# PRAKTIKUM BASIS DATA

# Pertemuan 11- DATA MANIPULATION LANGUAGE (DML)

Dosen Pengajar : Bapak Farid Angga Pribadi, S.Kom., M.Kom



Nama : Surya Rahmat Fatahillah

NIM : 2341760020

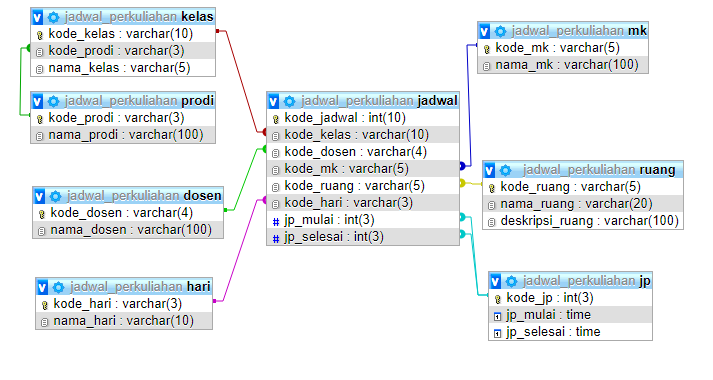
Prodi : Sistem Informasi Bisnis

# JURUSAN TEKNOLOGI INFORMASI POLITEKNIK NEGERI MALANG

# 2024

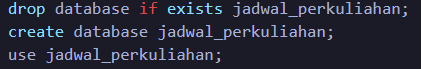
**Praktikum – Bagian 1: Membuat Database untuk Percobaan**

1. Perhatikan skema/model relasional/EER diagram dari database berikut.



1. Skema tersebut adalah sekema database pada sebuah sistem informasi penjadwalan di Jurusan Teknologi Informasi. Pertama-tama, **buatlah database tersebut dengan mengeksekusi baris-baris kode DDL berikut dan jelaskan maksud dari setiap tahapan tersebut.**

2a.



**DROP DATABASE IF EXISTS jadwal\_perkuliahan;**

* + Perintah ini bertujuan untuk menghapus database dengan nama "jadwal\_perkuliahan" jika sudah ada. Kata kunci **IF EXISTS** digunakan untuk memastikan bahwa perintah hanya dijalankan jika database tersebut sudah ada. Jika database tidak ada, perintah ini tidak akan melakukan apa pun.

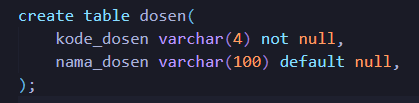
**CREATE DATABASE jadwal\_perkuliahan;**

* + Ini adalah perintah untuk membuat database baru dengan nama "jadwal\_perkuliahan". Setelah perintah ini dijalankan, sebuah database baru akan dibuat dengan nama tersebut.

**USE jadwal\_perkuliahan;**

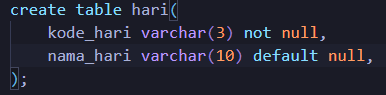
* + Perintah ini digunakan untuk beralih ke database "jadwal\_perkuliahan". Setelah perintah ini dijalankan, semua operasi database selanjutnya akan dilakukan di dalam database tersebut, kecuali jika secara eksplisit beralih ke database lain.

2b.



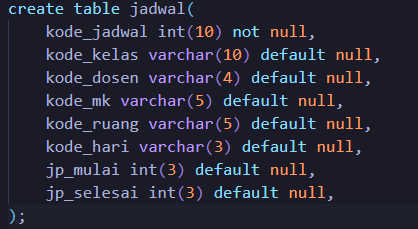
Query tersebut merupakan perintah SQL untuk membuat sebuah tabel bernama "dosen" dengan dua kolom: "kode\_dosen" dengan tipe data VARCHAR dengan panjang maksimum 4 karakter. Kata kunci **NOT NULL** menandakan bahwa kolom ini tidak dapat memiliki nilai kosong (NULL), sehingga setiap baris yang dimasukkan ke dalam tabel harus memiliki nilai untuk kolom ini. dan "nama\_dosen" dengan tipe data VARCHAR dengan panjang maksimum 100 karakter. Kata kunci DEFAULT NULL menandakan bahwa kolom ini dapat memiliki nilai kosong (NULL) secara default jika tidak diberikan nilai saat memasukkan data baru ke dalam tabel.

2c.



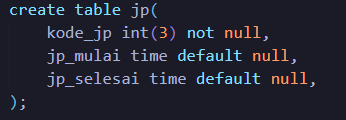
Query tersebut membuat tabel "hari" dengan dua kolom: "kode\_hari" yang harus diisi (NOT NULL) dan "nama\_hari" yang bisa kosong (NULL) dengan panjang maksimum 10 karakter.

2d.



