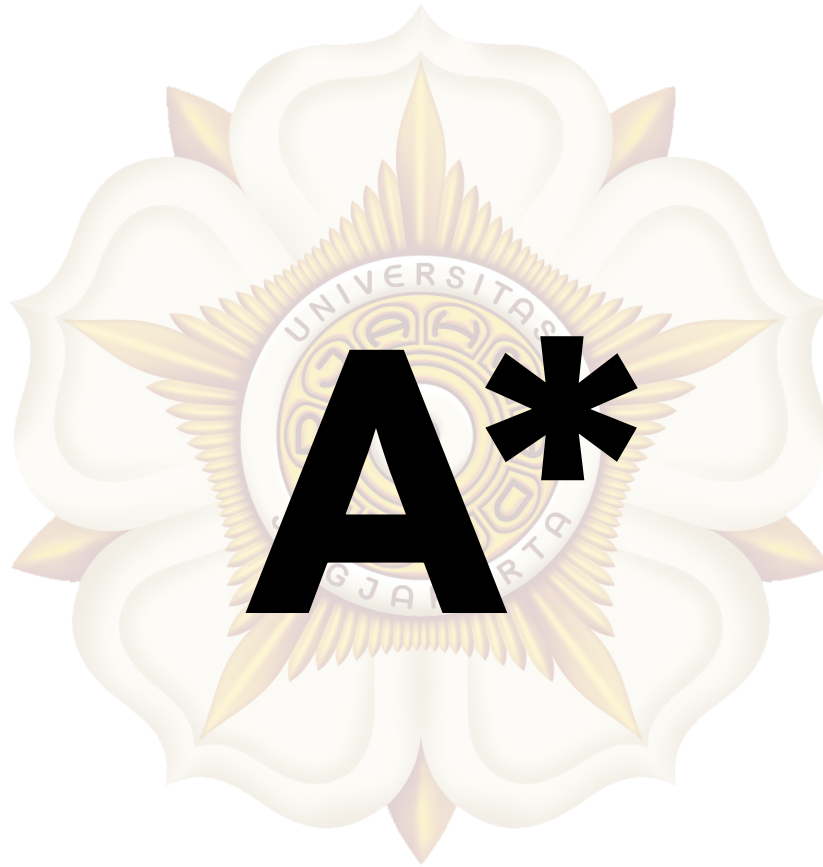


Proposal Project Database

Sistem Manajemen Inventori dan Penjualan untuk Usaha Kecil



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

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

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

Abstraksi

Dalam era digital yang semakin kompetitif, usaha kecil dan menengah dituntut untuk mampu mengelola produk, stok, dan transaksi secara efisien. Dengan kebutuhan tersebut, tim kami mengembangkan “A*”, sebuah sistem basis data dalam lingkup *Small Business Inventory and Sales* yang dirancang untuk membantu usaha dalam mengatur persediaan produk dan mencatat aktivitas penjualan secara terstruktur.

Nama “A*” diambil dari algoritma *A-star* (A*) yang terkenal dalam bidang ilmu komputer sebagai algoritma pencarian jalur paling optimal. Filosofi ini kami adopsi sebagai cerminan dari misi kami, yaitu membantu bisnis menemukan jalur terbaik dalam mengelola operasional dengan cara yang cepat, efisien, dan adaptif. Selain itu, simbol asterisk (*) merepresentasikan keberagaman dan inklusivitas, bahwa setiap pelanggan, produk, dan transaksi memiliki tempat penting dalam sistem kami.

Melalui proyek ini, kami tidak hanya berfokus pada pembangunan aplikasi berbasis web dan mobile yang terhubung dengan database relasional, tetapi juga pada penerapan prinsip-prinsip *learning by doing*. Tahapan pengembangan mencakup analisis permasalahan, perancangan model data, implementasi basis data ter-normalisasi hingga *Third Normal Form (3NF)*, dan pengembangan antarmuka sederhana yang mendukung operasi CRUD serta fitur pencarian stok.

1. Definisi Domain Permasalahan

Usaha kecil sering menghadapi tantangan besar dalam mengelola inventori dan transaksi penjualan secara efisien. Banyak pelaku usaha masih menggunakan pembukuan manual atau lembar kerja sederhana (misalnya Excel) yang rawan kesalahan, memakan waktu, serta sulit untuk diintegrasikan. Akibatnya, sering terjadi kekeliruan seperti kehabisan stok, stok berlebih, laporan penjualan yang tidak akurat, dan kesulitan dalam mengambil keputusan berbasis data.

