**LAPORAN PRAKTIKUM**

**Basis Data**

**Jobsheet-11: Data Manipulation Language**



|  |  |
| --- | --- |
| Nama | : Ghoffar Abdul Ja’far |
| NIM | : 2341720035 |
| Kelas | : 1E |

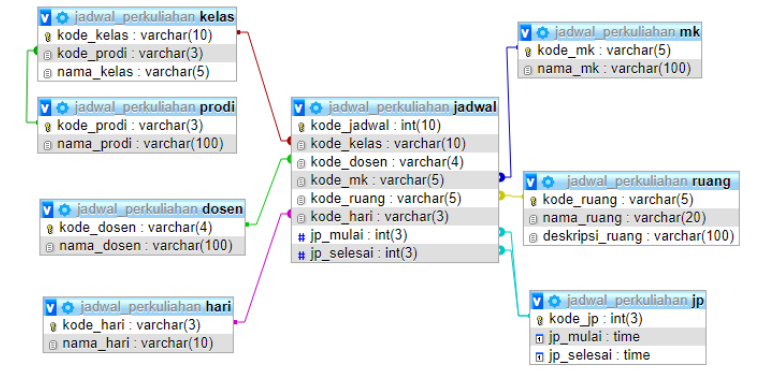
**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

**2023/2024**

**Praktikum – Bagian 1: Membuat Database untuk Percobaan**

1. Skema/model relasional/EER diagram dari database.



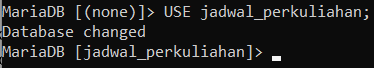
1. **DROP DATABASE IF EXISTS jadwal\_perkuliahan**; digunakan untuk menghapus database jadwal\_perkuliahan jika database tersebut sudah ada. Tujuannya adalah untuk memastikan bahwa kita memulai dengan database yang kosong.



1. **CREATE DATABASE jadwal\_perkuliahan**; digunakan untuk membuat database baru dengan nama jadwal\_perkuliahan. Database ini nantinya akan digunakan untuk menyimpan semua tabel dan data yang berkaitan dengan sistem informasi penjadwalan.



1. **USE jadwal\_perkuliahan**; digunakan untuk mengubah database aktif yang sedang digunakan oleh koneksi SQL saat ini menjadi jadwal\_perkuliahan. Setelah perintah ini dijalankan, semua operasi SQL yang dilakukan akan berlaku pada database jadwal\_perkuliahan.

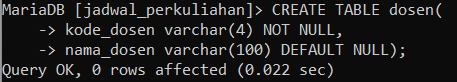


1. CREATE TABLE dosen: membuat tabel baru dengan nama dosen.

kode\_dosen varchar(4)

NOT NULL: mendefinisikan kolom kode\_dosen dengan tipe data varchar(4). Tipe data varchar(4) berarti kolom ini dapat menyimpan string dengan panjang hingga 4 karakter. NOT NULL berarti kolom ini harus selalu memiliki nilai; tidak boleh kosong atau NULL.

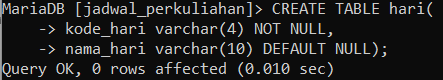
nama\_dosen varchar(100) DEFAULT NULL: mendefinisikan kolom nama\_dosen dengan tipe data varchar(100). Tipe data varchar(100) berarti kolom ini dapat menyimpan string dengan panjang hingga 100 karakter. DEFAULT NULL berarti jika tidak ada nilai yang disediakan saat memasukkan data, kolom ini akan secara otomatis diisi dengan NULL.



1. CREATE TABLE hari: membuat tabel baru dengan nama hari.

kode\_hari varchar(3) NOT NULL: mendefinisikan kolom kode\_hari dengan tipe data varchar(3). Tipe data varchar(3) berarti kolom ini dapat menyimpan string dengan panjang hingga 3 karakter. NOT NULL berarti kolom ini harus selalu memiliki nilai; tidak boleh kosong atau NULL.

nama\_hari varchar(10) DEFAULT NULL: mendefinisikan kolom nama\_hari dengan tipe data varchar(10). Tipe data varchar(10) berarti kolom ini dapat menyimpan string dengan panjang hingga 10 karakter. DEFAULT NULL berarti jika tidak ada nilai yang disediakan saat memasukkan data, kolom ini akan secara otomatis diisi dengan NULL.