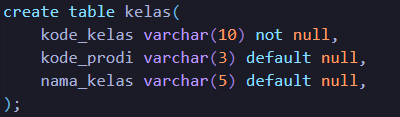
Query tersebut adalah perintah SQL untuk membuat sebuah tabel bernama "jadwal" yang digunakan untuk menyimpan informasi tentang jadwal perkuliahan. Tabel ini memiliki delapan kolom, yaitu "kode\_jadwal", "kode\_kelas", "kode\_dosen", "kode\_mk", "kode\_ruang", "kode\_hari", "jp\_mulai", dan "jp\_selesai". Kolom "kode\_jadwal" adalah kunci utama (primary key) yang bertipe data integer dengan panjang maksimum 10 digit dan harus diisi (NOT NULL). Kolom-kolom lainnya, seperti "kode\_kelas", "kode\_dosen", "kode\_mk", "kode\_ruang", dan "kode\_hari", memiliki tipe data varchar yang memungkinkan nilai kosong (NULL) secara default. Sementara itu, kolom "jp\_mulai" dan "jp\_selesai" adalah kolom integer yang mewakili jam pelajaran dimulai dan selesai dengan panjang maksimum 3 digit dan dapat memiliki nilai kosong (NULL) secara default.

2e.



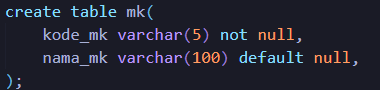
Query tersebut adalah perintah SQL untuk membuat sebuah tabel bernama "jp" yang digunakan untuk menyimpan informasi tentang jadwal perkuliahan. Tabel ini memiliki tiga kolom: "kode\_jp", "jp\_mulai", dan "jp\_selesai". Kolom "kode\_jp" adalah kunci utama (primary key) yang bertipe data integer dengan panjang maksimum 3 digit dan harus diisi (NOT NULL). Kolom "jp\_mulai" dan "jp\_selesai" adalah kolom waktu (time) yang mewakili jam mulai dan jam selesai dari jadwal perkuliahan. Keduanya memiliki default NULL, yang berarti mereka dapat memiliki nilai kosong jika tidak disediakan nilai saat memasukkan data baru ke dalam tabel.

2f.



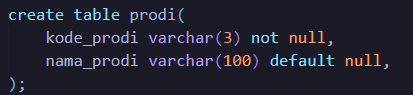
Query tersebut merupakan perintah SQL untuk membuat tabel "kelas" yang akan digunakan untuk menyimpan informasi tentang kelas-kelas dalam suatu program studi. Tabel ini terdiri dari tiga kolom: "kode\_kelas", "kode\_prodi", dan "nama\_kelas". Kolom "kode\_kelas" didefinisikan sebagai VARCHAR dengan panjang maksimum 10 karakter dan harus diisi (NOT NULL), dimana ini mungkin akan menjadi kunci utama untuk tabel ini. Kolom "kode\_prodi" adalah kode program studi yang merupakan referensi ke program studi mana kelas tersebut terkait, dengan default NULL, sehingga tidak harus diisi saat memasukkan data baru. Sedangkan kolom "nama\_kelas" adalah nama yang diberikan untuk kelas tersebut, dengan default NULL juga.

2g.

****

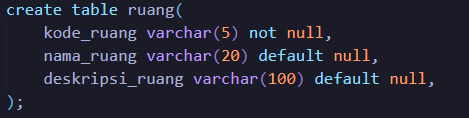
Query tersebut adalah perintah SQL untuk membuat sebuah tabel dengan nama "mk", yang bertujuan untuk menyimpan informasi mengenai mata kuliah. Tabel ini terdiri dari dua kolom: "kode\_mk" dan "nama\_mk". Kolom "kode\_mk" didefinisikan sebagai VARCHAR dengan panjang maksimum 5 karakter dan harus diisi (NOT NULL), yang kemungkinan akan menjadi kunci utama (primary key) dari tabel ini. Sedangkan kolom "nama\_mk" adalah nama lengkap dari mata kuliah tersebut, dengan default NULL, yang berarti dapat memiliki nilai kosong jika tidak disediakan saat memasukkan data baru ke dalam tabel.

2h.



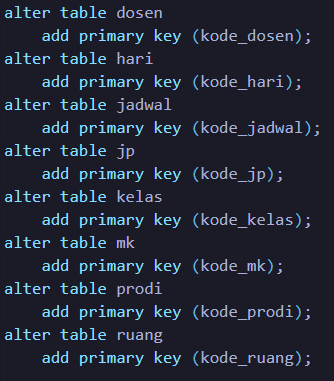
Query tersebut merupakan perintah SQL untuk membuat sebuah tabel dengan nama "prodi", yang bertujuan untuk menyimpan informasi mengenai program studi. Tabel ini terdiri dari dua kolom: "kode\_prodi" dan "nama\_prodi". Kolom "kode\_prodi" didefinisikan sebagai VARCHAR dengan panjang maksimum 3 karakter dan harus diisi (NOT NULL), yang kemungkinan akan menjadi kunci utama (primary key) dari tabel ini. Sedangkan kolom "nama\_prodi" adalah nama lengkap dari program studi tersebut, dengan default NULL, yang berarti dapat memiliki nilai kosong jika tidak disediakan saat memasukkan data baru ke dalam tabel.

2i.



