

# 1. Rancangan Database (MySQL)

Pada Database ini terdiri dari tabel `siswa`, `transaksi`, `produk`, dan `detail_transaksi` dengan relasi sebagai berikut:

- `siswa` berelasi 1-N dengan `transaksi` melalui `id_siswa`.
- `transaksi` berelasi 1-N dengan `detail_transaksi` melalui `id_transaksi`.
- `produk` berelasi 1-N dengan `detail_transaksi` melalui `id_produk`.

Tabel dibuat dengan foreign key untuk menjaga integritas data antar tabel.

**kode pembuatan tabel:**

## 1. TABEL SISWA

```
CREATE TABLE siswa (  
  id_siswa INT AUTO_INCREMENT PRIMARY KEY,  
  nama_siswa VARCHAR(100) NOT NULL,  
  kelas VARCHAR(50) NOT NULL,  
  saldo DECIMAL(10, 2) NOT NULL  
);
```

## 1. TABEL TRANSAKSI

```
CREATE TABLE transaksi (  
  id_transaksi INT AUTO_INCREMENT PRIMARY KEY,  
  id_siswa INT NOT NULL,  
  tanggal_transaksi DATE NOT NULL,  
  total_bayar DECIMAL(10, 2) NOT NULL,  
  FOREIGN KEY (id_siswa) REFERENCES siswa(id_siswa)  
);
```

## 3. TABEL PRODUK

```
CREATE TABLE produk (  
  id_produk INT AUTO_INCREMENT PRIMARY KEY,  
  nama_produk VARCHAR(100) NOT NULL,  
  harga DECIMAL(10, 2) NOT NULL,  
  stok INT NOT NULL  
);
```

#### 4. TABEL DETAIL\_TRANSAKSI

```
CREATE TABLE detail_transaksi (  
    id_detail INT AUTO_INCREMENT PRIMARY KEY,  
    id_transaksi INT NOT NULL,  
    id_produk INT NOT NULL,  
    jumlah INT NOT NULL,  
    total_harga DECIMAL(10, 2) NOT NULL,  
    FOREIGN KEY (id_transaksi) REFERENCES transaksi(id_transaksi),  
    FOREIGN KEY (id_produk) REFERENCES produk(id_produk)  
);
```

#### tampilan data yang berealisasi

```
Database changed  
MariaDB [kantin_sekolah]> SELECT * FROM transaksi;SELECT * FROM detail_transaksi;SELECT * FROM produk;SELECT * FROM siswa;  
+-----+-----+-----+-----+  
| id_transaksi | id_siswa | tanggal_transaksi | total_bayar |  
+-----+-----+-----+-----+  
| 1 | 1 | 2024-11-14 | 20000.00 |  
| 2 | 2 | 2024-11-14 | 12000.00 |  
+-----+-----+-----+-----+  
2 rows in set (0.001 sec)  
  
+-----+-----+-----+-----+-----+  
| id_detail | id_transaksi | id_produk | jumlah | total_harga |  
+-----+-----+-----+-----+-----+  
| 1 | 1 | 1 | 1 | 15000.00 |  
| 2 | 1 | 3 | 1 | 5000.00 |  
| 3 | 2 | 2 | 1 | 12000.00 |  
+-----+-----+-----+-----+-----+  
3 rows in set (0.001 sec)  
  
+-----+-----+-----+-----+  
| id_produk | nama_produk | harga | stok |  
+-----+-----+-----+-----+  
| 1 | Nasi Goreng | 15000.00 | 50 |  
| 2 | Mie Ayam | 12000.00 | 40 |  
| 3 | Es Teh | 5000.00 | 60 |  
+-----+-----+-----+-----+  
3 rows in set (0.001 sec)  
  
+-----+-----+-----+-----+  
| id_siswa | nama_siswa | kelas | saldo |  
+-----+-----+-----+-----+  
| 1 | Ahmad Zaki | X RPL 1 | 50000.00 |  
| 2 | Putri Lestari | XI TKJ 2 | 60000.00 |  
| 3 | Dedi Kurniawan | XII MM 1 | 45000.00 |  
+-----+-----+-----+-----+  
3 rows in set (0.001 sec)
```

## CONTOH

### tabel yang digunakan

#### 1. Tabel produk

Menyimpan data produk yang dijual di kantin, dengan kolom-kolom sebagai berikut:

- `id_produk` (PK): ID unik untuk produk.
- `nama_produk` : Nama produk.
- `harga` : Harga per produk.
- `stok` : Jumlah stok produk yang tersedia.