1. CREATE TABLE jadwal: membuat tabel baru dengan nama jadwal.

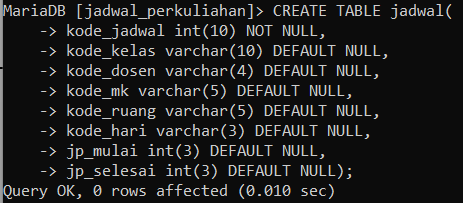
kode\_jadwal int(10) NOT NULL: mendefinisikan kolom kode\_jadwal dengan tipe data int(10). Tipe data int(10) berarti kolom ini dapat menyimpan angka integer dengan panjang hingga 10 digit. NOT NULL berarti kolom ini harus selalu memiliki nilai; tidak boleh kosong atau NULL.

kode\_kelas varchar(10) DEFAULT NULL, kode\_dosen varchar(4) DEFAULT NULL, kode\_mk varchar(5) DEFAULT NULL, kode\_ruang varchar(5)

DEFAULT NULL, dan kode\_hari varchar(3) DEFAULT NULL: mendefinisikan kolom-kolom dengan tipe data varchar dan panjang karakter yang berbeda-beda. DEFAULT NULL berarti jika tidak ada nilai yang disediakan saat memasukkan data, kolom ini akan secara otomatis diisi dengan NULL.

jp\_mulai int(3) DEFAULT NULL dan jp\_selesai int(3) DEFAULT NULL:

mendefinisikan kolom jp\_mulai dan jp\_selesai dengan tipe data int(3). Tipe data int(3) berarti kolom ini dapat menyimpan angka integer dengan panjang hingga 3 digit. DEFAULT NULL berarti jika tidak ada nilai yang disediakan saat memasukkan data, kolom ini akan secara otomatis diisi dengan NULL.



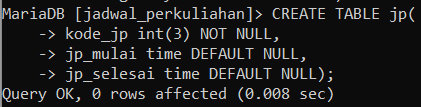
1. CREATE TABLE jp: Perintah ini membuat tabel baru dengan nama jp.

kode\_jp int(3) NOT NULL: mendefinisikan kolom kode\_jp dengan tipe data int(3). Tipe data int(3) berarti kolom ini dapat menyimpan angka integer dengan panjang hingga 3 digit.

NOT NULL berarti kolom ini harus selalu memiliki nilai; tidak boleh kosong atau NULL.

jp\_mulai time DEFAULT NULL dan jp\_selesai time DEFAULT NULL:

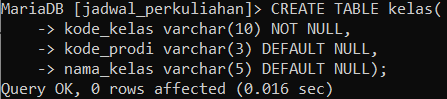
mendefinisikan kolom jp\_mulai dan jp\_selesai dengan tipe data time. Tipe data time berarti kolom ini dapat menyimpan waktu dalam format HH:MM:SS. DEFAULT NULL berarti jika tidak ada nilai yang disediakan saat memasukkan data, kolom ini akan secara otomatis diisi dengan NULL.



9. CREATE TABLE kelas: membuat tabel baru dengan nama kelas.

kode\_kelas varchar(10) NOT NULL: mendefinisikan kolom kode\_kelas dengan tipe data varchar(10). Tipe data varchar(10) berarti kolom ini dapat menyimpan string dengan panjang hingga 10 karakter. NOT NULL berarti kolom ini harus selalu memiliki nilai; tidak boleh kosong atau NULL.

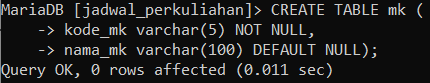
kode\_prodi varchar(3) DEFAULT NULL dan nama\_kelas varchar(5) DEFAULT NULL: mendefinisikan kolom kode\_prodi dan nama\_kelas dengan tipe data varchar dan panjang karakter yang berbeda-beda. DEFAULT NULL berarti jika tidak ada nilai yang disediakan saat memasukkan data, kolom ini akan secara otomatis diisi dengan NULL.



10. CREATE TABLE mk: membuat tabel baru dengan nama mk.

kode\_mk varchar(5) NOT NULL: mendefinisikan kolom kode\_mk dengan tipe data varchar(5). Tipe data varchar(5) berarti kolom ini dapat menyimpan string dengan panjang hingga 5 karakter. NOT NULL berarti kolom ini harus selalu memiliki nilai; tidak boleh kosong atau NULL.

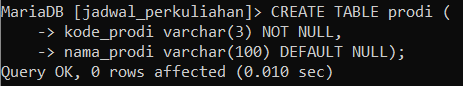
nama\_mk varchar(100) DEFAULT NULL: mendefinisikan kolom nama\_mk dengan tipe data varchar(100). Tipe data varchar(100) berarti kolom ini dapat menyimpan string dengan panjang hingga 100 karakter. DEFAULT NULL berarti jika tidak ada nilai yang disediakan saat memasukkan data, kolom ini akan secara otomatis diisi dengan NULL.