Perintah SQL tersebut digunakan untuk membuat tabel "ruang" yang bertujuan untuk menyimpan informasi tentang ruang kuliah. Tabel ini terdiri dari tiga kolom: "kode\_ruang", "nama\_ruang", dan "deskripsi\_ruang". Kolom "kode\_ruang" didefinisikan sebagai VARCHAR dengan panjang maksimum 5 karakter dan harus diisi (NOT NULL), yang kemungkinan akan menjadi kunci utama (primary key) dari tabel ini. Kolom "nama\_ruang" adalah nama ruang kuliah dengan default NULL, yang berarti dapat memiliki nilai kosong jika tidak disediakan saat memasukkan data baru ke dalam tabel. Sementara itu, kolom "deskripsi\_ruang" digunakan untuk memberikan deskripsi tambahan tentang ruang kuliah tersebut, dengan default NULL juga.

2j.



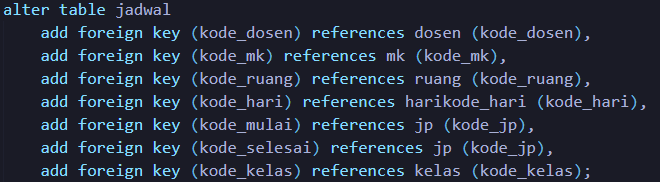
Query tersebut adalah serangkaian perintah SQL untuk menambahkan kunci utama (primary key) pada masing-masing tabel yang sudah ada dalam sebuah basis data. Setiap perintah "ALTER TABLE" diikuti dengan perintah "ADD PRIMARY KEY" yang menentukan kolom mana yang akan dijadikan kunci utama untuk tabel yang bersangkutan. Misalnya, perintah "ALTER TABLE dosen ADD PRIMARY KEY (kode\_dosen);" menambahkan kunci utama pada kolom "kode\_dosen" dalam tabel "dosen". Hal ini bertujuan untuk memastikan bahwa setiap baris dalam tabel memiliki nilai unik untuk kolom tersebut, sehingga mempermudah pengelolaan dan akses data.

2k.

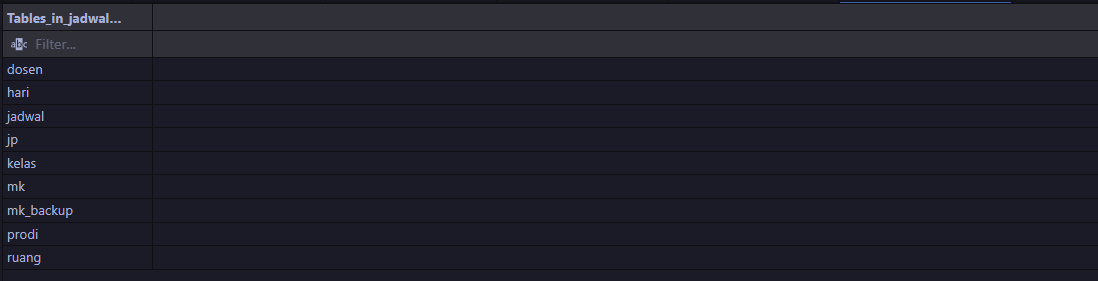


Perintah SQL tersebut merupakan sebuah perintah ALTER TABLE yang dimaksudkan untuk mengubah struktur tabel "jadwal". Dengan menggunakan klausa MODIFY, kolom "kode\_jadwal" yang sebelumnya didefinisikan sebagai integer dengan panjang maksimum 10 digit dan not null, kini juga ditambahkan dengan kata kunci AUTO\_INCREMENT. Penambahan ini membuat nilai pada kolom "kode\_jadwal" secara otomatis meningkat seiring dengan penambahan baris baru ke dalam tabel, dimulai dari nilai awal yang didefinisikan secara otomatis oleh sistem database. Dengan demikian, setiap kali sebuah baris baru dimasukkan ke dalam tabel "jadwal", nilai pada kolom "kode\_jadwal" akan secara otomatis ditetapkan oleh sistem.

2i



Perintah SQL tersebut adalah perintah ALTER TABLE yang ditujukan untuk menambahkan kunci asing (foreign key) ke dalam tabel "jadwal". Setiap perintah "ADD FOREIGN KEY" diikuti oleh spesifikasi kolom yang akan dijadikan kunci asing, diikuti oleh klausa "REFERENCES" yang menunjukkan tabel dan kolom yang menjadi acuan kunci asing tersebut. Misalnya, "ADD FOREIGN KEY (kode\_dosen) REFERENCES dosen (kode\_dosen)" menetapkan kunci asing pada kolom "kode\_dosen" dalam tabel "jadwal", yang merujuk ke kolom "kode\_dosen" dalam tabel "dosen". Proses ini memastikan bahwa setiap nilai dalam kolom "kode\_dosen" di tabel "jadwal" harus ada dalam kolom "kode\_dosen" di tabel "dosen", memastikan integritas referensial antar data. Selain itu, kunci asing juga ditambahkan untuk kolom "kode\_mk", "kode\_ruang", "kode\_hari", "kode\_mulai", "kode\_selesai", dan "kode\_kelas", yang merujuk ke tabel-tabel terkait untuk memperkuat hubungan antar data yang terdapat dalam tabel "jadwal".

