

LAPORAN UJIAN AKHIR SEMESTER
BASIS DATA
DATABASE AGEN TELUR



Disusun oleh :

Soja Purnamasari (4523210104)

Sonia Debora (4523210105)

Kezia Annabel Sinaga (4523210134)

Dosen pengampu :

Adi Wahyu Pribadi, S.Si., M.Kom

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK UNIVERSITAS PANCASILA

2024/2025

Link Github : https://github.com/sojaap/basisdata_agentelur.git

1. Penjelasan *Database* Agen Telur

Setiap bisnis pada umumnya memerlukan arsip data sebagai rupa pencatatan yang terstruktur. Demikian halnya dengan bisnis agen telur, basis data diperlukan untuk mencatat keterangan produk yang dijual, akumulasi transaksi serta rekap dari penjualannya. Meskipun ukuran dari bisnis masih tergolong kecil, *database* membantu menginformasikan alur kas perusahaan yang tentunya menjadi catatan penting bagi pemilik usaha. Hal inilah yang membuat kami mengusul gagasan *database* bernama “agen_telur” untuk mengelola kompleksitas selama bisnis telur berjalan. Dan bukan sebagai historis wirausaha saja, *database* “agen_telur” juga membantu pemilik usaha dalam berkeputusan di langkah selanjutnya.

2. Indetifikasi Entitas dan Relasi

Desain *database* ERD dibuat oleh dua konsep yang saling membangun, yaitu entitas dan relasi. Entitas merupakan objek yang mempunyai karakteristik tersendiri sehingga dapat dibedakan dengan objek lainnya. Sedangkan relasi merupakan hubungan yang terbentuk di antara beberapa entitas.

Entitas fundamental yang dapat ditetapkan di dalam *database* agen_telur ialah sebagai berikut:

a. *Seller*

Seller merupakan objek yang diisi informasi *shifter*/pekerja paruh waktu usaha. Entitas *seller* banyak berelasi dengan tabel lain dengan maksud pertanggungjawaban setiap *shifter* dalam setiap mencatat rekap per harinya. Tabel *seller* dirancang untuk menampung id *shifter*, nama lengkap *shifter* serta nomor telepon *shifter*. Tabel *shifter* dirancang untuk terisi manual untuk mencatat maksimal 3 data dummy.

b. *Kategori*

Kategori merupakan onbjek yang diisi informasi terkait produk-produk yang dijual di dalam agen. Data yang ditampung berupa id kategori, nama produknya, serta harga satuannya. Entitas kategori akan berinteraksi dengan tabel lain melalui ID untuk

pencatatan pembelian serta penjualan per kategori. Tabel kategori dirancang untuk diisi manual untuk mencatat maksimal 20 data dummy terkait produk.

c. Arus_kas

Arus_kas akan menjadi entitas yang menginformasikan alur keuangan perusahaan secara akumulatif. Namun, tabel arus_kas terjalin interaksi khusus dengan tabel relasi laporan_penjualan_harian serta transaksi_supplier untuk mendukung sumber pencatatannya. Arus_kas didukung oleh *trigger* sehingga pada setiap kali pemasukan data *dummy* ke dalam tabel laporan_penjualan_harian serta transaksi_supplier, akan terjadi pencatatan otomatis di tabel arus_kas. Meskipun terhubung dengan dua tabel relasi, arus_kas tetap menjadi entitas dikarenakan mempunyai informasi yang berdiri sendiri. Adapun data yang ditampilkan di dalam arus_kas adalah tanggal transaksi (baik penjualan maupun pembelian), kategori, kuantitas kategori, dan keterangan. Tabel arus_kas dirancang untuk terisi dan ter-update otomatis tatkala laporan_penjualan_harian dan transaksi_supplier menerima data dummy baru.

d. Informasi_keuangan

Informasi_keuangan merupakan entitas yang mencatat kondisi keuangan dalam *realtime*. Sama dengan arus_kas, tetapi bedanya entitas informasi_keuangan ini akan terjadi penambahan atau pengurangan secara langsung yang membuat pengguna bisa melihat saldo perusahaan yang tersedia. Entitas informasi_keuangan tidak bertugas sebagai tabel yang menyediakan banyak informasi. Namun, tabelnya didesain 1 row khusus untuk melihat informasi saldo tunggal saja yang diperbarui setiap kali ada data *dummy* masuk. Dari sini diketahui bahwa informasi_keuangan terisi otomatis selama laporan_penjualan_harian dan transaksi_supplier menerima data *dummy* baru.

Adapun relasi yang terbentuk akibat hubungan antar entitas-entitas ialah sebagai berikut :

a. Transaksi_supplier

Transaksi_supplier merupakan bentuk relasi dari entitas *seller* dan *shifter* yang terikat oleh *FOREIGN KEY* dari id_shifter (dari entitas *seller*) dan id_kategori (dari entitas *kategori*). Informasi tambahan yang ditampilkan di dalam tabel transaksi_supplier ialah jumlah barang yang dibeli, total pembayaran untuk mengetahui nominal total satu kategori, dan tanggal pembelian. Transaksi_supplier didesain untuk menjadi tabel

pengisian secara manual. Namun, pada setiap kali menginput data *dummy* ke tabel akan mempengaruhi pula pencatatan di entitas *arus_kas* dan *informasi_keuangan*

b. Laporan_penjualan_harian

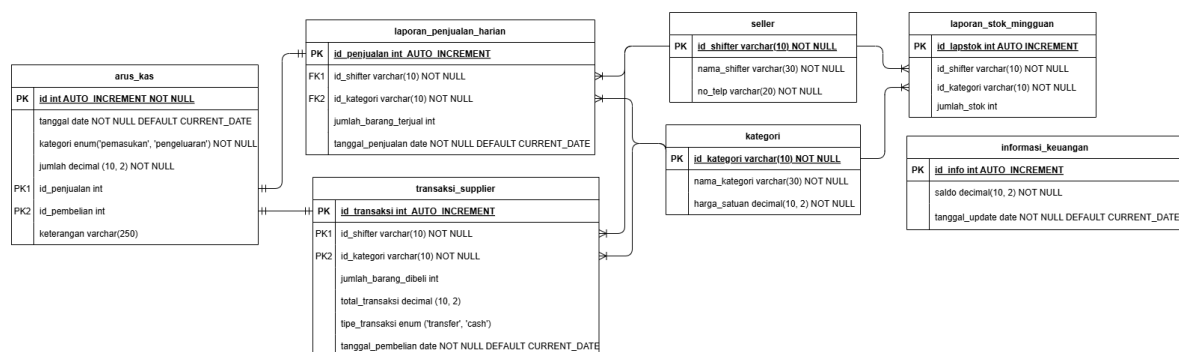
Laporan_penjualan_harian merupakan relasi akibat entitas *shifter* dan kategori untuk menerangkan jumlah yang terjual per produk, tanggal penjualan, serta total pendapatan per produk tiap harinya. Laporan_penjualan_harian dirancang untuk menjadi tabel pengisian secara manual. Tetapi, pada setiap kali menginput data *dummy* akan mempengaruhi pula pencatatan di entitas *arus_kas* dan *informasi_keuangan*

c. Laporan_stok_mingguan

Laporan_stok_mingguan menjadi relasi antara *shifter* dan kategori yang terikat oleh *FOREIGN KEY* dari *id_shifter* (dari entitas *seller*) dan *id_kategori* (dari entitas kategori). Laporan_stok_mingguan menginformasikan ketersediaan stok berdasarkan kategori secara mingguan sebagai catatan pemilik usaha dalam membeli stok berikutnya. Pencatatan laporan stok akan terhubung langsung dengan tabel laporan_penjualan_harian (mengeksrak kolom jumlah_barang_terjual) dan transaksi_supplier (mengeksrak kolom jumlah_barang_dibeli). Dengan demikian, laporan_stok_mingguan akan terisi otomatis selama laporan_penjualan_harian dan laporan transaksi_supplier menerima data *dummy*.

3. Diagram Entitas-Relasi

Diagram di bawah menggambarkan struktur *database* *agen_telur* secara terperinci



Analisis derajat relasi yang terbentuk melalui struktur *database* ialah sebagai berikut :

- a. **seller – laporan_penjualan_harian**, membentuk derajat relasi *one to many* , di mana satu *seller* bisa mengisi banyak laporan tetapi laporan tersebut hanya dimiliki oleh satu *seller*
- b. **seller – transaksi_supplier**, membentuk derajat relasi *one to many* , di mana satu *seller* bisa mencatat banyak transaksi dengan satu *supplier*, tetapi satu *supplier* cukup berurusan dengan satu *seller* saja
- c. **seller – laporan_stok_mingguan**, membentuk derajat relasi *one to many* , di mana satu *seller* bisa melakukan pembaruan informasi ke banyak stok barang
- d. **kategori – laporan_penjualan_harian**, membentuk derajat relasi *one to many*, di mana penjualan kategori dapat ditemukan banyak dan bahkan berulang kali di laporan_penjualan_harian, tetapi satu laporan penjualan hanya untuk satu kategori spesifik
- e. **kategori – transaksi_supplier**, membentuk derajat relasi *one to many*, di mana kategori ditemukan di banyak transaksi_supplier, tetapi satu transaksi hanya untuk satu kategori spesifik
- f. **kategori – laporan_stok_mingguan**, membentuk derajat relasi *one to many*, di mana kategori ditemukan banyak di laporan_stok_mingguan, tetapi satu laporan stok hanya untuk satu kategori spesifik
- g. **arus_kas – transaksi_supplier**, membentuk derajat *one to one* dikarenakan satu transaksi_supplier terhubung dengan satu catatan arus_kas
- h. **arus_kas – laporan_penjualan_harian**, membentuk derajat *one to one* dikarenakan satu laporan_penjualan_harian terhubung dengan satu catatan arus_kas

4. Proses Pembuatan Database Menggunakan MySQL

Database agen_telur dibuat dengan query SQL di dalam PHPmyAdmin. Tidak lupa pada setiap penyisipan query selalu disertai Engine InnoDB agar *Trigger transaction* nanti dapat berjalan lancar.

a. Membuat database agen_telur

Database agen_telur menjadi wadah bagi semua tabel-tabel yang berhubungan dengan operasional bisnis agen telur. Untuk membuatnya, dapat memasukkan query berikut :

```
CREATE DATABASE agen_telur;
```

✓ MySQL memberikan hasil kosong (atau nol baris). (Pencarian dilakukan dalam 0,0012 detik.)

```
CREATE DATABASE agen_telur;
```

[Edit dikotak] [Ubah] [Buat kode PHP]

b. Membuat tabel *seller*

Tabel *seller* merupakan tabel yang menampung data nama dan nomor telepon para *shifter* yang bekerja. Adapun atribut-atribut serta tipe datanya adalah, *id_shifter* (varchar PRIMARY KEY), *nama_shifter*(varchar), dan *no_telp*(varchar). Berikut ialah query-nya :

```
CREATE TABLE seller (  
    id_shifter VARCHAR(10) PRIMARY KEY,  
    nama_shifter VARCHAR(30) NOT NULL,  
    no_telp VARCHAR(20) NOT NULL  
) ENGINE=INNODB;
```