## 2. Tabel transaksi

Menyimpan data transaksi yang dilakukan siswa di kantin, dengan kolom-kolom sebagai berikut:

- `id_transaksi` (PK): ID transaksi.
- `id_siswa` (FK): ID siswa yang melakukan transaksi.
- `tanggal_transaksi` : Tanggal transaksi dilakukan.
- `total_bayar` : Total harga yang dibayar pada transaksi.

## 3. Tabel detail\_transaksi

Menyimpan detail produk yang dibeli dalam transaksi, dengan kolom-kolom sebagai berikut:

- `id_detail` : ID unik untuk detail transaksi.
- `id_transaksi` (FK): ID transaksi terkait.
- `id_produk` (FK): ID produk yang dibeli.
- `jumlah` : Jumlah produk yang dibeli.
- `total_harga` : Total harga produk yang dibeli ( $\text{jumlah} * \text{harga}$ ).

# 1. Rekap Data Penjualan Bulan Ini:

Tujuan: Menghitung total penjualan untuk setiap produk yang terjual pada bulan ini.

Query:

```
SELECT
    p.nama_produk,
    SUM(dt.jumlah) AS total_terjual,
    SUM(dt.total_harga) AS total_pendapatan
FROM
    transaksi t
JOIN
    detail_transaksi dt ON t.id_transaksi = dt.id_transaksi
JOIN
    produk p ON dt.id_produk = p.id_produk
WHERE
    MONTH(t.tanggal_transaksi) = MONTH(CURDATE())
    AND YEAR(t.tanggal_transaksi) = YEAR(CURDATE())
GROUP BY
    p.id_produk
HAVING
    total_terjual > 0
ORDER BY
    total_pendapatan DESC;
```

penjelasan:

- **Bagian SELECT:**

- `p.nama_produk` : Memilih nama produk dari tabel `produk` untuk ditampilkan.
- `SUM(dt.jumlah) AS total_terjual` : Menghitung total kuantitas produk yang terjual dari tabel `detail_transaksi`.
- `SUM(dt.total_harga) AS total_pendapatan` : Menghitung total pendapatan dari penjualan produk berdasarkan `total_harga` di tabel `detail_transaksi`.

- **Bagian FROM dan JOIN:**

- `transaksi t` : Mengambil data dari tabel `transaksi`.
- `detail_transaksi dt ON t.id_transaksi = dt.id_transaksi` : Melakukan join antara tabel `transaksi` dan `detail_transaksi` berdasarkan kolom `id_transaksi`. Hal ini untuk mendapatkan detail produk yang terjual dalam setiap transaksi.
- `produk p ON dt.id_produk = p.id_produk` : Melakukan join antara tabel `detail_transaksi` dan `produk` untuk mendapatkan nama produk dari ID produk (`id_produk`).

- **Bagian WHERE:**

- `MONTH(t.tanggal_transaksi) = MONTH(CURDATE())` : Memfilter data hanya untuk transaksi yang terjadi pada bulan saat ini.
- `YEAR(t.tanggal_transaksi) = YEAR(CURDATE())` : Memastikan hanya transaksi di tahun ini yang diproses.

- **Bagian GROUP BY:**

- `p.id_produk` : Mengelompokkan data berdasarkan ID produk, sehingga total terjual dan total pendapatan dihitung untuk setiap produk.

- **Bagian HAVING:**

- `total_terjual > 0` : Memastikan hanya produk yang memiliki penjualan (jumlah lebih dari 0) yang ditampilkan.

