**ULANGAN BASIS DATA**

**PADA SISTEM INFORMASI (APLIKASI)**

**PETUGASAN PERPUSTAKAAN SMK 1 KARANG BARU**

****

**OLEH:**

**RIZKA AULIA**

**NISN. 91287491261**

**REKAYASA PERANGKAT LUNAK**

**SMK NEGERI 1 KARANG BARU**

**PEMERINTAH PROVINSI ACEH**

**2025**

Pada studi kasus kali ini saya akan coba membuat perancangan basis data menggunakan pemodelan ERD dengan aturan-aturan berikut:

* Seorang petugas dapat menjual banyak layanan
* Seorang pembeli dapat membeli lebih dari satu layanan yang dijual oleh petugas
* Seorang petugas dapat merekap banyak transaksi
* Transaksi memiliki detiltransaksi
* Stok layanan akan berkurang sesuai jumlah layanan yang dibeli
* Nomor telepon petugas boleh lebih dari satu
* Petugas, pembeli, layanan dan transaksi dapat di identifikasi dengan id yang berbeda (unik)

**Langkah-langkah Membuat ERD**

1. Menentukan entitas
2. Menentukan atribut termasuk atribut kunci
3. Identifikasi relasi
4. Menentukan kardinalitas

**Menentukan Entitas**

Berdasarkan aturan-aturan yang di definisikan di atas dapat kita tentukan jumlah entitas ada sebanyak 3 yakni:

1. Admin
2. Peminjam
3. Buku
4. Kategori
5. Denda
6. Peminjaman
7. Detailpeminjaman

**Menentukan Atribut**

Selanjutnya dari ketiga entitas tersebut kita jabarkan atribut-atribut yang melekat pada masing-masing entitas. Atribut yang bersifat unik akan di jadikan sebagai atribut kunci (*primary key*).

**1 admin**

1. id\_admin: INT(Primary Key)
2. nama\_admin: VARCHAR(100)
3. username: VARCHAR(100)
4. password: VARCHAR(100)
5. Foto : VARCHAR(255)

**2. peminjam**

1. id\_peminjam: ITN(Primary Key)
2. nama\_peminjam: VARCHAR(100)
3. alamat: TEXT
4. no\_telp: VARCHAR(20)
5. Foto:VARCHAR(255)

3 **buku**

1. id\_buku: VARCHAR(50) (Primary Key)
2. judul: VARCHAR(255)
3. pengarang: VARCHAR(100)
4. penerbit: VARCHAR(100)
5. tahun\_terbit: YEAR atau INT
6. stok: INT
7. id\_kategori: INT atau VARCHAR(5) (Foreign Key)
8. foto:VARCHAR:(255)

**4 kategori**

1. id\_kategori: VARCHAR(50) (Primary Key)
2. nama\_kategori: VARCHAR(100)

**5. denda**

1. id\_denda: varchar(Primary Key
2. jumlah\_denda: DECIMAL(10,2)
3. status\_pembayaran: ENUM('Lunas', 'Belum Lunas') atau VARCHAR(50)

6 **peminjaman**

1. id\_peminjaman: INT atau VARCHAR(50) (Primary Key)
2. id\_peminjam: INT atau VARCHAR(50) (Foreign Key)
3. id\_admin: INT atau VARCHAR(50) (Foreign Key)
4. tanggal\_pinjam: DATE
5. tanggal\_kembali: DATE
6. status\_peminjaman: ENUM('Dipinjam', 'Selesai') atau VARCHAR(50)

7 **detailpeminjaman**

1. id\_detailpeminjaman: varchar (50) Primary Key
2. id\_peminjaman:VARCHAR(50) (Foreign Key)
3. id\_buku: VARCHAR(50) (Foreign Key)
4. status\_pengembalian: ENUM('Sudah Kembali', 'Belum Kembali') atau VARCHAR(50)

*Atribut dengan kode (PK) akan menjadi atribut kunci (primary key) pada masing-masing entitas.*

**Menentukan Relasi & Kardinalitasnya**

**1. pinjam**

* Merupakan relasi antara entitas peminjam dan buku yang berarti setiap peminjam boleh meminjam beberapaa buku.
* Kardinalitas relasi antara petugas dan layanan adalah*one to many* karena seorang peminjam dapat meminjam banyak buku.

**2. catat**

* Relasi catat adalah hubungan antara entitas Admin dan Buku, yang berarti Admin (*petugas*) bertanggung jawab mencatat atau mengelola data (seperti input data, stok, atau status) Buku.

**-**kardinalitas relasi antaraa admin dan buku adalah one two many karna admin dapat mencata banyak buku

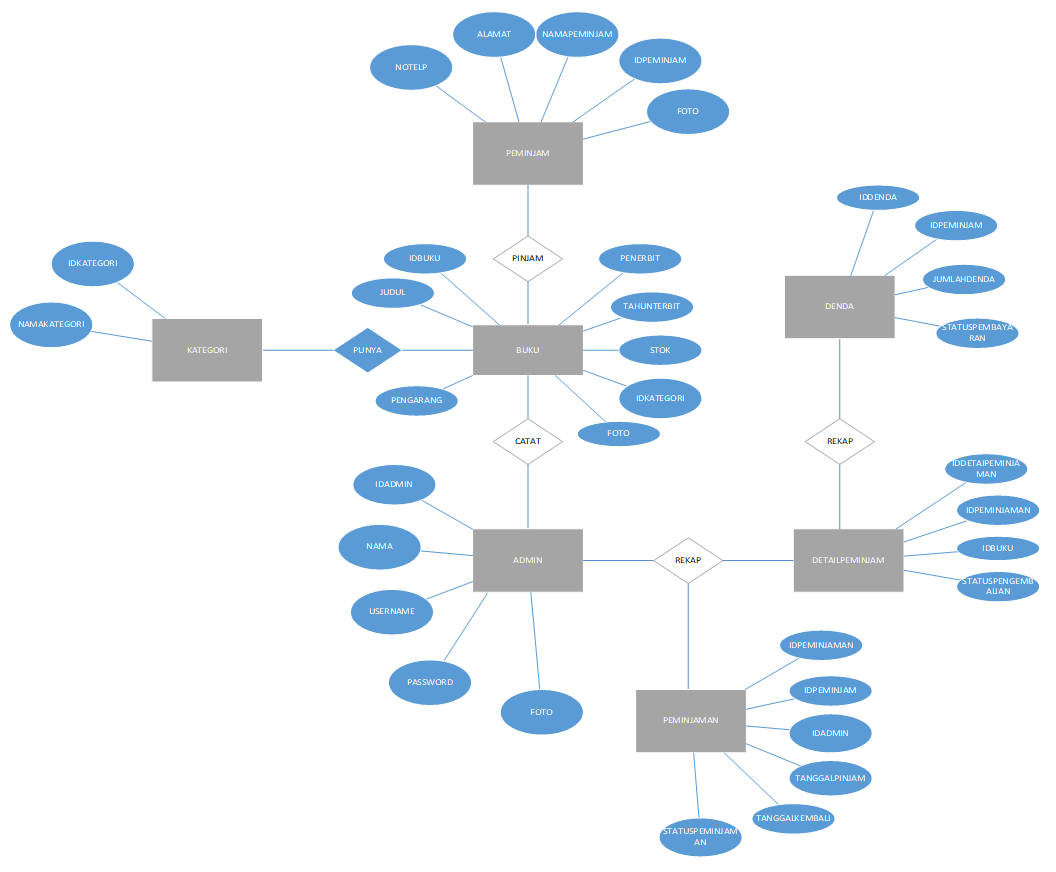
Sedangkan 1 buku hanya dicatat 1 admin yang bertanggung jawab

**3. Rekap**

* Merupakan relasi antara entitas admin dengan peminjaman ,detil transaksi dan denda yang berarti admin dapat melakukan rekap transaksi yang dibuat oleh admin
* Kardinalitas relasi antara peminjaman dengan transaksi dan detil transaksi adalah *one to many* karena seorang admin dapat merekap banyak transaksi dan detil transaksi atau dengan kata lain banyak transaksi dan detil transaksi boleh di rekap oleh seorang admin.