✓ MySQL memberikan hasil kosong (atau nol baris). (Pencarian dilakukan dalam 0,0317 detik.)

```
CREATE TABLE seller ( id_shifter VARCHAR(10) PRIMARY KEY, nama_shifter VARCHAR(30) NOT NULL,  
no_telp VARCHAR(20) NOT NULL ) ENGINE=INNODB;
```

[Edit dikotak] [Ubah] [Buat kode PHP]

c. Membuat Tabel Kategori

Tabel kategori merupakan tabel yang menampung data mengenai produk-produk yang dijual oleh agen. Atribut-atribut serta tipe data yang dideklarasikan adalah *id_kategori* (varchar PRIMARY KEY), *nama_kategori* (varchar), dan *harga_satuan* (decimal). Berikut ialah query-nya :

```
CREATE TABLE kategori (  
    id_kategori varchar(10) PRIMARY KEY,  
    nama_kategori varchar(30) NOT NULL,  
    harga_satuan DECIMAL(10,2) NOT NULL  
)ENGINE=INNODB;
```

✓ MySQL memberikan hasil kosong (atau nol baris). (Pencarian dilakukan dalam 0,0301 detik.)

```
CREATE TABLE kategori ( id_kategori varchar(10) PRIMARY KEY, nama_kategori varchar(30) NOT NULL,
harga_satuan DECIMAL(10,2) NOT NULL )ENGINE=INNODB;
```

[Edit kotak] [Ubah] [Buat kode PHP]

Selama pembuatan tabel kategori dan tabel *seller*, terjadi pengeditan query dikarenakan terlupanya menambahkan NOT NULL ke PRIMARY KEY masing-masing tabel. NOT NULL menjadi hal penting untuk menghindari *primary key* yang bersifat NULL, dan berujung mengakibatkan gagalnya fungsi *trigger*. Pengeditan dilakukan dengan query berikut:

```
ALTER TABLE seller
MODIFY id_shifter VARCHAR(10) NOT NULL;
```

```
ALTER TABLE kategori
MODIFY id_kategori VARCHAR(10) NOT NULL;
```

✓ MySQL memberikan hasil kosong (atau nol baris). (Pencarian dilakukan dalam 0,0203 detik.)

```
ALTER TABLE seller MODIFY id_shifter VARCHAR(10) NOT NULL;
```

[Edit kotak] [Ubah] [Buat kode PHP]

✓ MySQL memberikan hasil kosong (atau nol baris). (Pencarian dilakukan dalam 0,0071 detik.)

```
ALTER TABLE kategori MODIFY id_kategori VARCHAR(10) NOT NULL;
```

[Edit kotak] [Ubah] [Buat kode PHP]

d. Membuat Tabel *arus_kas*

Arus_kas merupakan tabel yang menampung alur kas perusahaan. Ketika tabel relasi laporan_penjualan_harian dan transaksi_supplier menerima data *dummy* baru, tabel *arus_kas* secara otomatis merekam juga data baru tersebut. Atribut-atribut serta tipe datanya adalah id (int PRIMARY KEY), tanggal (date), kategori (enum), jumlah (decimal), id_sumber (int FOREIGN KEY), dan keterangan (text). Berikut ialah query-nya :

```
CREATE TABLE arus_kas (
    id INT AUTO_INCREMENT PRIMARY KEY,
    tanggal DATE NOT NULL DEFAULT CURRENT_DATE,
    kategori ENUM('pemasukan', 'pengeluaran') NOT NULL,
    jumlah DECIMAL(10, 2) NOT NULL,
    id_sumber INT,
    tipe_sumber ENUM ('penjualan', 'pembelian') NOT NULL,
    keterangan VARCHAR(250)
)ENGINE=INNODB;
```

✓ MySQL memberikan hasil kosong (atau nol baris). (Pencarian dilakukan dalam 0,0299 detik.)

```
CREATE TABLE arus_kas( id INT AUTO_INCREMENT PRIMARY KEY, tanggal DATE NOT NULL DEFAULT CURRENT_DATE, kategori ENUM('pemasukan',  
'pengeluaran') NOT NULL, jumlah DECIMAL(10, 2) NOT NULL, id_sumber INT, tipe_sumber ENUM ('penjualan', 'pembelian') NOT NULL,  
keterangan VARCHAR(250) )ENGINE=INNODB;
```

[Edit dikotak] [Ubah] [Buat kode PHP]

Selama pembuatan arus_kas, terdapat beberapa pertimbangan sehingga menghasilkan keputusan untuk melakukan DROP (atau DELETE) kolom id_sumber dan tipe_sumber. Pengeditan dilakukan dengan query berikut :

```
ALTER TABLE arus_kas  
DROP COLUMN id_sumber,  
DROP COLUMN tipe_sumber;
```

✓ MySQL memberikan hasil kosong (atau nol baris). (Pencarian dilakukan dalam 0,0341 detik.)

```
ALTER TABLE arus_kas DROP COLUMN tipe_sumber;
```

[Edit dikotak] [Ubah] [Buat kode PHP]

Untuk mengganti fungsi awal id_sumber, ditambahkan dua kolom baru bernama id_penjualan dan id_pembelian supaya mengetahui secara jelas transaksi penjualan dan pembelian tersebut merujuk ke ID berapa pada tabel asalnya. Penambahan kolom dilakukan dengan query berikut :

```
ALTER TABLE arus_kas  
ADD COLUMN id_penjualan INT,  
ADD COLUMN id_pembelian INT;
```

Kemudian untuk melengkapi fungsi tabel arus_kas, dilakukan hubungan *foreign key* ke id_penjualan yang berasal dari tabel laporan_penjualan_harian dan id_pembelian yang berasal dari transaksi_supppier. Namun, query ini baru bisa ditambahkan ketika tabel laporan_penjualan_harian serta transaksi_supplier sudah jadi. Penambahan kolom dilakukan dengan query berikut :

```
ALTER TABLE arus_kas  
ADD CONSTRAINT fk_arus_kas_penjualan  
FOREIGN KEY (id_penjualan)  
REFERENCES laporan_penjualan_harian(id_penjualan)  
ON DELETE CASCADE;
```

```
ALTER TABLE arus_kas  
ADD CONSTRAINT fk_arus_kas_supplier  
FOREIGN KEY (id_pembelian)  
REFERENCES transaksi_supplier(id_transaksi)
```


ON DELETE CASCADE;
ON UPDATE CASCADE; (Sebaiknya query ini ditambahkan belakangan setelah trigger)

```
✓ MySQL memberikan hasil kosong (atau nol baris). (Pencarian dilakukan dalam 0,0899 detik.)

ALTER TABLE arus_kas ADD CONSTRAINT fk_arus_kas_penjualan FOREIGN KEY (id_sumber) REFERENCES
laporan_penjualan_harian(id_penjualan) ON DELETE CASCADE;

[ Edit dikotak ] [ Ubah ] [ Buat ]
```

```
✓ MySQL memberikan hasil kosong (atau nol baris). (Pencarian dilakukan dalam 0,0398 detik.)

ALTER TABLE arus_kas ADD CONSTRAINT fk_arus_kas_supplier FOREIGN KEY (id_sumber) REFERENCES
transaksi_supplier(id_transaksi) ON DELETE CASCADE;

[ Edit dikotak ] [ Ubah ] [ Buat ]
```

Penambahan query ON DELETE CASCADE bermaksud supaya ketika ada penghapusan data *dummy* dari tabel asal, data *dummy* yang tercatat di *arus_kas* juga turut terhapus.

e. Membuat Tabel informasi_keuangan

Tabel informasi_keuangan merupakan tabel yang menampung informasi akumulatif keuangan perusahaan. Baris dari tabel informasi_keuangan hanya akan tersedia 1 saja, dan kolom saldo akan mengalami UPDATE rutin dari tabel transaksi_supplier dan laporan_penjualan_harian. Atribut-atribut yang melengkapi tabel adalah id_info (int PRIMARY KEY), saldo (decimal), dan tanggal_update (date). Adapun query-nya dapat dilihat di bawah ini :

```
CREATE TABLE informasi_keuangan(
    id_info INT PRIMARY KEY,
    saldo DECIMAL(10, 2) NOT NULL,
    tanggal_update DATE NOT NULL DEFAULT CURRENT_DATE
)ENGINE=INNODB;
```

```
✓ MySQL memberikan hasil kosong (atau nol baris). (Pencarian dilakukan dalam 0,0975 detik.)

CREATE TABLE informasi_keuangan( id_info INT AUTO_INCREMENT PRIMARY KEY, saldo DECIMAL(10, 2) NOT NULL, tanggal_update DATE
NOT NULL DEFAULT CURRENT_DATE )ENGINE=INNODB;

[ Edit dikotak ] [ Ubah ] [ Buat kode PHP ]
```

f. Membuat Tabel Relasi transaksi_supplier

Tabel transaksi_supplier merupakan tabel relasi yang menjadi penghubung antara *shifter* dan kategori terkait pencatatan pembelian dengan *supplier*. Atribut yang melengkapi tabel transaksi_supplier adalah id_transaksi (int PRIMARY KEY), id_shifter (varchar), id_kategori

(varchar), jumlah_barang_dibeli (int), total_transaksi (decimal), tipe_transaksi (enum) dan tanggal_pembelian (date). Id_shifter dan id_kategori akan terhubung oleh tabel entitas *seller* dan kategori. Adapun query-nya dapat dilihat di bawah ini :

```
CREATE TABLE transaksi_supplier(  
    id_transaksi INT AUTO_INCREMENT PRIMARY KEY,  
    id_shifter VARCHAR(10) NOT NULL,  
    id_kategori VARCHAR(10) NOT NULL,  
    jumlah_barang_dibeli INT,  
    total_transaksi DECIMAL(10, 2),  
    tipe_transaksi ENUM('transfer', 'cash'),  
    tanggal_pembelian DATE NOT NULL DEFAULT CURRENT_DATE  
)ENGINE=INNODB;
```

✓ MySQL memberikan hasil kosong (atau nol baris). (Pencarian dilakukan dalam 0,0296 detik.)

```
CREATE TABLE transaksi_supplier( id_transaksi INT AUTO_INCREMENT PRIMARY KEY, id_shifter VARCHAR(10) NOT NULL, id_kategori  
VARCHAR(10) NOT NULL, jumlah_barang_dibeli INT, total_transaksi DECIMAL(10, 2), tipe_transaksi ENUM('transfer', 'cash'),  
tanggal_pembelian DATE NOT NULL DEFAULT CURRENT_DATE )ENGINE=INNODB;
```

[Edit dikotak] [Ubah] [Buat kode PHP]

