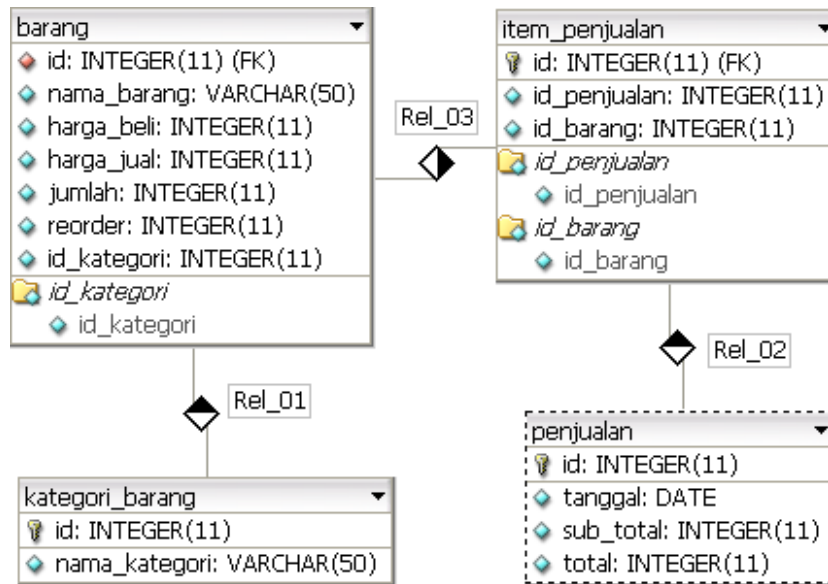


Materi Tambahan Praktikum PBO-II

LATIHAN SQL



Gambar 1. Relasi sistem POS sederhana

- Gambar 1 di atas adalah EER (*Enhanced Entity Relationship*) Diagram dari sistem POS sederhana. Untuk membuat relasi di atas, query yang digunakan adalah sebagai berikut :

```
CREATE TABLE kategori_barang(  
id INT AUTO_INCREMENT PRIMARY KEY NOT NULL,  
nama_kategori VARCHAR(50) NOT NULL  
);  
  
CREATE TABLE barang(  
id INT AUTO_INCREMENT PRIMARY KEY NOT NULL,  
nama_barang VARCHAR(50) NOT NULL,  
harga_beli INT NOT NULL,  
harga_jual INT NOT NULL,  
jumlah INT NOT NULL,  
reorder INT NOT NULL,  
id_kategori INT NOT NULL,  
FOREIGN KEY (id_kategori) REFERENCES kategori(id) ON DELETE CASCADE ON  
UPDATE CASCADE  
);
```

```
CREATE TABLE item_penjualan(
id INT AUTO_INCREMENT PRIMARY KEY NOT NULL,
id_penjualan INT NOT NULL,
id_barang INT NOT NULL,
FOREIGN KEY (id_penjualan) REFERENCES penjualan(id),
FOREIGN KEY (id_barang) REFERENCES barang(id)
);
```

```
CREATE TABLE penjualan(
id INT AUTO_INCREMENT PRIMARY KEY NOT NULL,
tanggal DATE NOT NULL,
sub_total INT,
total INT
);
```

- Untuk pengisian data, dilakukan pada tabel relasi yang paling luar (kategori_barang dan penjualan) baru kemudian dilakukan pada tabel berikutnya yang berelasi dengannya (misal kategori_barang, dilanjutkan dengan barang).

```
INSERT INTO kategori_barang(nama_kategori) VALUES('makanan');
INSERT INTO kategori_barang(nama_kategori) VALUES('minuman');
```

- Untuk seleksi data, dilakukan dengan perintah SELECT sebagai berikut

```
mysql> SELECT * FROM kategori_barang;
```

```
+-----+-----+
| id | nama_kategori |
+-----+-----+
|  1 | makanan      |
|  2 | minuman      |
+-----+-----+
2 rows in set (0.26 sec)
```

```
mysql> SELECT id AS "ID Kategori",
-> nama_kategori AS "Nama Kategori Barang"
-> FROM kategori_barang;
```

```
+-----+-----+
| ID Kategori | Nama Kategori Barang |
+-----+-----+
|           1 | makanan              |
|           2 | minuman              |
+-----+-----+
2 rows in set (0.01 sec)
```

```
mysql> SELECT * FROM kategori_barang WHERE id=1;
```

```
+-----+-----+
| id | nama_kategori |
+-----+-----+
|  1 | makanan      |
+-----+-----+
1 row in set (0.37 sec)
```

