Permodelan Data Normalisasi Terhadap Sebuah Domain ”Struk Swalayan”

Mata Kuliah Analisis dan Desain Basis Data

****

Anggota Kelompok:

1. 10123401 Lutfi Pramudya IF 10
2. 10123404 Gilang Aldiano IF 10
3. 10123411 Farhan Putra Pratama IF 10

Pendahuluan

**Latar Belakang**

Dalam era digital yang terus berkembang, efisiensi dalam pengelolaan data menjadi salah satu faktor kunci untuk keberhasilan bisnis. Sebuah sistem manajemen data yang baik harus mampu menyimpan, memproses, dan menyajikan data dengan cara yang mudah dipahami dan digunakan. Normalisasi data merupakan salah satu teknik penting yang digunakan dalam pemodelan data untuk mengurangi redundansi, mengelola ketergantungan atribut, dan memastikan integritas data dalam sebuah sistem.

Pada kasus Toko Cung proses pencatatan transaksi masih dilakukan secara manual, yang seringkali mengakibatkan data yang tidak konsisten, duplikasi, dan sulitnya pelacakan informasi. Untuk mengatasi masalah tersebut, dilakukan normalisasi terhadap data transaksi agar sistem manajemen data dapat dibangun dengan baik. Normalisasi ini dimulai dari bentuk tidak ternormalisasi (UNF), 1NF, hingga mencapai bentuk normalisasi ke-2 (2NF), di mana setiap atribut hanya bergantung pada kunci primer, dan tidak ada ketergantungan transitif.

**Tujuan**

1. **Menyusun struktur data yang optimal:** Dengan menggunakan teknik normalisasi, diharapkan data yang disimpan dalam tabel-tabel menjadi lebih terstruktur dan bebas dari redundansi.
2. **Menghilangkan ketergantungan yang tidak perlu:** Ketergantungan transitif dan parsial sering kali menyebabkan data sulit dikelola. Normalisasi bertujuan untuk mengeliminasi ketergantungan-ketergantungan ini.
3. **Mempermudah pengolahan dan integrasi data:** Setelah normalisasi, tabel-tabel menjadi lebih sederhana sehingga proses pengolahan data seperti penarikan laporan, perhitungan harga, dan manajemen inventaris menjadi lebih efisien.
4. **Memastikan fleksibilitas sistem:** Dengan normalisasi, sistem akan lebih mudah dikembangkan di masa depan, misalnya dengan menambah tabel atau atribut baru.

**Permasalahan**

1. Bagaimana membangun sistem yang terstruktur dan memenuhi kriteria normalisasi hingga 2NF untuk sistem transaksi Toko Cung?
2. Bagaimana memastikan bahwa data yang diinput pengguna (seperti kuantitas barang) tidak bergantung pada atribut lain selain kunci utama?

**Manfaat Normalisasi Data**

1. **Integritas Data:** Dengan struktur data yang sesuai normalisasi, integritas data lebih terjaga sehingga mengurangi kesalahan seperti duplikasi atau data yang tidak konsisten.
2. **Efisiensi Penyimpanan:** Normalisasi mengurangi redundansi data sehingga ukuran penyimpanan menjadi lebih kecil.
3. **Kemudahan Pemeliharaan Sistem:** Perubahan data atau struktur tabel dapat dilakukan tanpa memengaruhi tabel lain yang tidak terkait langsung.

