Implementasi Basis Data

Triggers

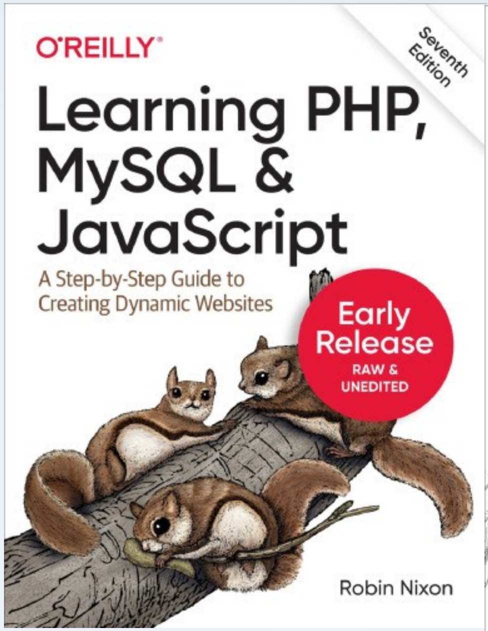
Prof. Dr. Ridwan Sanjaya

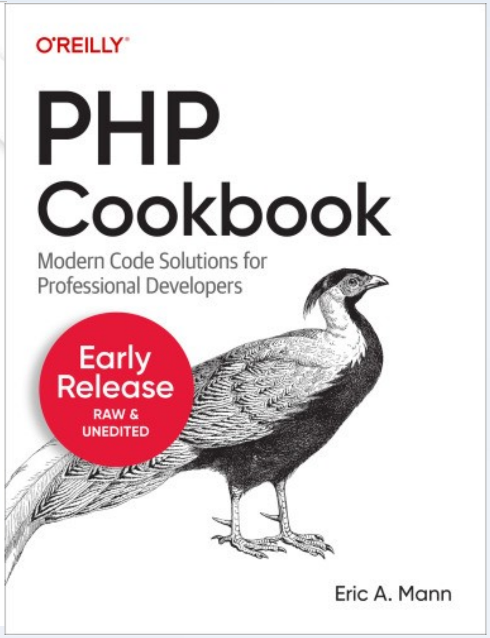
Unika
SOEGIJAPRANATA

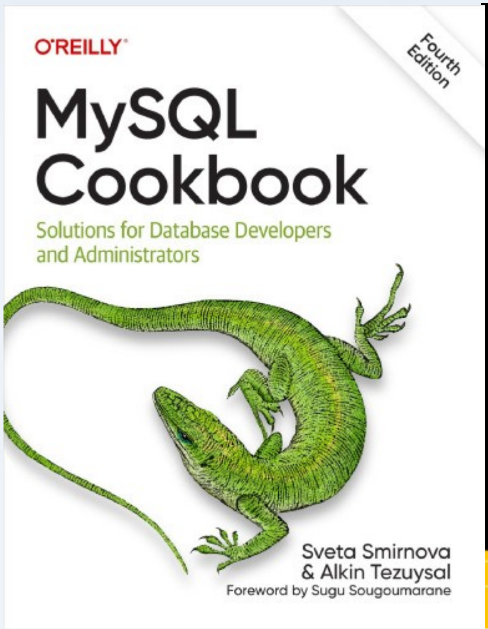
Scoring

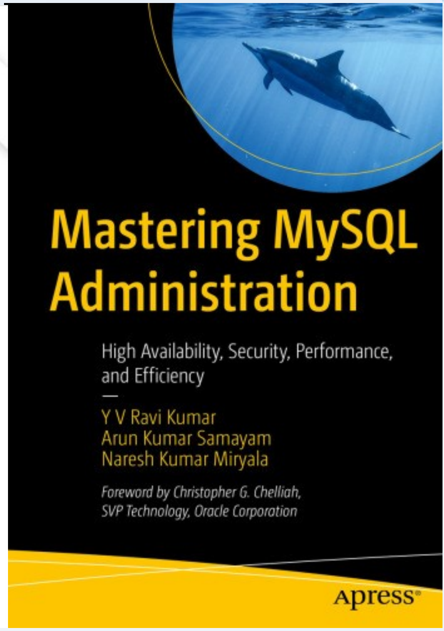
- Assignments and Presentations 40%
  - Mid Term 20%
  - After Mid Term 20%
- Mid-Term Exam 30%
- Final Exam 30%