Untuk menghubungkan id_shifter dan id_kategori ke dua tabel entitas, dilakukan ALTER TABLE untuk menghubungkan foreign key ke id_shifter yang berasal dari tabel *seller* dan id_kategori yang berasal dari tabel kategori. Berikut adalah query untuk menambahkan foreign key :

```
ALTER TABLE transaksi_supplier  
ADD CONSTRAINT fk_transaksi_supplier_shifter  
FOREIGN KEY (id_shifter)  
REFERENCES seller(id_shifter);
```

```
ALTER TABLE transaksi_supplier  
ADD CONSTRAINT fk_transaksi_supplier_kategori  
FOREIGN KEY (id_kategori)  
REFERENCES kategori(id_kategori);
```

g. Membuat Tabel Relasi laporan_penjualan_harian

Tabel laporan_penjualan_harian merupakan tabel relasi yang menjadi penghubung antara *shifter* dan kategori terkait pencatatan penjualan harian. Atribut yang melengkapi tabel laporan_penjualan_harian adalah id_penjualan (int PRIMARY KEY), id_shifter (varchar), id_kategori (varchar), jumlah_barang_terjual (int), total_transaksi (decimal), tipe_transaksi (enum) dan tanggal_pembelian (date). Id_shifter dan id_kategori akan terhubung oleh tabel entitas *seller* dan kategori. Adapun query-nya dapat dilihat di bawah ini :

```
CREATE TABLE laporan_penjualan_harian(  
    id_penjualan INT AUTO_INCREMENT PRIMARY KEY,  
    id_shifter VARCHAR(10) NOT NULL,  
    id_kategori VARCHAR(10) NOT NULL,  
    jumlah_barang_terjual INT,  
    total_transaksi DECIMAL(10, 2),  
    tipe_transaksi ENUM('transfer', 'cash'),  
    tanggal_pembelian DATE NOT NULL DEFAULT CURRENT_DATE  
)ENGINE=INNODB;
```

```

id_penjualan INT AUTO_INCREMENT PRIMARY KEY,
id_shifter VARCHAR(10) NOT NULL,
id_kategori VARCHAR(10) NOT NULL,
jumlah_barang_terjual INT,
tanggal_penjualan DATE NOT NULL DEFAULT CURRENT_DATE
)ENGINE=INNODB;

```

```

✓ MySQL memberikan hasil kosong (atau nol baris). (Pencarian dilakukan dalam 0,0315 detik.)

CREATE TABLE laporan_penjualan_harian( id_penjualan INT AUTO_INCREMENT PRIMARY KEY, id_shifter VARCHAR(10) NOT NULL, id_kategori
VARCHAR(10) NOT NULL, jumlah_barang_terjual INT, tanggal_penjualan DATE NOT NULL DEFAULT CURRENT_DATE )ENGINE=INNODB;

[ Edit dikotak ] [ Ubah ] [ Buat kode PHP ]

```

Untuk melengkapi fungsi tabel, kolom bernama total_penjualan ditambahkan untuk menyatakan total terjual secara nominal. Supaya tidak melakukan perhitungan manual antara total produk terjual dengan harga satuannya, dilakukan UPDATE untuk menghasilkan perhitungan total per kategori. Query-nya dapat dilihat di bawah ini :

```

ALTER TABLE laporan_penjualan_harian
ADD COLUMN total_penjualan DECIMAL(10, 2);

UPDATE laporan_penjualan_harian lph
JOIN kategori k
ON lph.id_kategori = k.id_kategori
SET lph.total_penjualan = lph.jumlah_barang_terjual * k.harga_satuan;

```

```

✓ MySQL memberikan hasil kosong (atau nol baris). (Pencarian dilakukan dalam 0,0238 detik.)

ALTER TABLE laporan_penjualan_harian ADD COLUMN total_penjualan DECIMAL(10, 2);

[ Edit dikotak ] [ Ubah ] [ Buat kode PHP ]

```

```

✓ 0 baris terpengaruh. (Pencarian dilakukan dalam 0,0016 detik.)

UPDATE laporan_penjualan_harian lph JOIN kategori k ON lph.id_kategori = k.id_kategori SET
lph.total_penjualan = lph.jumlah_barang_terjual * k.harga_satuan;

[ Edit dikotak ] [ Ubah ] [ Buat kode PHP ]

```

Selanjutnya melakukan ALTER TABLE untuk menghubungkan foreign key ke id_shifter yang berasal dari tabel *seller* dan id_kategori yang berasal dari tabel kategori. Berikut adalah query untuk menambahkan foreign key :

```

ALTER TABLE laporan_penjualan_harian
ADD CONSTRAINT fk_id_dari_shifter
FOREIGN KEY (id_shifter)
REFERENCES seller(id_shifter);

```

```

ALTER TABLE laporan_penjualan_harian

```

```
ADD CONSTRAINT fk_id_dari_kategori
FOREIGN KEY (id_kategori)
REFERENCES kategori(id_kategori);
```

✓ MySQL memberikan hasil kosong (atau nol baris). (Pencarian dilakukan dalam 0,0814 detik.)

```
ALTER TABLE laporan_penjualan_harian ADD CONSTRAINT fk_id_dari_shifter FOREIGN KEY (id_shifter) REFERENCES seller(id_shifter);
```

[Edit dikotak] [Ubah] [Buat kode PHP]

✓ MySQL memberikan hasil kosong (atau nol baris). (Pencarian dilakukan dalam 0,0860 detik.)

```
ALTER TABLE laporan_penjualan_harian ADD CONSTRAINT fk_id_dari_kategori FOREIGN KEY (id_kategori) REFERENCES kategori(id_kategori);
```

[Edit dikotak] [Ubah] [Buat kode PHP]

h. Membuat Tabel Relasi laporan_stok_mingguan

Tabel laporan_stok_mingguan merupakan tabel relasi yang menjadi penghubung antara *shifter* dan kategori terkait pencatatan stok secara mingguan. Atribut yang melengkapi tabel laporan_stok_mingguan adalah id_lapstok (int PRIMARY KEY), id_shifter (varchar), id_kategori (varchar), jumlah_stok (int), dan tanggal_penjualan (date). Adapun query-nya dapat dilihat di bawah ini :

```
CREATE TABLE laporan_stok_mingguan(
    id_lapstok INT AUTO_INCREMENT PRIMARY KEY,
    id_shifter VARCHAR(10) NOT NULL,
    id_kategori VARCHAR(10) NOT NULL,
    jumlah_stok INT,
    tanggal_penjualan DATE NOT NULL DEFAULT CURRENT_DATE
)ENGINE=INNODB;
```

✓ MySQL memberikan hasil kosong (atau nol baris). (Pencarian dilakukan dalam 0,0175 detik.)

```
CREATE TABLE laporan_stok_mingguan( id_lapstok INT AUTO_INCREMENT PRIMARY KEY, id_shifter VARCHAR(10) NOT NULL, id_kategori VARCHAR(10) NOT NULL, jumlah_stok INT, tanggal_penjualan DATE NOT NULL DEFAULT CURRENT_DATE )ENGINE=INNODB;
```

[Edit dikotak] [Ubah] [Buat kode PHP]

Selama pembuatan tabel, terdapat pertimbangan untuk menghilangkan salah satu atribut untuk memperingkas tabel. Untuk itu, dilakukan keputusan melakukan DROP (atau *delete*) kolom tanggal_penjualan :

```
ALTER TABLE laporan_stok_mingguan DROP COLUMN tanggal_penjualan;
```

✓ MySQL memberikan hasil kosong (atau nol baris). (Pencarian dilakukan dalam 0,0336 detik.)

```
ALTER TABLE laporan_stok_mingguan DROP COLUMN tanggal_penjualan;
```

[Edit dikotak] [Ubah] [Buat kode PHP]

Selanjutnya melakukan ALTER TABLE untuk menghubungkan foreign key ke id_shifter yang berasal dari tabel *seller* dan id_kategori yang berasal dari tabel kategori. Berikut adalah query untuk menambahkan foreign key :

```
ALTER TABLE laporan_stok_mingguan
ADD CONSTRAINT fk_id_asal_shifter
FOREIGN KEY (id_shifter)
REFERENCES seller(id_shifter);
```

```
ALTER TABLE laporan_stok_mingguan
ADD CONSTRAINT fk_id_asal_kategori
FOREIGN KEY (id_kategori)
REFERENCES kategori(id_kategori);
```

✓ MySQL memberikan hasil kosong (atau nol baris). (Pencarian dilakukan dalam 0,1054 detik.)

```
ALTER TABLE laporan_stok_mingguan ADD CONSTRAINT fk_id_asal_shifter FOREIGN KEY (id_shifter) REFERENCES seller(id_shifter);
```

[Edit dikotak] [Ubah] [Buat kode PHP]

✓ MySQL memberikan hasil kosong (atau nol baris). (Pencarian dilakukan dalam 0,0410 detik.)

```
ALTER TABLE laporan_stok_mingguan ADD CONSTRAINT fk_id_asal_kategori FOREIGN KEY (id_kategori) REFERENCES kategori(id_kategori);
```

[Edit dikotak] [Ubah] [Buat kode PHP]

5. Pengisian Data *Dummy* ke dalam Setiap Tabel

Data *dummy* merupakan data sementara untuk pengujian *database* agen_telur kali ini. Pengisian data *dummy* dilakukan sebanyak 20 kali di setiap tabel, terkecuali tabel *seller* dan informasi_keuangan, serta beberapa tabel lain yang memang dirancang untuk menerima pengisian otomatis dengan sistem *trigger*. Pengisian data dilakukan secara berurut supaya setiap tabel berjalan dengan lancar. Tetapi catatan penting di sini, pengisian data disarankan dilakukan ketika *trigger* sudah dimasukkan.

a. Pengisian Data *Dummy* ke Tabel *seller*

Tabel *seller* cukup diisi oleh 3 data *dummy* yang berisi tentang ID, nama dan nomor telepon para *shifter*. Berikut ialah data *dummy* yang dimasukkan ke dalam tabel *seller* :

```
INSERT INTO seller (id_shifter, nama_shifter, no_telp)
VALUES
```

```
('kezia', 'Kezia Annabel', '08123456789'),
('soja', 'Soja Purnamasari', '08234567890'),
('sonia', 'Sonia Debora', '08123456789');
```

✓ 3 baris ditambahkan. (Pencarian dilakukan dalam 0,0091 detik.)

```
INSERT INTO seller (id_shifter, nama_shifter, no_telp) VALUES ('kezia', 'Kezia Annabel', '08123456789'), ('soja', 'Soja Purnamasari', '08234567890'), ('sonia', 'Sonia Debora', '08123456789');
```

[Edit dikotak] [Ubah] [Buat kode PHP]

 				id_shifter	nama_shifter	no_telp
<input type="checkbox"/>	 Ubah	 Salin	 Hapus	kezia	Kezia Annabel	08123456789
<input type="checkbox"/>	 Ubah	 Salin	 Hapus	soja	Soja Purnamasari	08234567890
<input type="checkbox"/>	 Ubah	 Salin	 Hapus	sonia	Sonia Debora	08123456789