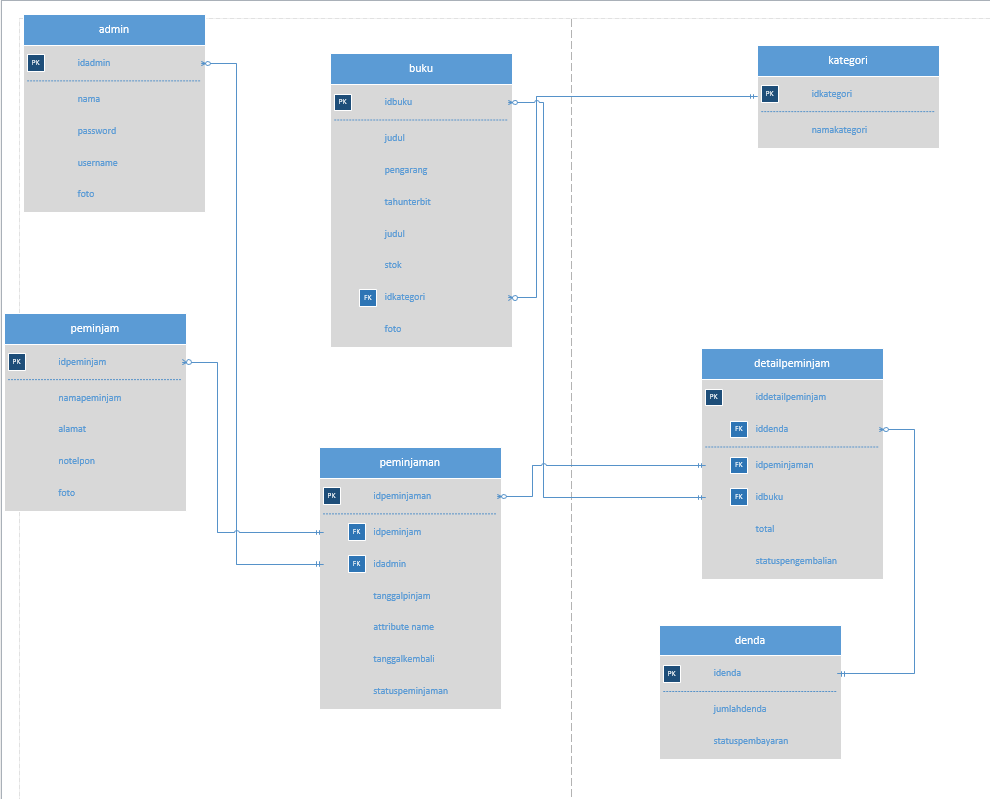
**Hasil ERD perpustakaan**

Dari tahap-tahap di atas maka dapat di buat rancangan ERD perpustakaan adalah sebagai berikut:

****

**DESAIN LOGIKAL**

Desain logikal yaitu proses pembuatan model dari informasi yang digunakan perusahaan berdasarkan model dan data spesifik. Deskripsi implementasi *database* berdasarkan hasil desain logikal dengan *Entity Relationship Diagram* (ERD) pada *Database Management System* (DBMS) menghasilkan ERT sebagai berikut



**DESAIN FISIKAL & SOURCE SQL**

**Menggambarkan Rancangan Entitas Pada Basisdata Secara Fisikal (Physical Data Disaign) serta Membuat Source SQL pembuatan Masing-Masing Tabel/Entitas**

**Tabel petugas/admin**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Nama kolom** | **Tipe data** | **Lebar** | **Null** | **kunci** | **keterangan** |
| 1 | **id\_admin** | int | 50 | not null | primarykey | auto increment |
| 2 | nama | varchar | 100 |  |  |  |
| 3 | username | varchar | 100 |  |  |  |
| 4 | password | varchar | 100 |  |  |  |
| 5 | poto | varchar | 255 |  |  |  |

CREATE TABLE admin (

id\_admin VARCHAR(50) PRIMARY KEY,

nama VARCHAR(100),

password VARCHAR(100),

username VARCHAR(100),

foto VARCHAR(255)

);

**Tabel peminjam**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Nama kolom** | **Tipe data** | **Lebar** | **Null** | **kunci** | **keterangan** |
| 1 | **id\_peminjam** | int | 50 | not null | primarykey | auto increment |
| 2 | nama\_peminjam | varchar | 100 |  |  |  |
| 3 | alamat | text | 255 |  |  |  |
| 4 | No,telp | varchar | 20 |  |  |  |
| 5 | foto | varchar | 255 |  |  |  |

CREATE TABLE peminjam (

id\_peminjam VARCHAR(50) PRIMARY KEY,

nama\_peminjam VARCHAR(100),

alamat VARCHAR(255),

no telpon VARCHAR(20),

foto VARCHAR(255)

);

**Table buku**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Nama kolom** | **Tipe data** | **Lebar** | **Null** | **kunci** | **keterangan** |
| 1 | **id\_buku** | int | 50 | not null | primarykey | auto increment |
| 2 | judul | varchar | 255 |  |  |  |
| 3 | pengarang | varchar | 255 |  |  |  |
| 4 | Tahun\_penerbit | Int/year | 100 |  |  |  |
| 5 | stok | int |  |  |  |  |
| 6 | foto | int | 255 |  |  |  |
| 7 | Id\_kategori | Int/varchar | 50 |  | Foreign key |  |

CREATE TABLE buku ( id\_buku VARCHAR(50) PRIMARY KEY, judul VARCHAR(255), pengarang VARCHAR(255), tahun\_terbit INT, stok INT, foto VARCHAR(255), id\_kategori VARCHAR(50), FOREIGN KEY (id\_kategori) REFERENCES kategori(id\_kategori) );

**Tabel kategori**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Nama kolom** | **Tipe data** | **Lebar** | **Null** | **kunci** | **keterangan** |
| 1 | **id\_kategori** | varchar | 50 | not null | primarykey | auto increment |
| 2 | Nama\_kategori | varchar | 100 |  |  |  |

CREATE TABLE kategori (

id\_kategori VARCHAR(50) PRIMARY KEY,

nama\_kategori VARCHAR(100)

);

**Tabel peminjaman**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Nama kolom** | **Tipe data** | **Lebar** | **Null** | **kunci** | **keterangan** |
| 1 | id\_peminjaman | varchar | 50 | not null | Primary key |  |
| 2 | id\_peminjam | varchar | 50 |  | Foreign key |  |
| 3 | Id\_admin | varchar | 50 |  | Foreign key |  |
| 4 | Tanggal\_pinjam | date |  |  |  |  |
| 5 | Tanggal\_kembali | date |  |  |  |  |
| 6 | Status\_peminjaman | varchar | 50 |  |  |  |

CREATE TABLE peminjaman ( id\_peminjaman VARCHAR(50) PRIMARY KEY, id\_peminjam VARCHAR(50), id\_admin VARCHAR(50), tanggal\_pinjam DATE, tanggal\_kembali DATE, status\_peminjaman VARCHAR(50), FOREIGN KEY (id\_peminjam) REFERENCES peminjam(id\_peminjam), FOREIGN KEY (id\_admin) REFERENCES admin(id\_admin) );

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Nama kolom** | **Tipe data** | **Lebar** | **Null** | **kunci** | **keterangan** |
| 1 | *Id\_denda* | varchar | 50 |  |  |  |
| 2 | *Jumlah\_denda* | desimal | 10,2 |  |  |  |
| 3 | Status\_pembayaran | varchar | 50 |  |  |  |

CREATE TABLE denda ( id\_denda VARCHAR(50) PRIMARY KEY, jumlah\_denda DECIMAL(10, 2), status\_pembayaran VARCHAR(50) );

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Nama kolom** | **Tipe data** | **Lebar** | **Null** | **kunci** | **keterangan** |
| 1 | Id\_detail\_peminjaman | varchar | 50 |  | Primary key |  |
| 2 | Id\_peminjaman | varchar | 50 |  | Foreign key |  |
| 3 | Id\_buku | varchar | 50 |  | Foreign key |  |
| 4 | Id\_denda | varchar | 50 |  | Foreign key |  |
| 5 | total | decimal | 10,2 |  |  |  |
| 6 | Status\_pengembalian | varchar | 50 |  |  |  |

