

**MAKALAH**  
**SISTEM MANAJEMEN BASIS DATA**  
**STUDI KASUS KAFE JANJI JIWA**

**Dosen Pengampu : Kamarudin M.Kom**



Nama : Imam Farisi  
NIM : 1855201110006  
Youtube url : <https://youtu.be/LtnS67rpqgY>

**PROGRAM STUDI INFORMATIKA**  
**FAKULTAS TEKNIK**  
**UNIVERSITAS MUHAMMADIYAH BANJARMASIN**  
**BANJARMASIN**  
**2020**

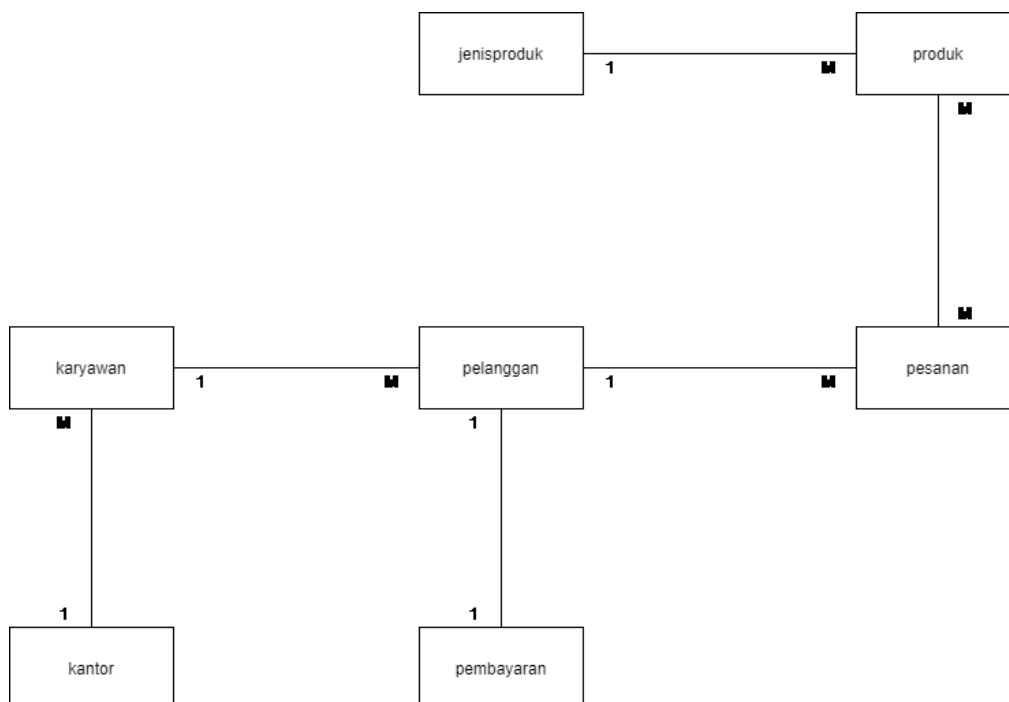
## 1. Desain Basis Data

Desain basis data adalah pengorganisasian data menurut model basis data. Perancang menentukan data apa yang harus disimpan dan bagaimana elemen data saling berhubungan. Dengan informasi ini, mereka dapat mulai menyesuaikan data dengan model basis data. [1]

Model konseptual, logis dan fisik atau ERD adalah tiga cara pemodelan data yang berbeda dalam suatu domain. Meskipun semuanya berisi entitas dan hubungan, keduanya berbeda dalam tujuan pembuatannya dan audiens yang menjadi target mereka. Pemahaman umum untuk ketiga model tersebut adalah bahwa, analis bisnis menggunakan model konseptual dan logis untuk memodelkan data yang dibutuhkan dan dihasilkan oleh sistem dari sudut bisnis, sedangkan perancang basis data menyempurnakan desain awal untuk menghasilkan model fisik untuk menyajikan struktur basis data fisik yang siap digunakan untuk konstruksi basis data. [2]

