



BUKU KERJA PRAKTIK MAHASISWA (BKPM)

**WORKSHOP BASIS DATA – MINGGU 14
TIF110708
SEMESTER 1**

TIM PENGAMPU :

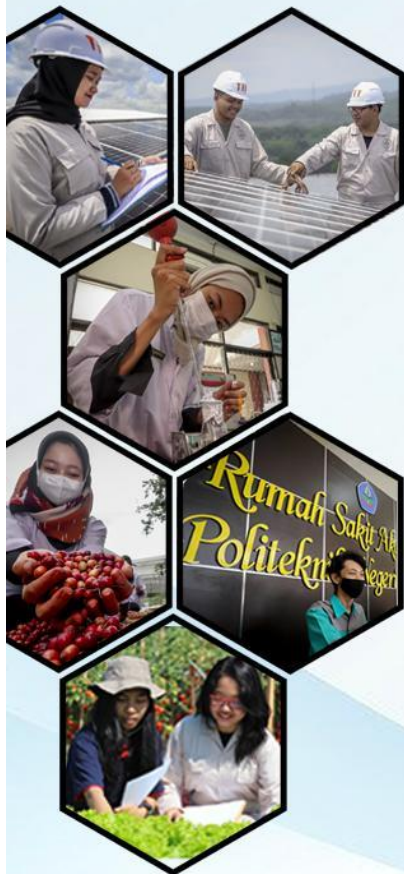
BETY ETIKASARI, S.Pd., M.Pd

LUKIE PERDANASARI, S.Kom., M.T

PRAWIDYA DESTARIANTO, S.Kom., M.T

LUKMAN HAKIM, S.Kom., M.Kom

**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI JEMBER
TAHUN 2022**



ACARA 49 dan 50

Materi Pembelajaran : Trigger
Acara Praktikum/Pertemuan : Minggu 14 / 1 dan 2
Tempat : Laboratorium Jurusan Teknologi Informasi, Politeknik
Negeri Jember
Alokasi Waktu : 100 menit

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu memahami penggunaan SQL *Trigger*.
2. Mahasiswa mampu mengimplementasikan penggunaan SQL *Trigger* pada database.

b. Penilaian Bertahap BNSP (Skill Passport)

Nama Skema Sertifikasi :

No	Kode Unit	Nama Unit Kompetensi	Elemen Kompetensi
1	J.620100.020.02	Menggunakan SQL	• Membuat trigger

c. Indikator

1. Mahasiswa memahami penggunaan SQL *Trigger*.
2. Mahasiswa mengimplementasikan penggunaan SQL *Trigger* pada database.

d. Dasar Teori

Trigger didefinisikan sebagai himpunan kode (procedural) yang dieksekusi secara otomatis sebagai respon atas suatu kejadian berkaitan dengan tabel basis data. Kejadian (event) yang dapat membangkitkan trigger umumnya berupa pernyataan INSERT, UPDATE, dan DELETE.

Berdasarkan ruang lingkupnya, trigger diklasifikasikan menjadi dua jenis: row trigger dan statement trigger. Trigger baris (row) mendefinisikan aksi untuk setiap baris tabel; trigger pernyataan hanya berlaku untuk setiap pernyataan INSERT, UPDATE, atau DELETE. Dari sisi perilaku (behavior) eksekusi, trigger dapat dibedakan menjadi beberapa jenis; namun umumnya ada dua jenis: trigger BEFORE dan AFTER.

Sesuai penamaannya, jenis-jenis ini merepresentasikan waktu eksekusi trigger, misalnya sebelum ataukah sesudah pernyataan-pernyataan yang

berkorespondensi. Trigger juga dipandang sebagai bentuk spesifik dari stored procedure (terkait pendefinisian body). Bagaimanapun, trigger akan dipanggil (secara otomatis) ketika event terjadi, sedangkan stored procedure harus dipanggil secara eksplisit.

Cara penulisan trigger:

```
DELIMITER $$
CREATE TRIGGER nama_trigger
{BEFORE | AFTER} {INSERT | UPDATE | DELETE}
ON nama_table
FOR EACH ROW
BEGIN
    KODE SQL
END$$
DELIMITER ;
```

- Untuk memulai menggunakan TRIGGER, gunakan **CREATE TRIGGER** dilanjutkan **nama TRIGGER** yang ingin dibuat.
- **{BEFORE | AFTER}** adalah waktu TRIGGER akan dijalankan, apakah sebelum atau sesudah database dimodifikasi oleh perintah DML.
- **{INSERT | UPDATE | DELETE}** adalah perintah DML yang mengaktifkan TRIGGER

