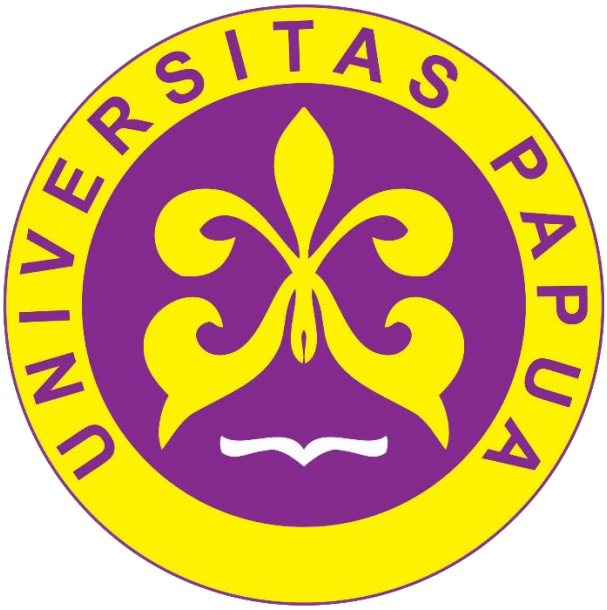
Laporan Praktikum

Basis Data Lanjutan

Normalisasi



DISUSUN OLEH

Nama : Iqbal Dwi Wijanarko

NIM : 202265046

Kelas : B

Github : https://github.com/IqbaLe0/normalisasi-DBL

S1 Teknik Informatika

Fakultas Teknik

Universitas Papua

A. Normalisasi

Normalisasi database adalah proses mengorganisir tabel dalam database relasional untuk mengurangi redudansi dan memastikan data dapat diakses dengan mudah dan efisien. Normalisasi database dibagi menjadi beberapa bentuk normalisasi, yaitu 1NF, 2NF, dan 3NF.

- 1NF (First Normal Form) memastikan bahwa setiap atribut dalam tabel hanya memiliki satu nilai dalam satu baris.
- 2NF (Second Normal Form) memastikan bahwa setiap atribut dalam tabel bergantung pada kunci utama tabel.
- 3NF (Third Normal Form) memastikan bahwa setiap atribut dalam tabel bergantung pada kunci utama tabel dan tidak bergantung pada atribut lain dalam tabel.

Untuk melakukan normalisasi database, kita harus mengidentifikasi seperti apa bentuk data yang saat ini disimpan. Setelah itu, kita dapat melakukan normalisasi 1NF, 2NF, dan 3NF secara berurutan. Tahap normalisasi 1NF memastikan bahwa setiap atribut dalam tabel hanya memiliki satu nilai dalam satu baris. Tahap normalisasi 2NF memastikan bahwa setiap atribut dalam tabel bergantung pada kunci utama tabel. Tahap normalisasi 3NF memastikan bahwa setiap atribut dalam tabel bergantung pada kunci utama tabel dan tidak bergantung pada atribut lain dalam tabel.

B. Penjelasan

1NF :

penjualan_brg

no_jual	tgl_jual	kode_brg	nama_brg	harga	id_customer	nama_customer	qty_penjualan
j001	02/03/2020	b001	Lemari	13000000	c001	Ryan	1
j001	12/02/2020	b002	Kulkas	23000000	c001	Ryan	2
j002	02/13/2020	b001	lemari	13000000	c002	Mozez	1

- Sudah memenuhi syarat 1NF karena setiap atribut dalam tabel hanya memiliki satu nilai dalam satu baris.
- Setiap nama atribut sudah unik maksudnya tidak ada yang sama

2NF :

penjualan_brg

no_jual	tgl_jual	kode_brg	nama_brg	harga	id_customer	nama_customer	qty_penjualan
j001	02/03/2020	b001	Lemari	13000000	c001	Ryan	1
j001	12/02/2020	b002	Kulkas	23000000	c001	Ryan	2
j002	02/13/2020	b001	lemari	13000000	c002	Mozez	1

- Kita bisa melihat bahwa tabel diatas sudah berada dalam 1NF
- Selanjutnya pisahkan kolom nama_barang dan harga ke tabel baru yang disebut barang, karena mereka bergantung pada kode_barang dan bukan pada no_jual. Buat kode_barang sebagai primary key tabel barang.
- Pisahkan kolom nama_customer ke tabel baru yang disebut customer, karena dia bergantung pada id_customer dan bukan pada no_jual. Buat id_customer sebagai primary key tabel customer.
- no_jual tidak bisa menjadi primary key karena tidak unik
- Tambahkan foreign key kode_barang dan id_customer ke tabel penjualan_brg untuk menghubungkannya dengan tabel barang dan customer.