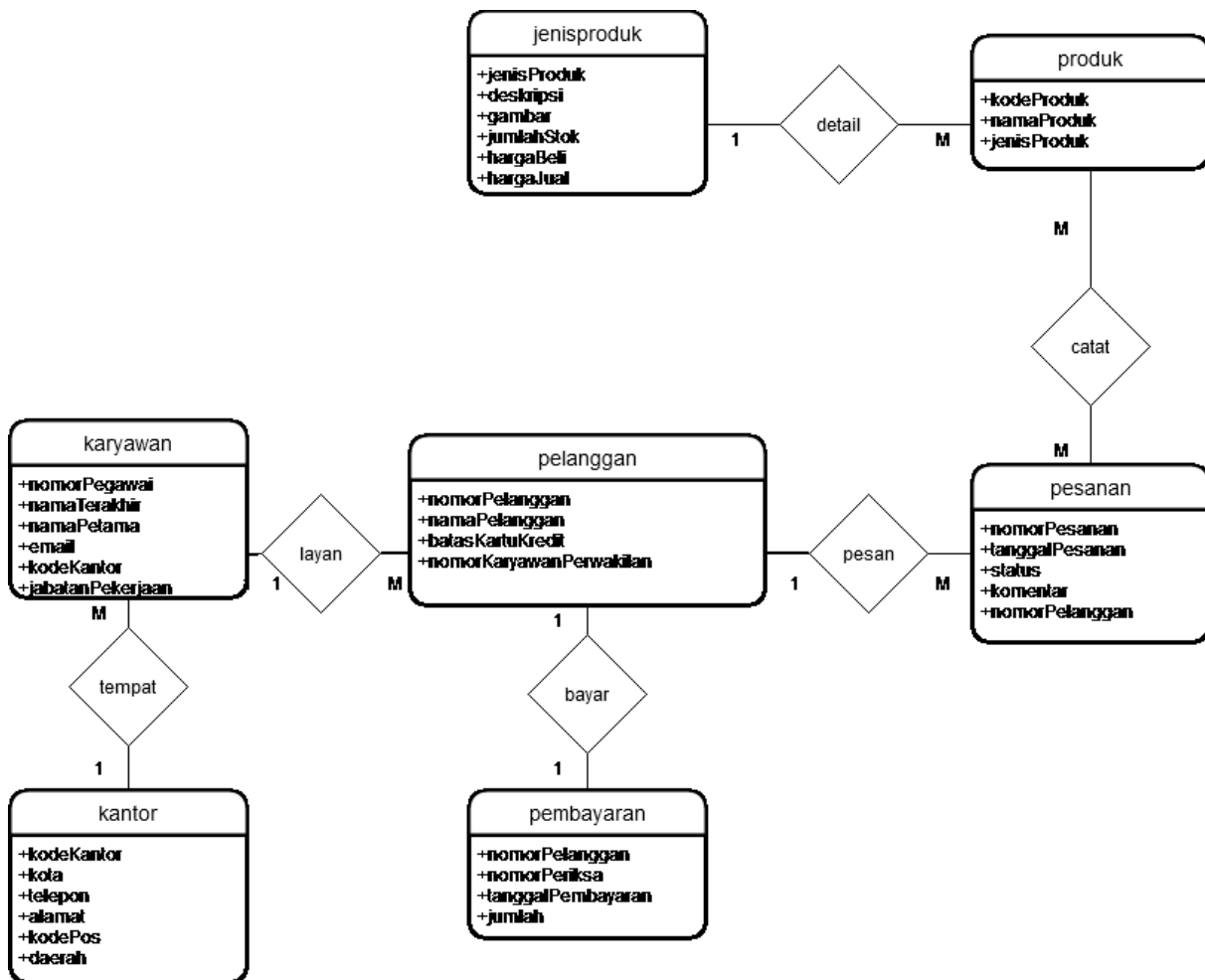
## 2. Desain Konseptual

Informasi model ERD konseptual yang dikumpulkan dari kebutuhan bisnis. Entitas dan hubungan yang dimodelkan dalam ERD tersebut didefinisikan di sekitar kebutuhan bisnis. Kebutuhan akan desain database yang memuaskan belum dipertimbangkan. ERD konseptual adalah model paling sederhana di antara semuanya. Berikut adalah gambar desain konseptual dari studi kasus cafe janji jiwa. [2]