Buku Referensi


O'REILLY
Seventh Edition
Learning PHP, MySQL & JavaScript
A Step-by-Step Guide to Creating Dynamic Websites
Early Release
RAW & UNEDITED
Robin Nixon




O'REILLY
Fourth Edition
MySQL Cookbook
Solutions for Database Developers and Administrators
Sveta Smirnova
& Alkin Tezuysal
Foreword by Sugu Sougoumarane


Mastering MySQL Administration
High Availability, Security, Performance, and Efficiency
Y V Ravi Kumar
Arun Kumar Samayam
Naresh Kumar Miryala
Foreword by Christopher G. Chelliah, SVP Technology, Oracle Corporation
Apress

- Learning PHP, MySQL & JavaScript: A Step-by-Step Guide to Creating Dynamic Websites, 7th Edition, O'Reilly Media, 2025
- PHP Cookbook, O'Reilly Media, 2023
- MySQL Cookbook, 4th Edition, O’Reilly Media, 2022
- Mastering MySQL Administration, Apress, 2024

Unika
SOEGIJAPRANATA

Using Triggers

Database triggers are stored programs that are executed in response to some kind of event that occurs within the database. In the current MySQL implementation of triggers, triggers fire in response to a DML statement (INSERT, UPDATE, DELETE) on a specified table.

Triggers are a powerful mechanism for ensuring the integrity of your data, as well as a useful means of automating certain operations in the database, such as denormalization and audit logging.

Syntax

```
CREATE TRIGGER trigger_name
{BEFORE|AFTER}
{UPDATE|INSERT|DELETE}
ON table_name
FOR EACH ROW
trigger_statements
```

Example Syntax

```
CREATE TRIGGER contoh1
  BEFORE INSERT ON sales_detail
  FOR EACH ROW


BEGIN
  ….
  ….
END$$
```

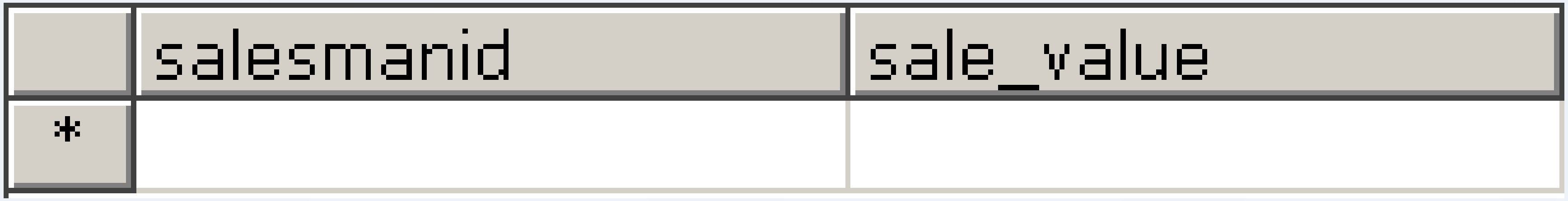
Simple Example

```
CREATE TRIGGER simple
  AFTER INSERT ON transaksi
  FOR EACH ROW

BEGIN
 INSERT INTO pembeli (KodePembeli, NamaPembeli)
      VALUES (NEW.KodePembeli, NEW.NamaPembeli);
END$$
```

Example

```
CREATE TRIGGER contoh1
  BEFORE INSERT ON sales_detail
  FOR EACH ROW


BEGIN
 DECLARE row_count INTEGER;


  SELECT COUNT(*)
    INTO row_count
    FROM sales_total
   WHERE
    salesmanid=NEW.salesmanid;
  IF row_count > 0 THEN
    UPDATE sales_total
       SET sale_value=
         sale_value+NEW.sale_value
     WHERE salesmanid=
         NEW.salesmanid;
  ELSE
    INSERT INTO sales_total
       (salesmanid,sale_value)
      VALUES (NEW.salesmanid,
       NEW.sale_value);
  END IF;
END$$
```