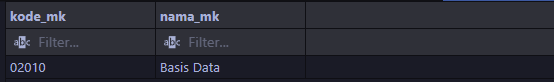


**Praktikum – Bagian 2: Percobaan Statement INSERT**

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



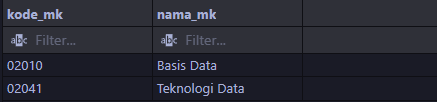
Untuk melihat hasil SQL yang kita eksekusi tersebut, gunakan statement **SELECT** seperti berikut.



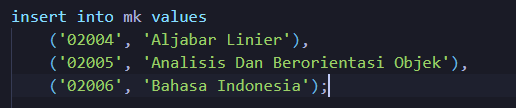
1. Apabila data di-insert-kan pada semua kolom tabel, maka kita dapat langsung menggunakan klausa VALUES tanpa harus menuliskan nama-nama kolom dahulu.



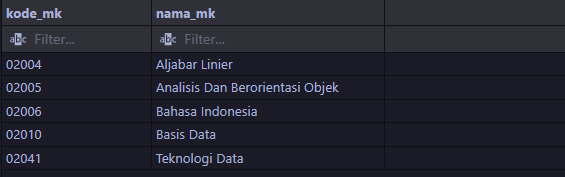
Untuk melihat hasil SQL yang kita eksekusi tersebut, gunakan statement **SELECT** seperti berikut.



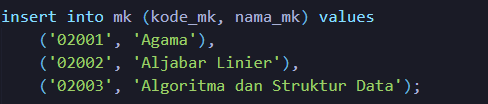
1. Untuk menambahkan beberapa kolom sekaligus dalam 1 statement digunakan statement dengan format seperti berikut.



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



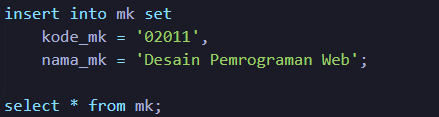
1. Dan seperti berikut, jika hanya kolom tertentu saja yang akan diberi nilai dengan cara menyebutkan nama kolomnya.



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



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

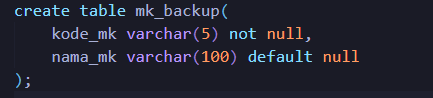


Untuk melihat hasil SQL yang kita eksekusi tersebut, gunakan statement **SELECT** seperti berikut.

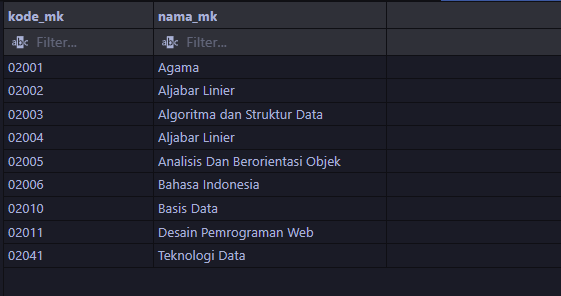


1. Pada statement INSERT juga dapat digunakan klausa SELECT.

Misalnya kita ingin menyalin semua baris pada tabel mk ke tabel mk\_backup, maka kita SQL berikut dapat digunakan. (Buat terlebih dahulu tabel “mk\_backup” dengan struktur tabel yang sama dengan tabel “mk”)