### 3. Desain Logis

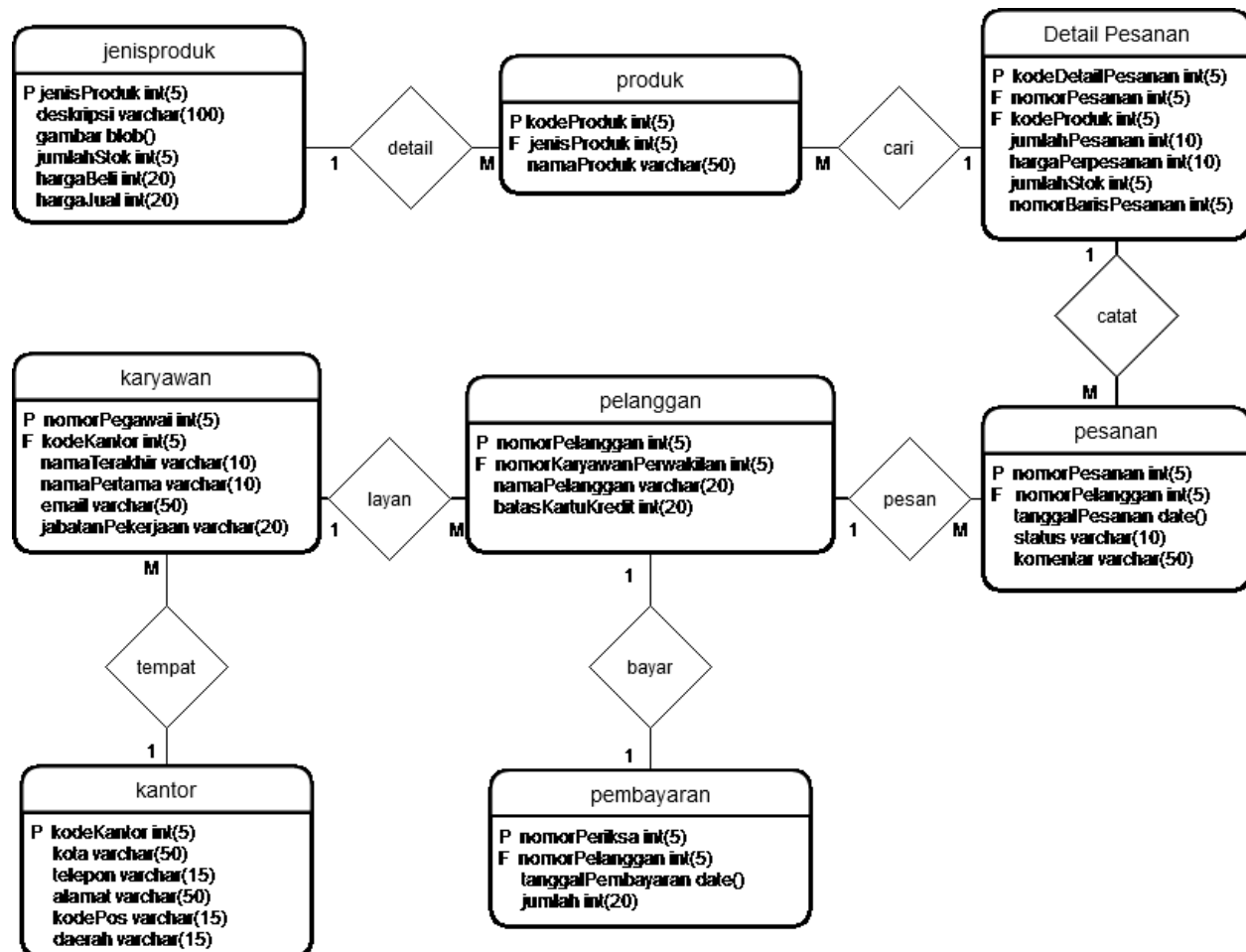
ERD logis juga memodelkan informasi yang dikumpulkan dari kebutuhan bisnis. Ini lebih kompleks daripada model konseptual dalam jenis kolom yang ditetapkan. Perhatikan bahwa pengaturan tipe kolom adalah opsional dan jika Anda melakukannya, Anda harus melakukannya untuk membantu analisis bisnis. Ini belum ada hubungannya dengan pembuatan database. Berikut adalah desain logis dari studi kasus cafe janji jiwa. [2]



