# Merancang Struktur Database

## Penjelasan Singkat Keyword yang akan Digunakan

Berikut adalah penjelasan dari keyword syntax pada **dbdiagram.io** yang akan digunakan :

1. **Table**. Digunakan untuk mendefinisikan sebuah tabel dalam database.

Contoh :

Table user {

//kolom-kolom yang dibutuhkan

}

1. **Ref**. Digunakan untuk mendefinisikan relasi antara dua tabel.

Contoh :

Ref {

  produk.id\_kategori > kategori.id\_kategori

}

1. **PK** atau singkatan dari **Primary Key**, digunakan untuk mendefinisikan kolom yang berfungsi sebagai identitas unik dalam sebuah tabel.

Contoh :

id int PK

1. **int** atau **Integer,** digunakan untuk mendefinisikan bahwa kolom tersebut memiliki tipe data *bilangan bulat.*
2. **varchar**, digunakan untuk mendefinisikan bahwa kolom tersebut memiliki tipe data string atau teks.
3. **tinyint**, digunakan untuk mendefinisikan bahwa kolom tersebut memiliki tipe data bilangan bulat yang sangat kecil, biasanya digunakan untuk menyimpan nilai numerik kecil, seperti **boolean** (0 atau 1).
4. **text,** digunakan untuk mendefinisikan bahwa kolom tersebut memiliki tipe data teks dalam jumlah besar.
5. **timestamp**, digunakan untuk mendefinisikan bahwa kolom tersebut memiliki tipe data informasi waktu (tanggal dan jam).
6. **unique**, digunakan untuk memastikan bahwa nilai dalam kolom tersebut harus unik, tidak boleh ada duplikasi.
7. **nullable**, menunjukkan bahwa kolom tersebut boleh kosong (null)
8. **default**, digunakan untuk menetapkan nilai bawaan untuk kolom jika tidak ada nilai yang diinput.

## Langkah – Langkah membuat struktur database untuk aplikasi Point of Sales ini adalah :

### Identifikasi Kebutuhan dan Fitur Aplikasi

Kebutuhan fitur yang akan ada dalam aplikasi POS antara lain :

1. Manajemen pengguna (user)
2. Manajemen kategori produk
3. Manajemen produk
4. Manajemen anggota (member)
5. Manajemen pemasok (supplier)
6. Manajemen pembelian
7. Pencatatan detail pembelian
8. Manajemen penjualan
9. Pencatatan detail pembelian
10. Pencatatan pengeluaran
11. Pengelolaan informasi toko

### Menggunakan dbdiagram.io untuk Mendesain Database

Untuk mengakses dbdiagram.io kamu bisa mengunjungi alamat berikut ini :

<https://dbdiagram.io/>

### Membuat Tabel User

Tabel **user** ini berfungsi untuk menyimpan data pengguna aplikasi.

Berikut adalah penulisan code sql tabel user :

Table users{

  id int PK

  nama\_user varchar

  email\_user varchar // unique

  password\_user varchar

  foto\_user varchar

  level\_user tinyint

remember\_token varchar // nullable

  created\_at timestamp // nullable

  deleted\_at timestamp // nullable

  }

* **Catatan** :
* ***remember\_token*** *digunakan untuk menyimpan token keamanan dalam tabel autentikasi, sering kali digunakan dalam sistem login untuk mengingat pengguna yang sudah login.*
* ***created\_at*** *adalah kolom untuk menyimpan waktu kapan data dibuat. Biasanya menggunakan tipe data* ***timestamp*** *atau* ***datetime.***
* ***deleted\_at*** *digunakan dalam konsep* ***soft delete****, yaitu data tidak benar-benar dihapus dari database, tapi hanya “disembunyikan” dengan memberikan nilai waktu pada kolom ini ketika data dihapus.*

### Membuat Tabel Kategori

Tabel **Kategori** ini berfungsi untuk mengelompokkan produk.

Berikut adalah penulisan untuk tabel kategori :

Table kategori {

  id\_kategori int PK

  nama\_kategori varchar // unique

  created\_at timestamp // nullable

  updated\_at timestamp // nullable

}

* **Catatan** :
* ***updated\_at*** *adalah kolom yang digunakan untuk menyimpan kapan data terakhir kali diperbarui.*

### Membuat Tabel Produk

Tabel **Produk** ini berfungsi untuk menyimpan informasi produk yang dijual.

Berikut adalah penulisan code sql untuk tabel kategori :

Table produk{

id\_produk int PK

id\_kategori int

nama\_produk varchar // unique

merk\_produk varchar // nullable

harga\_beli\_produk int

diskon\_produk int // nullable default 0

harga\_jual\_produk int

stok\_produk int // default 0

created\_at timestamp // nullable

updated\_at timestamp // nullable

}

Ref {

produk.id\_kategori > kategori.id\_kategori

}

### Membuat Tabel Member

Tabel **Member** ini berfungsi untuk menyimpan informasi anggota atau pelanggan yang mendaftar.