- Dari tabel di atas, diketahui makanan dan minuman mempunyai ID masing-masing 1 dan 2. Untuk mengisikan data barang (baik makanan maupun minuman) digunakan ID yang bersesuaian. Contoh

```
INSERT INTO
barang(nama_barang,harga_beli,harga_jual, jumlah,reorder, id_kategori)
VALUES( 'Chitato',1000,2000,10,2,1);
```

```
INSERT INTO
barang(nama_barang,harga_beli,harga_jual, jumlah,reorder, id_kategori)
VALUES( 'Pocari Sweat',1000,2000,10,2,2);
```

```
INSERT INTO
barang(nama_barang,harga_beli,harga_jual, jumlah,reorder, id_kategori)
VALUES( 'Calpico',1000,2000,10,2,2);
```

- Dari proses insert di atas, terlihat bahwa harga_jual dan harga_beli masing-masing produk di atas sama. Anda bisa meng-editnya (tanpa menghapus) dengan perintah UPDATE. Contoh :

```
UPDATE barang SET harga_beli=1500,harga_jual=3000 WHERE id=2;
```

SQL di atas adalah perintah untuk meng-update harga_beli dan harga_jual pada barang dengan ID=2 (silakan disesuaikan dengan ID pada tabel barang anda).

- Untuk menghapus sebuah data, bisa dilakukan dengan perintah DELETE. Misal untuk menghapus barang dengan ID=4 :

```
DELETE FROM barang WHERE id=4;
```

- Kita bisa menggabungkan beberapa tabel berdasarkan kunci yang menghubungkan tabel tersebut. Contoh :

```
mysql> SELECT b.nama_barang, kb.nama_kategori
-> FROM barang b, kategori_barang kb
-> WHERE b.id_kategori=kb.id;
```

```
+-----+-----+
| nama_barang      | nama_kategori |
+-----+-----+
| gethuk magelang | makanan      |
| Chitato          | makanan      |
| Pocari Sweat     | minuman      |
+-----+-----+
3 rows in set (0.00 sec)
```

SQL query di atas melakukan seleksi khusus untuk kolom nama_barang pada tabel barang (diidentifikasi dengan 'b') dan kolom nama_kategori pada tabel kategori_barang ((diidentifikasi dengan 'kb')).

- Anda bisa menggunakan statement JOIN untuk menggabungkan isi tabel sebagai berikut :

```
mysql> SELECT b.nama_barang, kb.nama_kategori
-> FROM barang b JOIN kategori_barang kb
```

```
-> ON b.id_kategori=kb.id;
```

nama_barang	nama_kategori
gethuk magelang	makanan
Chitato	makanan
Pocari Sweat	minuman

3 rows in set (0.00 sec)

- Anda bisa juga menggunakan statement LEFT dan RIGHT JOIN. Di sini kita akan menggunakan RIGHT JOIN untuk membedakan dengan statement JOIN sebelumnya, tambahkan satu data pada kategori_barang.

```
INSERT INTO kategori_barang(nama_kategori) VALUES('sabun cuci');
```

```
mysql> SELECT b.nama_barang, kb.nama_kategori
-> FROM barang b RIGHT JOIN kategori_barang kb
-> ON b.id_kategori=kb.id;
```

nama_barang	nama_kategori
gethuk magelang	makanan
Chitato	makanan
Pocari Sweat	minuman
NULL	sabun cuci

4 rows in set (0.00 sec)

Bisa menyimpulkan perbedaannya ?

LATIHAN

1. Masukkan data pada tabel “penjualan” dan “item_penjualan” (ikuti cara di atas). Perhatikan bahwa kolom sub_total adalah
2. Lakukan proses seleksi semua item (dengan SELECT * FROM...) dan beberapa item tertentu (dengan SELECT kolom1,kolom2 FROM)pada masing-masing tabel !
3. Lakukan seleksi gabungan tabel sehingga menampilkan :
 - a. Tanggal,total harga,id barang
 - b. Tanggal , nama barang dan total harga
 - c. Tanggal , nama barang, kategori barang dan total harga