Lebih detail untuk waktu TRIGGER dijelaskan sebagai berikut:

Waktu TRIGGER	Keterangan TRIGGER
BEFORE INSERT	TRIGGER dijalankan sebelum record dimasukkan ke database
AFTER INSERT	TRIGGER dijalankan sesudah record dimasukkan ke database
BEFORE UPDATE	TRIGGER dijalankan sebelum record dirubah di database
AFTER UPDATE	TRIGGER dijalankan sesudah record dirubah database
BEFORE DELETE	TRIGGER dijalankan sebelum record dihapus di database

AFTER DELETE

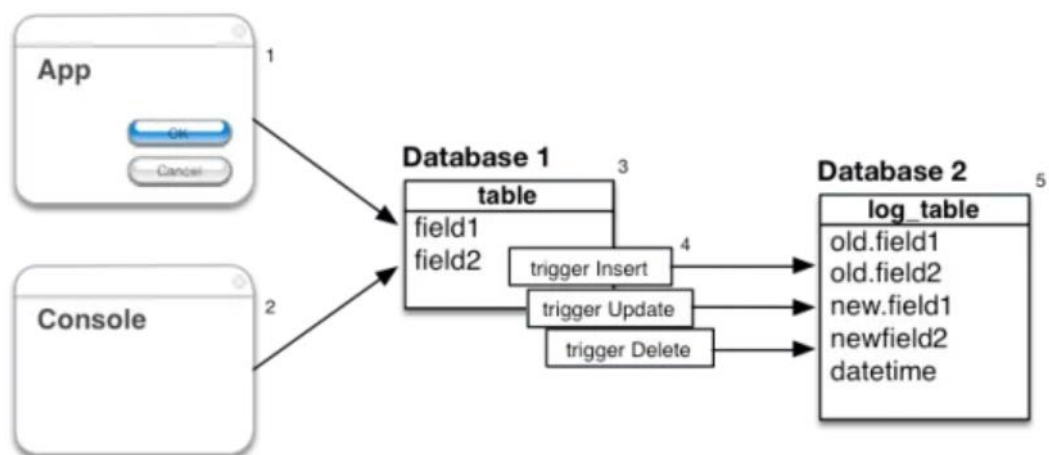
TRIGGER dijalankan sesudah record dihapus di database

- **ON** mendefinisikan table yang mengaktifkan TRIGGER
- **BEGIN END** adalah pernyataan yang membungkus kode TRIGGER
- Pastikan diawal gunakan **DELIMITER \$\$** dan diakhir dikembalikan ke **DELIMITER ;** seperti dalam membuat Stored Procedure

Referensi "OLD" dan "NEW"

Karena trigger digunakan pada saat terjadi perubahan row data, maka kita perlu referensi ke row sebelum dan sesudah perubahan. Untuk ini ada dua alias yang berfungsi untuk hal tersebut yaitu OLD dan NEW. Sesuai namanya, OLD digunakan untuk referensi sebelum perubahan dan NEW untuk referensi setelah perubahan.

Fungsi Trigger



Biasanya dalam relasional database, trigger dapat ditemui ketika melakukan perintah eksekusi table. Hal itu menjadi lebih bermanfaat trigger menjadikan penulisan pemrograman yang sederhana dan dapat menjaga informasi agar tetap konsisten dalam database. Selain itu, terdapat beberapa fungsi trigger yang akan dijelaskan dibawah ini.

1. Integritas Data

Dengan adanya trigger, dapat mempertahankan integritas table yang terdapat dalam database. Sebab, dapat melakukan operasi-operasi yang berkaitan dengan pengolahan database, seperti INSERT, UPDATE dan DELETE

2. Mencegah Error

Trigger dalam MySQL dapat mencegah terjadinya error dalam pengoperasian data. Jika terjadi error dalam pendefinisian trigger, error tersebut tidak mengganggu trigger yang sedang berjalan.

3. Membuat Tugas Kerja menjadi Terjadwal

Ketika trigger telah dijalankan, maka dapat menggunakan berbagai bahasa pemrograman tanpa harus bingung bagaimana cara mengaksesnya.