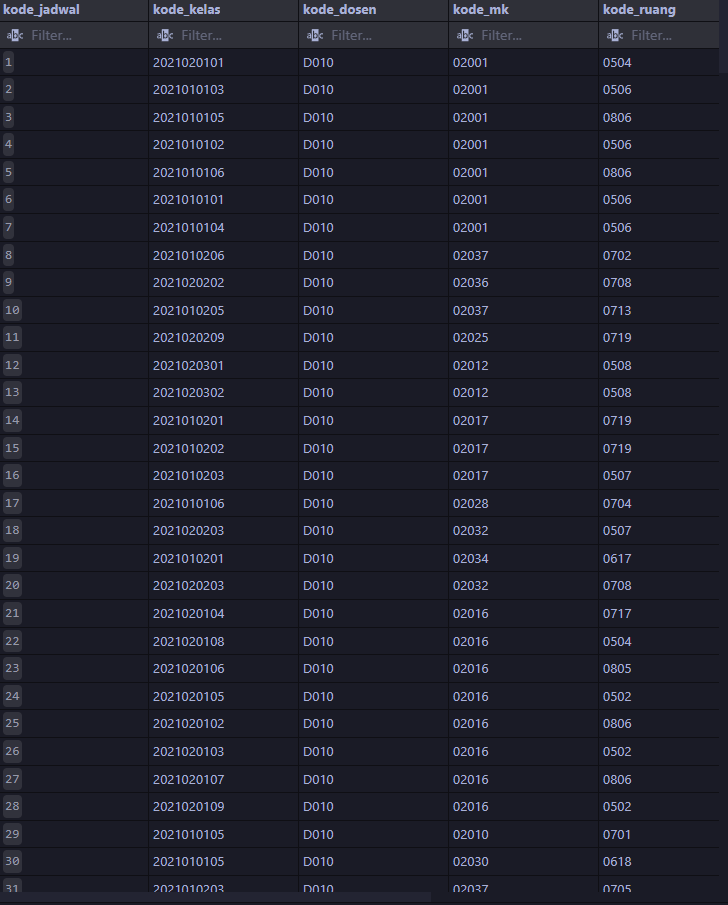


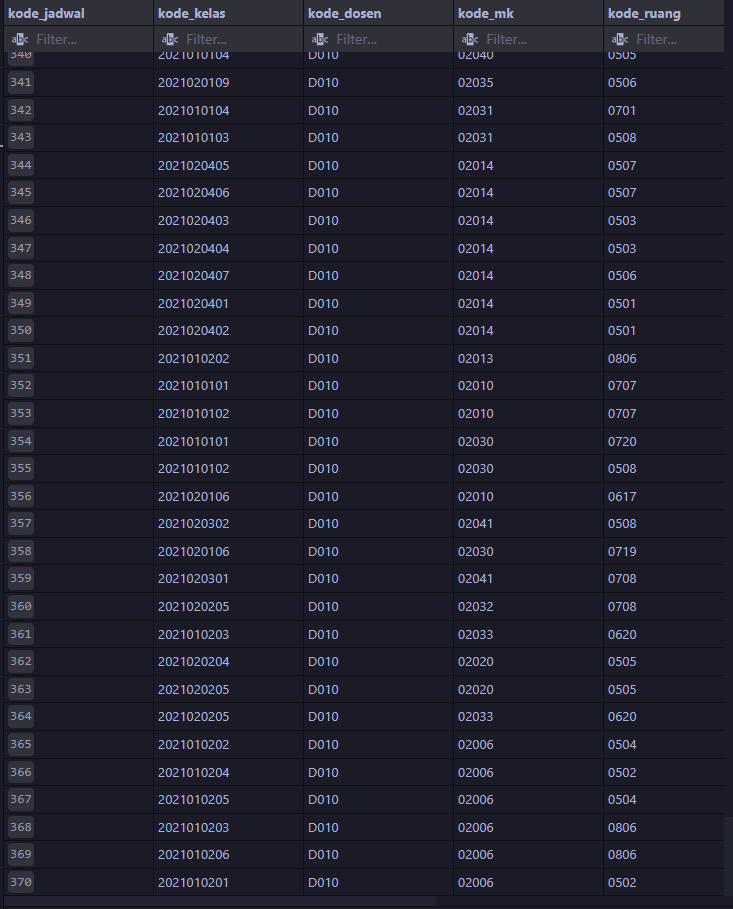
**Praktikum - Bagian 3: Percobaan Statement UPDATE**

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. Format dasar statement Update ini adalah sebagai berikut:

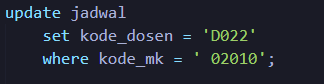


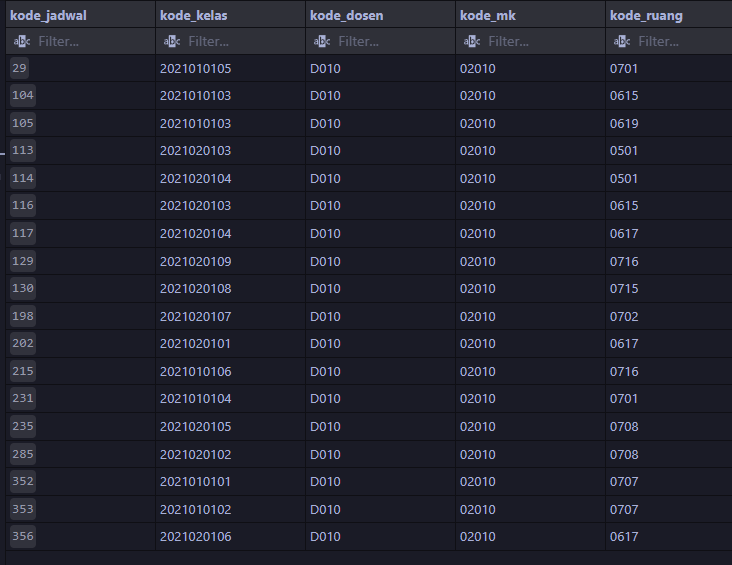
Tampilkan isi data menggunakan statement SELECT berikut:



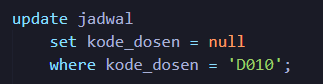
****

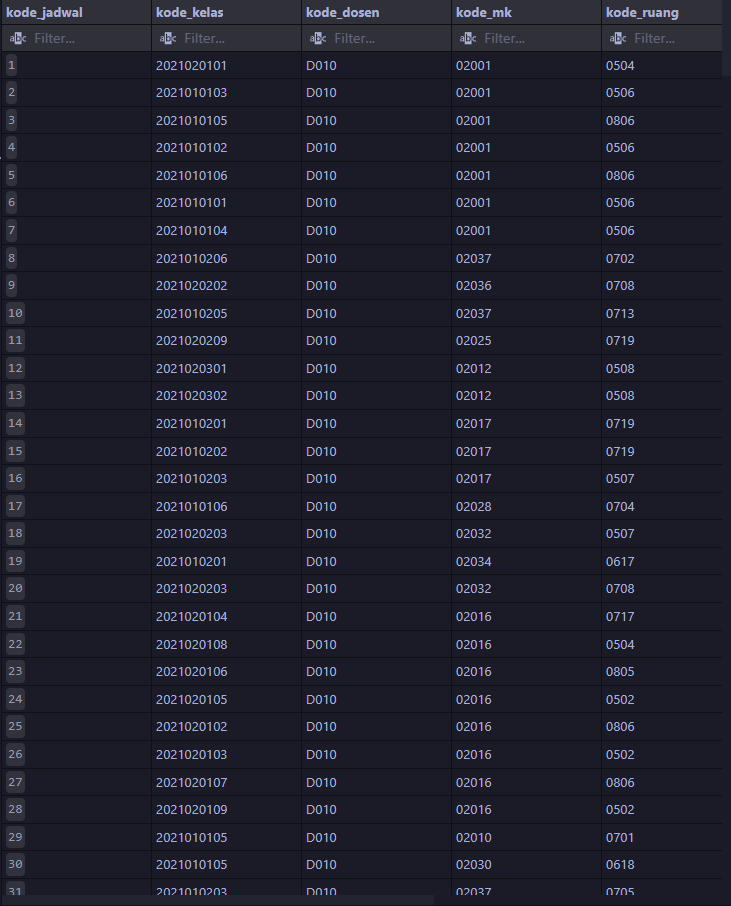
1. Untuk mengubah nilai pada baris tertentu saja, kita tambahkan klausa WHERE pada statement UPDATE. Misalkan kita akan menjadikan dosen dengan kode\_dosen D022 sebagai pengampu mata kuliah dengan kode\_mk 02010, maka dapat digunakan SQL sebagai berikut:



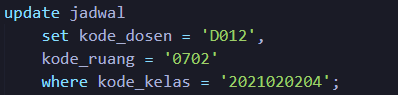
****

1. 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.





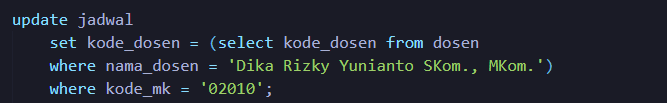
1. Untuk mengubah beberapa kolom sekaligus dalam satu kali eksekusi statement UPDATE, dapat digunakan format berikut.



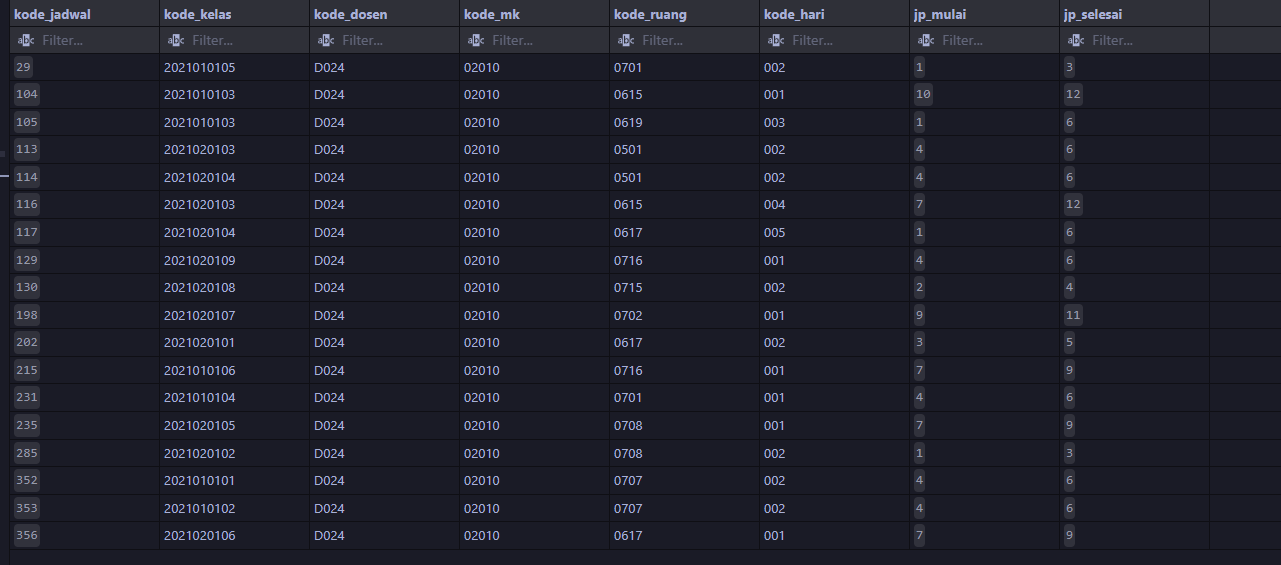
****

1. 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.’, maka dapat digunakan SQL dengan format berikut.



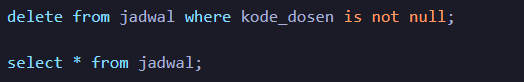


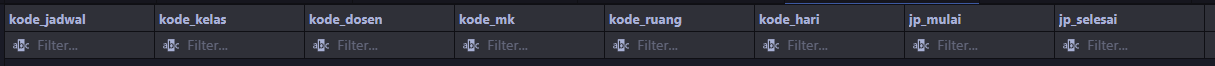


**Praktikum - Bagian 4: Percobaan Statement DELETE**

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:





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



1. Semua data dalam satu tabel jadwal akan hilang!



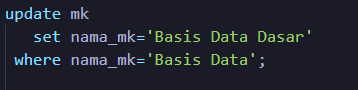
**Tugas**

* + Import kembali **isi\_data\_jadwal\_perkuliahan.sql**.
  + Screenshot sintaks dan hasil SELECT dari setiap soal dibawah ini!