11. CREATE TABLE prodi: membuat tabel baru dengan nama prodi.

kode\_prodi varchar(3) NOT NULL: mendefinisikan kolom kode\_prodi dengan tipe data varchar(3). Tipe data varchar(3) berarti kolom ini dapat menyimpan string dengan panjang hingga 3 karakter. NOT NULL berarti kolom ini harus selalu memiliki nilai; tidak boleh kosong atau NULL.

nama\_prodi varchar(100) DEFAULT NULL: mendefinisikan kolom nama\_prodi dengan tipe data varchar(100). Tipe data varchar(100) berarti kolom ini dapat menyimpan string dengan panjang hingga 100 karakter. DEFAULT NULL berarti jika tidak ada nilai yang disediakan saat memasukkan data, kolom ini akan secara otomatis diisi dengan NULL.

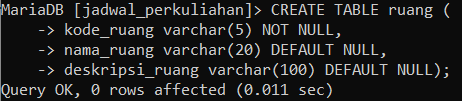


12. CREATE TABLE ruang: membuat tabel baru dengan nama ruang.

kode\_ruang varchar(5) NOT NULL: mendefinisikan kolom kode\_ruang dengan tipe data varchar(5). Tipe data varchar(5) berarti kolom ini dapat menyimpan string dengan panjang hingga 5 karakter. NOT NULL berarti kolom ini harus selalu memiliki nilai; tidak boleh kosong atau NULL.

nama\_ruang varchar(20) DEFAULT NULL: mendefinisikan kolom nama\_ruang dengan tipe data varchar(20). Tipe data varchar(20) berarti kolom ini dapat menyimpan string dengan panjang hingga 20 karakter. DEFAULT NULL berarti jika tidak ada nilai yang disediakan saat memasukkan data, kolom ini akan secara otomatis diisi dengan NULL.

deskripsi\_ruang varchar(100) DEFAULT NULL: Ini mendefinisikan kolom deskripsi\_ruang dengan tipe data varchar(100). Tipe data varchar(100) berarti kolom ini dapat menyimpan string dengan panjang hingga 100 karakter. DEFAULT NULL berarti jika tidak ada nilai yang disediakan saat memasukkan data, kolom ini akan secara otomatis diisi dengan NULL.



13. Penjelasan secara berurut, yaitu:

• menambahkan kunci utama (PRIMARY KEY) pada kolom kode\_dosen di tabel dosen. Kunci utama digunakan untuk memastikan data dalam kolom tersebut unik dan tidak boleh NULL.

• menambahkan kunci utama (PRIMARY KEY) pada kolom kode\_hari di tabel hari.

• menambahkan kunci utama (PRIMARY KEY) pada kolom kode\_jadwal di tabel jadwal.

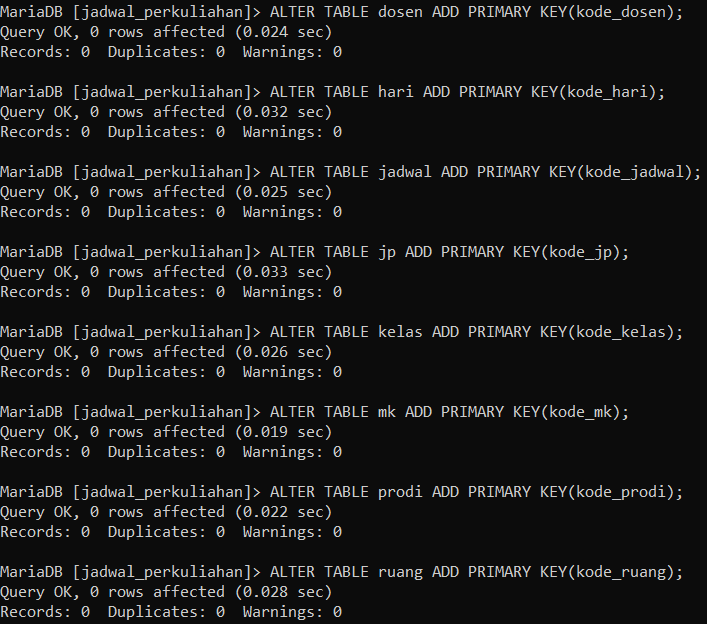
• menambahkan kunci utama (PRIMARY KEY) pada kolom kode\_jp di tabel jp.

• menambahkan kunci utama (PRIMARY KEY) pada kolom kode\_kelas di tabel kelas.

• menambahkan kunci utama (PRIMARY KEY) pada kolom kode\_mk di tabel mk.