sales_detail & sales_total



	salesmanid	sale_value
*		

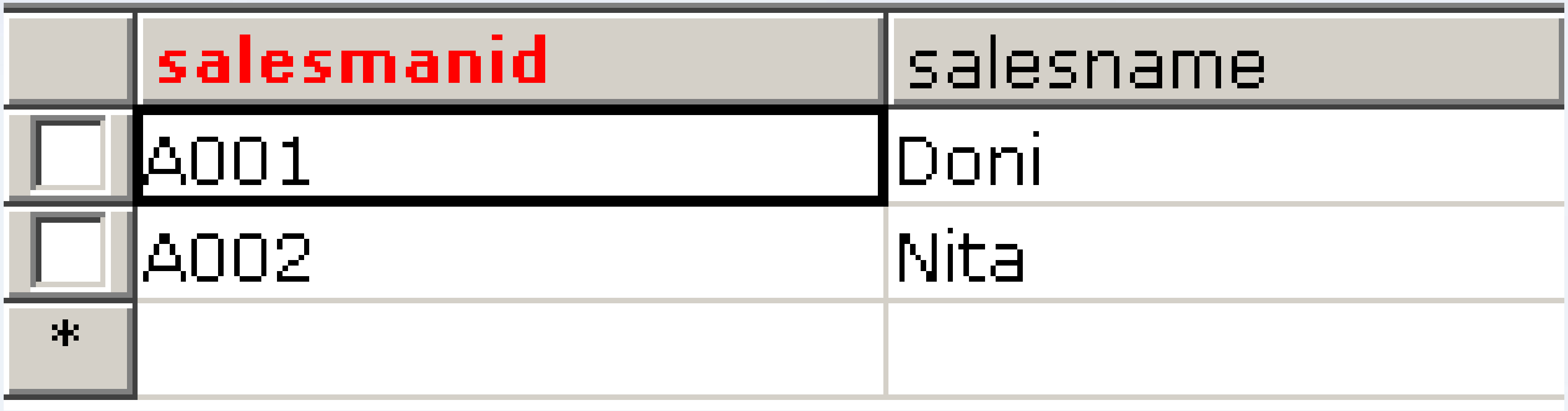
Result

```
mysql> insert into sales_detail values('A001',50000);
Query OK, 1 row affected (0.03 sec)

mysql> insert into sales_detail values('A002',75000);
Query OK, 1 row affected (0.03 sec)

mysql> insert into sales_detail values('A001',60000);
Query OK, 1 row affected (0.03 sec)

mysql> select * from sales_detail;
+------------+------------+
| salesmanid | sale_value |
+------------+------------+
| A001       |      50000 |
| A002       |      75000 |
| A001       |      60000 |
+------------+------------+
3 rows in set (0.00 sec)
```

Result

```
mysql> select * from sales_total;
+------------+------------+
| salesmanid | sale_value |
+------------+------------+
| A001       |     110000 |
| A002       |      75000 |
+------------+------------+
2 rows in set (0.00 sec)
```

sales_person



	salesmanid	salesname
□	A001	Doni
□	A002	Nita
*		

Example

```
CREATE TRIGGER contoh2
  BEFORE INSERT ON sales_person
  FOR EACH ROW
BEGIN
  DECLARE row_count INTEGER;


  SELECT COUNT(*)
    INTO row_count
    FROM sales_person
   WHERE
    salesmanid=NEW.salesmanid;
 IF row_count > 0 THEN
   SELECT "Already exist[illegible]s
Warning;
  ELSE
    INSERT INTO sales_person
       (salesmanid,salesname)
      VALUES (NEW.salesmanid,
       NEW.salesname);
  END IF;
END$$
```

TIDAK BOLEH “SELECT”

Result

```
mysql> source triggers2.sql;
ERROR 1415 (0A000): Not allowed to return
a result set from a trigger
```

Example

```
CREATE TRIGGER contoh2
  BEFORE INSERT ON sales_person
  FOR EACH ROW
BEGIN
  DECLARE row_count INTEGER;


  SELECT COUNT(*)
    INTO row_count
    FROM sales_person
   WHERE
    salesmanid=NEW.salesmanid;
```

```
 IF row_count = 0 THEN
   INSERT INTO sales_person
      (salesmanid,salesname)
     VALUES (NEW.salesmanid,
      NEW.salesname);
  END IF;
END$$
```

TIDAK BOLEH TRIGGER DI TABLE SENDIRI

Result

```
mysql> insert into sales_person values('A001','Karjo');

ERROR 1442 (HY000): Can't update table 'sales_person'
in stored function/trigger because it is already used by
statement which invoked this stored function/trigger.
```

Try out

- Create a table of stock and sales_activity;
- If the sales_activity is increased, the stock must be decreased.

stock :
prod_code, total

sales_activity:
trans_date, prod_code, price, total

Try out (2)