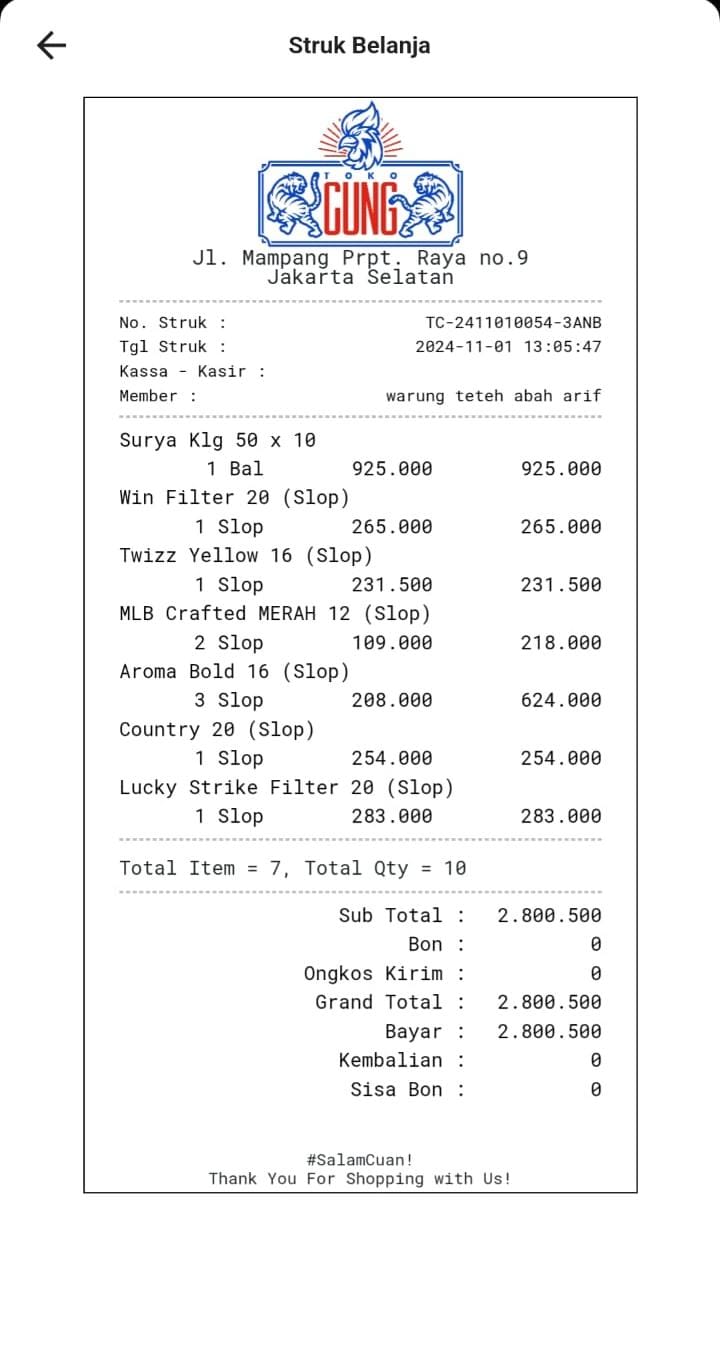
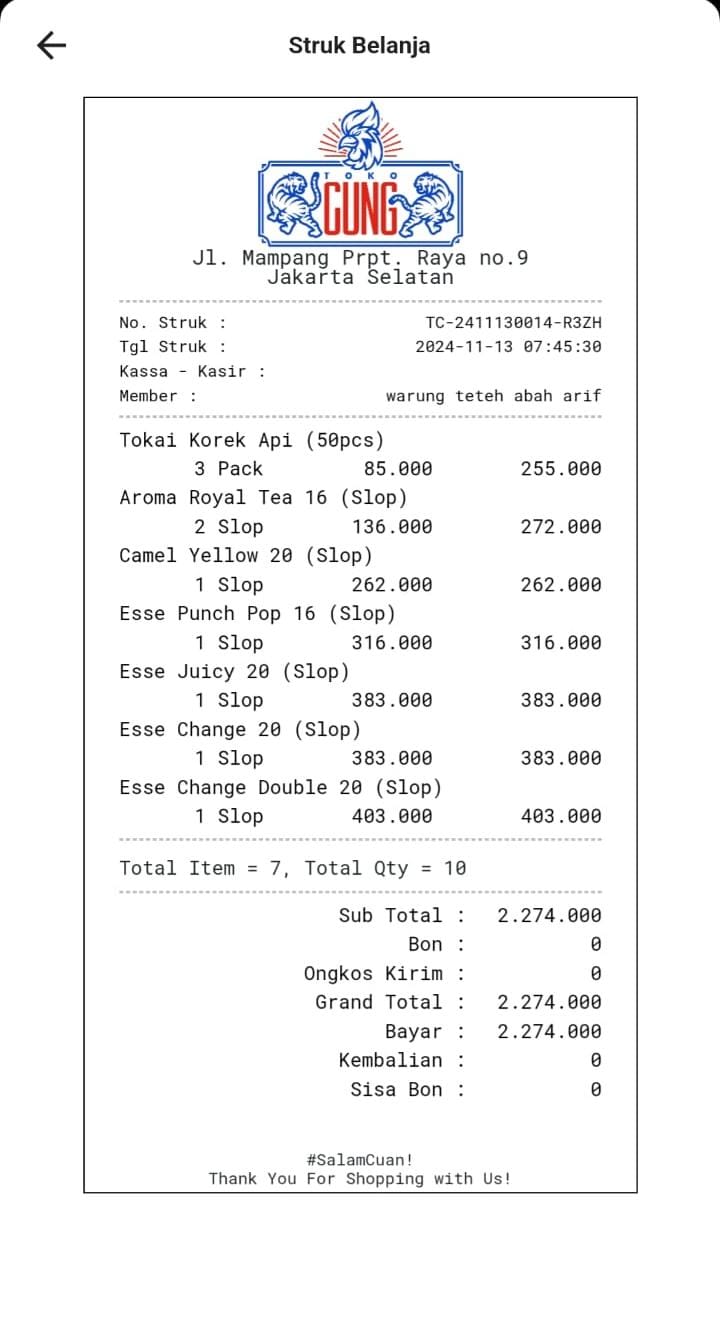
**Lingkup Proyek**