sehingga tabel yang dihasilkan adalah sebagai berikut :

barang

kode_brg	nama_brg	harga
b001	Lemari	13000000
b002	Kulkas	23000000

Pk:kode_barang

customer

id_customer	nama_customer
c001	Ryan
c002	Mozez

Pk:id_customer

penjualan_brg

no_jual	tgl_jual	kode_brg	id_customer	qty_penjualan
j001	02/03/2020	b001	c001	1
j001	12/02/2020	b002	c001	2
j002	02/13/2020	b001	c002	1

Fk : kode_barang, id_customer

dari tabel diatas no_jual tidak bisa menjadi primary key karena tidak unik

3NF :

barang

kode_brg	nama_brg	harga
b001	Lemari	13000000
b002	Kulkas	23000000

customer

id_customer	nama_customer
c001	Ryan
c002	Mozez

no_jual	tgl_jual	kode_brg	id_customer	qty_penjualan
j001	02/03/2020	b001	c001	1

j001	12/02/2020	b002	c001	2
j002	02/13/2020	b001	c002	1

Pk:kode_barang Pk:id_customer

penjualan_brg

- Kita bisa melihat bahwa tabel diatas sudah berada dalam 2NF
- Pada 2NF, tabel penjualan_brg memiliki atribut qty_penjualan yang bergantung secara transitif pada atribut no_jual. Ini melanggar aturan 3NF yang mengharuskan semua atribut non-primer hanya bergantung secara fungsional pada kunci primer.
- Untuk memenuhi 3NF, tabel penjualan_brg dibagi menjadi dua tabel: penjualan_brg dan detail_penjualan_brg.
- Tabel penjualan_brg hanya berisi atribut no_jual, tgl_jual, dan id_customer, yang dimana no_jual adalah primary key dan id_customer adalah foreign key.
- Tabel detail_penjualan_brg berisi atribut no_jual, kode_brg, dan qty_penjualan, yang merupakan primary key(no_jual & kode_brg) dan foreign key(no_jual & kode_brg).
- Dengan demikian, semua atribut non-primer hanya bergantung pada kunci primer secara langsung, dan tidak ada ketergantungan transitif.

Tabel yang Dihasilkan :

barang

kode_brg	nama_brg	harga
b001	Lemari	13000000
b002	Kulkas	23000000

PK : kode_brg

customer

id_customer	nama_customer
c001	Ryan
c002	Mozez

PK: id_customer

penjualan_brg

no_jual	tgl_jual	id_customer
j001	02/03/2020	c001
j001	12/02/2020	c001
j002	02/13/2020	c002

PK: no_jual

FK: id_customer

detail_penjualan_brg

no_jual	kode_brg	qty_penjualan
j001	b001	1
j001	b002	2
j002	b001	1

PK: no_jual , kode_brg

FK: kode_brg , no_jual

C. Relasi

- tabel barang, memiliki tiga kolom: kode_barang, nama_barang, dan harga. Kolom kode_barang adalah kunci utama dari tabel ini.
- Tabel customer, memiliki dua kolom: id_customer, dan nama_customer. Kolom id_customer adalah kunci utama dari tabel ini.
- Tabel detail_penjualan_brg, memiliki tiga kolom: no_jual, kode_barang, dan qty_penjualan. Kolom no_jual dan kode_barang adalah kunci utama dari tabel ini.
- Tabel penjualan_brg, memiliki tiga kolom : no_jual, tgl_jual, dan id_customer. Kolom no_jual adalah kunci utama dari tabel ini

Dari tabel tersebut, kita bisa melihat bahwa ada tiga relasi yang terbentuk, yaitu:

- Relasi satu-ke-banyak antara tabel customer dan tabel penjualan_brg, di mana setiap customer bisa memiliki banyak penjualan_brg, tetapi setiap penjualan_brg hanya dimiliki oleh satu customer. Relasi ini dibuat dengan menggunakan kolom id_customer sebagai kunci asing di tabel penjualan_brg yang merujuk pada kolom id_customer sebagai kunci utama di tabel customer.
- Relasi satu-ke-banyak antara tabel barang dan tabel detail_penjualan_brg, di mana setiap barang bisa memiliki banyak detail_penjualan_brg, tetapi setiap detail_penjualan_brg hanya dimiliki oleh satu barang. Relasi ini dibuat dengan menggunakan kolom kode_barang sebagai kunci asing di tabel detail_penjualan_brg yang merujuk pada kolom kode_barang sebagai kunci utama di tabel barang.
- Relasi satu-ke-banyak antara tabel penjualan_brg dan tabel detail_penjualan_brg, di mana setiap penjualan_brg bisa memiliki banyak detail_penjualan_brg, tetapi setiap detail_penjualan_brg hanya dimiliki oleh satu penjualan_brg. Relasi ini dibuat dengan menggunakan kolom no_jual sebagai kunci asing di tabel detail_penjualan_brg yang merujuk pada kolom no_jual sebagai kunci utama di tabel penjualan_brg.

D. Memasukkan data ke dalam tabel

1. Tabel barang

```
INSERT INTO barang (`kode_brg`, `nama_brg`, `harga`) VALUES
(‘b001’ , ’lemari’ , 13000000),
(‘b002’ , ‘kulkas’ , 23000000);
```

2. Tabel customer

```
INSERT INTO customer (`id_customer `, `nama_customer`) VALUES
(‘c001’ , ’Ryan’),
(‘c002’ , ‘Mozez’);
```

3. Tabel penjualan_brg
   INSERT INTO `penjualan_brg`(`no_jual`, `tgl_jual`, `id_customer`) VALUES
   (‘j001’ , ‘2020-02-03’ , ‘c001’),
   (‘j002’ , ‘2020-02-13 , ‘c002’);
4. Tabel detail_penjualan_brg
   INSERT INTO `detail_penjualan_brg`(`no_jual`, `kode_brg`, `qty_penjualan`) VALUES
   (‘j001’ , ‘b001’ , 1),
   (‘j001’ , ‘b002’ , 2),
   (‘j002’ , ‘b001’ , 1);

E. Output

- Query

  select penjualan_brg.no_jual, penjualan_brg.tgl_jual, barang.kode_brg, barang.nama_brg, barang.harga, customer.id_customer, customer.nama_customer, detail_penjualan_brg.qty_penjualan

  from penjualan_brg

  join detail_penjualan_brg on penjualan_brg.no_jual = detail_penjualan_brg.no_jual

  join barang on detail_penjualan_brg.kode_brg = barang.kode_brg

  join customer on penjualan_brg.id_customer = customer.id_customer;

- Hasil

no_jual	tgl_jual	kode_brg	nama_brg	harga	id_customer	nama_customer	qty_penjualan
j001	2020-02-03	b001	lemari	13000000	c001	Ryan	1
j001	2020-02-03	b002	kulkas	23000000	c001	Ryan	2
j002	2020-02-13	b001	lemari	13000000	c002	Mozez	1

F. Kesimpulan

Kesimpulan yang saya peroleh dari laporan ini adalah sebagai berikut :

- Normalisasi database adalah proses mengorganisir tabel dalam database relasional untuk mengurangi redudansi dan memastikan data dapat diakses dengan mudah dan efisien.
- Normalisasi database dibagi menjadi beberapa bentuk normalisasi, yaitu 1NF, 2NF, dan 3NF, yang harus dilakukan secara berurutan.
- 1NF memastikan bahwa setiap atribut dalam tabel hanya memiliki satu nilai dalam satu baris.
- 2NF memastikan bahwa setiap atribut dalam tabel bergantung pada kunci utama tabel.
- 3NF memastikan bahwa setiap atribut dalam tabel bergantung pada kunci utama tabel dan tidak bergantung pada atribut lain dalam tabel.
- Untuk melakukan normalisasi database, kita harus mengidentifikasi seperti apa bentuk data yang saat ini disimpan, dan kemudian memisahkan atribut-

atribut yang melanggar aturan normalisasi ke tabel-tabel baru yang terhubung dengan kunci asing.

- Keuntungan dari normalisasi database adalah mengurangi redundansi data, menghemat ruang penyimpanan, dan meningkatkan integritas data. Kerugian dari normalisasi database adalah meningkatkan kompleksitas query, karena memerlukan operasi join untuk menggabungkan tabel yang terpisah.