

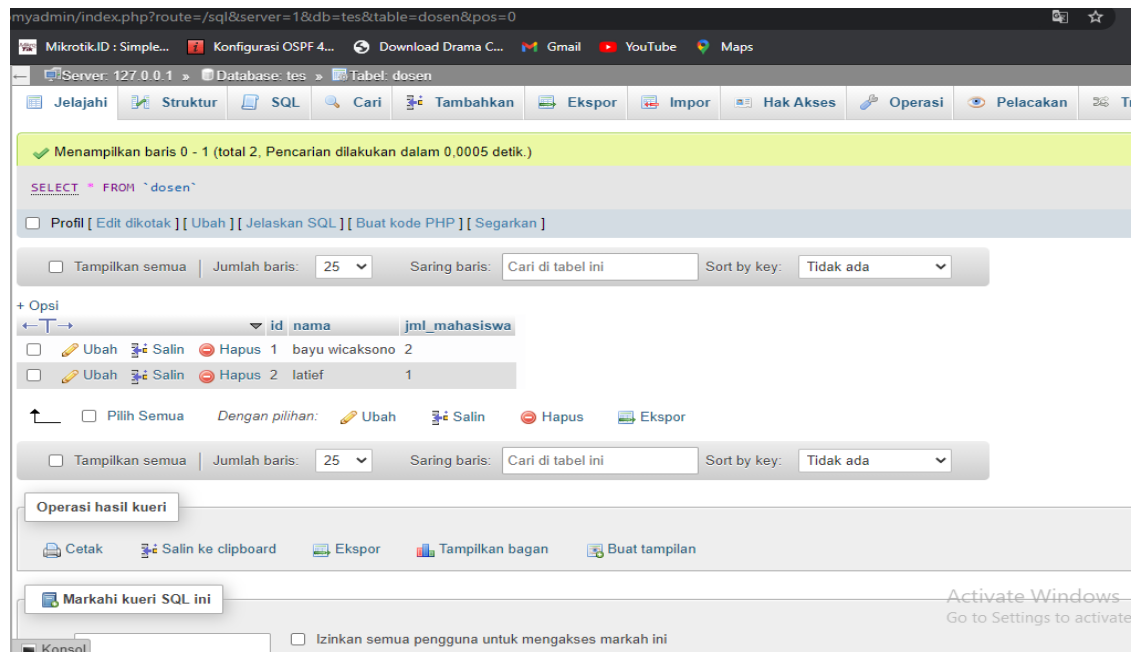
Nama : Fariz Fahmi Faturachmad

Kelas : SI 3B

LAPORAN TUGAS BASIS DATA LANJUTAN

❖ Kondisi Database Awal

➤ Tabel Dosen



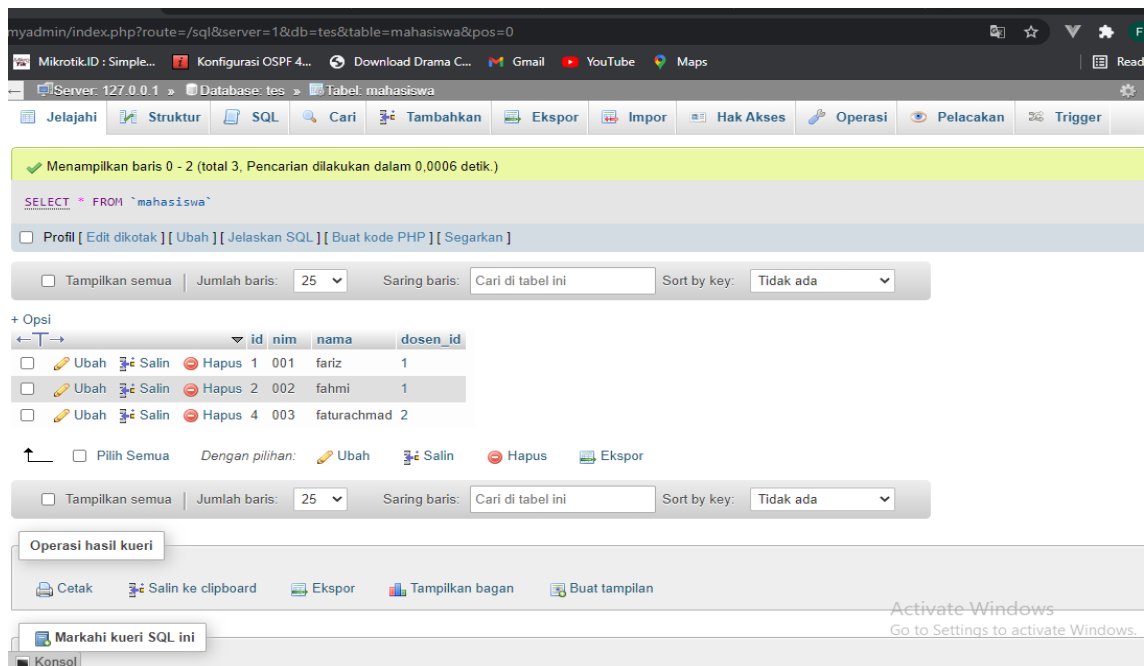
(Gambar 1.0)