Studi ini berfokus pada normalisasi data dari sistem transaksi toko Cung dengan cakupan:

* Data barang (kode, nama, harga).
* Data transaksi (nomor struk, tanggal transaksi).
* Detail transaksi (kode barang, kuantitas, dan harga total per barang).
* Member (id\_member,nama\_member,kontak\_member,alamat\_member)
* Kasir (id\_kasir,nama\_kasir)

Dokumen Manual

Dokumen 1 Dan Dokumen 2



4.Batasan/Aturan/Asumsi

**1. Batasan / Aturan / Asumsi**

a. Transaksi melibatkan satu atau lebih barang yang dibeli.

b. Setiap transaksi dilakukan oleh satu kasir, dan bisa melibatkan member (opsional).

c. Jasa pengiriman tersedia untuk barang tertentu yang membutuhkan pengiriman.

d. Berat barang dalam transaksi dihitung berdasarkan kuantitas barang yang dibeli.

e. Setiap barang memiliki harga dan berat per unit yang tetap.

f. Member memiliki informasi kontak yang unik.

**2. Entitas, Relasi, Atribut, dan Keys**

a. Transaksi

**Atribut**  **Keterangan**

no\_struk (PK) : Nomor unik untuk setiap transaksi

tgl\_struk : Tanggal transaksi dilakukan

jam\_belanja : Waktu transaksi dilakukan

id\_kasir (FK) : ID kasir yang melayani transaksi

id\_member (FK) : ID member yang melakukan transaksi (opsional)

b. Kasir

Atribut : Keterangan

id\_kasir (PK) : ID unik untuk setiap kasir

nama\_kasir : Nama kasir

c. Member

Atribut Keterangan

id\_member (PK) : ID unik untuk setiap member

nama\_member : Nama member

kontak\_member : Nomor telepon atau email member

alamat\_member : Alamat lengkap member

d. Barang

Atribut : Keterangan

kode\_barang (PK) : ID unik untuk setiap barang

nama\_barang : Nama barang

harga\_barang : Harga barang per unit

stok : Jumlah stok barang

e. Detail Transaksi

Atribut : Keterangan

no\_struk (PK, FK) : Nomor transaksi yang terkait

kode\_barang (PK, FK): Kode barang dalam transaksi

kuantitas\_barang : Jumlah barang yang dibeli

harga\_TotalBarang: Total harga barang (kuantitas × harga barang)

5.Table Unnormalized



6.Tahapan normalisasi lengkap dengan depedensi fungsional

**Normalisasi Tahap 1:**

1. Eliminasi nilai null pada repeating groups
2. Identifikasi primary key (Suggestion)
3. Identifikasi seluruh dependensi (Suggestion)



**Primary Key:**

{no\_struk,id\_kasir,id\_member,kode\_barang}

{no\_struk,id\_kasir,id\_member,kode\_barang}>tgl\_struk,jam\_belanja,nama\_kasir,nama\_member,kontak\_member,alamat\_member,nama\_barang,harga\_barang,stok,kuantitas\_barang,harga\_TotalBarang

**Dependensi Fungsional Sebagian:**

no\_struk → (tgl\_struk, jam\_belanja, id\_kasir, id\_member)

(no\_struk, kode\_barang, kuantitas\_barang) → harga\_TotalBarang

id\_kasir → nama\_kasir

id\_member → nama\_member, kontak\_member, alamat\_member

kode\_barang → nama\_barang, harga\_barang, stok

**Normalisasi Tahap 2:**

1. Pastikan tabel sudah dalam bentuk normal pertama
2. Eliminasi depedensi fungsional sebagian
3. Tentukan atribut yang berdepedensi terhadap atribut lain



**Tabel Transaksi** Menggunakan Depedensi:

Dependensi Fungsional Sebagian

no\_struk → (tgl\_struk, jam\_belanja, id\_kasir, id\_member)