4. Mencegah proses transaksi yang tidak sah

Dalam praktiknya, biasa trigger digunakan untuk melakukan proses transaksi. Dapat menyimpan record transaksi tersebut ke table lain (history) tanpa harus takut jika data tersebut di-update atau delete. Semua perubahan yang terjadi juga dapat diacak berdasarkan waktu pembuatannya.

e. Alat dan Bahan

1. Kertas folio
2. Spidol dan Bolpoin
3. Penggaris
4. Client server: Xampp
5. Text editor: Mysql Workbench atau sejenisnya

f. Prosedur Kerja

1. Perhatikan contoh berikut (**TRIGGER BEFORE UPDATE**)

Berdasarkan database dbmahasiswa, buatlah satu table dengan nama "log_mahasiswa". Tabel ini akan digunakan untuk menyimpan perubahan data yang dilakukan pada table mahasiswa. Sehingga dengan adanya log perubahan data mahasiswa ini akan memudahkan dalam melihat histori data mahasiswa yang pernah berubah dalam sistem.

Tabel log_mahasiswa:

```
id_log INT (10) AUTO_INCREMENT, PRIMARY KEY
nim CHAR (9)
alamat_lama VARCHAR (35)
alamat_baru VARCHAR (35)
waktu DATE
```


Membuat trigger untuk menyimpan data perubahan “alamat” sebelum perintah UPADTE dijalankan

```
DELIMITER $$
CREATE TRIGGER update_alamat_mahasiswa
    BEFORE UPDATE
    ON tbl_mhs
    FOR EACH ROW
BEGIN
    INSERT INTO tbl_log_mhs
    SET nim = OLD.nim,
        alamat_lama = OLD.alamat,
        alamat_baru = NEW.alamat,
        waktu = NOW();
END $$
DELIMITER;
```

*Jika eror, maka coba tanpa menggunakan delimiter; melainkan langsung END; diakhir

Awal tampilan table mahasiswa

nim	nama_mhs	alamat	phone	nip	kode_jur
E2021003	Sita	Temanggung	098745633	NULL	NULL
E20211001	Abizar	Jember	0821236789	199305102021032001	00JTI
E20211002	Bili	Jember	0821236789	199305102021032001	00JTI
E20211003	Ina	Mojokerto	0987654322	NULL	NULL
E20212001	Arka	Bondowoso	08212236789	199305102021032001	0JTEK
E20212002	Caca	Nganjuk	08212236789	199308312021032001	0JTEK
E20212003	Rama	Pasuruan	0987643323	NULL	NULL

Awal tampilan table log_mhs

id_log	nim	alamat_lama	alamat_baru	waktu
--------	-----	-------------	-------------	-------

Ujicoba dengan sintaks berikut dan jalankan:

```
1 UPDATE tbl_mhs
2 SET alamat = "Jember"
3 WHERE nim = "E2021003";|
```

BKPM – WORKSHOP BASIS DATA

Tampilan table mahasiswa setelah dijalankan:

```
SELECT * FROM tbl_mhs;
```

nim	nama_mhs	alamat	phone	nip	kode_jur
E2021003	Sita	Jember	098745633	NULL	NULL
E20211001	Abizar	Jember	0821236789	199305102021032001	00JTI
E20211002	Bili	Jember	0821236789	199305102021032001	00JTI
E20211003	Ina	Mojokerto	0987654322	NULL	NULL
E20212001	Arka	Bondowoso	08212236789	199305102021032001	0JTEK
E20212002	Caca	Nganjuk	08212236789	199308312021032001	0JTEK

Tampilan table log_mhs:

```
SELECT * FROM tbl_log_mhs;
```

id_log	nim	alamat_lama	alamat_baru	waktu
4	E2021003	Temanggung	Jember	2021-12-04

Buatlah penjelasan dari contoh diatas sesuai dengan yang dipahami oleh masing-masing mahasiswa !

g. Hasil dan Pembahasan

Dokumentasi hasil latihan *trigger BEFORE UPDATE* pada database dalam bentuk laporan disertai dengan analisa.

h. Rubrik Penilaian

No	INDIKATOR KINERJA	Bobot (%)	Penilaian	Nilai Akhir
1	Memberikan penjelasan dan analisa secara terstruktur yang disertai dengan hasil implementasi query <i>trigger Before Update</i> dan atau penerapan query <i>trigger Before Update</i> pada basis data di lingkungan sekitar	30		
2	Memberikan bukti referensi dari jawaban	15		
3	Mempertanggung jawabkan hasil pekerjaan dengan presentasi	25		
4	Kerapian dalam menulis (bahasa dan struktur penulisan)	10		
5	Ketepatan waktu mengumpulkan	20		
	Total	100		

ACARA 51 - 52

Materi Pembelajaran : Studi Kasus Trigger
Acara Praktikum/Pertemuan : Minggu 14 / 3 dan 4
Tempat : Laboratorium Jurusan Teknologi Informasi, Politeknik
Negeri Jember
Alokasi Waktu : 100 menit

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu memahami penggunaan SQL *Trigger* pada studi kasus.
2. Mahasiswa mampu mengimplementasikan penggunaan SQL *Trigger* pada studi kasus database.