Berikut adalah penulisan code sql untuk tabel member :

Table member {

id\_member int PK

kode\_member varchar // unique

nama\_member varchar

alamat\_member text // nullable

telepon\_member varchar

created\_at timestamp // nullable

updated\_at timestamp // nullable

}

### Membuat Tabel Supplier

Tabel **Supplier** ini berfungsi untuk menyimpan data pemasok barang.

Berikut adalah penulisan untuk tabel supplier :

Table supplier {

  id\_supplier int PK

  nama\_supplier varchar

  alamat\_supplier text

  telepon\_supplier varchar

  created\_at timestamp // nullable

  updated\_at timestamp // nullable

}

### Membuat Tabel Pembelian

Tabel **Pembelian** ini berfungsi untuk mencatat transaksi pembelian dari pemasok/supplier.

Berikut adalah penulisan code sql untuk tabel supplier :

Table pembelian {

  id\_pembelian int PK

  id\_supplier int

  total\_item\_pembelian int

  total\_harga\_pembelian int

  diskon\_pembelian int // default 0

  bayar\_pembelian int

  created\_at timestamp // nullable

  updated\_at timestamp // nullable

}

Ref {

  pembelian.id\_supplier > supplier.id\_supplier

}

### Membuat Tabel Pembelian Detail

Tabel **Pembelian Detail** ini berfungsi untuk mencatat rincian setiap barang yang dibeli dari pemasok/supplier.

Berikut adalah penulisan code sql untuk tabel detail\_pembelian :

Table pembelian\_detail {

  id\_pembelian\_detail int PK

  id\_pembelian int

  id\_produk int

  harga\_pembelian\_detail int

  jumlah\_pembelian\_detail int

  subtotal\_pembelian\_detail int

  created\_at timestamp // nullable

  updated\_at timestamp // nullable

}

Ref {

  pembelian\_detail.id\_pembelian > pembelian.id\_pembelian

}

Ref {

  pembelian\_detail.id\_produk > produk.id\_produk

}

### Membuat Tabel Penjualan

Tabel **Penjualan** ini berfungsi untuk mencatat transaksi penjualan kepada pelanggan.

Berikut adalah penulisan code sql untuk tabel penjualan :

Table penjualan {

  id\_penjualan int PK

  id\_member int

  total\_item\_penjualan int

  total\_harga\_penjualan int

  diskon\_penjualan int // default 0

  bayar\_penjualan int

  diterima\_penjualan int // default 0

  id\_user int

  created\_at timestamp // nullable

  updated\_at timestamp // nullable

}

Ref {

  penjualan.id\_member > member.id\_member

}

### Membuat Tabel Penjualan Detail

Tabel **Penjualan Detail** ini berfungsi untuk mencatat rincian barang yang dijual.

Berikut adalah penulisan code sql untuk tabel penjualan detail :

Table penjualan\_detail {

  id\_penjualan\_detail int PK

  id\_penjualan int

  id\_produk int

  harga\_penjualan\_detail int

  jumlah\_penjualan\_detail int

  diskon\_penjualan\_detail int // default 0

  subtotal\_penjualan\_detail int

  created\_at timestamp // nullable

  updated\_at timestamp // nullable

}

Ref {

  penjualan\_detail.id\_penjualan > penjualan.id\_penjualan

}

Ref {

  penjualan\_detail.id\_produk > produk.id\_produk

}

### Membuat Tabel Pengeluaran

Tabel **Pengeluaran** ini berfungsi untuk mencatat pengeluaran lain dalam bisnis, misalnya ‘biaya operasional’.

Berikut adalah penulisan code sql untuk tabel pengeluaran:

Table pengeluaran {

  id\_pengeluaran int PK

  deskripsi\_pengeluaran text

  nominal\_pengeluaran int

  created\_at timestamp // nullable

  updated\_at timestamp // nullable

}

### Membuat Tabel Setting

Tabel **Setting** ini berfungsi untuk menyimpan pengaturan aplikasi seperti nama toko, logo, alamat, dan lainnya.

Berikut adalah penulisan code sql untuk tabel setting :

Table setting {

  id\_setting int PK

  nama\_perusahaan varchar

  alamat text // nullable

  telepon varchar

  path\_logo varchar // nullable

  tipe\_nota tinyint

  path\_kartu\_member varchar

  created\_at timestamp // nullable

  updated\_at timestamp // nullable

}