Penjelasan : Pertama, membuat tabel dosen dan mengisi data yang ada di dalamnya.

Code :

```
CREATE TABLE dosen (  
  id INT PRIMARY KEY AUTO_INCREMENT,  
  nama varchar(225),  
  jml_mahasiswa INT  
);  
  
INSERT INTO `dosen` (`id`, `nama`, `jml_mahasiswa`) VALUES ('1', 'bayu wicaksono', '2'),  
(NULL, 'latief', '1');
```

➤ Tabel Mahasiswa



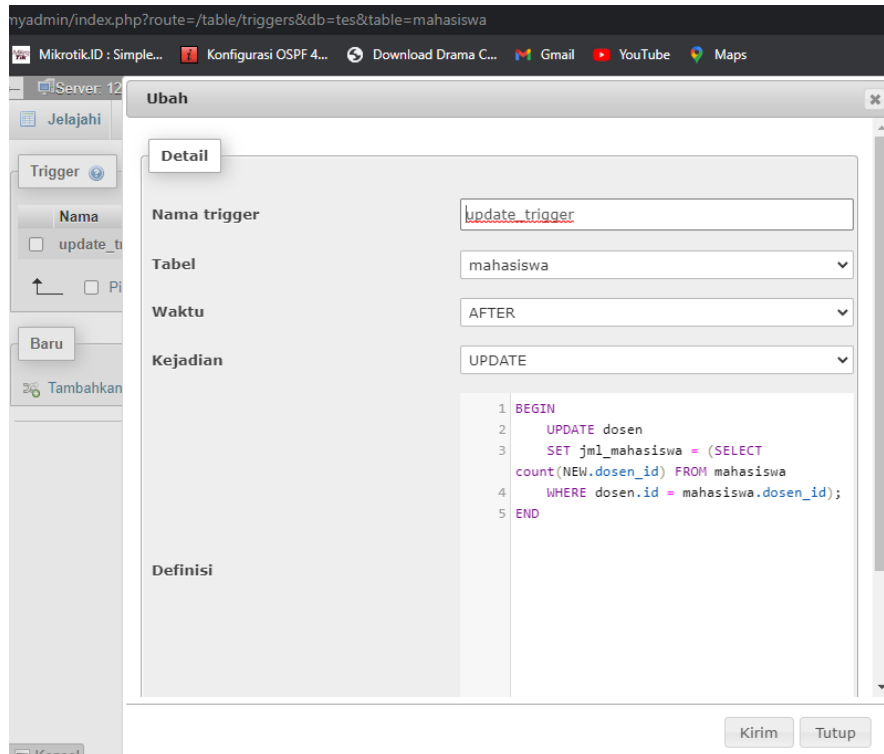
(Gambar 1.1)

Penjelasan : Kemudian, membuat tabel mahasiswa dan mengisi data yang ada di dalamnya.