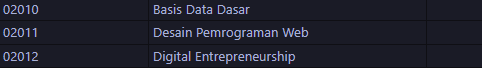
1. Ubah nama mata kuliah “Basis Data” menjadi “Basis Data Dasar”!
2. Ubah semua jadwal kuliah mata kuliah “Basis Data Dasar” menjadi hari Senin di jam pelajaran ke 5 sampai dengan jam pelajaran ke 10!
3. Hapus jadwal perkuliahan “Kewarganegaraan” pada tabel jadwal!
4. Tambahkan mata kuliah “Pancasila”, “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 !

Jawaban!

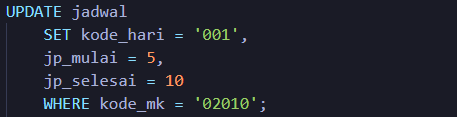
1. Ubah nama mata kuliah “Basis Data” menjadi “Basis Data Dasar”!



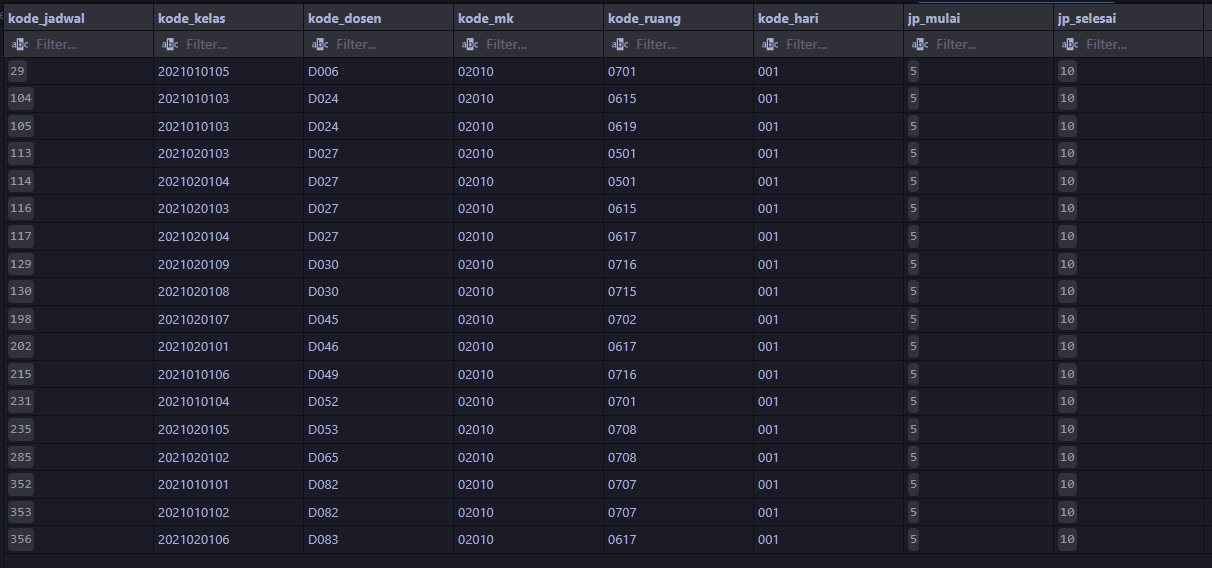
Output:



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



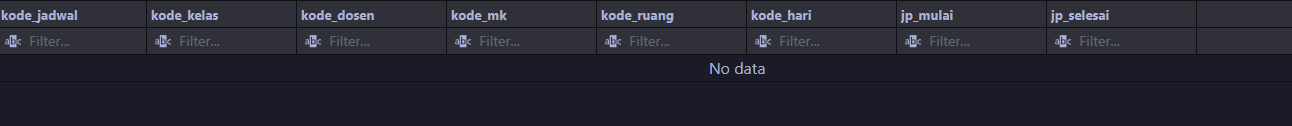




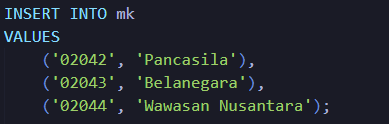
1. Hapus jadwal perkuliahan “Kewarganegaraan” pada tabel jadwal!