 ☐ Pilih Semua Dengan pilihan:  Ubah  Salin  Hapus  Ekspor

b. Pengisian Data *Dummy* ke Tabel kategori

Tabel kategori diisi oleh 20 data *dummy* yang berisi informasi mengenai ID barang, nama barang, dan harga satuannya. Berikut ialah data *dummy* yang dimasukkan ke dalam tabel kategori :

```
INSERT INTO kategori VALUES
('telur akm', 'Telur Ayam Kampung', 2500.00),
('telur an', 'Telur Ayam Negeri', 2000.00),
('telur bebek', 'Telur Bebek', 3000.00),
('telur puyuh', 'Telur Puyuh', 500.00),
('telur asin', 'Telur Asin', 5000.00),
('telur retak', 'Telur Retak', 1500.00),
('minyak 1l', 'Minyak 1 Liter', 15000.00),
('minyak 2l', 'Minyak 2 Liter', 30000.00),
('tissue', 'Tissue', 20000.00),
('bumbu', 'Bumbu Racik', 2000.00),
('beras 5kg', 'Beras 5 Kg', 60000.00),
('royco', 'Royco', 1000.00),
('garam', 'Garam', 5000.00),
('cuka', 'Cuka', 8000.00),
('indomie', 'Indomie', 3000.00),
('kecap', 'Kecap Manis', 10000.00),
('kunyit', 'Kunyit Bubuk', 1000.00),
('merica', 'Merica Bubuk', 1000.00),
('mie sedap', 'Mie Sedap', 2500.00),
('saos', 'Saos Sambal', 12000.00),
```

				id_kategori	nama_kategori	harga_satuan
<input type="checkbox"/>		Ubah		Salin		Hapus
					beras 5kg	Beras 5 Kg
						60000.00
<input type="checkbox"/>		Ubah		Salin		Hapus
					bumbu	Bumbu Racik
						2000.00
<input type="checkbox"/>		Ubah		Salin		Hapus
					cuka	Cuka
						8000.00
<input type="checkbox"/>		Ubah		Salin		Hapus
					garam	Garam
						5000.00
<input type="checkbox"/>		Ubah		Salin		Hapus
					indomie	Indomie
						3000.00
<input type="checkbox"/>		Ubah		Salin		Hapus
					kecap	Kecap Manis
						10000.00
<input type="checkbox"/>		Ubah		Salin		Hapus
					kunyit	Kunyit Bubuk
						1000.00
<input type="checkbox"/>		Ubah		Salin		Hapus
					merica	Merica Bubuk
						1000.00
<input type="checkbox"/>		Ubah		Salin		Hapus
					mie sedap	Mie Sedap
						2500.00
<input type="checkbox"/>		Ubah		Salin		Hapus
					minyak 1l	Minyak 1 Liter
						15000.00
<input type="checkbox"/>		Ubah		Salin		Hapus
					minyak 2l	Minyak 2 Liter
						30000.00
<input type="checkbox"/>		Ubah		Salin		Hapus
					royco	Royco
						1000.00
<input type="checkbox"/>		Ubah		Salin		Hapus
					saos	Saos Sambal
						12000.00
<input type="checkbox"/>		Ubah		Salin		Hapus
					telur akm	Telur Ayam Kampung
						2500.00
<input type="checkbox"/>		Ubah		Salin		Hapus
					telur an	Telur Ayam Negeri
						2000.00
<input type="checkbox"/>		Ubah		Salin		Hapus
					telur asin	Telur Asin
						5000.00
<input type="checkbox"/>		Ubah		Salin		Hapus
					telur bebe	Telur Bebek
						3000.00
<input type="checkbox"/>		Ubah		Salin		Hapus
					telur puyu	Telur Puyuh
						500.00
<input type="checkbox"/>		Ubah		Salin		Hapus
					telur reta	Telur Retak
						1500.00
<input type="checkbox"/>		Ubah		Salin		Hapus
					tissue	Tissue
						20000.00

☐ Pilih Semua Dengan pilihan: Ubah Salin Hapus Ekspor

c. Pengisian Data *Dummy* ke Tabel informasi_keuangan

Tabel informasi_keuangan dibuat khusus hanya untuk memiliki 1 baris untuk menampilkan informasi saldo dan tanggal terakhir data diperbarui. Berikut ialah data *dummy* yang dimasukkan ke dalam tabel informasi_keuangan :

```
INSERT INTO informasi_keuangan (id_info, saldo, tanggal_update)
VALUES
(1, 0.00, CURRENT_DATE);
```

				id_info	saldo	tanggal_update
<input type="checkbox"/>		Ubah		Salin		Hapus
				1	0.00	2025-01-02

☐ Pilih Semua Dengan pilihan: Ubah Salin Hapus Ekspor

d. Pengisian Data *Dummy* ke Tabel laporan_stok_mingguan

Tabel laporan_stok_mingguan perlu diatur terlebih dahulu agar ketika data dari tabel transaksi_supplier maupun dari tabel_penjualan_harian masuk, laporan_stok_mingguan

otomatis menerima data terbarunya. Pada tabel laporan_stok_mingguan, *shifter* dibuat untuk bertanggungjawab pada masing-masing pembagian barang. Ini juga bertujuan untuk konsistensi data supaya ketika pembelian barang yang sama oleh dua *shifter* tidak menimbulkan konflik. Berikut ialah data *dummy* yang dimasukkan ke dalam tabel laporan_stok_mingguan :

```
INSERT INTO laporan_stok_mingguan (id_shifter, id_kategori, jumlah_stok)
VALUES
```





























































```
('sonia','telur akm', 0),
('sonia','telur an', 0),
('sonia','telur bebek', 0),
('sonia','telur puyuh', 0),
('sonia','telur asin', 0),
('sonia','telur retak', 0),
('soja','minyak 1l', 0),
('soja','minyak 2l', 0),
('soja','tissue', 0),
('soja','beras 5kg', 0),
('soja','indomie', 0),
('soja','mie sedap', 0),
('soja','bumbu', 0),
('kezia','royco', 0),
('kezia','garam', 0),
('kezia','cuka', 0),
('kezia','kecap', 0),
('kezia','kunyit', 0),
('kezia','merica', 0),
('kezia','saos', 0);
```


✔ 20 baris ditambahkan.

Nomor baris baru: 20 (Pencarian dilakukan dalam 0,0006 detik.)

```
INSERT INTO laporan_stok_mingguan (id_shifter, id_kategori, jumlah_stok) VALUES ('sonia','telur
akm', 0), ('sonia','telur an', 0), ('sonia','telur bebek', 0), ('sonia','telur puyuh', 0),
('sonia','telur asin', 0), ('sonia','telur retak', 0), ('soja','minyak 1l', 0), ('soja','minyak
2l', 0), ('soja','tissue', 0), ('soja','beras 5kg', 0), ('soja','indomie', 0), ('soja','mie
sedap', 0), ('soja','bumbu', 0), ('kezia','royco', 0), ('kezia','garam', 0), ('kezia','cuka',
0), ('kezia','kecap', 0), ('kezia','kunyit', 0), ('kezia','merica', 0), ('kezia','saos', 0);
```


[Edit dikotak] [Ubah] [Buat kode PHP]


				id_lapstok	id_shifter	id_kategori	jumlah_stok
<input type="checkbox"/>	 Ubah	 Salin	 Hapus	1	sonia	telur akm	0
<input type="checkbox"/>	 Ubah	 Salin	 Hapus	2	sonia	telur an	0
<input type="checkbox"/>	 Ubah	 Salin	 Hapus	3	sonia	telur bebe	0
<input type="checkbox"/>	 Ubah	 Salin	 Hapus	4	sonia	telur puyu	0
<input type="checkbox"/>	 Ubah	 Salin	 Hapus	5	sonia	telur asin	0
<input type="checkbox"/>	 Ubah	 Salin	 Hapus	6	sonia	telur reta	0
<input type="checkbox"/>	 Ubah	 Salin	 Hapus	7	soja	minyak 1l	0
<input type="checkbox"/>	 Ubah	 Salin	 Hapus	8	soja	minyak 2l	0
<input type="checkbox"/>	 Ubah	 Salin	 Hapus	9	soja	tissue	0
<input type="checkbox"/>	 Ubah	 Salin	 Hapus	10	soja	beras 5kg	0
<input type="checkbox"/>	 Ubah	 Salin	 Hapus	11	soja	indomie	0
<input type="checkbox"/>	 Ubah	 Salin	 Hapus	12	soja	mie sedap	0
<input type="checkbox"/>	 Ubah	 Salin	 Hapus	13	soja	bumbu	0
<input type="checkbox"/>	 Ubah	 Salin	 Hapus	14	kezia	royco	0
<input type="checkbox"/>	 Ubah	 Salin	 Hapus	15	kezia	garam	0
<input type="checkbox"/>	 Ubah	 Salin	 Hapus	16	kezia	cuka	0
<input type="checkbox"/>	 Ubah	 Salin	 Hapus	17	kezia	kecap	0
<input type="checkbox"/>	 Ubah	 Salin	 Hapus	18	kezia	kunyit	0
<input type="checkbox"/>	 Ubah	 Salin	 Hapus	19	kezia	merica	0
<input type="checkbox"/>	 Ubah	 Salin	 Hapus	20	kezia	saos	0





☐ Pilih Semua

Dengan pilihan:

 Ubah

 Salin

 Hapus

 Ekspor

(Mungkin beberapa id_kategori terlihat terpotong. Hal ini dikarenakan varchar yang menampung atribut id_kategori hanya dirancang untuk mempunyai array sepanjang 10 saja.)

e. Pengisian Data *Dummy* ke Tabel transaksi_supplier

Tabel transaksi_supplier diisi oleh 20 data *dummy* yang berisi informasi penting mengenai ID barang, jumlah barang dibeli, total transaksi serta tipe transaksi. Berikut ialah data *dummy* yang dimasukkan ke dalam tabel transaksi_supplier:

```

INSERT INTO transaksi_supplier (id_shifter, id_kategori, jumlah_barang_dibeli,
total_transaksi, tipe_transaksi, tanggal_pembelian) VALUES
('sonia','telur akm', 120, 276000, 'transfer', CURRENT_DATE),
('sonia','telur an', 120, 160000, 'transfer', CURRENT_DATE),
('sonia','telur bebek', 60, 170000, 'transfer', CURRENT_DATE),
('sonia','telur puyuh', 60, 50000, 'transfer', CURRENT_DATE),
('sonia','telur asin', 30, 140000, 'transfer', CURRENT_DATE),
('sonia','telur retak', 50, 0, 'transfer', CURRENT_DATE),
('soja','minyak 1l', 30, 300000, 'transfer', CURRENT_DATE),
('soja','minyak 2l', 30, 600000, 'transfer', CURRENT_DATE),
('soja','tissue', 10, 170000, 'transfer', CURRENT_DATE),
('soja','beras 5kg', 10, 500000, 'transfer', CURRENT_DATE),
('soja','indomie', 30, 75000, 'transfer', CURRENT_DATE),

```

```

('soja','mie sedap', 30, 65000, 'transfer', CURRENT_DATE),
('soja','bumbu', 15, 30000, 'transfer', CURRENT_DATE),
('kezia','royco', 30, 20000, 'cash', CURRENT_DATE),
('kezia','garam', 15, 60000, 'cash', CURRENT_DATE),
('kezia','cuka', 10, 60000, 'cash', CURRENT_DATE),
('kezia','kecap', 20, 180000, 'cash', CURRENT_DATE),
('kezia','kunyit', 30, 15000, 'cash', CURRENT_DATE),
('kezia', 'merica', 30, 15000, 'cash', CURRENT_DATE),
('kezia', 'saos', 20, 200000, 'cash', CURRENT_DATE);