Code :

```
CREATE TABLE mahasiswa(  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    nim varchar(3),  
    nama varchar(225),  
    dosen_id INT,  
    FOREIGN KEY (dosen_id) REFERENCES dosen(id)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE  
);  
  
INSERT INTO `dosen` (`id`, `nim`, `nama`, `dosen_id`) VALUES ('1', '001', 'fariz', '1'), (NULL, '002', 'fahmi', '1'), (NULL, '003', 'faturachmad', '2');
```

A. Membuat TRIGGER



nyadmin/index.php?route=/table/triggers&db=tes&table=mahasiswa

Mikrotik.ID : Simple... Konfigurasi OSPF 4... Download Drama C... Gmail YouTube Maps

Server: 12

Jelajahi

Trigger

Nama

☐ update_t

Baru

Tambahkan

Konsol

Ubah

Detail

Nama trigger:

Tabel:

Waktu:

Kejadian:

```
1 BEGIN
2   UPDATE dosen
3   SET jml_mahasiswa = (SELECT
4     count(NEW.dosen_id) FROM mahasiswa
5     WHERE dosen.id = mahasiswa.dosen_id);
6 END
```

Definisi

Kirim Tutup

(Gambar 1.2)

Penjelasan : Lalu, membuat TRIGGER dengan kondisi ketika ada **mahasiswa** yang di UPDATE **dosen_id** nya, maka kolom **jml_mahasiswa** yang ada di **table dosen** akan ikut terUPDATE juga. Dalam kasus ini, menggunakan AFTER UPDATE karena kolom **jml_mahasiswa** akan terUPDATE **sesudah** kolom **dosen_id** terjadi UPDATE.

Untuk melakukan TRIGGER tersebut, klik Trigger yang ada di pojok kanan atas pada navbar phpmyadmin,



(Gambar 1.3)

Kemudian, isi data yang dibutuhkan seperti pada Gambar 1.2 . Pada kolom Nama trigger, isi terserah (Optional), namun lebih disarankan untuk disesuaikan dengan TRIGGER yang akan dilakukan agar penamaan tidak membingungkan diri sendiri maupun orang lain nantinya.

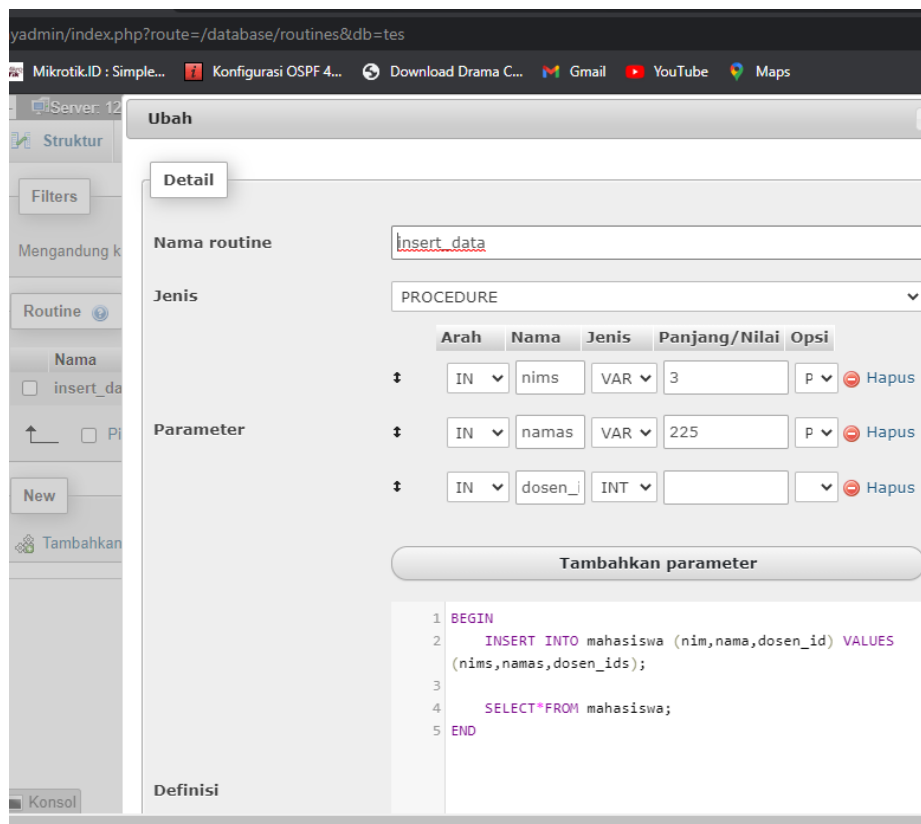
1. Pada kolom Tabel, pilih tabel yang akan dilakukan TRIGGER nya, yaitu tabel mahasiswa.
2. Setelah itu, pada kolom Waktu, pilih waktu yang akan diterapkan, yaitu AFTER, karena nantinya tabel dosen akan otomatis terupdate setelah dilakukan update pada tabel mahasiswa.
3. Pilih Kejadian yang akan diterapkan pada TRIGGER, yaitu UPDATE karena pemicu terjadinya update pada tabel dosen dikarenakan adanya update data pada tabel mahasiswa.