CREATE TABLE detail\_peminjaman ( id\_detail\_peminjaman VARCHAR(50) PRIMARY KEY, id\_peminjaman VARCHAR(50), id\_buku VARCHAR(50), id\_denda VARCHAR(50), total DECIMAL(10, 2), status\_pengembalian VARCHAR(50), FOREIGN KEY (id\_peminjaman) REFERENCES peminjaman(id\_peminjaman), FOREIGN KEY (id\_buku) REFERENCES buku(id\_buku), FOREIGN KEY (id\_denda) REFERENCES denda(id\_denda) );

**Implementasi Syntax Sql Database Melalui Cmd**

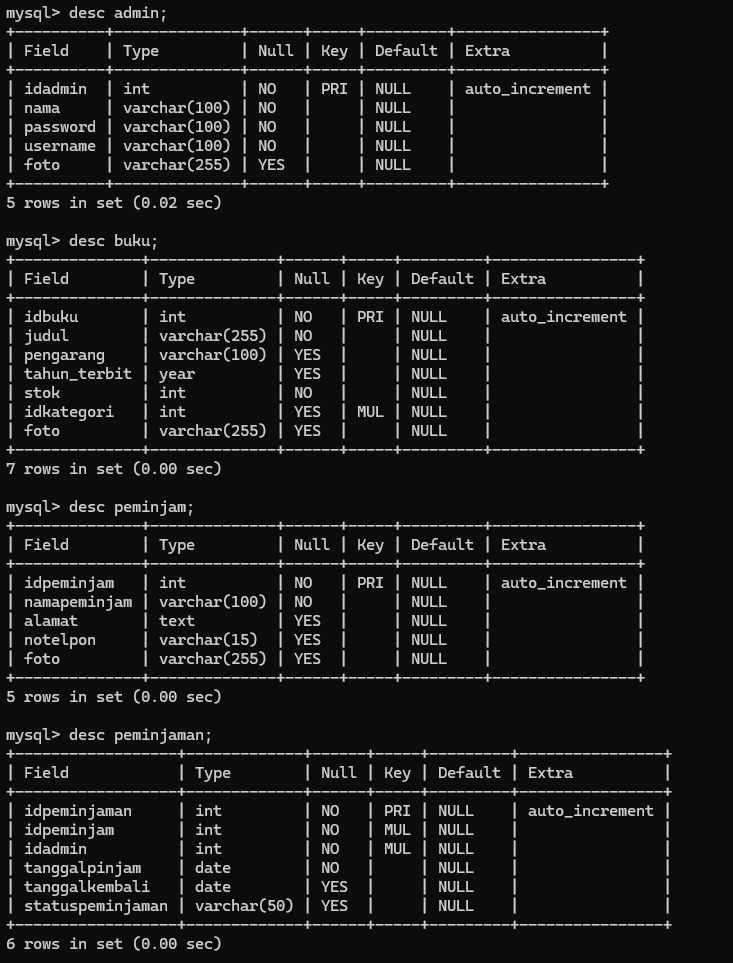
**Membuat Basisdata**

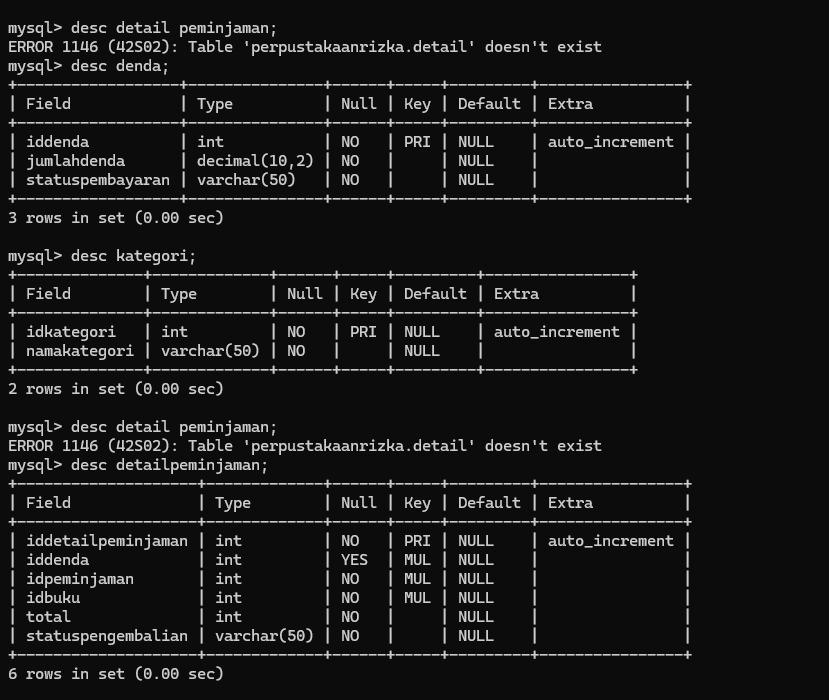
****

**Membuat Tabel**

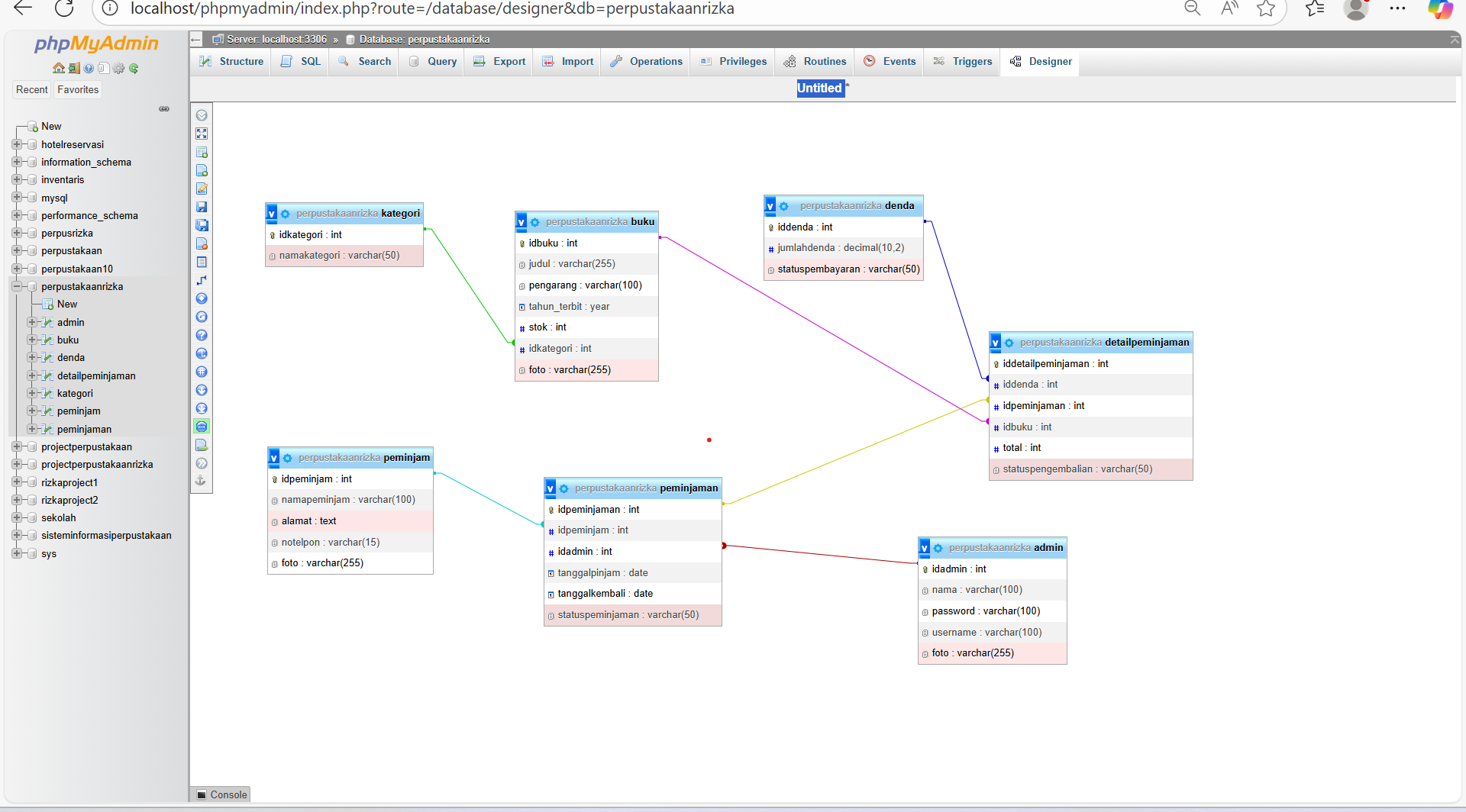


**Medeskripsikan tabel**





**Hasil Desain Konseptual Skema Relasi Database Di Phpmyadmin**

****

**MENGISI DATA RECORD**

**Cara Manual Dengan Source Code**

Memasukkan data dalam jumlah banyak

Insert into namatabel1