- If the sales_activity is increased, the sales_record must be updated.
- total_rp = price x volume

sales_record :
salesman_id, total_rp

sales_activity:
trans_date, salesman_id, prod_code, price, volume

Contoh Kasus YPD

- Setelah update diterima, maka data pelamar masuk ke table pegawai
  - Tabel pelamar punya kolom id_pelamar, nama, alamat, status.
  - Tabel pegawai punya kolom id_pegawai, nama, alamat, tgl_masuk.
  - Kalau status pelamar berubah menjadi bernilai 1, maka insert otomatis ke tabel pegawai

Contoh Kasus YPD

```
DELIMITER $$

CREATE TRIGGER pelamar_diterima
AFTER UPDATE ON pelamar
FOR EACH ROW
BEGIN
   IF NEW.status = 1 AND OLD.status <> 1 THEN
      INSERT INTO pegawai (nama, alamat, tgl_masuk)
      VALUES (NEW.nama, NEW.alamat, NOW());
   END IF;
END$$

DELIMITER ;
```

Contoh Kasus Koperasi

- Setelah anguran sesuai jumlah pinjaman, maka status pinjaman berubah 1
  - Tabel pinjaman: berisi id_pinjaman, id_anggota, jumlah, status.
  - Tabel pembayaran: berisi id_bayar, id_pinjaman, jumlah, bukti_transfer, tgl_bayar.
  - Tabel pinjaman harus otomatis update status = 1 kalau total pembayaran sama dengan jumlah pinjaman

Contoh Kasus Koperasi

```
DELIMITER $$

CREATE TRIGGER pinjaman_lunas
AFTER INSERT ON pembayaran
FOR EACH ROW
BEGIN
    DECLARE total_bayar DECIMAL(12,2);

    SELECT SUM(jumlah) INTO total_bayar FROM pembayaran
    WHERE id_pinjaman = NEW.id_pinjaman;

    UPDATE pinjaman SET status = 1
    WHERE id_pinjaman = NEW.id_pinjaman AND total_bayar >= jumlah;
END$$

DELIMITER ;
```

Contoh Kasus Bruderan

- Setelah anguran sesuai jumlah pinjaman, maka status pinjaman berubah 1
  - Tabel rencana_anggaran: id_rencana, id_bruder, bulan, nominal.
  - Tabel realisasi_anggaran: id_realisasi, id_rencana, nominal, bukti, tgl_realisasi.
  - Tabel rencana_anggaran otomatis update kolom status 1 kalau realisasi sudah sama atau staus 2 kalau melebihi rencana

Contoh Kasus Bruderan

```
DELIMITER $$

CREATE TRIGGER cek_realisasi_anggaran
AFTER INSERT ON realisasi_anggaran
FOR EACH ROW
BEGIN
    DECLARE total_realisasi DECIMAL(12,2);

    SELECT SUM(nominal) INTO total_realisasi FROM realisasi_anggaran
    WHERE id_rencana = NEW.id_rencana;

    UPDATE rencana_anggaran SET status = 1
    WHERE id_rencana = NEW.id_rencana AND total_realisasi >= nominal;
END$$

DELIMITER ;
```

Contoh Kasus Dokter Gigi

- Setelah periksa, jika ada follow up periksa kembali, maka follow up dinilai 1 jika pasien mendaftar lagi sesuai tanggal atau maksimal 1 bulan setelahnya
  - Tabel pendaftaran: id_daftar, id_pasien, tgl_daftar, status.
  - Tabel antrian: id_antrian, id_daftar, no_urut, status.
  - Tabel catatan_medis: id_catatan, id_daftar, id_pasien, diagnosis, tindakan, resep, follow_up_date, follow_up_status.

Contoh Kasus Dokter Gigi

```
DELIMITER $$

CREATE TRIGGER cek_followup
AFTER INSERT ON pendaftaran
FOR EACH ROW
BEGIN
    UPDATE catatan_medis cm
    SET cm.follow_up_status = 1 WHERE cm.id_pasien = NEW.id_pasien
      AND cm.follow_up_status = 0 AND NEW.tgl_daftar BETWEEN cm.follow_up_date
      AND DATE_ADD(cm.follow_up_date, INTERVAL 1 MONTH);

    UPDATE catatan_medis cm
    SET cm.follow_up_status = 2 WHERE cm.id_pasien = NEW.id_pasien
      AND cm.follow_up_status = 0
      AND NEW.tgl_daftar > DATE_ADD(cm.follow_up_date, INTERVAL 1 MONTH);
END$$
DELIMITER ;
```

Thank You

ridwan@unika.ac.id