```

20 baris ditambahkan.
 Nomor baris baru: 27 (Pencarian dilakukan dalam 0,0007 detik.)

```

INSERT INTO transaksi_supplier (id_shifter, id_kategori, jumlah_barang_dibeli, total_transaksi,
tipe_transaksi, tanggal_pembelian) VALUES ('sonia','telur akm', 120, 276000, 'transfer',
CURRENT_DATE), ('sonia','telur an', 120, 160000, 'transfer', CURRENT_DATE), ('sonia','telur bebek',
60, 170000, 'transfer', CURRENT_DATE), ('sonia','telur puyuh', 60, 50000, 'transfer',
CURRENT_DATE), ('sonia','telur asin', 30, 140000, 'transfer', CURRENT_DATE), ('sonia','telur
retak', 50, 0, 'transfer', CURRENT_DATE), ('soja','minyak 1l', 30, 300000, 'transfer',

```

[Ubah]

<div><div><div><div></div></div><div></div></div><div></div></div>				id_transaksi	id_shifter	id_kategori	jumlah_barang_dibeli	total_transaksi	tipe_transaksi	tanggal_pembelian
<div><div><div></div></div><div><div>Ubah</div><div><div><div></div></div><div></div></div><div><div>Salin</div><div><div><div></div></div><div></div></div><div><div>Hapus</div><div><div><div></div></div><div></div></div></div></div></div></div>	1	sonia	telur akm	120	276000.00	transfer	2025-01-05			
<div><div><div></div></div><div><div>Ubah</div><div><div><div></div></div><div></div></div><div><div>Salin</div><div><div><div></div></div><div></div></div><div><div>Hapus</div><div><div><div></div></div><div></div></div></div></div></div></div>	2	sonia	telur an	120	160000.00	transfer	2025-01-05			
<div><div><div></div></div><div><div>Ubah</div><div><div><div></div></div><div></div></div><div><div>Salin</div><div><div><div></div></div><div></div></div><div><div>Hapus</div><div><div><div></div></div><div></div></div></div></div></div></div>	3	sonia	telur bebe	60	170000.00	transfer	2025-01-05			
<div><div><div></div></div><div><div>Ubah</div><div><div><div></div></div><div></div></div><div><div>Salin</div><div><div><div></div></div><div></div></div><div><div>Hapus</div><div><div><div></div></div><div></div></div></div></div></div></div>	4	sonia	telur puyu	60	50000.00	transfer	2025-01-05			
<div><div><div></div></div><div><div>Ubah</div><div><div><div></div></div><div></div></div><div><div>Salin</div><div><div><div></div></div><div></div></div><div><div>Hapus</div><div><div><div></div></div><div></div></div></div></div></div></div>	5	sonia	telur asin	30	140000.00	transfer	2025-01-05			
<div><div><div></div></div><div><div>Ubah</div><div><div><div></div></div><div></div></div><div><div>Salin</div><div><div><div></div></div><div></div></div><div><div>Hapus</div><div><div><div></div></div><div></div></div></div></div></div></div>	6	sonia	telur reta	50	0.00	transfer	2025-01-05			
<div><div><div></div></div><div><div>Ubah</div><div><div><div></div></div><div></div></div><div><div>Salin</div><div><div><div></div></div><div></div></div><div><div>Hapus</div><div><div><div></div></div><div></div></div></div></div></div></div>	7	soja	minyak 1l	30	300000.00	transfer	2025-01-05			
<div><div><div></div></div><div><div>Ubah</div><div><div><div></div></div><div></div></div><div><div>Salin</div><div><div><div></div></div><div></div></div><div><div>Hapus</div><div><div><div></div></div><div></div></div></div></div></div></div>	8	soja	minyak 2l	30	600000.00	transfer	2025-01-05			
<div><div><div></div></div><div><div>Ubah</div><div><div><div></div></div><div></div></div><div><div>Salin</div><div><div><div></div></div><div></div></div><div><div>Hapus</div><div><div><div></div></div><div></div></div></div></div></div></div>	9	soja	tissue	10	170000.00	transfer	2025-01-05			
<div><div><div></div></div><div><div>Ubah</div><div><div><div></div></div><div></div></div><div><div>Salin</div><div><div><div></div></div><div></div></div><div><div>Hapus</div><div><div><div></div></div><div></div></div></div></div></div></div>	10	soja	beras 5kg	10	500000.00	transfer	2025-01-05			
<div><div><div></div></div><div><div>Ubah</div><div><div><div></div></div><div></div></div><div><div>Salin</div><div><div><div></div></div><div></div></div><div><div>Hapus</div><div><div><div></div></div><div></div></div></div></div></div></div>	11	soja	indomie	30	75000.00	transfer	2025-01-05			
<div><div><div></div></div><div><div>Ubah</div><div><div><div></div></div><div></div></div><div><div>Salin</div><div><div><div></div></div><div></div></div><div><div>Hapus</div><div><div><div></div></div><div></div></div></div></div></div></div>	12	soja	mie sedap	30	65000.00	transfer	2025-01-05			
<div><div><div></div></div><div><div>Ubah</div><div><div><div></div></div><div></div></div><div><div>Salin</div><div><div><div></div></div><div></div></div><div><div>Hapus</div><div><div><div></div></div><div></div></div></div></div></div></div>	13	soja	bumbu	15	30000.00	transfer	2025-01-05			
<div><div><div></div></div><div><div>Ubah</div><div><div><div></div></div><div></div></div><div><div>Salin</div><div><div><div></div></div><div></div></div><div><div>Hapus</div><div><div><div></div></div><div></div></div></div></div></div></div>	14	kezia	royco	30	20000.00	cash	2025-01-05			
<div><div><div></div></div><div><div>Ubah</div><div><div><div></div></div><div></div></div><div><div>Salin</div><div><div><div></div></div><div></div></div><div><div>Hapus</div><div><div><div></div></div><div></div></div></div></div></div></div>	15	kezia	garam	15	60000.00	cash	2025-01-05			
<div><div><div></div></div><div><div>Ubah</div><div><div><div></div></div><div></div></div><div><div>Salin</div><div><div><div></div></div><div></div></div><div><div>Hapus</div><div><div><div></div></div><div></div></div></div></div></div></div>	16	kezia	cuka	10	60000.00	cash	2025-01-05			
<div><div><div></div></div><div><div>Ubah</div><div><div><div></div></div><div></div></div><div><div>Salin</div><div><div><div></div></div><div></div></div><div><div>Hapus</div><div><div><div></div></div><div></div></div></div></div></div></div>	17	kezia	kecap	20	180000.00	cash	2025-01-05			
<div><div><div></div></div><div><div>Ubah</div><div><div><div></div></div><div></div></div><div><div>Salin</div><div><div><div></div></div><div></div></div><div><div>Hapus</div><div><div><div></div></div><div></div></div></div></div></div></div>	18	kezia	kunyit	30	15000.00	cash	2025-01-05			
<div><div><div></div></div><div><div>Ubah</div><div><div><div></div></div><div></div></div><div><div>Salin</div><div><div><div></div></div><div></div></div><div><div>Hapus</div><div><div><div></div></div><div></div></div></div></div></div></div>	19	kezia	merica	30	15000.00	cash	2025-01-05			
<div><div><div></div></div><div><div>Ubah</div><div><div><div></div></div><div></div></div><div><div>Salin</div><div><div><div></div></div><div></div></div><div><div>Hapus</div><div><div><div></div></div><div></div></div></div></div></div></div>	20	kezia	saos	20	200000.00	cash	2025-01-05			

Pilih Semua

Dengan nilai:

Ubah

Salin

Hapus

Clear

f. Pengisian Data *Dummy* ke Tabel laporan_penjualan_harian

Tabel laporan_penjualan_harian diisi oleh 20 data *dummy* yang berisi informasi mengenai ID barang, jumlah barang terjual, dan total pendapatan per barangnya, yang merupakan hasil

kali antara harga satuan (diatur di tabel kategori) dengan jumlah barang terjual di tanggal itu. Berikut ialah data *dummy* yang dimasukkan ke dalam tabel laporan_penjualan_harian :

```
INSERT INTO laporan_penjualan_harian
(id_shifter, id_kategori, jumlah_barang_terjual, tanggal_penjualan)
VALUES
('sonia','telur akm', 100, CURRENT_DATE),
('sonia','telur an', 120, CURRENT_DATE),
('sonia','telur bebek', 40, CURRENT_DATE),
('sonia','telur puyuh', 40, CURRENT_DATE),
('sonia','telur asin', 30, CURRENT_DATE),
('sonia','telur retak', 45, CURRENT_DATE),
('soja','minyak 1l', 10, CURRENT_DATE),
('soja','minyak 2l', 7, CURRENT_DATE),
('soja','tissue', 5, CURRENT_DATE),
('soja','beras 5kg', 6, CURRENT_DATE),
('soja','indomie', 26, CURRENT_DATE),
('soja','mie sedap', 14, CURRENT_DATE),
('soja','bumbu', 8, CURRENT_DATE),
('kezia','royco', 27, CURRENT_DATE),
('kezia','garam', 4, CURRENT_DATE),
('kezia','cuka', 2, CURRENT_DATE),
('kezia','kecap', 3, CURRENT_DATE),
('kezia','kunyit', 13, CURRENT_DATE),
('kezia','merica', 21, CURRENT_DATE),
('kezia','saos', 6, CURRENT_DATE);
```

✓ 20 baris ditambahkan.