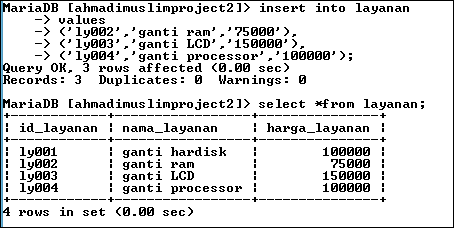
Values

(‘variable1’,’variable2’,’variabel3’,’dst’),

(‘variable1’,’variable2’,’variabel3’,’dst’),

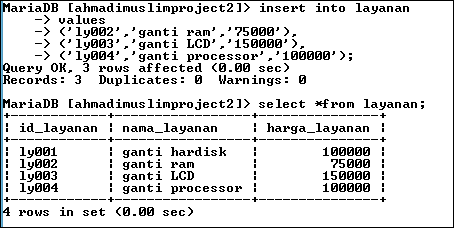
(‘variable1’,’variable2’,’variabel3’,’dst’),

(‘variable1’,’variable2’,’variabel3’,’dst’);



Melihat isi data / record yang telah dimasukkan

Select \*from namatabel1;



INSERT INTO peminjaman (id\_peminjaman, id\_peminjam, id\_admin, tanggal\_pinjam, tanggal\_kembali, status\_peminjaman)

values

( “ u007', 'u007', 'ADM002', '2025-10-16', '2025-10-21'),

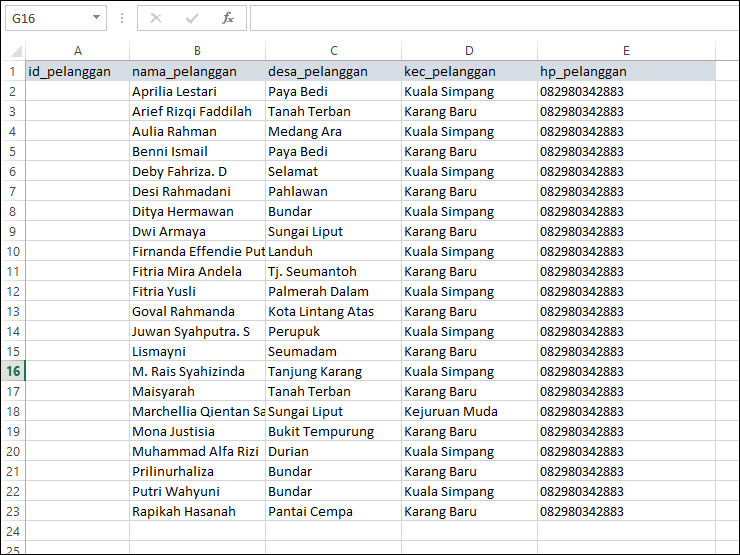
(“ u008', 'u008', 'ADM002', '2025-11-16', '2025-11-21'),

(“ u008', 'u008', 'ADM002', '2025-11-17', '2025-11-22');

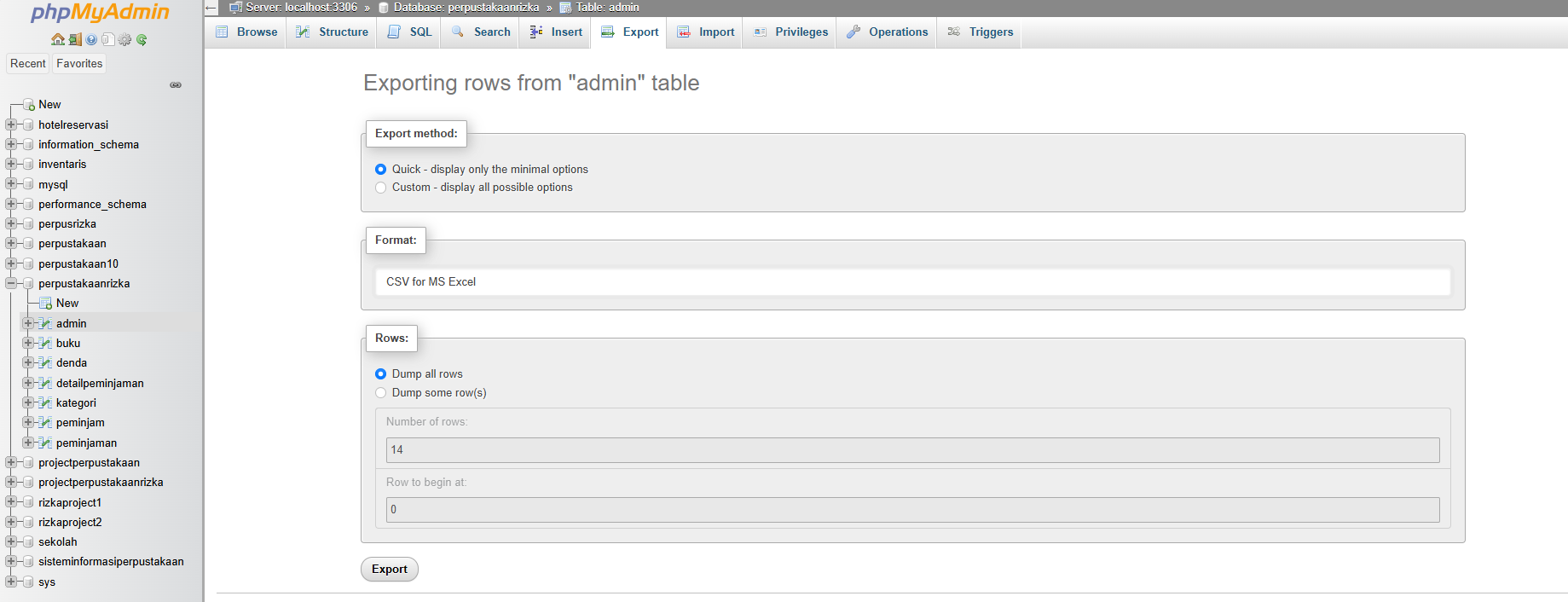
('u007', 'u007', 'ADM002', '2025-10-16', '2025-10-21', 'tepat waktu'), ('', NULL, '', NULL, NULL, NULL);

