Laporan Basis Data Lanjutan 1NF 2NF 3NF

Nama : Ilham Tatayo Lie

Nim : 202265056

Repository Github : <https://github.com/BOcort>

**Laporan 3NF (Third Normal Form):**

1. **Deskripsi 3NF:**

* Sudah memenuhi syarat 2NF.
* Tidak ada ketergantungan transitif non-kunci.

1. **Penjelasan:**
2. **Tabel barang:**
   * Sudah memenuhi syarat 3NF karena tidak ada ketergantungan transitif non-kunci.
   * id adalah kolom dengan tipe data VARCHAR(3) yang berfungsi sebagai primary key.
   * nama\_barang adalah kolom dengan tipe data VARCHAR(80) yang menyimpan nama barang.
   * harga adalah kolom dengan tipe data DECIMAL(10,2) yang menyimpan harga barang dengan dua digit di belakang koma.
   * **Normalisasi:**
     + Tidak ada perubahan yang diperlukan.

CREATE TABLE barang (

id VARCHAR(3) PRIMARY KEY,

nama\_barang VARCHAR(80),

harga DECIMAL(10,2)

);

1. **Tabel customer:**

* Sudah memenuhi syarat 3NF karena tidak ada ketergantungan transitif non-kunci.
* id adalah kolom dengan tipe data VARCHAR(3) yang berfungsi sebagai primary key.
* nama adalah kolom dengan tipe data VARCHAR(80) yang menyimpan nama customer.
* **Normalisasi:**
  + Tidak ada perubahan yang diperlukan.

CREATE TABLE customer (

id VARCHAR(3) PRIMARY KEY,

nama VARCHAR(80)

);

1. **Tabel penjualan\_barang:**

* Sudah memenuhi syarat 3NF karena tidak ada ketergantungan transitif non-kunci.
* id adalah kolom dengan tipe data VARCHAR(4) yang memiliki nilai default 'j001' dan berfungsi sebagai primary key.
* tgl\_jual adalah kolom dengan tipe data DATE yang menyimpan tanggal penjualan.
* id\_customer adalah kolom dengan tipe data VARCHAR(3) yang memiliki nilai default 'j001' dan merupakan foreign key yang merujuk ke kolom id di tabel customer.
* **Normalisasi:**
  + Tidak ada perubahan yang diperlukan.

CREATE TABLE penjualan\_barang (

id VARCHAR(4) DEFAULT 'j001' PRIMARY KEY,

tgl\_jual DATE,

id\_customer VARCHAR(3) DEFAULT 'j001',

FOREIGN KEY (id\_customer) REFERENCES customer(id)

);

1. **Tabel detail\_penjualan:**

* Sudah memenuhi syarat 3NF karena tidak ada ketergantungan transitif non-kunci.
* id\_jual adalah kolom dengan tipe data VARCHAR(4) yang berfungsi sebagai primary key dan foreign key yang merujuk ke kolom id di tabel penjualan\_barang.
* id\_barang adalah kolom dengan tipe data VARCHAR(3) yang merupakan foreign key yang merujuk ke kolom id di tabel barang.
* qty adalah kolom dengan tipe data INT yang menyimpan jumlah barang yang dijual.
* **Normalisasi:**
  + Tidak ada perubahan yang diperlukan.