Domain permasalahan proyek ini berfokus pada pengelolaan inventori dan penjualan untuk bisnis ritel atau perdagangan skala kecil dengan permasalahan utama sebagai berikut:

- Sulit memantau ketersediaan produk dan tren penjualan.
- Tidak ada integrasi antara data stok dan data transaksi.
- Tidak dapat menghasilkan laporan atau ringkasan penjualan dengan cepat.
- Tidak adanya basis data terpusat yang menyimpan seluruh informasi produk, stok, dan transaksi.

Untuk mengatasi masalah tersebut, tim akan mengembangkan aplikasi web dan mobile berbasis basis data yang mampu mengelola data produk, stok, dan transaksi penjualan dengan prinsip integritas data dan normalisasi hingga Third Normal Form (3NF).

2. Pengguna Sistem

Sistem ini akan digunakan oleh dua jenis pengguna utama, yaitu Admin dan Pelanggan. Admin merupakan pemilik atau manajer usaha yang memiliki akses penuh terhadap sistem, termasuk pengelolaan produk, penyesuaian stok, serta peninjauan laporan penjualan. Pelanggan, yang merupakan pengguna eksternal, dapat mengakses sistem untuk melihat ketersediaan produk dan harga, serta mengajukan permintaan pembelian. Akses hak admin akan difokuskan dengan aplikasi berbasis mobile, sedangkan pelanggan akan difokuskan dengan aplikasi berbasis web.

3. Tujuan Sistem

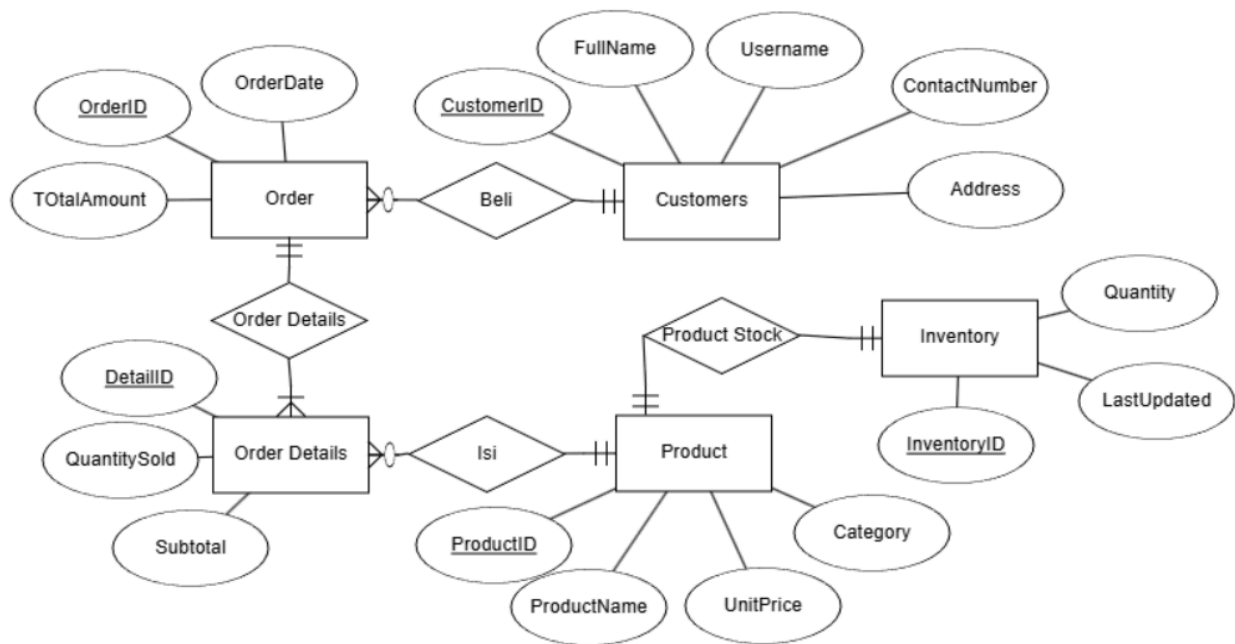
Tujuan dari proyek ini bersifat praktis dan edukatif, yaitu mengembangkan sistem berbasis basis data yang merepresentasikan skenario dunia nyata.