**Cara IMPORT Dari Excel**

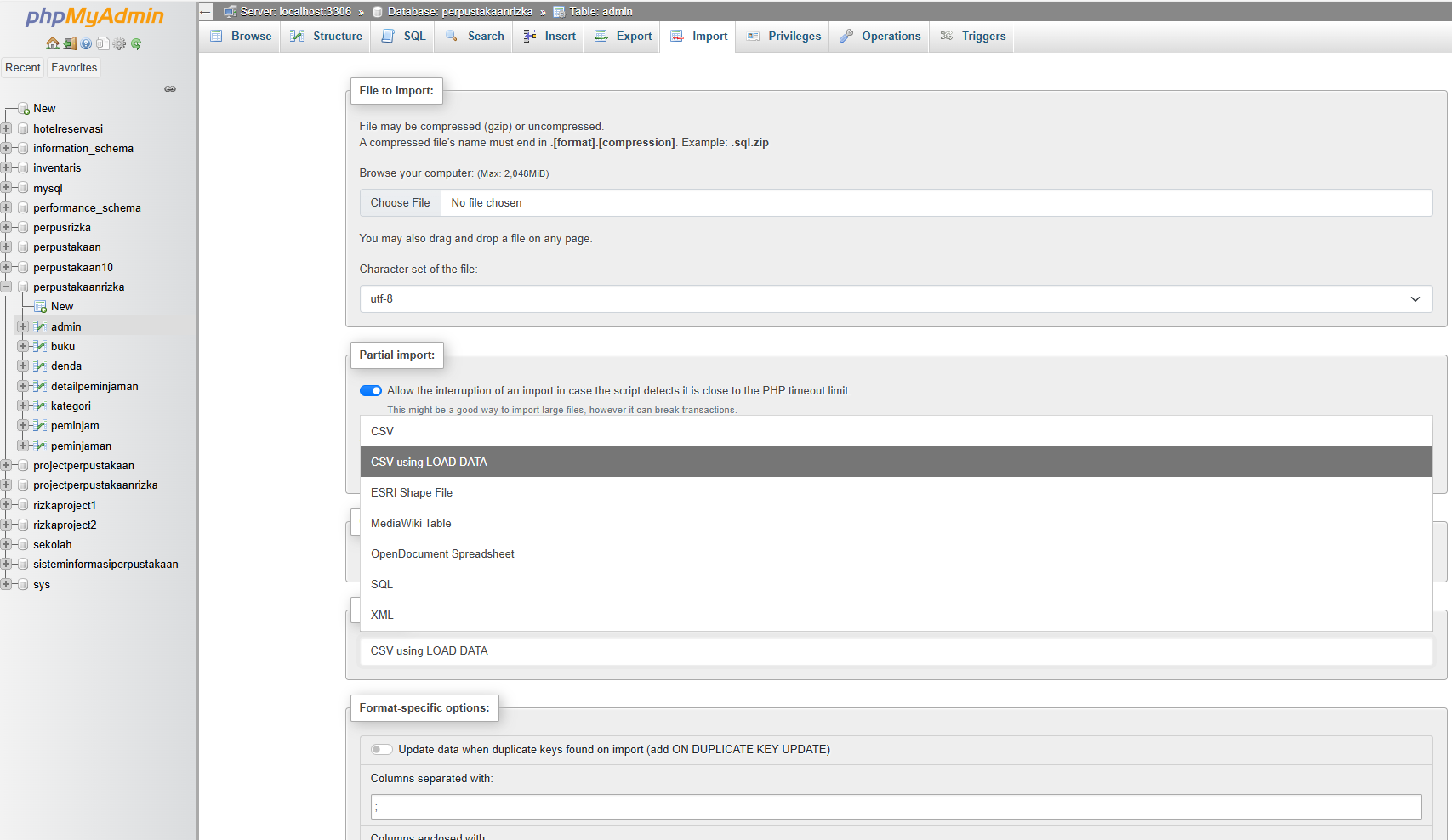
Buka file yang sudah dieksport tadi



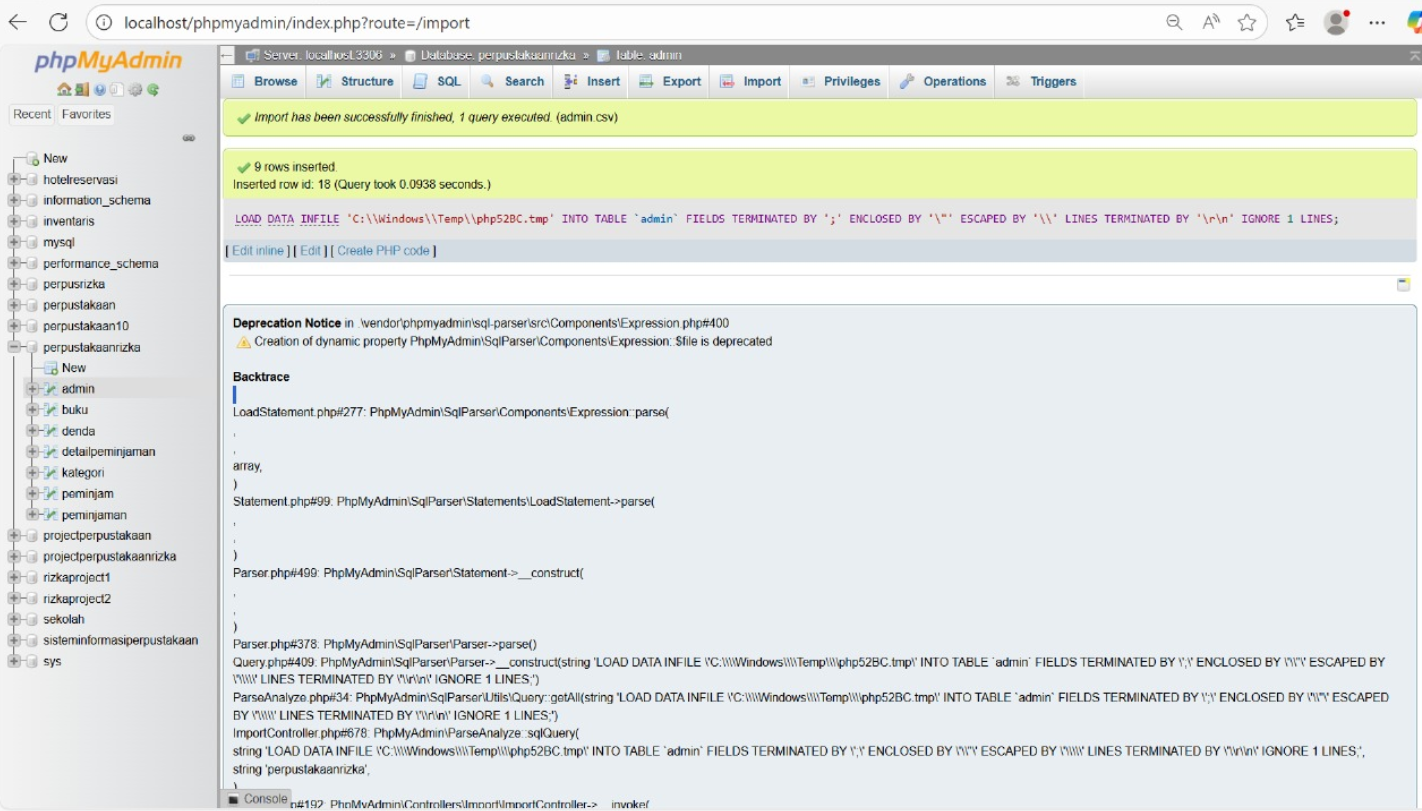
Import record yang sudah diisikan ke file csv tadi dengan cara: Masuk ke database dan buka tabel yang akan diimport datanya



Lalu cari file yang sudah diisi datanya tadi dan ubah format filenya menjadi csv Using Load Data