• menambahkan kunci utama (PRIMARY KEY) pada kolom kode\_prodi di tabel prodi.

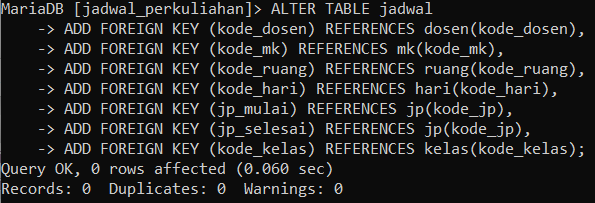
• menambahkan kunci utama (PRIMARY KEY) pada kolom kode\_ruang di tabel ruang.



14. digunakan untuk memodifikasi struktur tabel yang sudah ada dalam database.



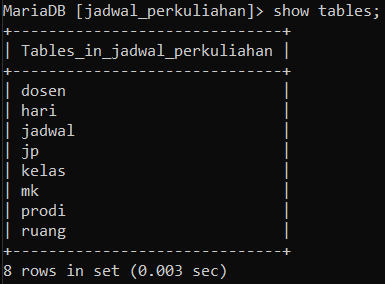
15. menambahkan kunci asing pada beberapa kolom di tabel jadwal. Kunci asing digunakan untuk memastikan integritas data antara tabel jadwal dan tabel-tabel lainnya (dosen, mk, ruang, hari, jp, dan kelas). Misalnya, ADD FOREIGN KEY (kode\_dosen) REFERENCES dosen (kode\_dosen) berarti setiap nilai dalam kolom kode\_dosen di tabel jadwal harus ada di kolom kode\_dosen di tabel dosen.



16. menambahkan kunci asing pada kolom kode\_prodi di tabel kelas. Kunci asing ini memastikan bahwa setiap nilai dalam kolom kode\_prodi di tabel kelas harus ada di kolom kode\_prodi di tabel prodi.



17. Cek database Anda dengan perintah ‘SHOW TABLES’ untuk memastikan bahwa semua tabel sudah dibuat. Setelah selesai membuat database diatas, lanjutkan ke Praktikum – Bagian 2.

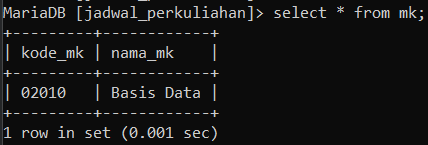


**Praktikum – Bagian 2: Percobaan Statement INSERT**

1. Untuk menambahkan data (mengisi) suatu tabel, digunakan statement (pernyataan) INSERT. Eksekusi SQL berikut untuk menambahkan 1 baris (record) baru pada tabel mk



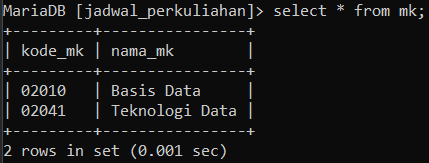
2. Untuk melihat hasil SQL yang kita eksekusi tersebut, gunakan statement SELECT seperti berikut.



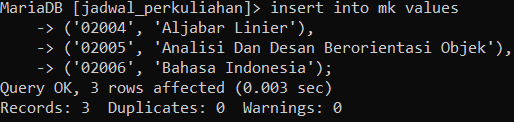
3. Apabila data di-insert-kan pada semua kolom tabel, maka kita dapat langsung menggunakan klausa VALUES tanpa harus menuliskan nama-nama kolom dahulu.



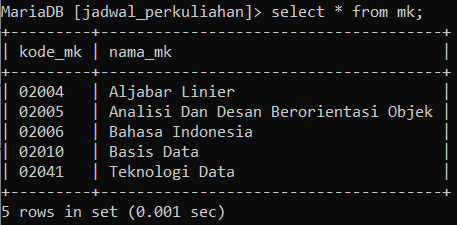
4. Statement SQL tersebut menambahkan 1 baris baru ke tabel mk pada kolom yang ada pada struktur tabel mk tanpa menyebutkan nama kolomnya.



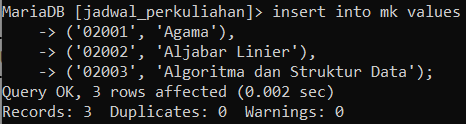
5. Untuk menambahkan beberapa kolom sekaligus dalam 1 statement digunakan statement dengan format seperti berikut.