- **Bagian ORDER BY:**

- `total_pendapatan DESC` : Mengurutkan produk berdasarkan total pendapatan dari yang terbesar ke yang terkecil.

**hasilnya:**

```

Database changed
MariaDB [kantin_sekolah]> SELECT
  ->     p.nama_produk,
  ->     SUM(dt.jumlah) AS total_terjual,
  ->     SUM(dt.total_harga) AS total_pendapatan
  -> FROM
  ->     transaksi t
  -> JOIN
  ->     detail_transaksi dt ON t.id_transaksi = dt.id_transaksi
  -> JOIN
  ->     produk p ON dt.id_produk = p.id_produk
  -> WHERE
  ->     MONTH(t.tanggal_transaksi) = MONTH(CURDATE())
  ->     AND YEAR(t.tanggal_transaksi) = YEAR(CURDATE())
  -> GROUP BY
  ->     p.id_produk
  -> HAVING
  ->     total_terjual > 0
  -> ORDER BY
  ->     total_pendapatan DESC;
+-----+-----+-----+
| nama_produk | total_terjual | total_pendapatan |
+-----+-----+-----+
| Nasi Goreng |          1 |       15000.00 |
| Mie Ayam    |          1 |       12000.00 |
| Es Teh      |          1 |        5000.00 |
+-----+-----+-----+
3 rows in set (0.003 sec)

```

## 2.Barang yang Paling Banyak Laku:

Tujuan: Menampilkan produk yang paling banyak terjual selama bulan ini.

Query:

```

SELECT
    p.id_produk,
    p.nama_produk,
    SUM(dt.jumlah) AS total_terjual
FROM
    detail_transaksi dt
JOIN
    produk p ON dt.id_produk = p.id_produk
JOIN
    transaksi t ON dt.id_transaksi = t.id_transaksi
WHERE
    MONTH(t.tanggal_transaksi) = MONTH(CURDATE())
    AND YEAR(t.tanggal_transaksi) = YEAR(CURDATE())
GROUP BY
    p.id_produk
ORDER BY
    total_terjual DESC

```

```
LIMIT 1;
```

### penjelasan:

- `SUM(dt.jumlah)` : Menghitung total jumlah barang yang terjual untuk setiap produk.
- `MONTH(t.tanggal_transaksi)` : Mengambil bulan dari kolom `tanggal_transaksi` untuk memfilter transaksi berdasarkan bulan saat ini.
- `YEAR(t.tanggal_transaksi)` : Mengambil tahun dari kolom `tanggal_transaksi` untuk memastikan hanya transaksi pada tahun saat ini yang diproses.
- `WHERE MONTH(t.tanggal_transaksi) = MONTH(CURDATE()) AND YEAR(t.tanggal_transaksi) = YEAR(CURDATE())` : Digunakan untuk memfilter transaksi yang hanya terjadi pada bulan dan tahun ini.
- `GROUP BY p.id_produk` : Mengelompokkan data berdasarkan ID produk, sehingga setiap produk memiliki total jumlah terjualnya.
- `ORDER BY total_terjual DESC` : Mengurutkan produk berdasarkan jumlah terjual dari yang terbanyak ke yang paling sedikit.
- `LIMIT 1` : Menampilkan hanya satu produk dengan jumlah terjual tertinggi.

### hasilnya:

```
MariaDB [kantin_sekolah]> SELECT
->     p.id_produk,
->     p.nama_produk,
->     SUM(dt.jumlah) AS total_terjual
-> FROM
->     detail_transaksi dt
-> JOIN
->     produk p ON dt.id_produk = p.id_produk
-> JOIN
->     transaksi t ON dt.id_transaksi = t.id_transaksi
-> WHERE
->     MONTH(t.tanggal_transaksi) = MONTH(CURDATE())
->     AND YEAR(t.tanggal_transaksi) = YEAR(CURDATE())
-> GROUP BY
->     p.id_produk
-> ORDER BY
->     total_terjual DESC
-> LIMIT 1;
```

id_produk	nama_produk	total_terjual
3	Es Teh	5

1 row in set (0.001 sec)