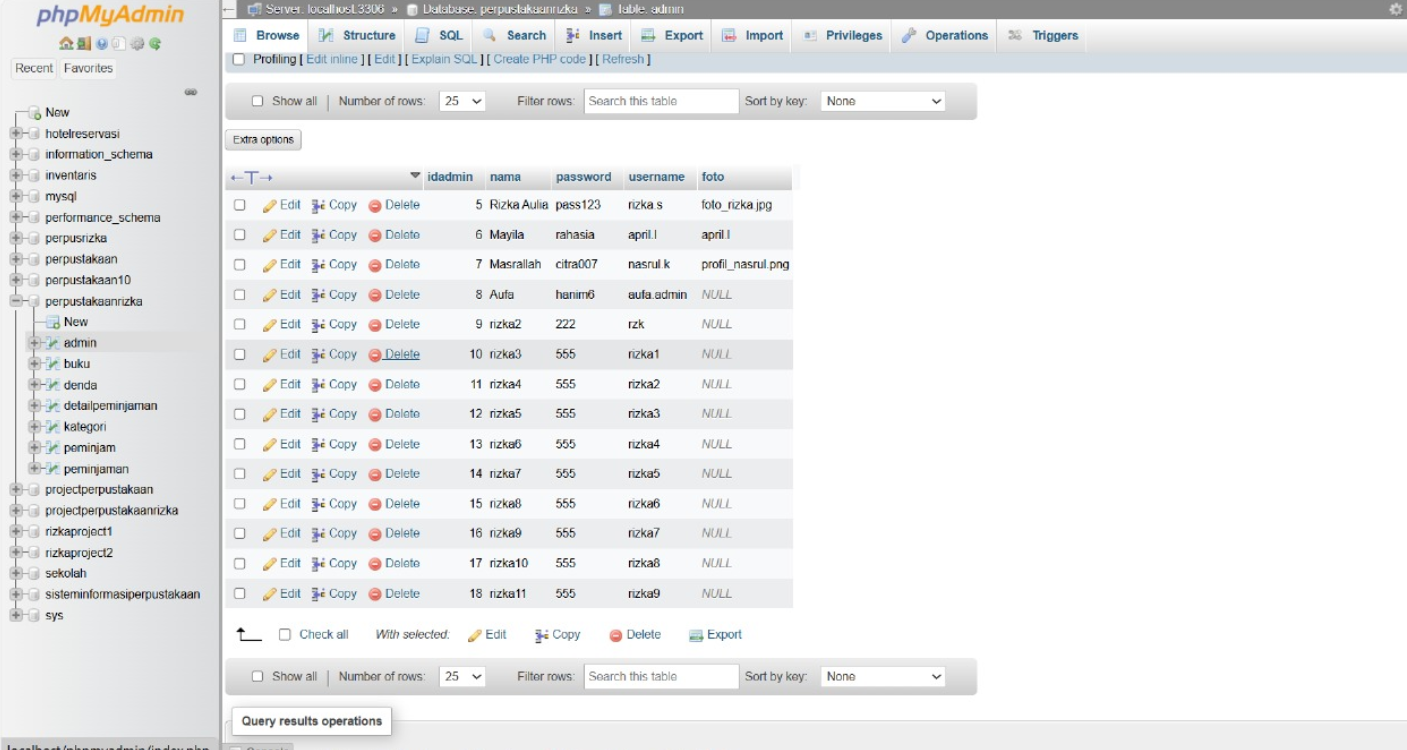


Jika sudah tekan go/kirim

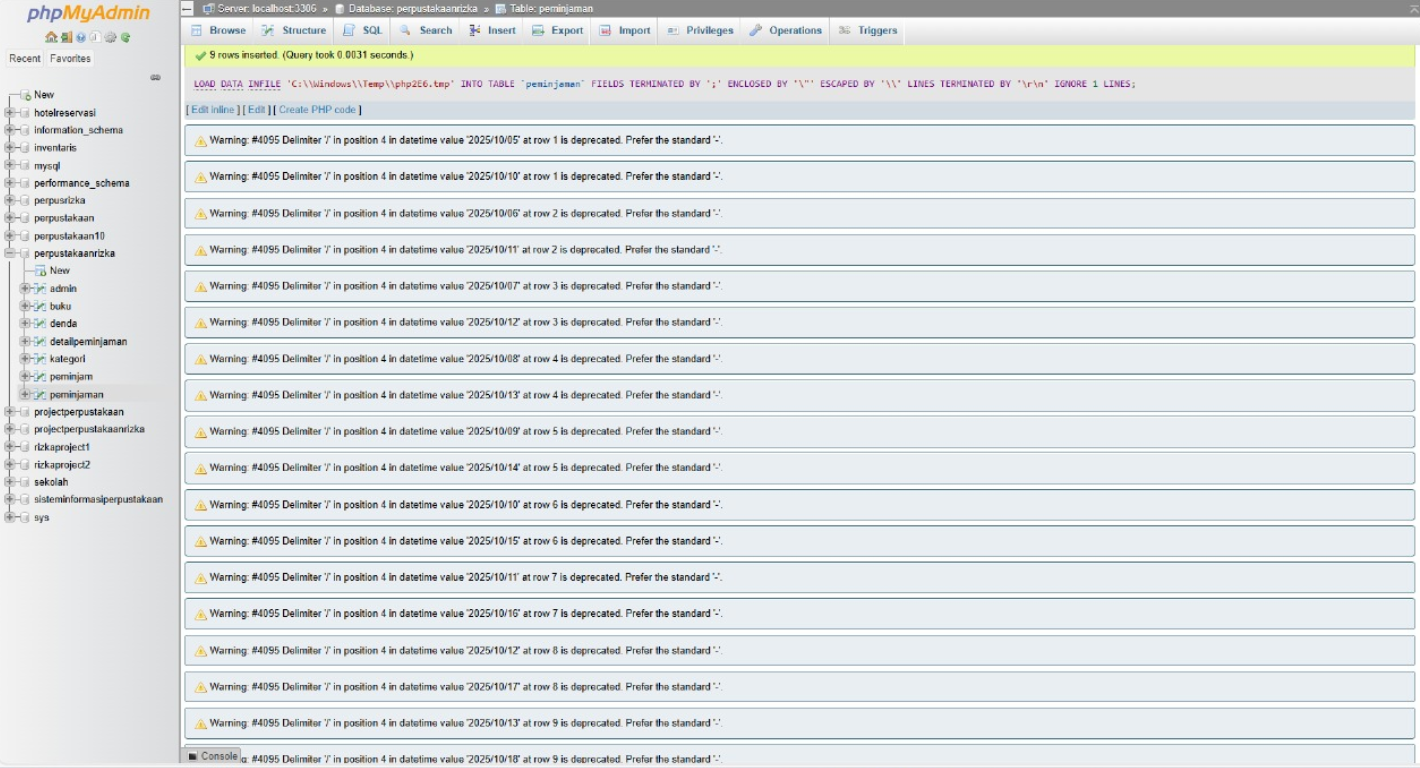


Jika berhasil, maka akan tampil semua data record dari file csv yang sudah diisi tadi

Dan lakukan ke semua tabel yang ada

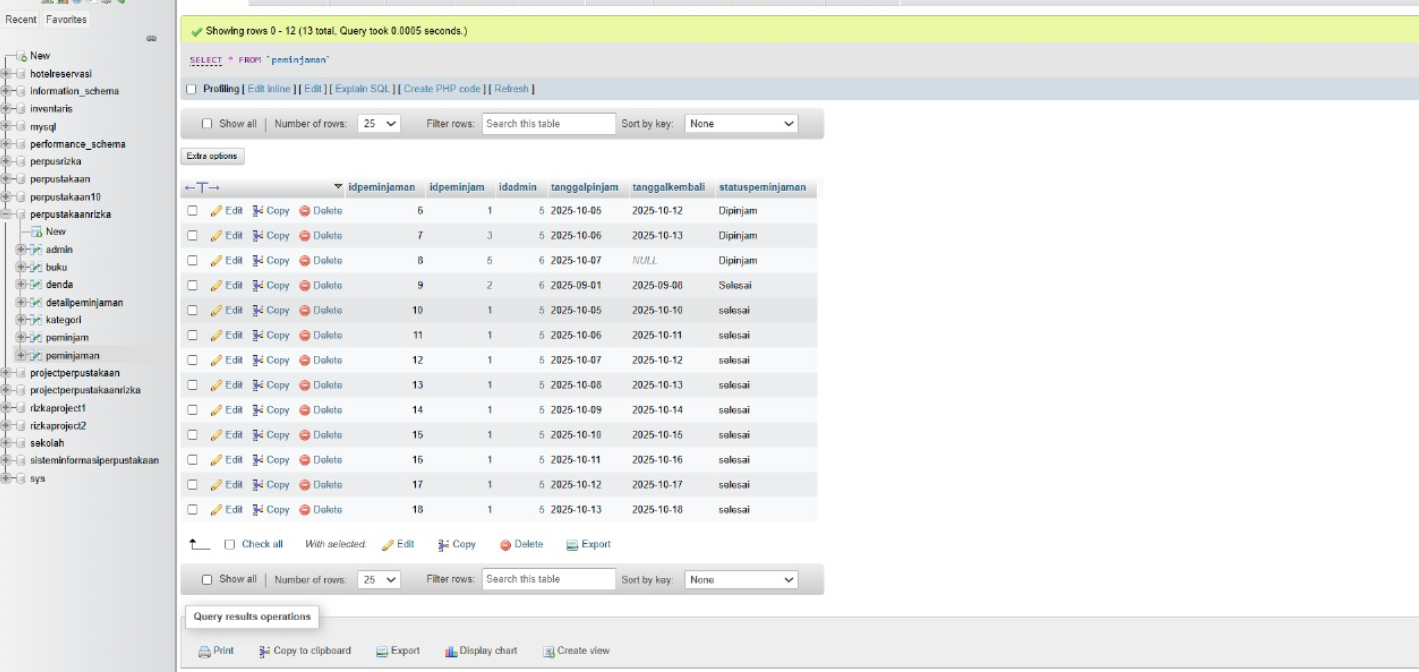


Jika sudah tekan go/kirim



Jika berhasil, maka akan tampil semua data record dari file csv yang sudah diisi tadi

Dan lakukan ke semua tabel yang ada



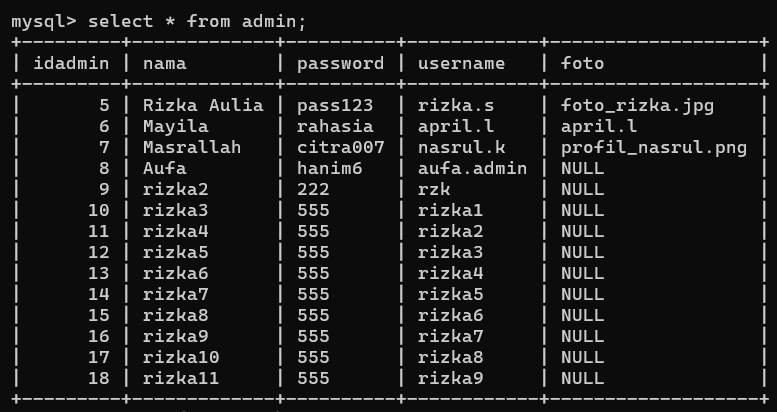
**SELECT:**

Bentuk umum.