4. Langkah terakhir, pada kolom Definisi digunakan untuk membuat statement yang akan dijalankan ketika sudah terjadi update di tabel mahasiswa. Codenya seperti di bawah ini,

Code :

```
BEGIN
  UPDATE dosen
  SET jml_mahasiswa = (SELECT count(NEW.dosen_id) FROM mahasiswa
  WHERE dosen.id = mahasiswa.dosen_id);
END
```

B. Membuat PROCEDURE



(Gambar 1.4)

Penjelasan : Membuat PROCEDURE dengan kondisi ketika ada data baru yang **dimasukkan** pada **tabel mahasiswa**, maka data tersebut akan **otomatis ditambahkan** ke dalam **tabel mahasiswa**. Dengan data yang **dimasukkan** meliputi **kolom nim, nama, dan dosen_id**.

Untuk melakukan PROCEDURE tersebut, klik Routine yang ada di pojok kanan atas pada navbar phpmyadmin,



(Gambar 1.5)

Kemudian, isi data yang dibutuhkan seperti pada Gambar 1.4 . Pada kolom Nama routine, isi terserah (Optional), namun lebih disarankan untuk disesuaikan dengan TRIGGER yang akan dilakukan agar penamaan tidak membingungkan diri sendiri maupun orang lain nantinya.

1. Pada kolom Jenis, pilih jenis routine yang akan dilakukan, yaitu PROCEDURE.
2. Setelah itu, pada kolom Parameter, tambahkan parameter sehingga menjadi 3 parameter.
3. Kemudian, pada kolom Arah, pilih IN pada semua parameter. Karena data yang ditambahkan nanti akan otomatis dimasukkan ke dalam tabel mahasiswa.
4. Untuk kolom Nama dibuat untuk penamaan parameternya. Karena, penamaan tersebut akan digunakan untuk values dari statement yang dibuat.
5. Lalu, pada kolom Jenis, pilih tipe data dari parameter yang akan digunakan nanti. (Sesuai dengan tipe data di tabel mahasiswa yang sudah dibuat sebelumnya).
6. Pada kolom Panjang/Nilai, diisi dengan nilai dari tipe data yang sudah dipilih tadi. (Tidak melebihi Nilai yang ada di tabel mahasiswa).
7. Langkah terakhir, pada kolom Definisi digunakan untuk membuat statement yang akan dijalankan ketika PROCEDURE diexecute atau dijalankan. Codenya seperti di bawah ini,

Code :