Tujuan Utama

1. Membangun basis data relasional yang menyimpan data produk, inventori, dan transaksi penjualan dengan struktur yang ternormalisasi (hingga 3NF).
2. Mengimplementasikan operasi CRUD (Create, Read, Update, Delete) untuk setiap entitas utama (Product, Inventory, Order, Order Details, dan Customers).
3. Menyediakan fitur pencarian dan pelaporan, seperti:
 - Pencarian produk berdasarkan nama, kategori, atau jumlah stok.
 - Pembuatan laporan penjualan dan status inventori.
4. Menjamin konsistensi dan integritas data dengan penerapan primary key, foreign key, serta constraint relasional.
5. Membangun antarmuka web sederhana dan interaktif menggunakan HTML, CSS, JavaScript, dan backend sederhana (misalnya Node.js atau PHP) untuk berinteraksi dengan basis data.
6. Membangun antarmuka mobile sederhana dan interaktif menggunakan Flutter dan backend sederhana (misalnya Go atau Python) untuk berinteraksi dengan basis data.

4. Diagram Hubungan Entitas (ERD)

ERD dirancang untuk menggambarkan hubungan antar entitas yang terlibat dalam proses manajemen inventori dan penjualan.



Berikut adalah desain konseptual dalam bentuk teks dalam diagram formal.

Entitas dan Atribut

1. Product
 - ProductID (PK)
 - ProductName
 - Category
 - UnitPrice
2. Inventory
 - InventoryID (PK)
 - ProductID (FK)
 - Quantity
 - LastUpdated
3. Orders

- OrderID (PK)
 - OrderDate
 - CustomerID (FK)
 - TotalAmount
4. OrderDetails
- DetailID (PK)
 - TransactionID (FK)
 - ProductID (FK)
 - QuantitySold
 - Subtotal
5. Customers
- CustomerID (PK)
 - FullName
 - Username
 - ContactNumber
 - Address

Hubungan Antar Entitas

1. Product ↔ Inventory (One-to-One)

Setiap produk memiliki satu catatan stok di tabel Inventory. Hubungan ini bersifat satu ke satu (1:1) karena setiap produk hanya memiliki satu nilai stok yang mewakili ketersediaan barang di sistem.

- Relasi: Product.ProductID → Inventory.ProductID
- Makna: Setiap produk memiliki jumlah stok dan tanggal pembaruan terakhir yang tersimpan di tabel *Inventory*.

Contoh: Produk “Tuff t-shirt” memiliki satu entri stok di *Inventory* yang menunjukkan jumlah persediaan dan tanggal update terakhir.

2. Product ↔ OrderDetails (One-to-Many)

Satu produk dapat muncul di banyak transaksi melalui tabel *OrderDetails*. Hubungan ini bersifat satu ke banyak (1:N) karena satu produk dapat dijual berkali-kali, sementara satu baris di *OrderDetails* hanya mengacu pada satu produk tertentu dalam satu transaksi.

- Relasi: *Product.ProductID* → *OrderDetails.ProductID*
- Makna: Produk yang sama bisa menjadi bagian dari beberapa transaksi yang berbeda.

Contoh: Produk “Dhimsum Knitwear” dapat muncul di beberapa transaksi pelanggan seperti #T001, #T007, dan #T018.

3. *Orders* ↔ *OrderDetails* (One-to-Many)

Setiap transaksi (*Orders*) dapat memiliki lebih dari satu item produk di dalamnya. Hubungan ini adalah satu ke banyak (1:N) karena satu transaksi bisa mencakup beberapa produk yang dibeli sekaligus.

- Relasi: *Orders.TransactionID* → *OrderDetails.TransactionID*
- Makna: Setiap transaksi pelanggan memiliki detail barang yang dibeli, termasuk kuantitas dan subtotal per item.

Contoh: Transaksi #T010 bisa berisi dua item, yaitu “Faris Paris Hoodie” (2 pcs) dan “Far East Jeans” (1 pcs).

4. *Customers* ↔ *Orders* (One-to-Many)

Setiap pelanggan dapat melakukan banyak transaksi pembelian, namun setiap transaksi hanya dimiliki oleh satu pelanggan. Hubungan ini bersifat satu ke banyak (1:N).

- Relasi: *Customers.CustomerID* → *Orders.CustomerID*
- Makna: Riwayat pembelian pelanggan dapat dilacak melalui daftar transaksi yang mereka lakukan di sistem.

Contoh: Pelanggan “Erdi Plas Budiyo” mungkin memiliki tiga transaksi dengan kode #T001, #T004, dan #T009.