Select \*from namatabel1

Select \*from admin;

**Output :**

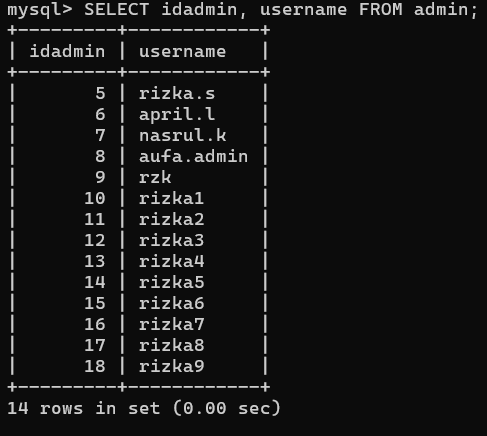
****

Memilih kolom tertentu.

SELECT nama\_kolom1, nama\_kolom2 FROM nama\_tabel;

SELECT idadmin, username FROM admin;

**Output :**

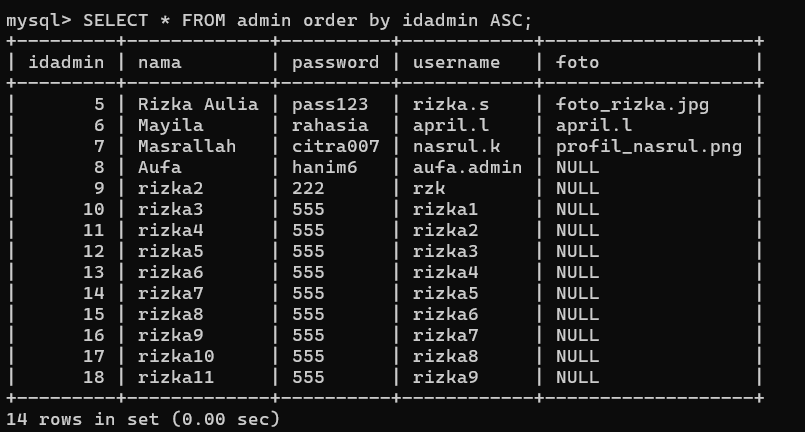


Memilih kolom dengan mengurutkan dari terkecil.

SELECT \* FROM nama\_tabel order by kolom\_dipilih ASC;

SELECT \* FROM admin order by idadmin ASC;

**Output :**



Memilih kolom dengan mengurutkan dari terbesar.

SELECT \* FROM nama\_tabel order by kolom\_dipilih DESC;

SELECT \* FROM admin order by idadmin DESC;

**Output :**



**UPDATE**

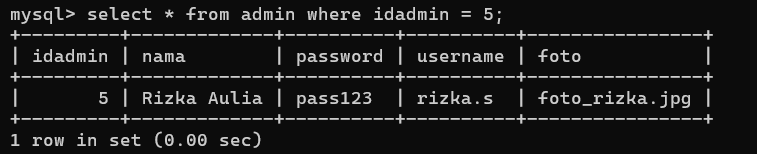
Update sebaris data record.

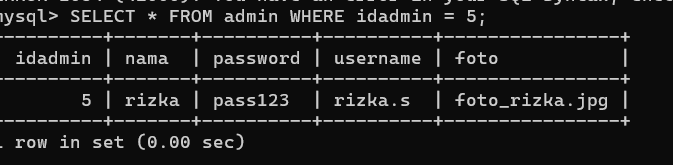
UPDATE nama\_tabel SET kolom1=data1, kolom2=data2,... WHERE kolom=data;

UPDATE admin SET namaadmin =’rizka’ WHERE id peminjam = 5;

**Output :**

Sebelum





**DELETE**

Delete sebaris data record.

DELETE FROM nama\_tabel WHERE kolom=data;

**Output :**

Delete sebuah tabel.

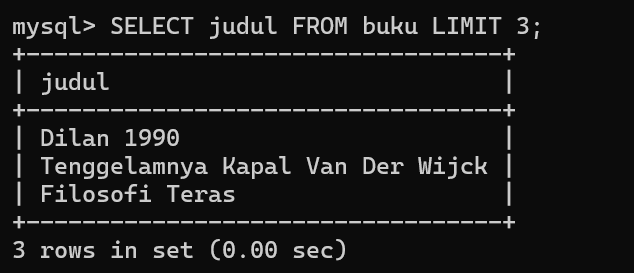
DELETE FROM nama\_tabel;

**Output :**

**SELECT SUBQUERY:**

SELECT nama\_buku FROM ms\_produk LIMIT 3;

SELECT judul FROM buku LIMIT 3;



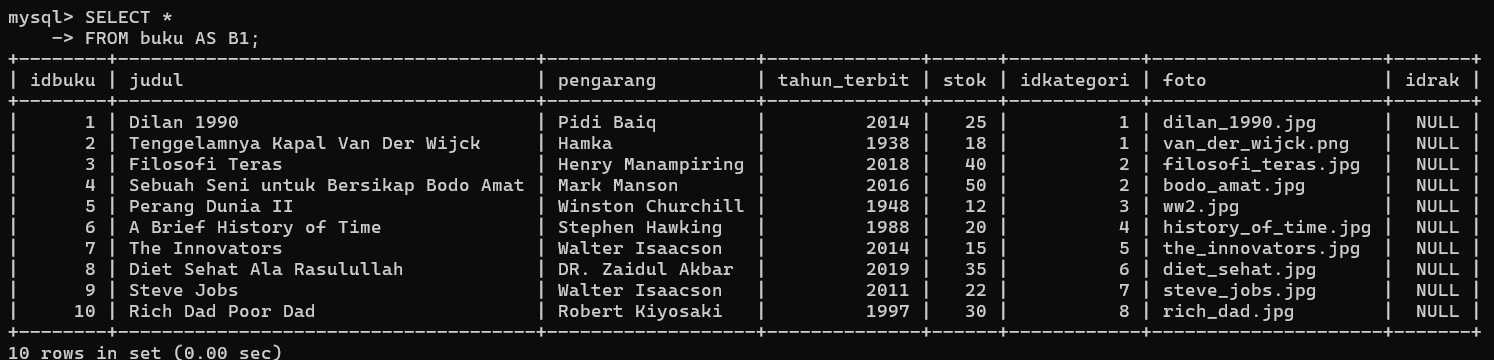
SELECT ms\_produk.nama\_produk AS nama from ms\_produk;

SELECT layanan.nama\_layanan AS layanan\_yang\_tersedia from layanan;

**Output :**

SELECT \* FROM buku AS t1;

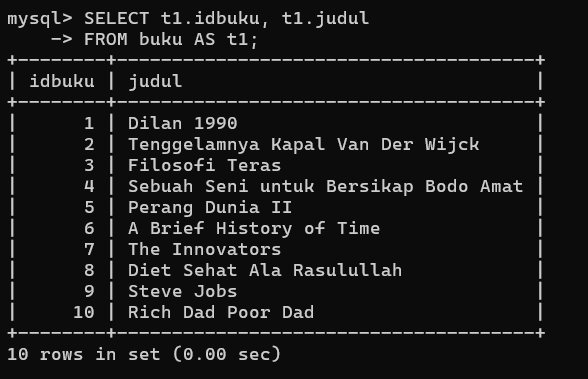
## SELECT \* FROM buku AS P1;