Nomor baris baru: 21 (Pencarian dilakukan dalam 0,0006 detik.)

```
INSERT INTO laporan_penjualan_harian (id_shifter, id_kategori, jumlah_barang_terjual,
tanggal_penjualan) VALUES ('sonia','telur akm', 100, CURRENT_DATE), ('sonia','telur an', 120,
CURRENT_DATE), ('sonia','telur bebek', 40, CURRENT_DATE), ('sonia','telur puyuh', 40,
CURRENT_DATE), ('sonia','telur asin', 30, CURRENT_DATE), ('sonia','telur retak', 45, CURRENT_DATE),
('soja','minyak 1l', 10, CURRENT_DATE), ('soja','minyak 2l', 7, CURRENT_DATE), ('soja','tissue', 5,
CURRENT_DATE), ('soja','beras 5kg', 6, CURRENT_DATE), ('soja','indomie', 26, CURRENT_DATE),
```

[Edit dikotak] [Ubah] [Buat kode PHP]

<div><div><div>←</div><div>T</div><div>→</div></div></div>				id_penjualan	id_shifter	id_kategori	jumlah_barang_terjual	tanggal_penjualan	total_penjualan
<div><div><div><div></div></div></div><div><div>Ubah</div><div>Salin</div><div>Hapus</div></div></div>		1	sonia	telur akm	100	2025-01-05	250000.00		
<div><div><div><div></div></div></div><div><div>Ubah</div><div>Salin</div><div>Hapus</div></div></div>		2	sonia	telur an	120	2025-01-05	240000.00		
<div><div><div><div></div></div></div><div><div>Ubah</div><div>Salin</div><div>Hapus</div></div></div>		3	sonia	telur bebe	40	2025-01-05	120000.00		
<div><div><div><div></div></div></div><div><div>Ubah</div><div>Salin</div><div>Hapus</div></div></div>		4	sonia	telur puyu	40	2025-01-05	20000.00		
<div><div><div><div></div></div></div><div><div>Ubah</div><div>Salin</div><div>Hapus</div></div></div>		5	sonia	telur asin	30	2025-01-05	150000.00		
<div><div><div><div></div></div></div><div><div>Ubah</div><div>Salin</div><div>Hapus</div></div></div>		6	sonia	telur reta	45	2025-01-05	67500.00		
<div><div><div><div></div></div></div><div><div>Ubah</div><div>Salin</div><div>Hapus</div></div></div>		7	soja	minyak 1l	10	2025-01-05	150000.00		
<div><div><div><div></div></div></div><div><div>Ubah</div><div>Salin</div><div>Hapus</div></div></div>		8	soja	minyak 2l	7	2025-01-05	210000.00		
<div><div><div><div></div></div></div><div><div>Ubah</div><div>Salin</div><div>Hapus</div></div></div>		9	soja	tissue	5	2025-01-05	100000.00		
<div><div><div><div></div></div></div><div><div>Ubah</div><div>Salin</div><div>Hapus</div></div></div>		10	soja	beras 5kg	6	2025-01-05	360000.00		
<div><div><div><div></div></div></div><div><div>Ubah</div><div>Salin</div><div>Hapus</div></div></div>		11	soja	indomie	26	2025-01-05	78000.00		
<div><div><div><div></div></div></div><div><div>Ubah</div><div>Salin</div><div>Hapus</div></div></div>		12	soja	mie sedap	14	2025-01-05	35000.00		
<div><div><div><div></div></div></div><div><div>Ubah</div><div>Salin</div><div>Hapus</div></div></div>		13	soja	bumbu	8	2025-01-05	16000.00		
<div><div><div><div></div></div></div><div><div>Ubah</div><div>Salin</div><div>Hapus</div></div></div>		14	kezia	royco	27	2025-01-05	27000.00		
<div><div><div><div></div></div></div><div><div>Ubah</div><div>Salin</div><div>Hapus</div></div></div>		15	kezia	garam	4	2025-01-05	20000.00		
<div><div><div><div></div></div></div><div><div>Ubah</div><div>Salin</div><div>Hapus</div></div></div>		16	kezia	cuka	2	2025-01-05	16000.00		
<div><div><div><div></div></div></div><div><div>Ubah</div><div>Salin</div><div>Hapus</div></div></div>		17	kezia	kecap	3	2025-01-05	30000.00		
<div><div><div><div></div></div></div><div><div>Ubah</div><div>Salin</div><div>Hapus</div></div></div>		18	kezia	kunyit	13	2025-01-05	13000.00		
<div><div><div><div></div></div></div><div><div>Ubah</div><div>Salin</div><div>Hapus</div></div></div>		19	kezia	merica	21	2025-01-05	21000.00		
<div><div><div><div></div></div></div><div><div>Ubah</div><div>Salin</div><div>Hapus</div></div></div>		20	kezia	saos	6	2025-01-05	72000.00		

g. Impact ke Tabel laporan_stok_mingguan

Seperti yang sudah disinggung sebelumnya, tabel laporan_stok_mingguan bersifat otomatis terisi dengan data jumlah_stok yang selalu UPDATE selama data terbaru laporan_penjualan_harian senantiasa berkurang atau data terbaru transaksi_supplier senantiasa bertambah.

				id_lapstok	id_shifter	id_kategori	jumlah_stok
<input type="checkbox"/>	Ubah	Salin	Hapus	1	sonia	telur akm	20
<input type="checkbox"/>	Ubah	Salin	Hapus	2	sonia	telur an	0
<input type="checkbox"/>	Ubah	Salin	Hapus	3	sonia	telur bebe	20
<input type="checkbox"/>	Ubah	Salin	Hapus	4	sonia	telur puyu	20
<input type="checkbox"/>	Ubah	Salin	Hapus	5	sonia	telur asin	0
<input type="checkbox"/>	Ubah	Salin	Hapus	6	sonia	telur reta	5
<input type="checkbox"/>	Ubah	Salin	Hapus	7	soja	minyak 1l	20
<input type="checkbox"/>	Ubah	Salin	Hapus	8	soja	minyak 2l	23
<input type="checkbox"/>	Ubah	Salin	Hapus	9	soja	tissue	5
<input type="checkbox"/>	Ubah	Salin	Hapus	10	soja	beras 5kg	4
<input type="checkbox"/>	Ubah	Salin	Hapus	11	soja	indomie	4
<input type="checkbox"/>	Ubah	Salin	Hapus	12	soja	mie sedap	16
<input type="checkbox"/>	Ubah	Salin	Hapus	13	soja	bumbu	7
<input type="checkbox"/>	Ubah	Salin	Hapus	14	kezia	royco	3
<input type="checkbox"/>	Ubah	Salin	Hapus	15	kezia	garam	11
<input type="checkbox"/>	Ubah	Salin	Hapus	16	kezia	cuka	8
<input type="checkbox"/>	Ubah	Salin	Hapus	17	kezia	kecap	17
<input type="checkbox"/>	Ubah	Salin	Hapus	18	kezia	kunyit	17
<input type="checkbox"/>	Ubah	Salin	Hapus	19	kezia	merica	9
<input type="checkbox"/>	Ubah	Salin	Hapus	20	kezia	saos	14

h. *Impact* Tabel arus_kas

Seperti yang sudah disinggung sebelumnya, tabel arus_kas bersifat otomatis terisi dengan data yang bersumber dari laporan_penjualan_harian serta transaksi_supplier. Data-data tersebut dapat dibedakan dengan melihat id_penjualan maupun id_pembelian, serta kategorinya. Apabila pada percobaan sebelumnya sudah memasukkan data *dummy* sebanyak 20 data pada masing-masing tabel, maka arus_kas mencatat kini mencatat sebanyak 40 data *dummy*. Berikut adalah isi tabel dari arus_kas.

				id	tanggal	kategori	jumlah	id_penjualan	id_pembelian	1	keterangan
<input type="checkbox"/>	Ubah	Salin	Hapus	41	2025-01-05	pemasukan	250000.00	1	NULL		Penjualan oleh sonia - Kategori: telur akm
<input type="checkbox"/>	Ubah	Salin	Hapus	42	2025-01-05	pemasukan	240000.00	2	NULL		Penjualan oleh sonia - Kategori: telur an
<input type="checkbox"/>	Ubah	Salin	Hapus	43	2025-01-05	pemasukan	120000.00	3	NULL		Penjualan oleh sonia - Kategori: telur bebe
<input type="checkbox"/>	Ubah	Salin	Hapus	44	2025-01-05	pemasukan	20000.00	4	NULL		Penjualan oleh sonia - Kategori: telur puyu
<input type="checkbox"/>	Ubah	Salin	Hapus	45	2025-01-05	pemasukan	150000.00	5	NULL		Penjualan oleh sonia - Kategori: telur asin
<input type="checkbox"/>	Ubah	Salin	Hapus	46	2025-01-05	pemasukan	67500.00	6	NULL		Penjualan oleh sonia - Kategori: telur reta
<input type="checkbox"/>	Ubah	Salin	Hapus	47	2025-01-05	pemasukan	150000.00	7	NULL		Penjualan oleh soja - Kategori: minyak 1l
<input type="checkbox"/>	Ubah	Salin	Hapus	48	2025-01-05	pemasukan	210000.00	8	NULL		Penjualan oleh soja - Kategori: minyak 2l
<input type="checkbox"/>	Ubah	Salin	Hapus	49	2025-01-05	pemasukan	100000.00	9	NULL		Penjualan oleh soja - Kategori: tissue
<input type="checkbox"/>	Ubah	Salin	Hapus	50	2025-01-05	pemasukan	360000.00	10	NULL		Penjualan oleh soja - Kategori: beras 5kg
<input type="checkbox"/>	Ubah	Salin	Hapus	51	2025-01-05	pemasukan	78000.00	11	NULL		Penjualan oleh soja - Kategori: indomie
<input type="checkbox"/>	Ubah	Salin	Hapus	52	2025-01-05	pemasukan	35000.00	12	NULL		Penjualan oleh soja - Kategori: mie sedap
<input type="checkbox"/>	Ubah	Salin	Hapus	53	2025-01-05	pemasukan	16000.00	13	NULL		Penjualan oleh soja - Kategori: bumbu
<input type="checkbox"/>	Ubah	Salin	Hapus	54	2025-01-05	pemasukan	27000.00	14	NULL		Penjualan oleh kezia - Kategori: royco
<input type="checkbox"/>	Ubah	Salin	Hapus	55	2025-01-05	pemasukan	20000.00	15	NULL		Penjualan oleh kezia - Kategori: garam
<input type="checkbox"/>	Ubah	Salin	Hapus	56	2025-01-05	pemasukan	16000.00	16	NULL		Penjualan oleh kezia - Kategori: cuka
<input type="checkbox"/>	Ubah	Salin	Hapus	57	2025-01-05	pemasukan	30000.00	17	NULL		Penjualan oleh kezia - Kategori: kecap
<input type="checkbox"/>	Ubah	Salin	Hapus	58	2025-01-05	pemasukan	13000.00	18	NULL		Penjualan oleh kezia - Kategori: kunyit
<input type="checkbox"/>	Ubah	Salin	Hapus	59	2025-01-05	pemasukan	21000.00	19	NULL		Penjualan oleh kezia - Kategori: merica
<input type="checkbox"/>	Ubah	Salin	Hapus	60	2025-01-05	pemasukan	72000.00	20	NULL		Penjualan oleh kezia - Kategori: saos
<input type="checkbox"/>	Ubah	Salin	Hapus	21	2025-01-05	pengeluaran	276000.00	NULL	1		Pembelian oleh sonia - Kategori: telur akm
<input type="checkbox"/>	Ubah	Salin	Hapus	22	2025-01-05	pengeluaran	160000.00	NULL	2		Pembelian oleh sonia - Kategori: telur an
<input type="checkbox"/>	Ubah	Salin	Hapus	23	2025-01-05	pengeluaran	170000.00	NULL	3		Pembelian oleh sonia - Kategori: telur bebe
<input type="checkbox"/>	Ubah	Salin	Hapus	24	2025-01-05	pengeluaran	50000.00	NULL	4		Pembelian oleh sonia - Kategori: telur puyu
<input type="checkbox"/>	Ubah	Salin	Hapus	25	2025-01-05	pengeluaran	140000.00	NULL	5		Pembelian oleh sonia - Kategori: telur asin