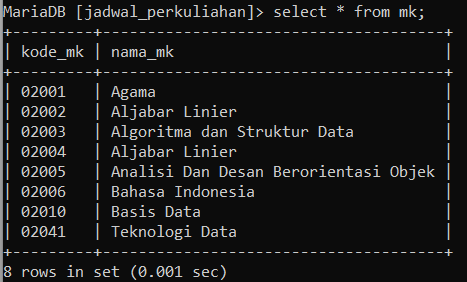
6. Statement SQL tersebut menambahkan 3 baris baru ke tabel mk pada kolom yang ada pada struktur tabel mk tanpa menyebutkan nama kolomnya.



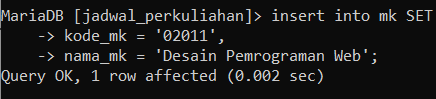
7. Dan seperti berikut, jika hanya kolom tertentu saja yang akan diberi nilai dengan cara menyebutkan nama kolomnya.



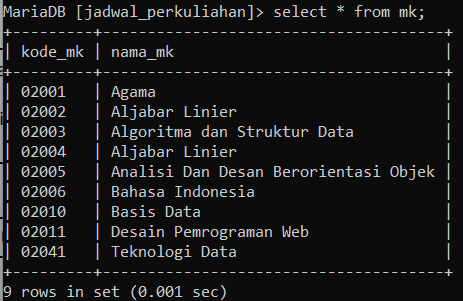
8. Statement SQL tersebut menambahkan 3 baris baru ke tabel mk pada kolom yang ada pada struktur tabel mk.



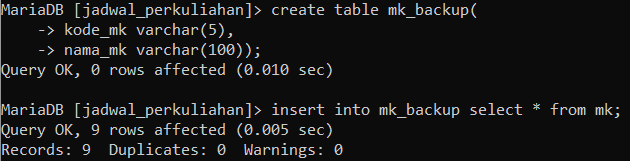
9. Statement INSERT juga dapat dieksekusi dengan menggunakan klausa SET alih-alih VALUES.



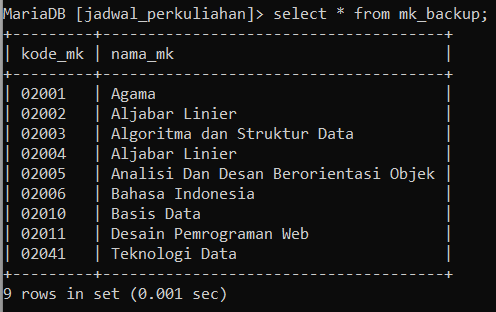
10. Statement SQL tersebut menambahkan 1 baris baru ke tabel mk pada kolom yang ada pada struktur tabel mk.



11. Pada statement INSERT juga dapat digunakan klausa SELECT.

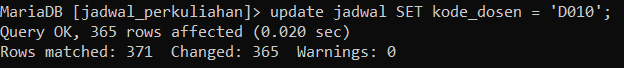


12. Statement SQL tersebut menambahkan data baru dari tabel mk ke tabel mk\_backup.

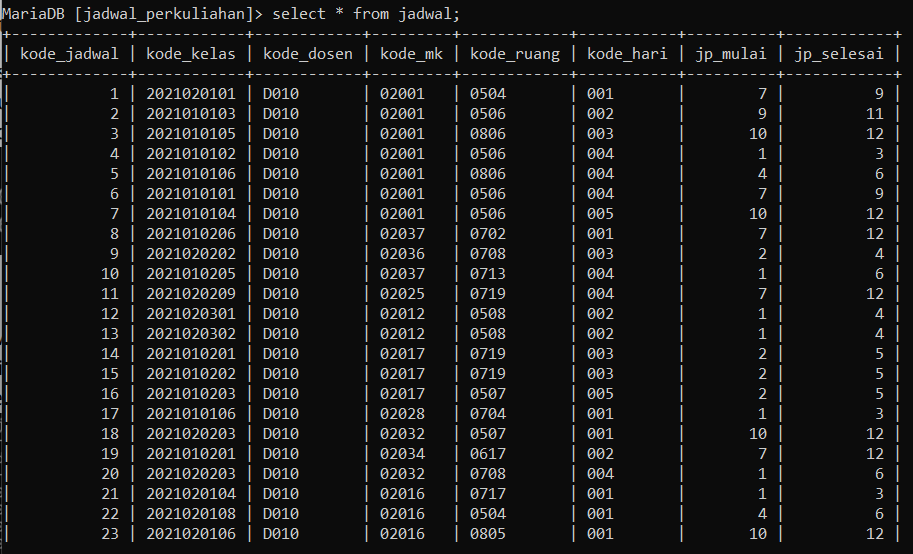


**Praktikum - Bagian 3: Percobaan Statement UPDATE**

1. UPDATE digunakan untuk mengubah nilai suatu baris pada sebuah tabel. Sebelum memulai praktikum bagian 3, Import terlebih dahulu file isi\_data\_jadwal\_perkuliahan.sql pada database jadwal\_perkuliahan yang sudah dibuat pada Bagian 1.