**Output :**

SELECT t1.idbuku, t1.judul FROM buku AS t1;

SELECT t1.idbuku, t1.judul FROM buku AS t1;

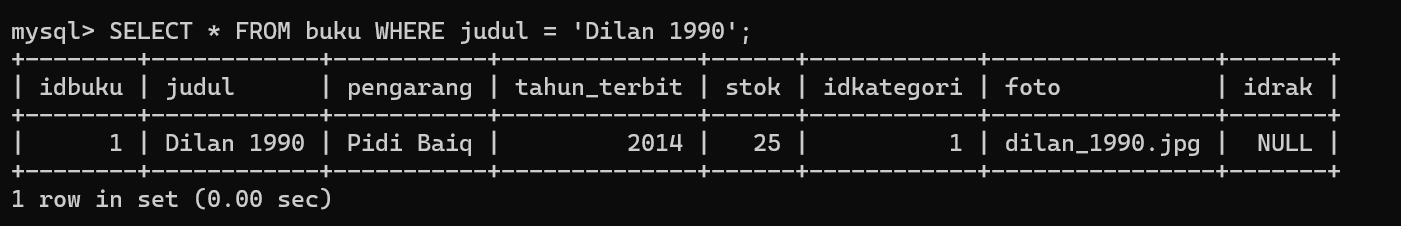


**Output :**

SELECT \* FROM buku WHERE judul = 'Gantungan Kunci DQLab';

SELECT \* FROM buku WHERE judul = 'Dilan 1990';

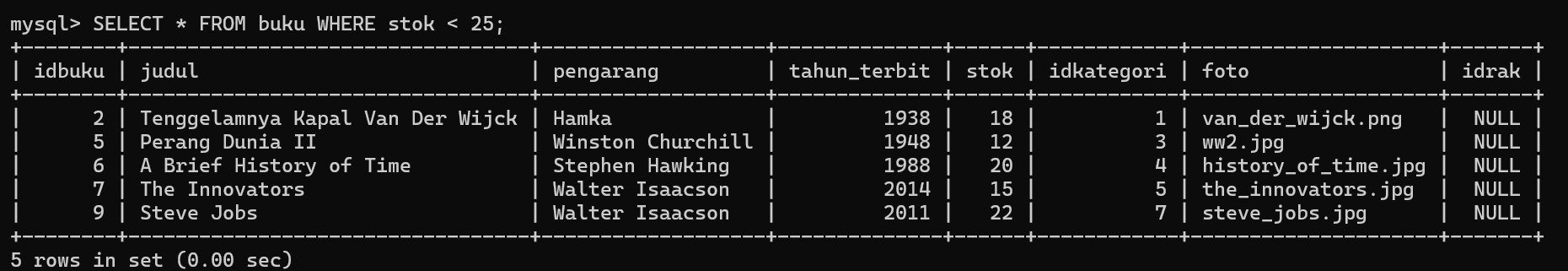
**Output :**

****

SELECT \* FROM buku WHERE stok <25;

SELECT \* FROM buku WHERE stok < 25;

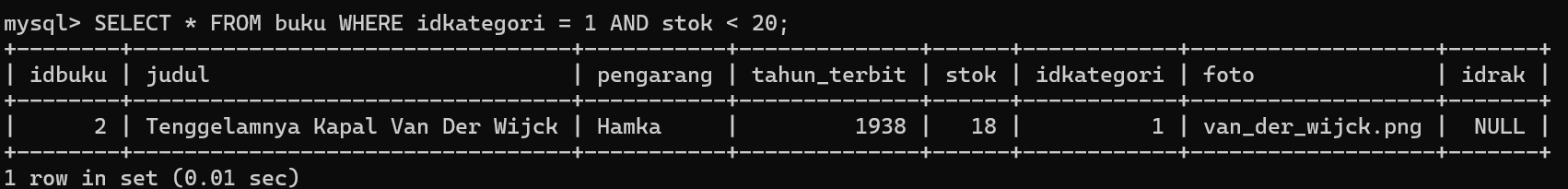
**Output :**

****

SELECT \* FROM ms\_buku WHERE nama\_kategori = 1 AND stok < 20;

SELECT \* FROM buku WHERE idkategori = 1 AND stok < 20;

Output :

****

**SELECT nama\_layanan, COUNT(id\_layanan) as jumlah FROM layanan WHERE harga\_layanan=’50000’ GROUP BY harga\_layanan HAVING COUNT(id\_layanan)>2;**

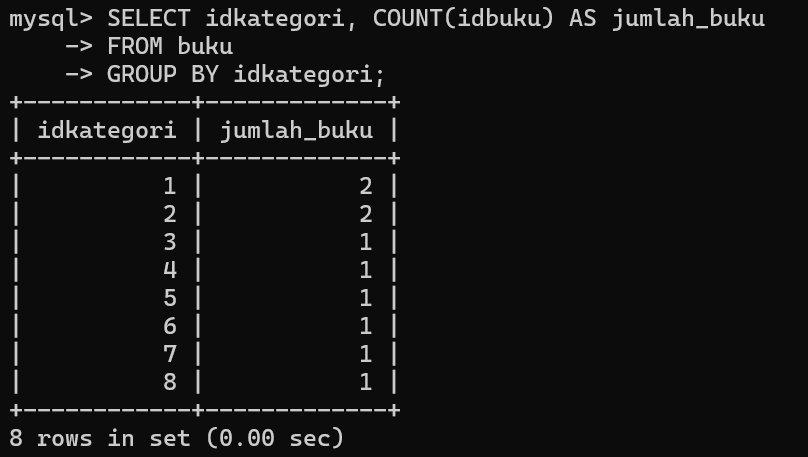
**SELECT idkategori, COUNT(idbuku) AS jumlah\_buku FROM buku GROUP BY idkategori HAVING COUNT(idbuku) > 2;**

**SELECT idkategori, COUNT(idbuku) AS jumlah\_buku**

**FROM buku**

**GROUP BY idkategori;**

**Output :**

****

**SELECT JOINT MULTITABLE relasi secara langsung :**

SELECT pelanggan.nama\_pelanggan, pembayaran.total\_bayar

FROM pelanggan INNER JOIN pembayaran ON pelanggan.id\_pelanggan=pembayaran.id\_pelanggan;

**Output :**

SELECT petugas.nama\_petugas, layanan.nama\_layanan, transaksi.jumlah

FROM petugas

INNER JOIN transaksi ON petugas.id\_petugas = transaksi.id\_petugas

INNER JOIN layanan ON transaksi.id\_layanan = layanan.id\_layanan;

SELECT pelanggan.id\_pelanggan, pelanggan.nama\_pelanggan, pembayaran.id\_bayar, pembayaran.tanggal\_bayar FROM pelanggan LEFT JOIN pembayaran ON pelanggan.id\_pelanggan=pembayaran.id\_pelanggan;

**Output :**

SELECT pelanggan.id\_pelanggan, pelanggan.nama\_pelanggan, pembayaran.id\_bayar, pembayaran.tanggal\_bayar FROM pelanggan RIGHT JOIN pembayaran ON pelanggan.id\_pelanggan=pembayaran.id\_pelanggan;

**Output :**

select a.\*, b.\*

from pelanggan a

join pembayaran b using (id\_pelanggan);

**Output :**

select a.\*, b.\*

from pelanggan a

left join pembayaran b

using (id\_pelanggan);

**Output :**

select a.\*, b.\*

from pelanggan a

right join pembayaran b

using (id\_pelanggan);

**Output :**

select a.\*, b.\*

from pelanggan a

right join pembayaran b

using (id\_pelanggan)

union all

select a.\*, b.\*

from pelanggan a

left join pembayaran b

using (id\_pelanggan);

**Output :**

select a.nama\_pelanggan, b.total\_bayar

from pelanggan a

right join pembayaran b

using (id\_pelanggan)

union all

select a.nama\_pelanggan, b.total\_bayar

from pelanggan a

left join pembayaran b

using (id\_pelanggan);

**Output :end**

Menambah table dan relasi

Menambah tabel

Tabel Rak

CREATE TABLE rak (

Idrak INT AUTO\_INCREMENT PRIMARY KEY,

Namarak VARCHAR (100) NOT NULL

);