b. Penilaian Bertahap BNSP (Skill Passport)

Nama Skema Sertifikasi :

No	Kode Unit	Nama Unit Kompetensi	Elemen Kompetensi
1	J.620100.020.02	Menggunakan SQL	Membuat trigger

c. Indikator

1. Mahasiswa memahami penggunaan SQL *Trigger*.
2. Mahasiswa mengimplementasikan penggunaan SQL *Trigger* pada di database.

d. Dasar Teori

-

e. Alat dan Bahan

1. Kertas folio
2. Spidol
3. Bolpoin
4. Penggaris
5. Client server: Xampp
6. Text editor: Mysql Workbench atau sejenisnya

f. Prosedur Kerja

1. Perhatikan contoh berikut (**TRIGGER BEFORE DELETE**)

Berdasarkan database dbmahasiswa, buatlah satu table dengan nama "history_hapus_dosen". Tabel ini akan digunakan untuk menyimpan history data yang dihapus pada table dosen. Sehingga dengan adanya history_hapus_dosen ini akan memudahkan dalam melihat histori data dosen yang pernah ada dalam sistem.

Tabel history_hapus_dosen:

```
id_log INT (10) AUTO_INCREMENT, PRIMARY KEY
nip CHAR(18)
nama_dosen VARCHAR (25)
alamat CHAR(25)
phone CHAR(13)
```

Membuat trigger untuk menyimpan data dosen yang akan dihapus sebelum perintah DELETE dijalankan

```
DELIMITER $$
CREATE TRIGGER before_delete_dosen
    BEFORE DELETE
    ON tbl_dosen
    FOR EACH ROW
BEGIN
    INSERT INTO history_hapus_dosen
    SET nip = old.nip,
    nama_dosen = old.nama_dosen,
    alamat=old.alamat,
    phone = old.phone,
    waktu = NOW();
END;
```

Awal tampilan tabel dosen

BKPM – WORKSHOP BASIS DATA

nip	nama_dosen	alamat	phone
198005172008121002	Dwi Putro Sarwo Setyohadi	jember	085111222333
198804042021032001	Pramudhita Shinta Dewi Pu	Surabaya	081246456789
199002272021032001	Trismayanti Dwi Puspitasa	Yogyakarta	081231456789
199205292021032001	Bety Etikasari	Trenggalek	081456789123
199305102021032001	Lukie Perdanasari	Jember	081123456789
199308312021032001	Arvita Agus Kurniasari	Malang	081123987654

Note: tambahkan satu data dosen dengan nama Dwi Putro Sarwo Setyohadi

Awal tampilan tabel history_hapus_dosen

id_log	nip	nama_dosen	alamat	phone	waktu
--------	-----	------------	--------	-------	-------

Ujicoba dengan sintaks berikut dan jalankan:

```
1 DELETE FROM tbl_dosen WHERE nip = "198005172008121002";
```

Tampilan tabel dosen

nip	nama_dosen	alamat	phone
198804042021032001	Pramudhita Shinta Dewi Pu	Surabaya	081246456789
199002272021032001	Trismayanti Dwi Puspitasa	Yogyakarta	081231456789
199205292021032001	Bety Etikasari	Trenggalek	081456789123
199305102021032001	Lukie Perdanasari	Jember	081123456789
199308312021032001	Arvita Agus Kurniasari	Malang	081123987654

Tampilan tabel history_hapus_dosen

id_log	nip	nama_dosen	alamat	phone	waktu
0	198005172008121002	Dwi Putro Sarwo Setyohadi	jember	085111222333	2021-12-06

Buatlah penjelasan dari contoh diatas sesuai dengan yang dipahami oleh masing-masing mahasiswa !

g. Hasil dan Pembahasan

Dokumentasi hasil latihan query *trigger BEFORE DELETE* pada pada studi kasus di database dalam bentuk laporan disertai dengan analisa.

h. Rubrik Penilaian

No	INDIKATOR KINERJA	Bobot (%)	Penilaian	Nilai Akhir
1	Memberikan penjelasan dan analisa secara terstruktur yang disertai dengan hasil implementasi query <i>trigger Before Delete</i> dan atau penerapan query <i>trigger Before Delete</i> pada basis data di lingkungan sekitar	30		
2	Memberikan bukti referensi dari jawaban	15		
3	Mempertanggung jawabkan hasil pekerjaan dengan presentasi	25		
4	Kerapian dalam menulis (bahasa dan struktur penulisan)	10		
5	Ketepatan waktu mengumpulkan	20		
	Total	100		