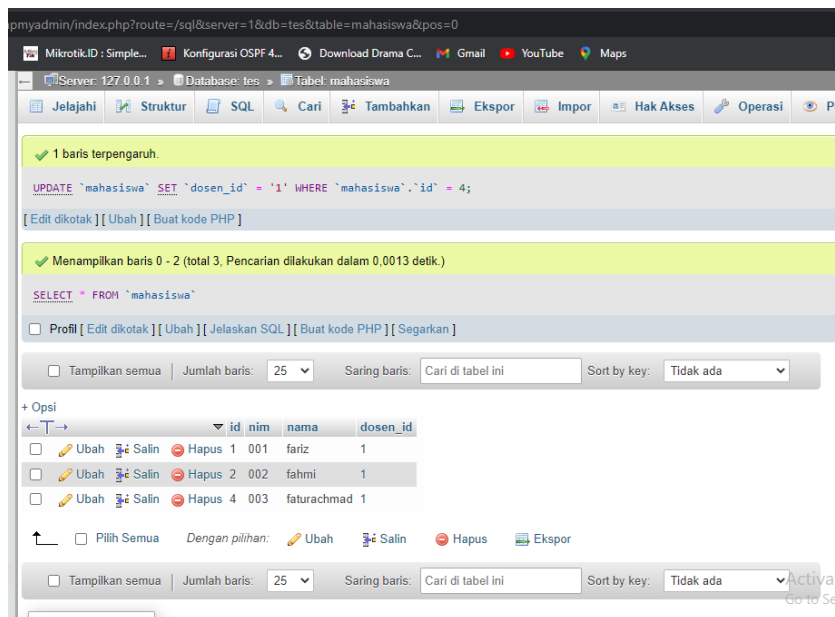
```
BEGIN
    INSERT INTO mahasiswa (nim,nama,dosen_id) VALUES
    (nims,namas,dosen_ids);

    SELECT*FROM mahasiswa;
END
```

❖ Kondisi Database Akhir

1. AFTER UPDATE dosen_id COLUMN FOR UPDATE TRIGGER

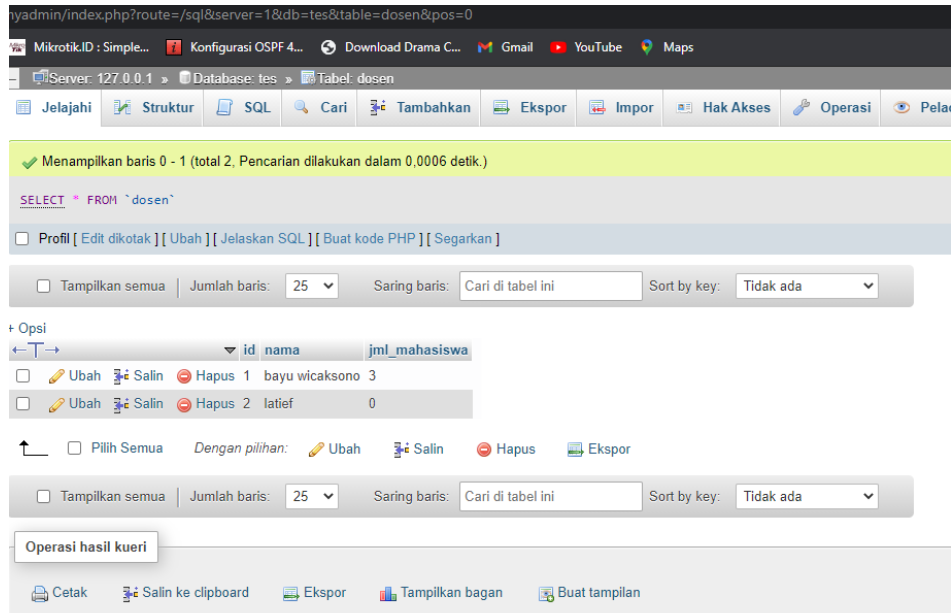
➤ Tabel Mahasiswa



(Gambar 1.5)

Penjelasan : Setelah membuat trigger update, uji coba dilakukan pada table mahasiswa dengan mengedit dosen_id dari salah satu data mahasiswa yaitu faturachmad. Yang awalnya dosen_id = 2 diedit menjadi dosen_id = 1.

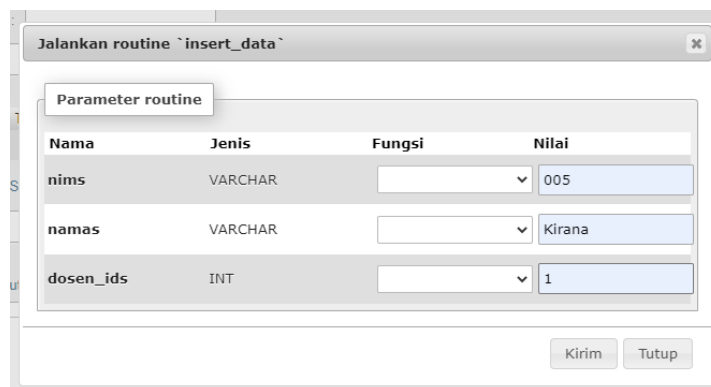
➤ Tabel Dosen



(Gambar 1.6)