### 4. Desain Fisik

ERD fisik mewakili cetak biru desain aktual dari database relasional. Ini mewakili bagaimana data harus terstruktur dan terkait dalam DBMS tertentu sehingga penting untuk mempertimbangkan konvensi dan pembatasan DBMS yang Anda gunakan saat Anda merancang ERD fisik. Ini berarti

bahwa penggunaan tipe data yang akurat diperlukan untuk kolom entitas dan penggunaan kata-kata khusus harus dihindari dalam penamaan entitas dan kolom. Selain itu, perancang database juga dapat menambahkan kunci utama, kunci asing, dan batasan ke desain. Berikut adalah desain logis dari studi kasus cafe janji jiwa. [2]



## 5. Perintah – Perintah MySQL

Pada menu command prompt, masuk pada bagian directory xampp>mysql>bin kemudian masukan perintah mysql.exe -u root lalu tekan enter. Kemudian dari sini akan tampil prompt untuk memasukan syntax *DML (Data Manipulation Language)* dan kita langsung membuat database dengan perintah create database janjijiwa. Kemudian kita akan masuk di database janji jiwa dengan perintah use janjijiwa. Mulai dari sini kita akan memasukan perintah syntax *DDL (Data Definition Language)*.

Perintah syntax DDL yang dimasukan adalah sebagai berikut :

Perintah yang dimasukan akan membuat *table* kantor dengan *primary key* kodeKantor.

```
→ create table kantor(  
→ kodeKantor int(5) not null auto_increment,  
→ kota varchar(50),  
→ telepon varchar(15),  
→ alamat varchar(50),  
→ kodePos varchar(15),  
→ daerah varchar(15),  
→ primary key(kodeKantor)  
→ );
```

Perintah yang dimasukan ini akan membuat *table* karyawan dengan *primary key* nomorPegawai dan *foreign key* kodeKantor.

```
→ create table karyawan(  
→ nomorPegawai int(5) not null auto_increment,  
→ kodeKantor int(5) not null,  
→ namaTerakhir varchar(10),  
→ namaPertama varchar(10),  
→ email varchar(50),  
→ jabatanPekerjaan varchar(20),  
→ index(kodeKantor),  
→ foreign key(kodeKantor) references kantor(kodeKantor)  
→ on delete restrict on update cascade,  
→ primary key(nomorPegawai)  
→ );
```

Perintah yang dimasukan ini akan membuat *table* pelanggan dengan *primary key* nomorPelanggan dan *foreign key* nomorKaryawanPerwakilan.

```
→ create table pelanggan(  
→ nomorPelanggan int(5) not null auto_increment,
```

- ➔ nomorKaryawanPerwakilan int(5) not null,
- ➔ namaPelanggan varchar(20),
- ➔ batasKartuKredit int(20),
- ➔ index(nomorKaryawanPerwakilan),
- ➔ foreign key(nomorKaryawanPerwakilan) references
- ➔ karyawan(nomorPegawai)
- ➔ on delete restrict on update cascade,
- ➔ primary key(nomorPelanggan)
- ➔ );

Perintah yang dimasukan ini akan membuat *table* pembayaran dengan *primary key* nomorPeriksa dan *foreign key* nomorPelanggan\_fk.