**Tabel Detail Transaksi** Menggunakan Depedensi:

Depedensi Fungsional Penuh

(no\_struk, kode\_barang) → kuantitas\_barang, harga\_TotalBarang



**Tabel Member** Menggunakan Depedensi:

Depedensi Fungsional

id\_member → nama\_member, kontak\_member, alamat\_member



**Table Kasir** Menggunakan Depedensi:

Depedensi Fungsional Penuh

id\_kasir → nama\_kasir



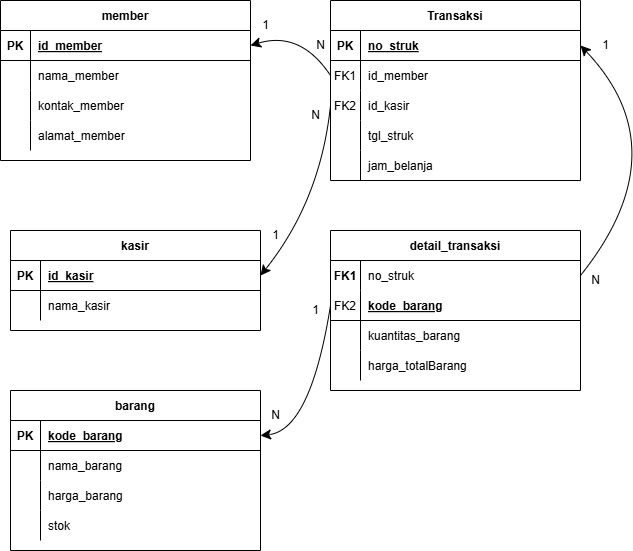
**Table Barang** Menggunakan Depedensi

Depedensi Depedensi Fungsional Penuh

kode\_barang → nama\_barang, harga\_barang, stok

|  |
| --- |
| **Relasi Tabel member ke transaksi**   * **Deskripsi: Relasi antara tabel member dan transaksi menunjukkan bahwa setiap *member* dapat melakukan satu atau lebih transaksi. Relasi ini terjadi melalui atribut id\_member, yang menjadi Primary Key di tabel member dan menjadi Foreign Key di tabel transaksi.** * **Jenis Relasi: One-to-Many** * **Interpretasi: Satu *member* dapat melakukan beberapa transaksi, tetapi satu transaksi hanya dapat dimiliki oleh satu *member*.**   **Relasi Tabel kasir ke transaksi**   * **Deskripsi: Relasi ini menjelaskan bahwa setiap transaksi dikelola oleh seorang *kasir*. Atribut id\_kasir menjadi Primary Key di tabel kasir dan menjadi Foreign Key di tabel transaksi.** * **Jenis Relasi: One-to-Many** * **Interpretasi: Seorang *kasir* dapat menangani banyak transaksi, tetapi satu transaksi hanya dikelola oleh satu *kasir*.**   **Relasi Tabel transaksi ke detail\_transaksi**   * **Deskripsi: Tabel transaksi terhubung dengan tabel detail\_transaksi melalui atribut no\_struk. Dalam hal ini, no\_struk adalah Primary Key di tabel transaksi dan menjadi salah satu bagian dari Composite Primary Key di tabel detail\_transaksi.** * **Jenis Relasi: One-to-Many** * **Interpretasi: Satu transaksi dapat mencakup banyak detail transaksi, tetapi satu detail transaksi hanya terkait dengan satu transaksi.**   **Relasi Tabel barang ke detail\_transaksi**   * **Deskripsi: Tabel barang dan detail\_transaksi terhubung melalui atribut kode\_barang. Atribut ini adalah Primary Key di tabel barang dan menjadi bagian dari Composite Primary Key di tabel detail\_transaksi.** * **Jenis Relasi: One-to-Many** * **Interpretasi: Satu barang dapat muncul pada banyak detail transaksi (karena dapat dijual berkali-kali dalam transaksi berbeda), tetapi satu detail transaksi hanya merujuk pada satu barang.**   **Relasi Tabel member, kasir, dan barang melalui detail\_transaksi**   * **Deskripsi: Tabel detail\_transaksi merupakan penghubung tidak langsung antara tabel member, kasir, dan barang. Data transaksi pelanggan (melalui member) dan kasir yang bertugas dikelola di tabel transaksi, sedangkan detail pembelian barang dicatat di tabel detail\_transaksi.** * **Jenis Relasi:**   + **member → transaksi → detail\_transaksi: 1**   + **kasir → transaksi → detail\_transaksi: 1**   + **barang → detail\_transaksi: 1** * **Interpretasi: Alur relasi ini mencerminkan bahwa setiap transaksi yang dilakukan pelanggan mencakup rincian barang yang dibeli dan dikelola oleh kasir tertentu.** |

**7.Diagram Relasi**

****