2. Statement tersebut mengubah nilai SEMUA baris dari tabel jadwal pada kolom kode\_dosen dengan nilai D010. Apabila kita tampilkan isi tabel, maka sekarang semua mata kuliah akan diampu oleh dosen dengan kode\_dosen tersebut.



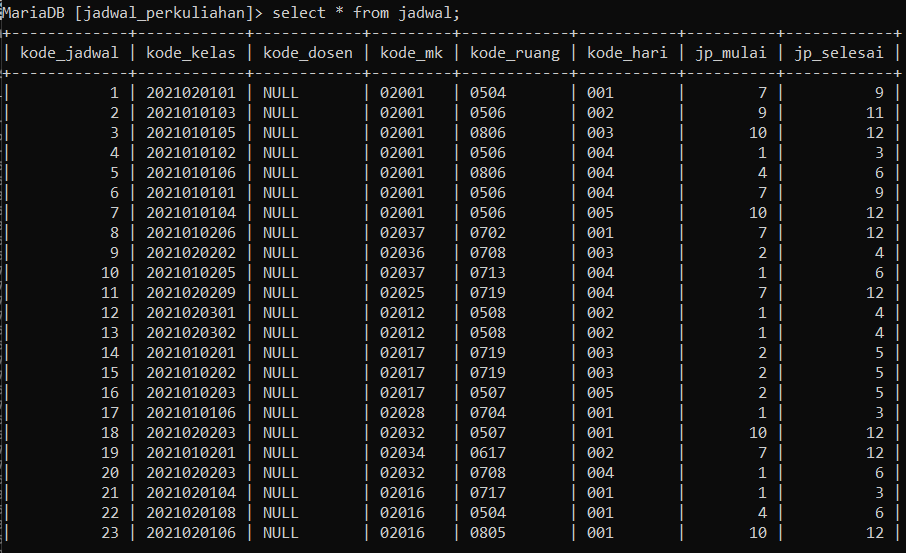
3. Untuk mengubah nilai pada baris tertentu saja, kita tambahkan klausa WHERE pada statement UPDATE. Misalkan kita akan menjadikan dosen dengan kode\_dosen D022 sebagai pengampu mata kuliah dengan kode\_mk 02010.



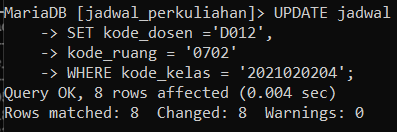


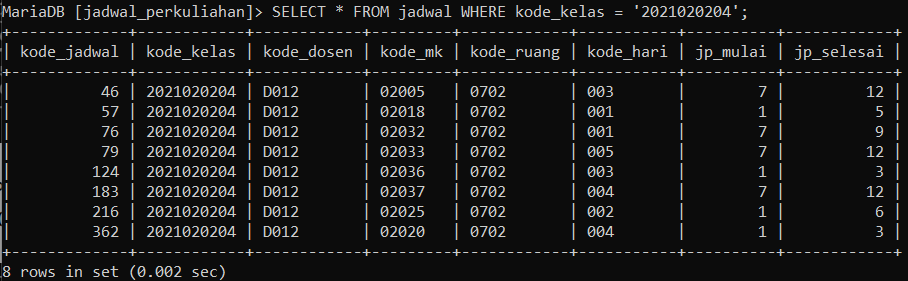
4. Klausa WHERE tidak selalu hanya membatasi UPDATE pada 1 baris saja, ia juga bisa memberlakukan UPDATE pada banyak baris sekaligus. Semuanya tergantung pada kondisi yang kita tentukan. Statement berikut ini akan mengosongkan kode\_dosen untuk semua mata kuliah yang diampu oleh dosen dengan kode\_dosen D010.