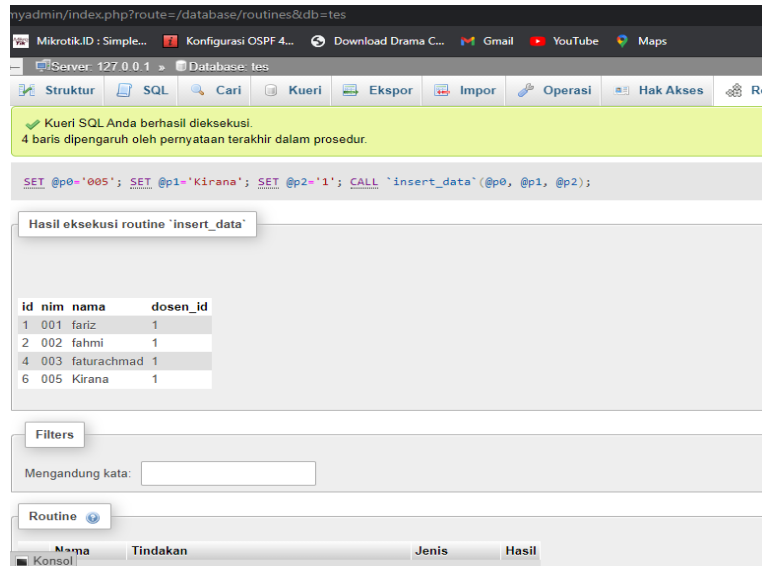
Penjelasan : Setelah dilakukan edit di column dosen_id pada table mahasiswa, maka otomatis tabel dosen juga ikut terupdate di column jml_mahasiswa. Yang awalnya dosen dengan id = 1 mempunyai 2 mahasiswa, sekarang punya 3 mahasiswa, begitu pula dengan dosen id = 2. Dari 1 mahasiswa menjadi 0 mahasiswa.

2. AFTER PROCEDURE ON TABLE MAHASISWA

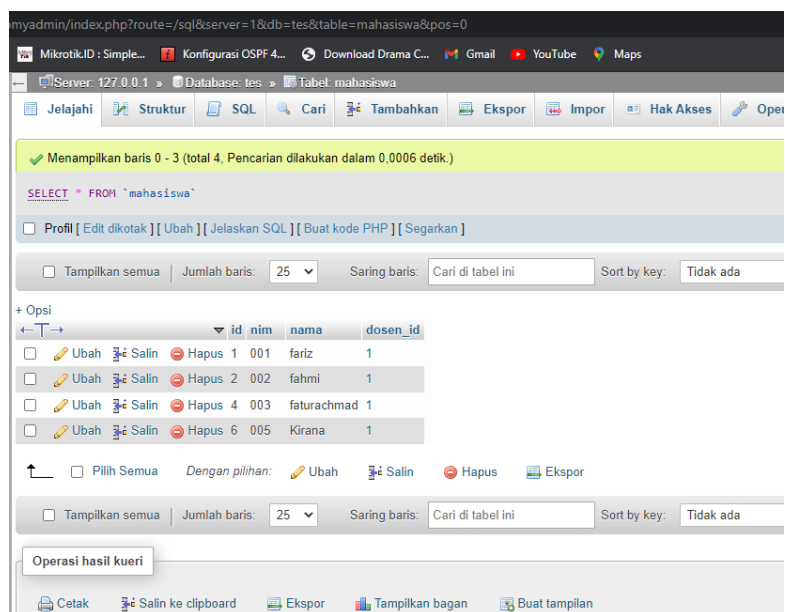


(Gambar 1.7)

Penjelasan : Setelah membuat prosedur, uji coba dilakukan pada table mahasiswa dengan memasukkan data baru melalui prosedur.



(Gambar 1.8)



(Gambar 1.9)

Penjelasan : Setelah membuat data baru, data tersebut akan otomatis masuk ke dalam table mahasiswa melalui prosedur yang sudah dibuat.