- ➔ create table pembayaran(
- ➔ nomorPeriksa int(5) not null auto\_increment,
- ➔ nomorPelanggan\_fk int(5),
- ➔ tanggalPembayaran date(),
- ➔ jumlah int(20) not null,
- ➔ index(nomorPelanggan\_fk),
- ➔ foreign key(nomorPelanggan\_fk) references
- ➔ pelanggan(nomorPelanggan)
- ➔ on delete restrict on update cascade,
- ➔ primary key(nomorPeriksa)
- ➔ );

Perintah yang dimasukan ini akan membuat *table* pesanan dengan *primary key* nomorPesanan dan *foreign key* nomorPelanggan\_fk.

- ➔ create table pesanan(
- ➔ nomorPesanan int(5) not null auto\_increment,
- ➔ nomorPelanggan\_fk int(5) not null,
- ➔ tanggalPesanan date(),
- ➔ status varchar(10),
- ➔ komentar varchar(50),

- ➔ index(nomorPelanggan\_fk),
- ➔ foreign key(nomorPelanggan\_fk) references
- ➔ pelanggan(nomorPelanggan)
- ➔ on delete restrict on update cascade,
- ➔ primary key(nomorPesanan)
- ➔ );

Perintah yang dimasukan ini akan membuat *table* jenisproduk dengan *primary key* jenisproduk.

- ➔ create table jenisproduk(
- ➔ jenisProduk int(5) not null auto\_increment,
- ➔ deskripsi varchar(100),
- ➔ gambar blob(),
- ➔ jumlahStok int(5),
- ➔ hargaBeli int(20),
- ➔ hargaJual int(20),
- ➔ primary key(jenisProduk)
- ➔ );

Perintah yang dimasukan ini akan membuat *table* produk dengan *primary key* kodeProduk.

- ➔ create table produk(
- ➔ kodeProduk int(5) not null auto\_increment,
- ➔ jenisProduk int(5),
- ➔ namaProduk varchar(50),
- ➔ primary key(kodeProduk)
- ➔ );

Perintah yang dimasukan ini akan membuat *table* detailpesanan dengan *primary key* kodeDetailPesanan dan *foreign key* nomorPesanan\_fk dan kodeProduk\_fk.

- ➔ create table detailpesanan(
- ➔ kodeDetailPesanan int(5) not null auto\_increment,
- ➔ nomorPesanan\_fk int(5),
- ➔ kodeProduk\_fk int(5),

- ➔ jumlahPesanan int(10),
- ➔ hargaPerpesanan int(10),
- ➔ jumlahStok int(5),
- ➔ nomorBarisPesanan int(5),
- ➔ index(nomorPesanan\_fk),
- ➔ index(kodeProduk\_fk),
- ➔ foreign key(nomorPesanan\_fk) references pesanan(nomorPesanan)
- ➔ on delete restrict on update cascade
- ➔ foreign key(kodeProduk\_fk) references produk(kodeProduk)
- ➔ on delete restrict on update cascade
- ➔ primary key(kodeDetailPesanan)
- ➔ );



## DAFTAR PUSTAKA

- [1] "wikipedia," 17 December 2020. [Online]. Available:  
[https://en.wikipedia.org/wiki/Database\\_design](https://en.wikipedia.org/wiki/Database_design).
- [2] "visual-paradigm," 17 December 2020. [Online]. Available: [https://www.visual-paradigm.com/support/documents/vpuserguide/3563/3564/85378\\_conceptual,1.html](https://www.visual-paradigm.com/support/documents/vpuserguide/3563/3564/85378_conceptual,1.html).