				id	tanggal	kategori	jumlah	id_penjualan	id_pembelian	1	keterangan
<input type="checkbox"/>	Ubah	Salin	Hapus	26	2025-01-05	pengeluaran	0.00	NULL	6		Pembelian oleh sonia - Kategori: telur reta
<input type="checkbox"/>	Ubah	Salin	Hapus	27	2025-01-05	pengeluaran	300000.00	NULL	7		Pembelian oleh soja - Kategori: minyak 1l
<input type="checkbox"/>	Ubah	Salin	Hapus	28	2025-01-05	pengeluaran	600000.00	NULL	8		Pembelian oleh soja - Kategori: minyak 2l
<input type="checkbox"/>	Ubah	Salin	Hapus	29	2025-01-05	pengeluaran	170000.00	NULL	9		Pembelian oleh soja - Kategori: tissue
<input type="checkbox"/>	Ubah	Salin	Hapus	30	2025-01-05	pengeluaran	500000.00	NULL	10		Pembelian oleh soja - Kategori: beras 5kg
<input type="checkbox"/>	Ubah	Salin	Hapus	31	2025-01-05	pengeluaran	75000.00	NULL	11		Pembelian oleh soja - Kategori: indomie
<input type="checkbox"/>	Ubah	Salin	Hapus	32	2025-01-05	pengeluaran	65000.00	NULL	12		Pembelian oleh soja - Kategori: mie sedap
<input type="checkbox"/>	Ubah	Salin	Hapus	33	2025-01-05	pengeluaran	30000.00	NULL	13		Pembelian oleh soja - Kategori: bumbu
<input type="checkbox"/>	Ubah	Salin	Hapus	34	2025-01-05	pengeluaran	20000.00	NULL	14		Pembelian oleh kezia - Kategori: royco
<input type="checkbox"/>	Ubah	Salin	Hapus	35	2025-01-05	pengeluaran	60000.00	NULL	15		Pembelian oleh kezia - Kategori: garam
<input type="checkbox"/>	Ubah	Salin	Hapus	36	2025-01-05	pengeluaran	60000.00	NULL	16		Pembelian oleh kezia - Kategori: cuka
<input type="checkbox"/>	Ubah	Salin	Hapus	37	2025-01-05	pengeluaran	180000.00	NULL	17		Pembelian oleh kezia - Kategori: kecap
<input type="checkbox"/>	Ubah	Salin	Hapus	38	2025-01-05	pengeluaran	15000.00	NULL	18		Pembelian oleh kezia - Kategori: kunyit
<input type="checkbox"/>	Ubah	Salin	Hapus	39	2025-01-05	pengeluaran	15000.00	NULL	19		Pembelian oleh kezia - Kategori: merica
<input type="checkbox"/>	Ubah	Salin	Hapus	40	2025-01-05	pengeluaran	200000.00	NULL	20		Pembelian oleh kezia - Kategori: saos

i. *Impact* Tabel informasi_keuangan

Tabel informasi_keuangan bagaikan merangkum isi dari tabel arus_kas. Tabel ini memiliki fungsi yang sama seperti query berikut :

```
SELECT * FROM informasi_keuangan;  
SELECT SUM(jumlah) as total_pemasukan FROM arus_kas WHERE kategori =  
'pemasukan';  
SELECT SUM(jumlah) as total_pengeluaran FROM arus_kas WHERE kategori =  
'pengeluaran';
```



	id_info	saldo	tanggal_update
<input type="checkbox"/> Ubah <input type="checkbox"/> Salin <input type="checkbox"/> Hapus	1	-1090500.00	2025-01-05

Jadi, sistematisasi tabel informasi_keuangan adalah melakukan *summary* masing-masing tabel arus_kas yang mempunyai kategori = 'pemasukan' dan kategori = 'pengeluaran'. kemudian kedua *summary* tersebut dioperasikan dalam operasi pengurangan, dan akhirnya menghasilkan saldo yang aktual. Pada tangkapan layar di atas, mungkin terlihat ambigu karena saldo bersifat minus. Tetapi ini tidak menjadi masalah sebagaimana penjualan belum menutupi total belanja dari *supplier*.

6. Pengimplementasian *Trigger*

Trigger menjadi konsep yang dapat menyokong untuk kemudahan modifikasi data *dummy* secara tepat waktu. Dengan penerapan *trigger* akan mempermudah siklus transaksi dalam waktu efisien.

Konsep *Trigger* diterapkan pada skenario-skenario berikut :

- laporan_stok_mingguan.** Kolom jumlah_stok Membutuhkan UPDATE otomatis dari tabel laporan_penjualan_harian dan transaksi_supplier. Jika terdapat data baru dari transaksi_supplier, jumlah_stok akan bertambah. Jika terdapat data baru dari laporan_penjualan_harian jumlah stok akan berkurang
- arus_kas.** Kolom jumlah membutuhkan UPDATE otomatis dari tabel laporan_penjualan_harian dan transaksi_supplier. Baris akan selalu bertambah jika terdapat pencatatan baru dari tabel relasinya, serta mengalami modifikasi apabila terdapat perubahan catatan.
- informasi_keuangan.** Kolom saldo membutuhkan UPDATE otomatis dari hasil *summary* antara tabel laporan_penjualan_harian dan transaksi_supplier
- laporan_penjualan_harian.** Terutama pada kolom jumlah_barang_terjual, didesain UPDATE sehingga setiap ada perubahan kuantitas barang terjual, total pendapatan

barang tersebut otomatis berubah, dan perubahan juga dijalankan di tabel lain, seperti di laporan_stok_mingguan dan arus_kas

- e. **transaksi_supplier**. Terutama pada kolom jumlah_barang_dibeli, didesain UPDATE sehingga setiap ada perubahan kuantitas barang dibeli, atau perubahan di total pembelian, perubahan tersebut dijalankan juga di tabel lain, seperti di laporan_stok_mingguan dan arus_kas

Adapun query *trigger* sebagai hasil implementasi skenario di atas adalah sebagai berikut :

Kueri SQL Anda berhasil dieksekusi.										
SHOW TRIGGERS;										
<input type="checkbox"/> Profil <input type="checkbox"/> Edit dikotak <input type="checkbox"/> Ubah <input type="checkbox"/> Buat kode PHP <input type="checkbox"/> Segarkan										
Extra options										
Trigger	Event	Table	Statement	Timing	Created	sql_mode	Definer	character_set_client	collation_connection	Database Collation
after_arus_kas_delete	DELETE	arus_kas	BEGIN UPDATE informasi_keuangan SET	AFTER	2025-01-05 03:14:33.18	NO_ZERO_IN_DATE,NO_ZERO_DATE,NO_ENGINE_SUBSTITUTIO...	root@localhost	utf8mb4	utf8mb4_unicode_ci	utf8mb4_general_ci
stok_sebelum_penjualan	INSERT	laporan_penjualan_harian	BEGIN DECLARE stok_tersedia INT; SE...	BEFORE	2025-01-02 14:33:35.99	NO_ZERO_IN_DATE,NO_ZERO_DATE,NO_ENGINE_SUBSTITUTIO...	root@localhost	utf8mb4	utf8mb4_unicode_ci	utf8mb4_general_ci
before_insert_penjualan	INSERT	laporan_penjualan_harian	BEGIN DECLARE harga DECIMAL(10,2); DECLA...	BEFORE	2025-01-02 17:04:26.96	NO_ZERO_IN_DATE,NO_ZERO_DATE,NO_ENGINE_SUBSTITUTIO...	root@localhost	utf8mb4	utf8mb4_unicode_ci	utf8mb4_general_ci
setelah_input_penjualan_ke_arus_kas	INSERT	laporan_penjualan_harian	BEGIN -- Insert ke arus kas INSERT INTO ...	AFTER	2025-01-05 02:33:10.76	NO_ZERO_IN_DATE,NO_ZERO_DATE,NO_ENGINE_SUBSTITUTIO...	root@localhost	utf8mb4	utf8mb4_unicode_ci	utf8mb4_general_ci
update_setelah_penjualan_harian	UPDATE	laporan_penjualan_harian	BEGIN -- 1. Perbarui stok di laporan_stok_min...	AFTER	2025-01-02 14:41:53.98	NO_ZERO_IN_DATE,NO_ZERO_DATE,NO_ENGINE_SUBSTITUTIO...	root@localhost	utf8mb4	utf8mb4_unicode_ci	utf8mb4_general_ci
setelah_hapus_penjualan	DELETE	laporan_penjualan_harian	BEGIN -- Kembalikan stok UPDATE laporan_...	AFTER	2025-01-05 03:14:33.21	NO_ZERO_IN_DATE,NO_ZERO_DATE,NO_ENGINE_SUBSTITUTIO...	root@localhost	utf8mb4	utf8mb4_unicode_ci	utf8mb4_general_ci
setelah_input_pembelian_ke_arus_kas	INSERT	transaksi_supplier	BEGIN -- Insert ke arus kas INSERT INTO ...	AFTER	2025-01-05 02:33:10.72	NO_ZERO_IN_DATE,NO_ZERO_DATE,NO_ENGINE_SUBSTITUTIO...	root@localhost	utf8mb4	utf8mb4_unicode_ci	utf8mb4_general_ci
update_setelah_transaksi_supplier	UPDATE	transaksi_supplier	BEGIN -- Update stok UPDATE laporan_stok...	AFTER	2025-01-05 02:33:10.79	NO_ZERO_IN_DATE,NO_ZERO_DATE,NO_ENGINE_SUBSTITUTIO...	root@localhost	utf8mb4	utf8mb4_unicode_ci	utf8mb4_general_ci
setelah_hapus_pembelian	DELETE	transaksi_supplier	BEGIN -- Kurangi stok UPDATE laporan_sto...	AFTER	2025-01-05 03:08:52.13	NO_ZERO_IN_DATE,NO_ZERO_DATE,NO_ENGINE_SUBSTITUTIO...	root@localhost	utf8mb4	utf8mb4_unicode_ci	utf8mb4_general_ci

a. *Trigger* arus_kas

Trigger yang dibuat untuk tabel arus_kas adalah “after_arus_kas_delete”, di mana *trigger* ini bekerja supaya ketika tabel arus_kas melakukan DELETE di salah satu data *dummy*, pengurangan data tersebut mempengaruhi pula isi tabel informasi_keuangan. Jika semisal tabel arus_kas menghapus salah satu data, maka tabel informasi_keuangan ikut mengalami pengurangan saldo. Query-nya dapat dilihat di bawah ini :

DELIMITER \$\$

CREATE TRIGGER after_arus_kas_delete

AFTER DELETE ON arus_kas

FOR EACH ROW

BEGIN

UPDATE informasi_keuangan

SET

saldo = (

SELECT

COALESCE(

(SELECT SUM(jumlah) FROM arus_kas WHERE kategori = 'pemasukan'),


```

        0
    ) -
    COALESCE(
        (SELECT SUM(jumlah) FROM arus_kas WHERE kategori = 'pengeluaran'),
        0
    )
),
tanggal_update = CURRENT_DATE
WHERE id_info = 1;
END$$