CREATE TABLE detail\_penjualan (

id\_jual VARCHAR(4) PRIMARY KEY,

id\_barang VARCHAR(3),

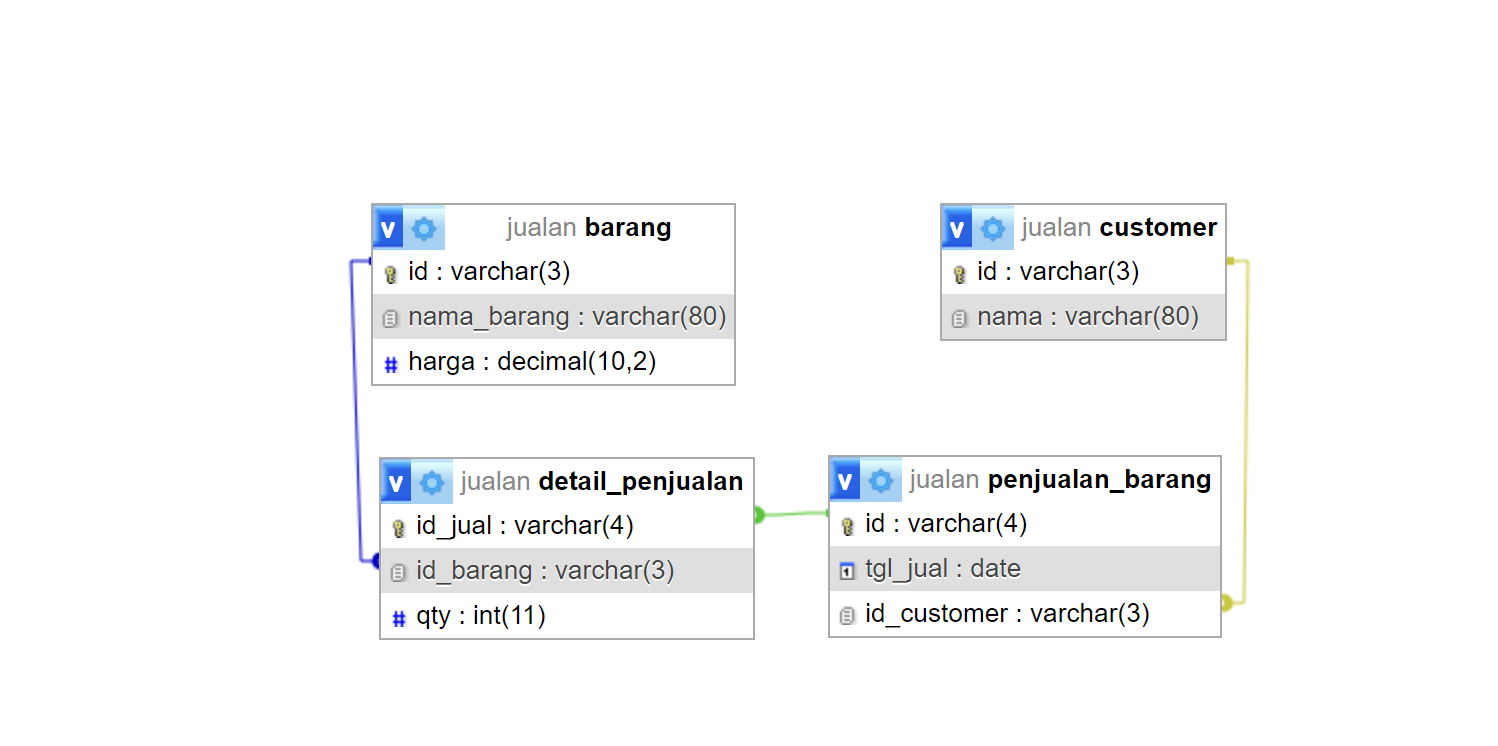
qty INT,

FOREIGN KEY (id\_jual) REFERENCES penjualan\_barang(id),

FOREIGN KEY (id\_barang) REFERENCES barang(id)

);

Disini kita dapat melihat erd hasil data yang kita buat pada phpMyAdmin



Data tersebut kita buat dalam dari hasil code yang kita buat

Selanjutnya kita menginput data yang akan kita berikan

1. **Menginput data yang telah diberikan**
2. **Contoh INSERT untuk Barang:**

* Pernyataan INSERT ini memasukkan tiga baris data ke dalam tabel barang.

INSERT INTO barang (id, nama\_barang, harga) VALUES

('A01', 'Naruto', 20000),

('A02', 'Chitato', 40000),

('A03', 'BatagorKu', 10000);

1. **Contoh INSERT untuk Customer:**

* Pernyataan INSERT ini memasukkan tiga baris data ke dalam tabel customer.

