**LAPORAN**

**DATABASE**

**UNTUK MEMENUHI UAS**

**Mata kuliah : Basis Data**

**Dosen Pengampu : Mohamad Firdaus, M.Kom**



**Penyusun**

Dimas Dwi Rianto (623C0004)

**INSTITUT TEKNOLOGI DAN KESEHATAN MAHARDIKA**

**PROGRAM STUDI INFORMATIKA S1**

**2024**

## Rancangan Struktur Database

Perancangan struktur database kali ini saya lakukan dimulai dengan memberi nama database saya sebagai “deltaromeo1”, database ini saya buat guna mendukung dan memenuhi kebutuhan aplikasi web atau aplikasi desktop (dengan python). Dalam perancangan database ini juga saya melakukan Normalisasi, Pembuatan ERD dan Pembuatan Struktur Tabel menggunakan SQL. Berikut adalah hasil normalisasi tabel dari 1NF sampai dengan 3NF

 Berikut adalah tabel yang belum dinormalisasi untuk selanjutnya dilakukan normalisasi tabel mulai dari 1NF sampai dengan 3NF

 Tabel di atas merupakan hasil Normalisasi 1NF, Normalisasi 1NF dilakukan untuk memisahkan alat camping dan jasa guide dalam baris terpisah, setiap kolom hanya memiliki satu nilai per sel.



Pada tahap ini kita memastikan bahwa setiap atribut non-primer sepenuhnya bergantung pada primary key. Berdasarkan tabel yang sudah ada pada 1NF, kita pisahkan menjadi dua tabel utama.

Untuk selanjutnya kita ke normalisasi 3NF, di tabel transactions kita memiliki kolom status yang berisi informasi tentang status transaksi. Ini menyebabkan ketergantungan transitif, karena status dalam tabel transactions bisa tergantung pada atribut non-primer lainnya, seperti status\_name yang seharusnya ada dalam tabel terpisah.

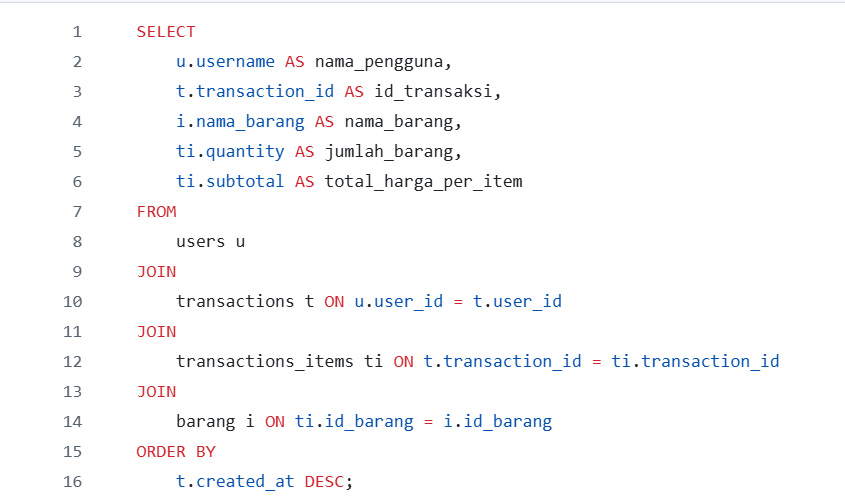
Perbaikan untuk di 3NF Untuk memenuhi 3NF, kita perlu memecah informasi yang berhubungan dengan status transaksi ke dalam tabel yang terpisah. Tabel transaction\_statuses akan menyimpan status dan status\_id, yang kemudian akan digunakan sebagai foreign key di tabel transactions.

Dengan perubahan ini, kita memastikan bahwa tidak ada ketergantungan transitif. Dalam tabel transactions, status\_id sekarang langsung merujuk ke tabel transaction\_statuses, yang menjelaskan status transaksi dengan menggunakan foreign key status\_id. Sebelumnya (2NF), kita memiliki kolom status dalam tabel transactions, yang bisa tergantung pada informasi lain, dan ini merupakan ketergantungan transitif. Setelah (3NF), kita memisahkan informasi status ke tabel terpisah (transaction\_statuses), yang menjamin bahwa setiap kolom dalam tabel transactions hanya bergantung langsung pada primary key transaction\_id.