```

Trigger arus_kas menggunakan fungsi COALESCE untuk menangani kasus ketika tidak ada data di tabel arus_kas. Jadi, andaikan ketika kategori = ‘pemasukan’ atau ‘pengeluaran’ benar-benar tidak ada data yang masuk sama sekali, maka akan mengembalikan nilai NULL.

b. *Trigger* laporan_penjualan_harian

Trigger-trigger di bawah ini berfungsi untuk *update* laporan_penjualan_harian setelah melakukan penjualan. *Trigger* dibuat dengan maksud memengaruhi tabel transaksi_supplier, laporan_stok_mingguan dan informasi_keuangan dalam pencatatannya. Query-nya dapat dilihat di bawah ini :

```

DELIMITER $$

```

```

CREATE TRIGGER stok_sebelum_penjualan
BEFORE INSERT ON laporan_penjualan_harian
FOR EACH ROW
BEGIN
    DECLARE stok_tersedia INT;

    SELECT jumlah_stok INTO stok_tersedia
    FROM laporan_stok_mingguan
    WHERE id_kategori = NEW.id_kategori;

    IF stok_tersedia < NEW.jumlah_barang_terjual THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Stok tidak cukup!';
    END IF;
END$$

```

Trigger di atas berfungsi untuk konfirmasi stok. Jadi ketika jumlah permintaan barang lebih besar dari jumlah stok, maka SQL akan mengirimkan pesan “Stok tidak cukup!”.

```

DELIMITER $$

```



```

CREATE TRIGGER `before_insert_penjualan`
BEFORE INSERT ON `laporan_penjualan_harian`
FOR EACH ROW BEGIN
    DECLARE harga DECIMAL(10,2);
    DECLARE total_stok INT;

    SELECT harga_satuan INTO harga
    FROM kategori
    WHERE id_kategori = NEW.id_kategori;

    SELECT COALESCE(SUM(jumlah_stok), 0) INTO total_stok
    FROM laporan_stok_mingguan
    WHERE id_kategori = NEW.id_kategori;

    IF total_stok < NEW.jumlah_barang_terjual THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Stok tidak cukup!';
    END IF;

    SET NEW.total_penjualan = NEW.jumlah_barang_terjual * harga;
END $$

DELIMITER ;

```

Trigger di atas berfungsi untuk persiapan sebelum penjualan dilakukan. Tujuan dari *trigger* tersebut adalah untuk menghitung otomatis jumlah barang terjual dengan harga satuan per kategorinya. Selain itu, *trigger* ini juga mengecek terlebih dahulu apakah jumlah barang cukup untuk memenuhi permintaan atau tidak.

```

DELIMITER $$
CREATE TRIGGER setelah_input_penjualan_ke_arus_kas
AFTER INSERT ON laporan_penjualan_harian
FOR EACH ROW
BEGIN

    INSERT INTO arus_kas (tanggal, kategori, jumlah, id_penjualan, keterangan)
    VALUES (
        NEW.tanggal_penjualan,
        'pemasukan',
        NEW.total_penjualan,
        NEW.id_penjualan,
        CONCAT('Penjualan oleh ', NEW.id_shifter, ' - Kategori: ', NEW.id_kategori)
    );

```

```

UPDATE informasi_keuangan
SET
saldo = (
SELECT (
COALESCE((SELECT SUM(jumlah) FROM arus_kas WHERE kategori =
'pemasukan'), 0) -
COALESCE((SELECT SUM(jumlah) FROM arus_kas WHERE kategori =
'pengeluaran'), 0));
tanggal_update = CURRENT_DATE
WHERE id_info = 1;
END$$

```

```

saldo + NEW.total_penjualan,
tanggal_update = CURRENT_DATE
WHERE id_info = 1;

```

```

UPDATE laporan_stok_mingguan
SET jumlah_stok = jumlah_stok - NEW.jumlah_barang_terjual
WHERE id_kategori = NEW.id_kategori;
END$$

```

DELIMITER ;

Trigger di atas adalah untuk melakukan UPDATE ke tabel arus_kas yang di mana langsung memasukkan data baru ke tabelnya, informasi_keuangan yang di mana langsung melakukan pembaruan akumulasi saldo, dan laporan_stok_mingguan yang di mana langsung mengonfirmasi jumlah stok tersisa setelah penjualan.

DELIMITER \$\$

```

CREATE TRIGGER setelah_hapus_penjualan
AFTER DELETE ON laporan_penjualan_harian
FOR EACH ROW
BEGIN

```

```

UPDATE laporan_stok_mingguan
SET jumlah_stok = jumlah_stok + OLD.jumlah_barang_terjual
WHERE id_kategori = OLD.id_kategori;

```

```

UPDATE informasi_keuangan
SET
saldo = (
SELECT (

```

```

        COALESCE((SELECT SUM(jumlah) FROM arus_kas WHERE kategori =
'pemasukan'), 0) -
        COALESCE((SELECT SUM(jumlah) FROM arus_kas WHERE kategori =
'pengeluaran'), 0))),
    tanggal_update = CURRENT_DATE
    WHERE id_info = 1;
END$$

```

DELIMITER ;

Trigger di atas berfungsi ketika penjualan dibatalkan. Di mana jumlah_stok dari laporan_stok_mingguan mengalami UPDATE kembali ke jumlah sediakala. Pada bagian informasi_keuangan cukup terjadi pengulangan *trigger* saja untuk menyesuaikan data di arus_kas.

DELIMITER \$\$

```

CREATE TRIGGER setelah_input_pembelian_ke_arus_kas
AFTER INSERT ON transaksi_supplier
FOR EACH ROW
BEGIN

```

```

    INSERT INTO arus_kas (tanggal, kategori, jumlah, keterangan)
    VALUES (
        NEW.tanggal_pembelian,
        'pengeluaran',
        NEW.total_transaksi,
        CONCAT('Pembelian oleh ', NEW.id_shifter, ' - Kategori: ', NEW.id_kategori)
    );

```

```

    UPDATE informasi_keuangan
    SET
    saldo = (
    SELECT (
        COALESCE((SELECT SUM(jumlah) FROM arus_kas WHERE kategori =
'pemasukan'), 0) -
        COALESCE((SELECT SUM(jumlah) FROM arus_kas WHERE kategori =
'pengeluaran'), 0))),
    tanggal_update = CURRENT_DATE
    WHERE id_info = 1;

```

```

IF EXISTS (
SELECT 1 FROM laporan_stok_mingguan
WHERE id_kategori = NEW.id_kategori )

```

```

THEN
    UPDATE laporan_stok_mingguan
    SET jumlah_stok = jumlah_stok + NEW.jumlah_barang_dibeli
    WHERE id_kategori = NEW.id_kategori;
ELSE
    INSERT INTO laporan_stok_mingguan (id_shifter, id_kategori, jumlah_stok)
    VALUES (NEW.id_shifter, NEW.id_kategori, NEW.jumlah_barang_dibeli);
END IF;
END$$

```

DELIMITER ;

Trigger di atas berfungsi setelah melakukan pembelian dengan *supplier*. Tabel yang berpengaruh adalah tabel *arus_kas*, *informasi_keuangan* dan *laporan_stok_mingguan*. Pada *laporan arus_kas* terjadi penambahan data baru ke dalam tabelnya, pada bagian *informasi_keuangan* terjadi pengulangan *trigger* lagi menyesuaikan dengan data di *arus_kas* terbaru, dan pada tabel *laporan_stok_mingguan* terjadi penambahan *jumlah_stok*.

c. *Trigger transaksi_supplier*

Trigger-trigger di bawah ini berfungsi untuk melakukan UPDATE setelah melakukan pembelian. *Trigger* dibuat dengan maksud memengaruhi tabel *laporan_stok_mingguan*, *arus_kas* dan *informasi_keuangan* dalam pencatatannya. Query-nya dapat dilihat di bawah ini:

```

CREATE TRIGGER update_setelah_transaksi_supplier
AFTER UPDATE ON transaksi_supplier
FOR EACH ROW
BEGIN

    UPDATE laporan_stok_mingguan
    SET jumlah_stok = jumlah_stok - OLD.jumlah_barang_dibeli + NEW.jumlah_barang_dibeli
    WHERE id_kategori = NEW.id_kategori;

    UPDATE arus_kas
    SET
        jumlah = NEW.total_transaksi,
        tanggal = NEW.tanggal_pembelian,
        keterangan = CONCAT('Update pembelian oleh ', NEW.id_shifter, ' - Kategori: ',
        NEW.id_kategori)
    WHERE id_pembelian = NEW.id_transaksi;

    UPDATE informasi_keuangan
    SET

```

```

        saldo = (
        SELECT (
            COALESCE((SELECT SUM(jumlah) FROM arus_kas WHERE kategori =
'pemasukan'), 0) -
            COALESCE((SELECT SUM(jumlah) FROM arus_kas WHERE kategori =
'pengeluaran'), 0)),
        tanggal_update = CURRENT_DATE
        WHERE id_info = 1;
END$$

```

DELIMITER ;

Trigger di atas berfungsi untuk melakukan UPDATE setelah melakukan pembelian. *Trigger* dibuat dengan maksud memengaruhi tabel laporan_stok_mingguan dalam pembaruan jumlah_stok, arus_kas dalam pembaruan catatan datanya, dan informasi_keuangan dalam pembaruan saldonya.

```

CREATE TRIGGER setelah_hapus_pembelian
AFTER DELETE ON transaksi_supplier
FOR EACH ROW
BEGIN

```

```

    UPDATE laporan_stok_mingguan
    SET jumlah_stok = jumlah_stok - OLD.jumlah_barang_dibeli
    WHERE id_kategori = OLD.id_kategori;

```

```

    UPDATE informasi_keuangan
    SET

```

```

        saldo = (
        SELECT
            COALESCE((SELECT SUM(jumlah) FROM arus_kas WHERE kategori =
'pemasukan'), 0) -
            COALESCE((SELECT SUM(jumlah) FROM arus_kas WHERE kategori =
'pengeluaran'), 0)),
        tanggal_update = CURRENT_DATE
        WHERE id_info = 1;
END$$

```

DELIMITER ;

Trigger di atas adalah sebagai pelengkap, karena pada *trigger* sebelumnya kurang tanggap ketika salah satu data pembelian dihapus atau dibatalkan. Pada *trigger* tersebut, akan terjadi pembaruan di mana jumlah_stok dari laporan_stok_mingguan mengalami UPDATE kembali

ke jumlah sediakala. Lalu pada bagian informasi_keuangan cukup terjadi pengulangan *trigger* saja untuk menyesuaikan data di arus_kas.

7. Penutup

Demikian laporan tugas besar ini kami susun. Kami menyadari bahwa masih terdapat banyak kekurangan selama pembuatan *database* agen_telur ini, terutama pada implementasi *trigger* yang terlihat banyak mengalami duplikasi proses dan perlu perbaikan lebih lanjut. Kami mengucapkan maaf atas segala kekurangan yang terdapat di dalam laporan ini. Kami berharap masukan dan saran dari pembaca dapat membantu kami meningkatkan pemahaman serta kemampuan proyek kami. Sekian laporan ini kami sampaikan. Terima kasih atas kesempatannya.

8. Dokumentasi