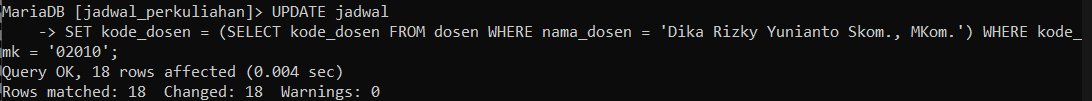


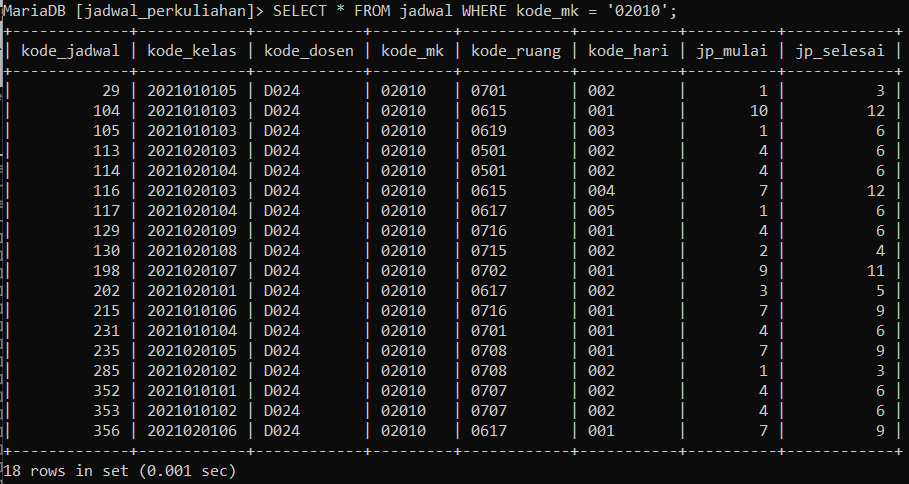
5. Untuk mengubah beberapa kolom sekaligus dalam satu kali eksekusi statement UPDATE, dapat digunakan format berikut.





6. Kita juga dapat menggunakan statement UPDATE dengan SELECT. Misalkan kita ingin mengeset kode\_dosen dari kode\_mk ‘02010’ dengan kode\_dosen dari dosen yang bernama ‘Dika Rizky Yunianto SKom., MKom.’

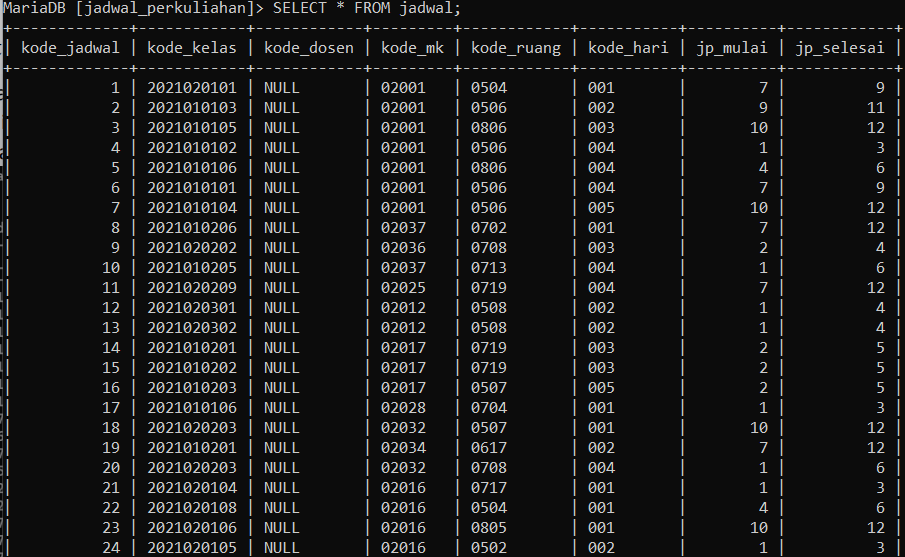




**Praktikum - Bagian 4: Percobaan Statement DELETE**

1. DELETE digunakan untuk menghapus satu atau lebih baris dari sebuah tabel. Misalkan kita ingin menghapus jadwal yang memiliki nilai pada kolom kode\_dosen, maka format dasar statement seperti berikut dapat kita gunakan:





2. HATI-HATI apabila kita menggunakan statement DELETE tanpa WHERE! Cobalah eksekusi syntax SQL berikut:

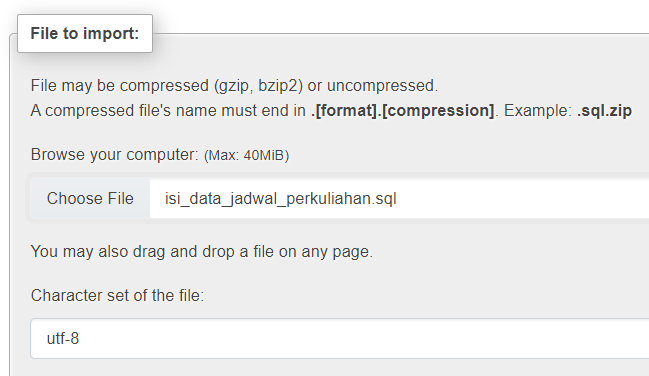


3. Semua data dalam satu tabel jadwal akan hilang!



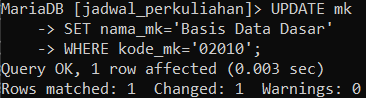
Tugas

- Import kembali isi\_data\_jadwal\_perkuliahan.sql.