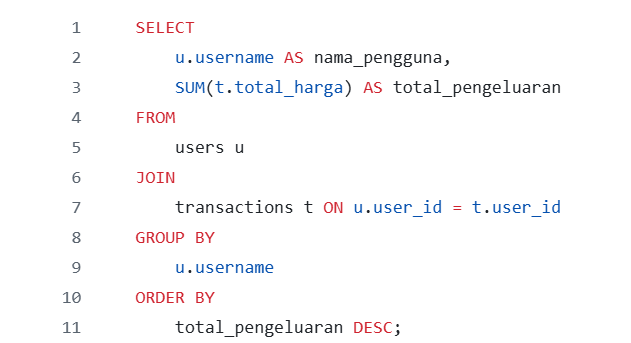
## Query SQL Untuk Memproses Data Kasus Bisnis

Saya menggunakan 2 Query SQL yaitu JOIN dan Subquery untuk memproses data pada kasus bisnis saya adapun query yang saya buat dengan JOIN adalah sebagai berikut :

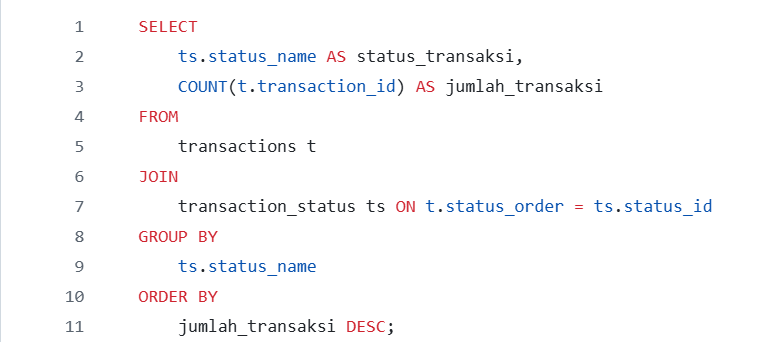
2.1 JOIN

1. Menampilkan daftar transaksi dan detail barang yang dibeli oleh pengguna

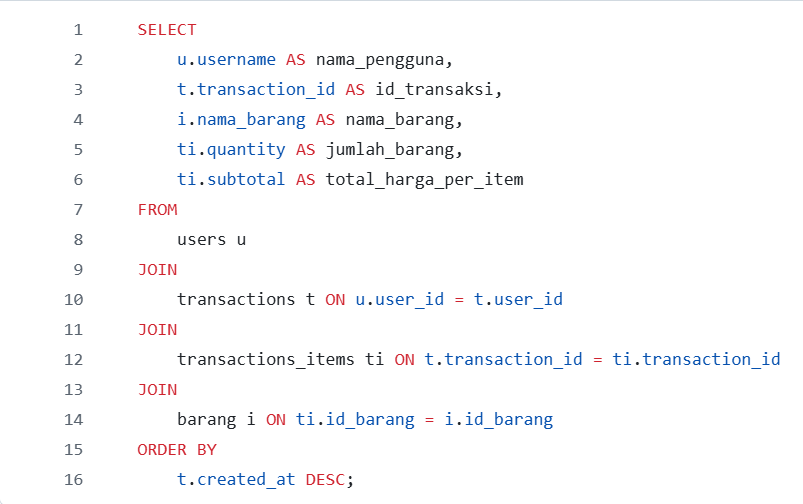
* Tabel yang Digunakan : users, transactions, transaction\_items, dan items.
* JOIN digunakan untuk menggabungkan tabel users, transactions, transaction\_items, dan items untuk mendapatkan informasi lengkap tentang transaksi dan barang yang dibeli.
* Hasil: Menampilkan nama pengguna, ID transaksi, nama barang, jumlah barang, dan subtotal untuk setiap item.

1. Menampilkan Total Pengeluaran Setiap Pengguna

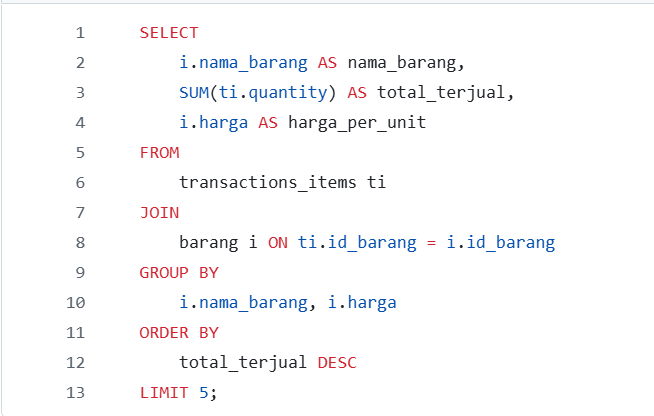
* Tabel yang Digunakan: users dan transactions.
* JOIN menghubungkan tabel users dengan tabel transactions untuk mendapatkan total pengeluaran dari setiap pengguna.
* GROUP BY digunakan untuk mengelompokkan data berdasarkan pengguna.
* Hasil: Menampilkan nama pengguna dan total pengeluaran yang diurutkan dari yang terbesar ke terkecil.

1. Menampilkan status transaksi beserta jumlah nya

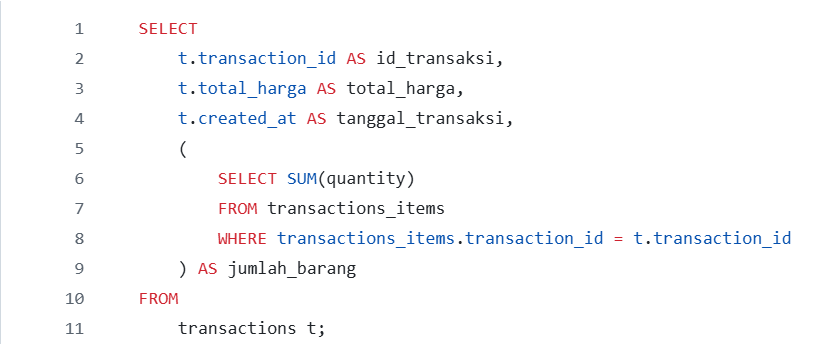
* Tabel yang Digunakan: transactions dan transaction\_status.
* JOIN menghubungkan tabel transactions dengan tabel transaction\_status untuk mendapatkan nama status transaksi.
* GROUP BY digunakan untuk mengelompokkan data berdasarkan status transaksi.
* Hasil: Menampilkan jumlah transaksi untuk setiap status transaksi.

1. Menampilkan Transaksi Terakhir yang dilakukan oleh setiap pengguna

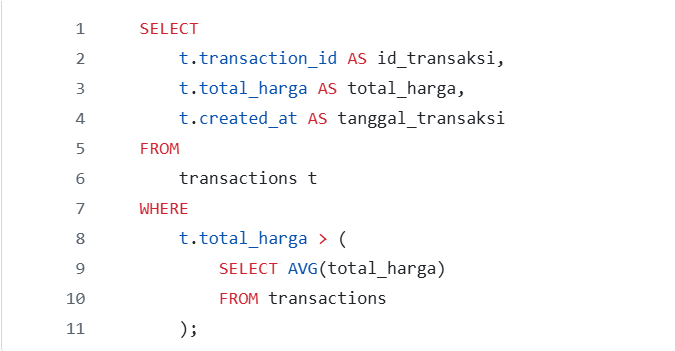
* Tabel yang Digunakan: transactions dan users.
* JOIN menghubungkan tabel transactions dengan tabel users untuk mendapatkan informasi pengguna.
* Subquery digunakan untuk mendapatkan waktu transaksi terakhir (MAX(created\_at)) untuk setiap pengguna.
* Hasil: Menampilkan ID transaksi terakhir, nama pengguna, total transaksi, dan waktu transaksi.

1. Menampilan barang dengan penjualan tertinggi

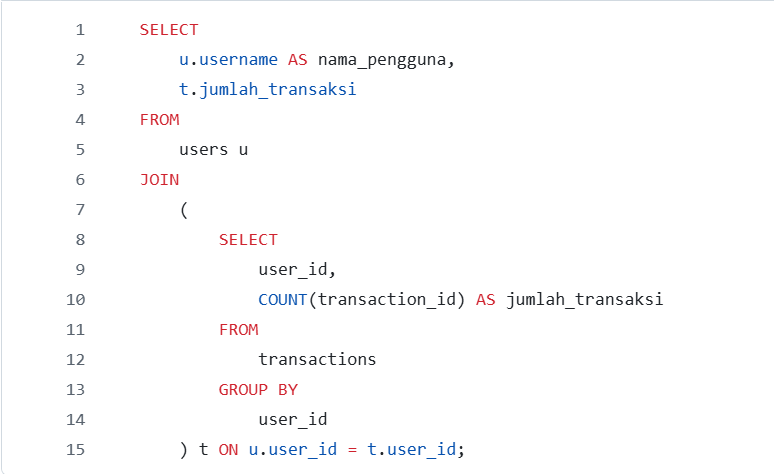
* Tabel yang Digunakan: transaction\_items dan items.
* JOIN menghubungkan tabel transaction\_items dengan tabel items untuk mendapatkan nama barang dan harga per unit.
* GROUP BY digunakan untuk mengelompokkan data berdasarkan nama barang.
* Hasil: Menampilkan nama barang, total barang yang terjual, dan harga per unit, diurutkan dari yang paling laku.
  1. Subquery

1. Subquery Pada SELECT

* Subquery pada bagian SELECT menghitung jumlah barang (SUM(quantity)) dari tabel transaction\_items berdasarkan transaction\_id pada tabel transactions.
* Hasilnya akan menunjukkan data transaksi beserta jumlah barang yang dibeli.

1. Subquery pada WHERE

* Subquery pada bagian WHERE menghitung rata-rata total harga (AVG(total\_harga)) dari tabel transactions.
* Query utama hanya akan menampilkan transaksi yang total harganya lebih besar dari rata-rata tersebut.

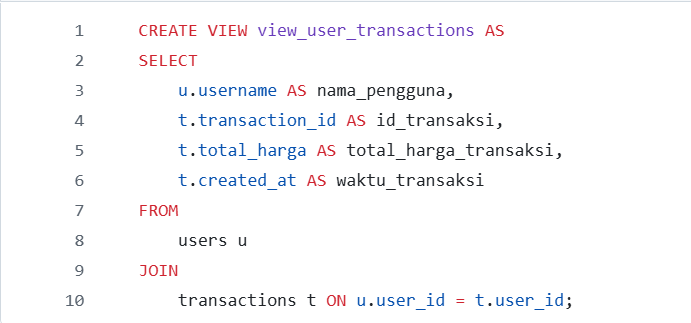
1. Subquery pada FROM

* Subquery pada bagian FROM menghitung jumlah transaksi (COUNT(transaction\_id)) untuk setiap pengguna (user\_id).
* Query utama kemudian menggabungkan data dari tabel users untuk menampilkan nama pengguna beserta jumlah transaksi mereka.

## Implementasi Database Object

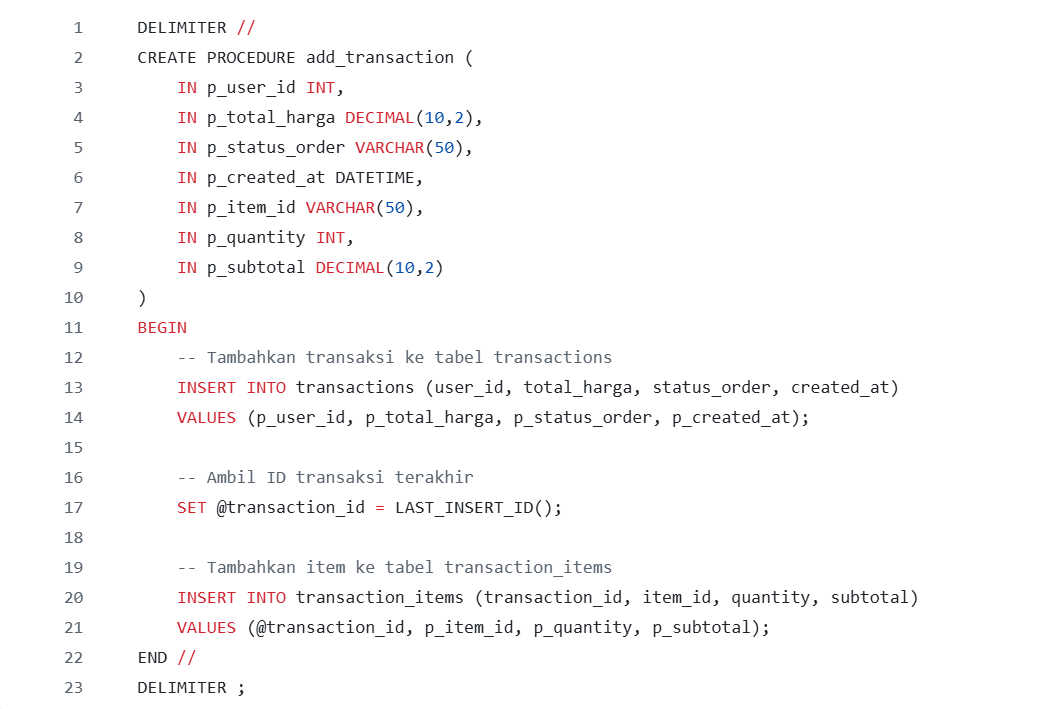
Berikut adalah implementasi untuk memenuhi minimal tiga jenis database object pada database yang Anda gunakan. Saya akan memberikan contoh penggunaan View, Procedure, Function, Trigger, dan Event.

1. View

Membuat view untuk menampilkan daftar transaksi beserta nama pengguna dan total harga transaksi

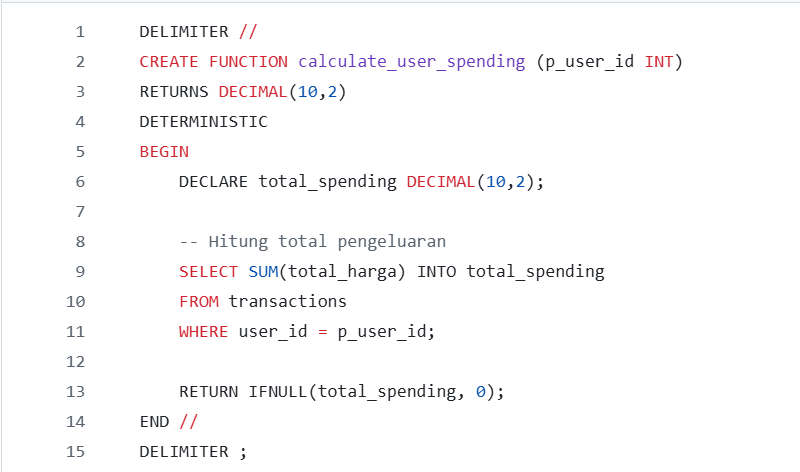
View ini akan menampilkan data transaksi beserta nama pengguna, total harga, dan waktu transaksi tanpa perlu menuliskan JOIN Query berulang kali.

1. Procedure

Membuat procedure untuk menambahkan data transaksi baru beserta item yang dibeli dalam satu proses.

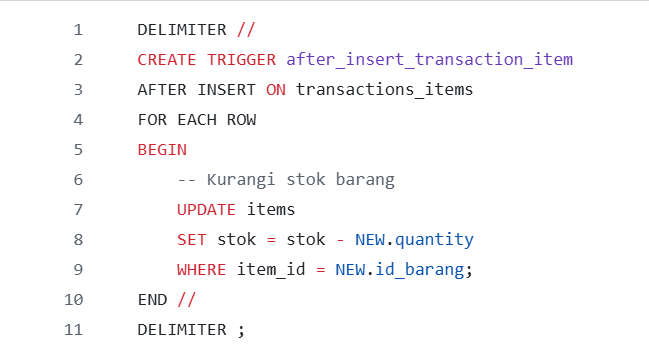
Procedure ini secara otomatis akan menambahkan transaksi ke tabel transactions dan item terkait ke tabel transaction\_items.

1. Function

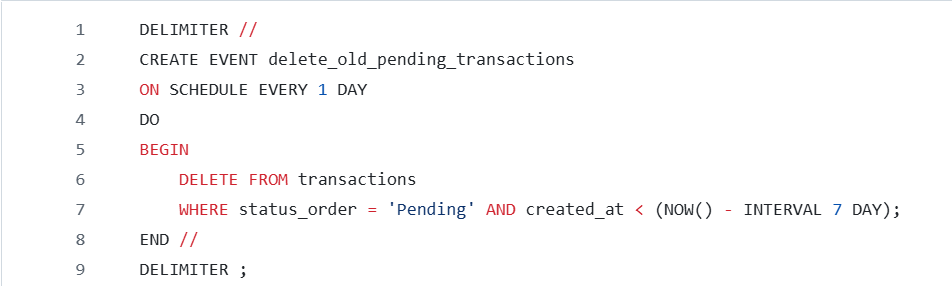
Membuat function untuk menghitung total pengeluaran dari seorang pengguna berdasarkan user\_id.

Function ini akan menghitung total pengeluaran dari pengguna dengan user\_id tertentu.

1. Trigger

Membuat trigger untuk mengurangi stok barang secara otomatis setiap kali ada transaksi barang baru.

Trigger ini secara otomatis akan mengurangi stok barang di tabel items ketika ada data baru yang dimasukkan ke tabel transaction\_items.

1. Event

Event ini akan berjalan setiap hari dan menghapus transaksi yang berstatus "Pending" lebih dari 7 hari secara otomatis.