksi kebutuhan inventori di masa mendatang berdasarkan riwayat penjualan. Lebih lanjut, integrasi sistem notifikasi otomatis melalui

email atau pesan instan terkait ambang batas stok minimum akan memberikan perlindungan proaktif terhadap risiko kehabisan barang. Dengan menyempurnakan aspek-aspek tersebut, platform A*Star diharapkan tidak hanya berfungsi sebagai alat administratif, tetapi juga dapat berevolusi menjadi ekosistem pendukung keputusan bisnis yang cerdas dan adaptif terhadap dinamika pasar.

4.3 Catatan

Kami juga menyediakan video demo, deploy backend, dan release APK pada repository github kami (https://github.com/yuumaSSS/DBProject_Group4_Astar/tree/main). Apabila penilai ingin mendapat akses secara langsung admin mobile app kami (hanya untuk pengujian), bisa menghubungi email dibawah ini:

thufailbahiralbariq2006@mail.ugm.ac.id

Video demo juga tersedia, dapat diakses melalui README pada github maupun langsung menggunakan link ini:

 AStar Admin Demo.mp4