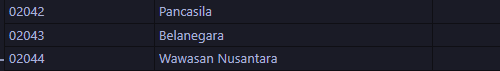




1. Tambahkan mata kuliah “Pancasila”, “Bela Negara”, “Wawasan Nusantara” pada tabel mk!



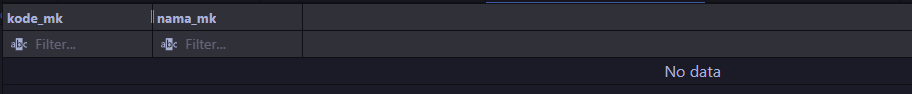




1. Hapus semua isi data pada tabel mk\_backup!



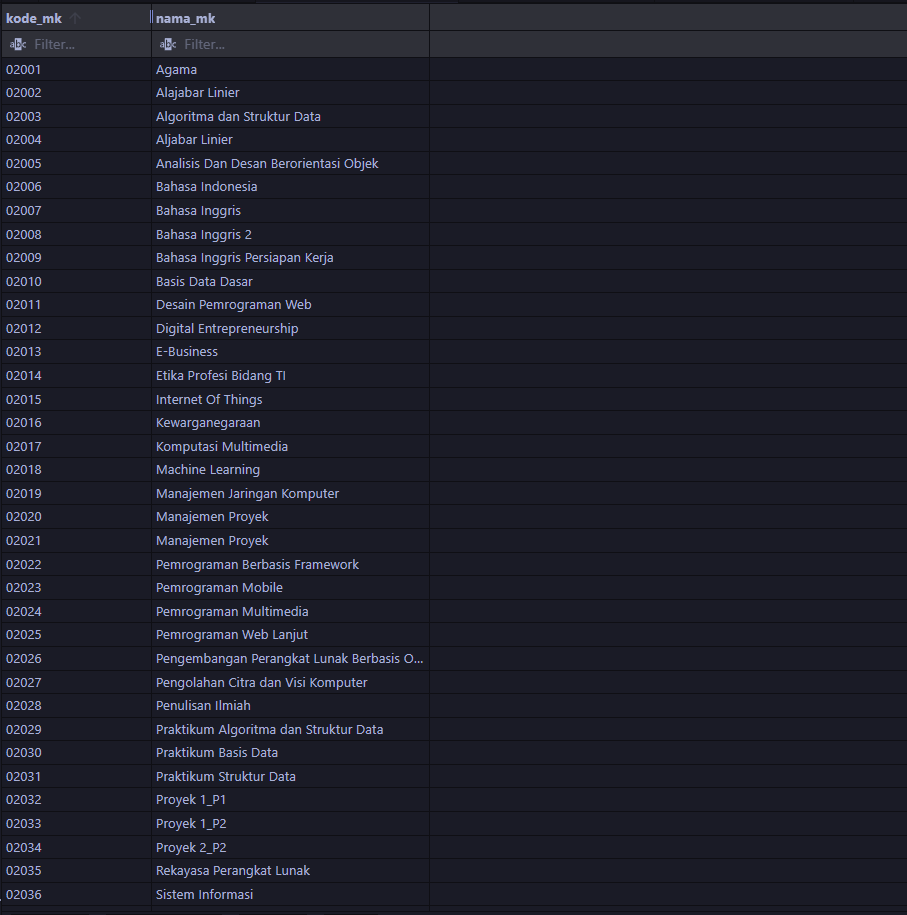




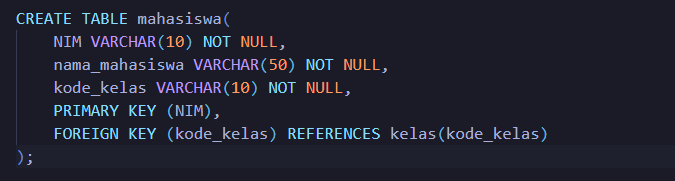
1. Isi data tabel mk\_backup dengan isi dari tabel mk!



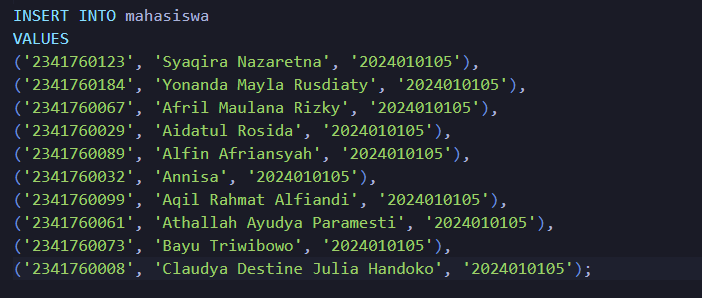




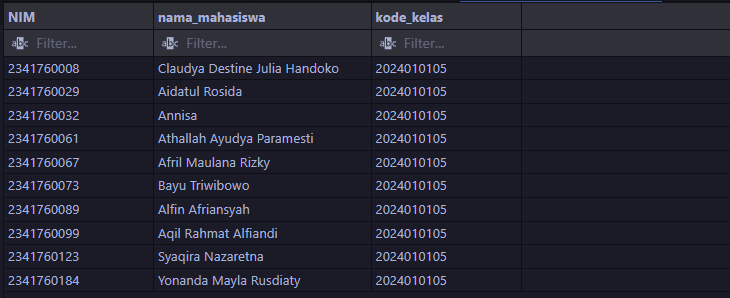
1. Membuat Tabel mahasiswa



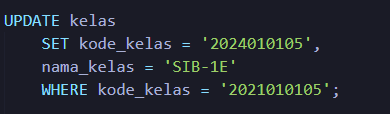
Mengisi tabel dengan 10 nama mahasiswa yang memiliki nomor presensi berturt-turut setelah saya di kelas.







Update kode kelas sesuai kode kelas.



Mengubah kode\_dosen mata kuliah “Basis Data Dasar” kelas anda pada tabel jadwal sesuai dengan dosen pengampu mata kuliah basis data anda saat ini.