INSERT INTO customer (id, nama) VALUES

('001', 'Adi'),

('002', 'Dian'),

('003', 'Dadi');

Disini kita harus menginput data barang dan customer terlebih dahulu karena penjulan\_barang memiliki foreign key dengan data id customer

1. **Contoh INSERT untuk Penjualan\_barang:**
   * Pernyataan INSERT ini memasukkan tiga baris data ke dalam tabel penjualan\_barang.

INSERT INTO penjualan\_barang (id, tgl\_jual, id\_customer) VALUES

('j001', '2023-11-01', '001'),

('j002', '2023-11-01', '002'),

('j003', '2023-11-03', '003');

Disini kita baru harus menginput data pada penjualan barang terlebih dahulu karena table penjualan memiliki foreign key dengan data id penjualan\_barang dan id barang

1. **Contoh INSERT untuk Detail\_penjualan:**

* Pernyataan INSERT ini memasukkan tiga baris data ke dalam tabel detail\_penjualan.

INSERT INTO detail\_penjualan (id\_jual, id\_barang, qty) VALUES

('j001', 'A01', 1),

('j002', 'A02', 2),

('j003', 'A03', 1);

1. **Output dari data tersebut**
   * Kita pelu mengecek data yang dihasil kan data disini kita select.

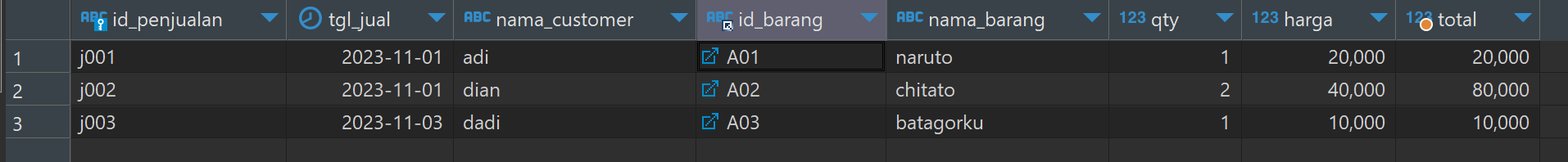
SELECT p.id AS id\_penjualan, p.tgl\_jual, c.nama AS nama\_customer, d.id\_barang, b.nama\_barang, d.qty, b.harga, (d.qty \* b.harga) AS total

FROM penjualan\_barang p

JOIN detail\_penjualan d ON p.id = d.id\_jual

JOIN barang b ON d.id\_barang = b.id

JOIN customer c ON p.id\_customer = c.id;



1. **Kesimpulan:**

Dari hasil analisis dan normalisasi basis data, dapat diambil beberapa kesimpulan:

1. **Struktur Tabel:**
   * Tabel barang, customer, penjualan\_barang, dan detail\_penjualan telah dirancang dengan baik dan mematuhi prinsip normalisasi 3NF.
   * Setiap tabel memiliki primary key yang unik dan tipe data yang sesuai.
2. **1NF, 2NF, 3NF:**
   * Setiap tabel memenuhi persyaratan 1NF, 2NF, dan 3NF.
   * Tidak ada data yang redundan atau anomali dalam basis data.
3. **Foreign Key:**
   * Hubungan antara tabel-tabel dijaga dengan baik menggunakan foreign key. Tabel penjualan\_barang menggunakan foreign key yang merujuk ke tabel customer, dan tabel detail\_penjualan menggunakan foreign key yang merujuk ke tabel penjualan\_barang dan barang.
4. **Normalisasi:**
   * Normalisasi telah dilakukan dengan memastikan bahwa setiap tabel dalam bentuk tereduksi dan tidak ada ketergantungan transitif non-kunci.
5. **Insert Data:**
   * Proses input data diatur dengan baik, mengikuti urutan yang benar untuk menghindari pelanggaran integritas referensial.

Dengan demikian, basis data ini telah dirancang untuk menghindari duplikasi data, meminimalkan anomali, dan mendukung integritas data. Normalisasi hingga 3NF memberikan struktur yang efisien dan konsisten untuk menyimpan dan mengelola informasi terkait barang, customer, penjualan\_barang, dan detail\_penjualan.