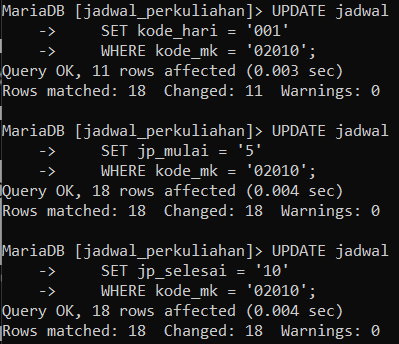


- Screenshot sintaks dan hasil SELECT dari setiap soal dibawah ini!

1. Ubah nama mata kuliah “Basis Data” menjadi “Basis Data Dasar”!



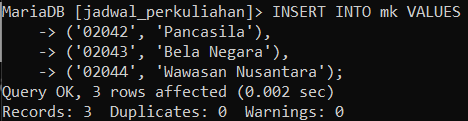
2. Ubah semua jadwal kuliah mata kuliah “Basis Data Dasar” menjadi hari Senin di jam pelajaran ke 5 sampai dengan jam pelajaran ke 10!



3. Hapus jadwal perkuliahan “Kewarganegaraan” pada tabel jadwal!



4. Tambahkan mata kuliah “Pancasila”, “Bela Negara”, “Wawasan Nusantara” pada tabel mk!



5. Hapus semua isi data pada tabel mk\_backup!

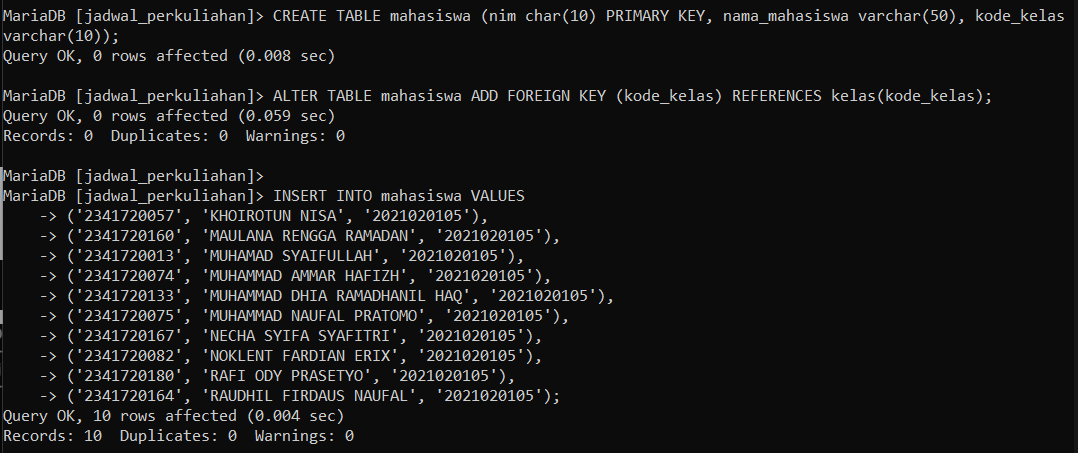


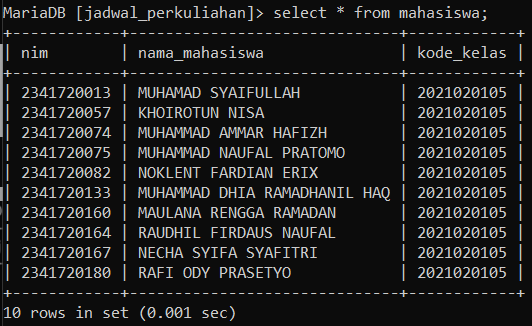
6. Isi data tabel mk\_backup dengan isi dari tabel mk!



7. Buatlah tabel mahasiswa dengan atribut nim, nama\_mahasiswa, kode\_kelas. Dimana kode\_kelas mereferensi kepada tabel kelas. Isi tabel tersebut dengan 10 nama mahasiswa yang memiliki nomor presensi berturt-turut setelah anda di kelas anda. Isi kode\_kelas sesuai kode kelas anda saat ini. Ubahlah kode\_dosen mata kuliah “Basis Data Dasar” kelas anda pada tabel jadwal sesuai dengan dosen pengampu mata kuliah basis data anda saat ini !

Mahasiswa:





Dosen:

