

LAPORAN PRAKTIKUM

Basis Data

Jobsheet-11: Data Manipulation Language



Nama : Ghoffar Abdul Ja'far

NIM : 2341720035

Kelas : 1E

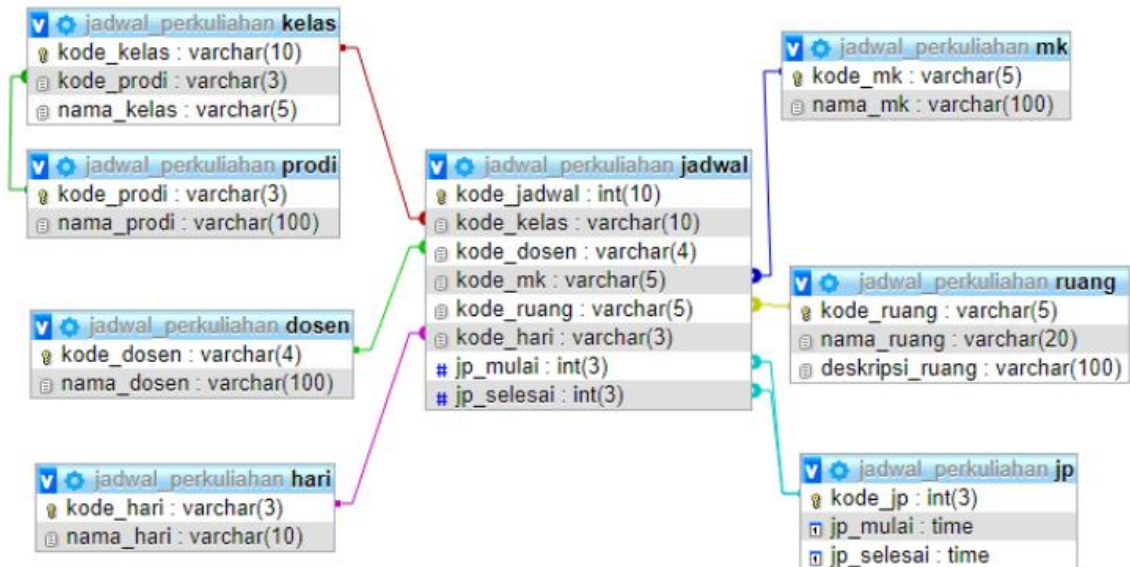
JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

2023/2024

Praktikum – Bagian 1: Membuat Database untuk Percobaan

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



2. **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.

```
MariaDB [(none)]> DROP DATABASE IF EXISTS jadwal_perkuliahan;  
Query OK, 0 rows affected, 1 warning (0.001 sec)
```

3. **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.

```
MariaDB [(none)]> create database jadwal_perkuliahan;  
Query OK, 1 row affected (0.002 sec)
```

4. **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.

```
MariaDB [(none)]> USE jadwal_perkuliahan;  
Database changed  
MariaDB [jadwal_perkuliahan]> _
```

5. **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.

```
MariaDB [jadwal_perkuliahan]> CREATE TABLE dosen(  
  -> kode_dosen varchar(4) NOT NULL,  
  -> nama_dosen varchar(100) DEFAULT NULL);  
Query OK, 0 rows affected (0.022 sec)
```

6. 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.

```
MariaDB [jadwal_perkuliahan]> CREATE TABLE hari(  
  -> kode_hari varchar(4) NOT NULL,  
  -> nama_hari varchar(10) DEFAULT NULL);  
Query OK, 0 rows affected (0.010 sec)
```

7. 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.

```

MariaDB [jadwal_perkuliahahan]> CREATE TABLE jadwal(
-> kode_jadwal int(10) NOT 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,
-> kode_hari varchar(3) DEFAULT NULL,
-> jp_mulai int(3) DEFAULT NULL,
-> jp_selesai int(3) DEFAULT NULL);
Query OK, 0 rows affected (0.010 sec)

```

8. 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.

```

MariaDB [jadwal_perkuliahahan]> CREATE TABLE jp(
-> kode_jp int(3) NOT NULL,
-> jp_mulai time DEFAULT NULL,
-> jp_selesai time DEFAULT NULL);
Query OK, 0 rows affected (0.008 sec)

```

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.

```

MariaDB [jadwal_perkuliahahan]> CREATE TABLE kelas(
-> kode_kelas varchar(10) NOT NULL,
-> kode_prodi varchar(3) DEFAULT NULL,
-> nama_kelas varchar(5) DEFAULT NULL);
Query OK, 0 rows affected (0.016 sec)

```

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.

```
MariaDB [jadwal_perkuliahan]> CREATE TABLE mk (  
-> kode_mk varchar(5) NOT NULL,  
-> nama_mk varchar(100) DEFAULT NULL);  
Query OK, 0 rows affected (0.011 sec)
```

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.

```
MariaDB [jadwal_perkuliahan]> CREATE TABLE prodi (  
-> kode_prodi varchar(3) NOT NULL,  
-> nama_prodi varchar(100) DEFAULT NULL);  
Query OK, 0 rows affected (0.010 sec)
```

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.

```
MariaDB [jadwal_perkuliah] > CREATE TABLE ruang (  
  -> kode_ruang varchar(5) NOT NULL,  
  -> nama_ruang varchar(20) DEFAULT NULL,  
  -> deskripsi_ruang varchar(100) DEFAULT NULL);  
Query OK, 0 rows affected (0.011 sec)
```

13. Penjelasan secara berurutan, 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.

```

MariaDB [jadwal_perkuliahan]> ALTER TABLE dosen ADD PRIMARY KEY(kode_dosen);
Query OK, 0 rows affected (0.024 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [jadwal_perkuliahan]> ALTER TABLE hari ADD PRIMARY KEY(kode_hari);
Query OK, 0 rows affected (0.032 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [jadwal_perkuliahan]> ALTER TABLE jadwal ADD PRIMARY KEY(kode_jadwal);
Query OK, 0 rows affected (0.025 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [jadwal_perkuliahan]> ALTER TABLE jp ADD PRIMARY KEY(kode_jp);
Query OK, 0 rows affected (0.033 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [jadwal_perkuliahan]> ALTER TABLE kelas ADD PRIMARY KEY(kode_kelas);
Query OK, 0 rows affected (0.026 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [jadwal_perkuliahan]> ALTER TABLE mk ADD PRIMARY KEY(kode_mk);
Query OK, 0 rows affected (0.019 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [jadwal_perkuliahan]> ALTER TABLE prodi ADD PRIMARY KEY(kode_prodi);
Query OK, 0 rows affected (0.022 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [jadwal_perkuliahan]> ALTER TABLE ruang ADD PRIMARY KEY(kode_ruang);
Query OK, 0 rows affected (0.028 sec)
Records: 0 Duplicates: 0 Warnings: 0

```

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

```

MariaDB [jadwal_perkuliahan]> ALTER TABLE jadwal MODIFY kode_jadwal int(10) NOT NULL AUTO_INCREMENT;
Query OK, 0 rows affected (0.036 sec)
Records: 0 Duplicates: 0 Warnings: 0

```

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.


```

MariaDB [jadwal_perkuliahan]> ALTER TABLE jadwal
  -> ADD FOREIGN KEY (kode_dosen) REFERENCES dosen(kode_dosen),
  -> ADD FOREIGN KEY (kode_mk) REFERENCES mk(kode_mk),
  -> ADD FOREIGN KEY (kode_ruang) REFERENCES ruang(kode_ruang),
  -> ADD FOREIGN KEY (kode_hari) REFERENCES hari(kode_hari),
  -> ADD FOREIGN KEY (jp_mulai) REFERENCES jp(kode_jp),
  -> ADD FOREIGN KEY (jp_selesai) REFERENCES jp(kode_jp),
  -> ADD FOREIGN KEY (kode_kelas) REFERENCES kelas(kode_kelas);
Query OK, 0 rows affected (0.060 sec)
Records: 0 Duplicates: 0 Warnings: 0

```

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.

```

MariaDB [jadwal_perkuliahan]> ALTER TABLE kelas ADD FOREIGN KEY (kode_prodi) REFERENCES prodi(kode_prodi);
Query OK, 0 rows affected (0.040 sec)
Records: 0 Duplicates: 0 Warnings: 0

```

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.

```

MariaDB [jadwal_perkuliahan]> show tables;
+-----+
| Tables_in_jadwal_perkuliahan |
+-----+
| dosen                          |
| hari                          |
| jadwal                        |
| jp                            |
| kelas                         |
| mk                            |
| prodi                         |
| ruang                         |
+-----+
8 rows in set (0.003 sec)

```

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

```

MariaDB [jadwal_perkuliahan]> insert into mk(kode_mk, nama_mk) values ('02010', 'Basis Data');
Query OK, 1 row affected (0.004 sec)

```

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


```
MariaDB [jadwal_perkuliahan]> select * from mk;
+-----+-----+
| kode_mk | nama_mk |
+-----+-----+
| 02010   | Basis Data |
+-----+-----+
1 row in set (0.001 sec)
```

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

```
MariaDB [jadwal_perkuliahan]> insert into mk values ('02041', 'Teknologi Data');
Query OK, 1 row affected (0.003 sec)
```

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

```
MariaDB [jadwal_perkuliahan]> select * from mk;
+-----+-----+
| kode_mk | nama_mk |
+-----+-----+
| 02010   | Basis Data |
| 02041   | Teknologi Data |
+-----+-----+
2 rows in set (0.001 sec)
```

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

```
MariaDB [jadwal_perkuliahan]> insert into mk values
-> ('02004', 'Aljabar Linier'),
-> ('02005', 'Analisi Dan Desan Berorientasi Objek'),
-> ('02006', 'Bahasa Indonesia');
Query OK, 3 rows affected (0.003 sec)
Records: 3 Duplicates: 0 Warnings: 0
```

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

```
MariaDB [jadwal_perkuliahan]> select * from mk;
+-----+-----+
| kode_mk | nama_mk |
+-----+-----+
| 02004   | Aljabar Linier |
| 02005   | Analisi Dan Desan Berorientasi Objek |
| 02006   | Bahasa Indonesia |
| 02010   | Basis Data |
| 02041   | Teknologi Data |
+-----+-----+
5 rows in set (0.001 sec)
```

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

```
MariaDB [jadwal_perkuliahan]> insert into mk values
-> ('02001', 'Agama'),
-> ('02002', 'Aljabar Linier'),
-> ('02003', 'Algoritma dan Struktur Data');
Query OK, 3 rows affected (0.002 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

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

```
MariaDB [jadwal_perkuliahan]> select * from mk;
+-----+-----+
| kode_mk | nama_mk |
+-----+-----+
| 02001   | Agama   |
| 02002   | Aljabar Linier |
| 02003   | Algoritma dan Struktur Data |
| 02004   | Aljabar Linier |
| 02005   | Analisi Dan Desan Berorientasi Objek |
| 02006   | Bahasa Indonesia |
| 02010   | Basis Data |
| 02041   | Teknologi Data |
+-----+-----+
8 rows in set (0.001 sec)
```

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

```
MariaDB [jadwal_perkuliahan]> insert into mk SET
-> kode_mk = '02011',
-> nama_mk = 'Desain Pemrograman Web';
Query OK, 1 row affected (0.002 sec)
```

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

```
MariaDB [jadwal_perkuliahan]> select * from mk;
```

kode_mk	nama_mk
02001	Agama
02002	Aljabar Linier
02003	Algoritma dan Struktur Data
02004	Aljabar Linier
02005	Analisi Dan Desan Berorientasi Objek
02006	Bahasa Indonesia
02010	Basis Data
02011	Desain Pemrograman Web
02041	Teknologi Data

```
9 rows in set (0.001 sec)
```

11. Pada statement INSERT juga dapat digunakan klausa SELECT.

```
MariaDB [jadwal_perkuliahan]> create table mk_backup(
  -> kode_mk varchar(5),
  -> nama_mk varchar(100));
Query OK, 0 rows affected (0.010 sec)

MariaDB [jadwal_perkuliahan]> insert into mk_backup select * from mk;
Query OK, 9 rows affected (0.005 sec)
Records: 9 Duplicates: 0 Warnings: 0
```

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

```
MariaDB [jadwal_perkuliahan]> select * from mk_backup;
```

kode_mk	nama_mk
02001	Agama
02002	Aljabar Linier
02003	Algoritma dan Struktur Data
02004	Aljabar Linier
02005	Analisi Dan Desan Berorientasi Objek
02006	Bahasa Indonesia
02010	Basis Data
02011	Desain Pemrograman Web
02041	Teknologi Data

```
9 rows in set (0.001 sec)
```

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.

```
MariaDB [jadwal_perkuliahan]> update jadwal SET kode_dosen = 'D010';
Query OK, 365 rows affected (0.020 sec)
Rows matched: 371 Changed: 365 Warnings: 0
```

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.

```
MariaDB [jadwal_perkuliahan]> select * from jadwal;
```

kode_jadwal	kode_kelas	kode_dosen	kode_mk	kode_ruang	kode_hari	jp_mulai	jp_selesai
1	2021020101	D010	02001	0504	001	7	9
2	2021010103	D010	02001	0506	002	9	11
3	2021010105	D010	02001	0806	003	10	12
4	2021010102	D010	02001	0506	004	1	3
5	2021010106	D010	02001	0806	004	4	6
6	2021010101	D010	02001	0506	004	7	9
7	2021010104	D010	02001	0506	005	10	12
8	2021010206	D010	02037	0702	001	7	12
9	2021020202	D010	02036	0708	003	2	4
10	2021010205	D010	02037	0713	004	1	6
11	2021020209	D010	02025	0719	004	7	12
12	2021020301	D010	02012	0508	002	1	4
13	2021020302	D010	02012	0508	002	1	4
14	2021010201	D010	02017	0719	003	2	5
15	2021010202	D010	02017	0719	003	2	5
16	2021010203	D010	02017	0507	005	2	5
17	2021010106	D010	02028	0704	001	1	3
18	2021020203	D010	02032	0507	001	10	12
19	2021010201	D010	02034	0617	002	7	12
20	2021020203	D010	02032	0708	004	1	6
21	2021020104	D010	02016	0717	001	1	3
22	2021020108	D010	02016	0504	001	4	6
23	2021020106	D010	02016	0805	001	10	12

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.

```
MariaDB [jadwal_perkuliahan]> update jadwal SET kode_dosen = 'D022' WHERE kode_mk = '02010';
Query OK, 18 rows affected (0.003 sec)
Rows matched: 18 Changed: 18 Warnings: 0
```

kode_jadwal	kode_kelas	kode_dosen	kode_mk	kode_ruang	kode_hari	jp_mulai	jp_selesai
28	2021020109	D010	02010	0502	002	1	3
29	2021010105	D022	02010	0701	002	1	3
30	2021010105	D010	02030	0618	004	1	6

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.

```
MariaDB [jadwal_perkuliahan]> update jadwal SET kode_dosen = NULL WHERE kode_dosen = 'D010';
Query OK, 353 rows affected (0.014 sec)
Rows matched: 353 Changed: 353 Warnings: 0
```

```
MariaDB [jadwal_perkuliahan]> select * from jadwal;
```

kode_jadwal	kode_kelas	kode_dosen	kode_mk	kode_ruang	kode_hari	jp_mulai	jp_selesai
1	2021020101	NULL	02001	0504	001	7	9
2	2021010103	NULL	02001	0506	002	9	11
3	2021010105	NULL	02001	0806	003	10	12
4	2021010102	NULL	02001	0506	004	1	3
5	2021010106	NULL	02001	0806	004	4	6
6	2021010101	NULL	02001	0506	004	7	9
7	2021010104	NULL	02001	0506	005	10	12
8	2021010206	NULL	02037	0702	001	7	12
9	2021020202	NULL	02036	0708	003	2	4
10	2021010205	NULL	02037	0713	004	1	6
11	2021020209	NULL	02025	0719	004	7	12
12	2021020301	NULL	02012	0508	002	1	4
13	2021020302	NULL	02012	0508	002	1	4
14	2021010201	NULL	02017	0719	003	2	5
15	2021010202	NULL	02017	0719	003	2	5
16	2021010203	NULL	02017	0507	005	2	5
17	2021010106	NULL	02028	0704	001	1	3
18	2021020203	NULL	02032	0507	001	10	12
19	2021010201	NULL	02034	0617	002	7	12
20	2021020203	NULL	02032	0708	004	1	6
21	2021020104	NULL	02016	0717	001	1	3
22	2021020108	NULL	02016	0504	001	4	6
23	2021020106	NULL	02016	0805	001	10	12

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

```
MariaDB [jadwal_perkuliahan]> UPDATE jadwal
-> SET kode_dosen = 'D012',
-> kode_ruang = '0702'
-> WHERE kode_kelas = '2021020204';
Query OK, 8 rows affected (0.004 sec)
Rows matched: 8 Changed: 8 Warnings: 0
```

```
MariaDB [jadwal_perkuliahan]> SELECT * FROM jadwal WHERE kode_kelas = '2021020204';
```

kode_jadwal	kode_kelas	kode_dosen	kode_mk	kode_ruang	kode_hari	jp_mulai	jp_selesai
46	2021020204	D012	02005	0702	003	7	12
57	2021020204	D012	02018	0702	001	1	5
76	2021020204	D012	02032	0702	001	7	9
79	2021020204	D012	02033	0702	005	7	12
124	2021020204	D012	02036	0702	003	1	3
183	2021020204	D012	02037	0702	004	7	12
216	2021020204	D012	02025	0702	002	1	6
362	2021020204	D012	02020	0702	004	1	3

8 rows in set (0.002 sec)

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 Yudianto SKom., MKom.'

```
MariaDB [jadwal_perkuliahan]> UPDATE jadwal
-> SET kode_dosen = (SELECT kode_dosen FROM dosen WHERE nama_dosen = 'Dika Rizky Yudianto SKom., MKom.') WHERE kode_mk = '02010';
Query OK, 18 rows affected (0.004 sec)
Rows matched: 18 Changed: 18 Warnings: 0
```

```
MariaDB [jadwal_perkuliahahan]> SELECT * FROM jadwal WHERE kode_mk = '02010';
```

kode_jadwal	kode_kelas	kode_dosen	kode_mk	kode_ruang	kode_hari	jp_mulai	jp_selesai
29	2021010105	D024	02010	0701	002	1	3
104	2021010103	D024	02010	0615	001	10	12
105	2021010103	D024	02010	0619	003	1	6
113	2021020103	D024	02010	0501	002	4	6
114	2021020104	D024	02010	0501	002	4	6
116	2021020103	D024	02010	0615	004	7	12
117	2021020104	D024	02010	0617	005	1	6
129	2021020109	D024	02010	0716	001	4	6
130	2021020108	D024	02010	0715	002	2	4
198	2021020107	D024	02010	0702	001	9	11
202	2021020101	D024	02010	0617	002	3	5
215	2021010106	D024	02010	0716	001	7	9
231	2021010104	D024	02010	0701	001	4	6
235	2021020105	D024	02010	0708	001	7	9
285	2021020102	D024	02010	0708	002	1	3
352	2021010101	D024	02010	0707	002	4	6
353	2021010102	D024	02010	0707	002	4	6
356	2021020106	D024	02010	0617	001	7	9

```
18 rows in set (0.001 sec)
```

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:

```
MariaDB [jadwal_perkuliahahan]> DELETE FROM jadwal WHERE kode_dosen IS NOT NULL;
Query OK, 26 rows affected (0.004 sec)
```

```
MariaDB [jadwal_perkuliahahan]> SELECT * FROM jadwal;
```

kode_jadwal	kode_kelas	kode_dosen	kode_mk	kode_ruang	kode_hari	jp_mulai	jp_selesai
1	2021020101	NULL	02001	0504	001	7	9
2	2021010103	NULL	02001	0506	002	9	11
3	2021010105	NULL	02001	0806	003	10	12
4	2021010102	NULL	02001	0506	004	1	3
5	2021010106	NULL	02001	0806	004	4	6
6	2021010101	NULL	02001	0506	004	7	9
7	2021010104	NULL	02001	0506	005	10	12
8	2021010206	NULL	02037	0702	001	7	12
9	2021020202	NULL	02036	0708	003	2	4
10	2021010205	NULL	02037	0713	004	1	6
11	2021020209	NULL	02025	0719	004	7	12
12	2021020301	NULL	02012	0508	002	1	4
13	2021020302	NULL	02012	0508	002	1	4
14	2021010201	NULL	02017	0719	003	2	5
15	2021010202	NULL	02017	0719	003	2	5
16	2021010203	NULL	02017	0507	005	2	5
17	2021010106	NULL	02028	0704	001	1	3
18	2021020203	NULL	02032	0507	001	10	12
19	2021010201	NULL	02034	0617	002	7	12
20	2021020203	NULL	02032	0708	004	1	6
21	2021020104	NULL	02016	0717	001	1	3
22	2021020108	NULL	02016	0504	001	4	6
23	2021020106	NULL	02016	0805	001	10	12
24	2021020105	NULL	02016	0502	002	1	3

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


```
MariaDB [jadwal_perkuliahan]> DELETE FROM jadwal;  
Query OK, 345 rows affected (0.005 sec)
```

3. Semua data dalam satu tabel jadwal akan hilang!

```
MariaDB [jadwal_perkuliahan]> SELECT * FROM jadwal;  
Empty set (0.001 sec)
```

Tugas

- Import kembali isi_data_jadwal_perkuliahan.sql.

File to import:

File may be compressed (gzip, bzip2) or uncompressed.
A compressed file's name must end in **[format].[compression]**. Example: **.sql.zip**

Browse your computer: (Max: 40MiB)

Choose File

isi_data_jadwal_perkuliahan.sql

You may also drag and drop a file on any page.

Character set of the file:

utf-8

- Screenshot sintaks dan hasil SELECT dari setiap soal dibawah ini!
1. Ubah nama mata kuliah "Basis Data" menjadi "Basis Data Dasar"!

```
MariaDB [jadwal_perkuliahan]> UPDATE mk  
-> SET nama_mk='Basis Data Dasar'  
-> WHERE kode_mk='02010';  
Query OK, 1 row affected (0.003 sec)  
Rows matched: 1 Changed: 1 Warnings: 0
```

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


```

MariaDB [jadwal_perkuliahan]> UPDATE jadwal
-> SET kode_hari = '001'
-> WHERE kode_mk = '02010';
Query OK, 11 rows affected (0.003 sec)
Rows matched: 18  Changed: 11  Warnings: 0

MariaDB [jadwal_perkuliahan]> UPDATE jadwal
-> SET jp_mulai = '5'
-> WHERE kode_mk = '02010';
Query OK, 18 rows affected (0.004 sec)
Rows matched: 18  Changed: 18  Warnings: 0

MariaDB [jadwal_perkuliahan]> UPDATE jadwal
-> SET jp_selesai = '10'
-> WHERE kode_mk = '02010';
Query OK, 18 rows affected (0.004 sec)
Rows matched: 18  Changed: 18  Warnings: 0

```

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

```

MariaDB [jadwal_perkuliahan]> DELETE FROM jadwal WHERE kode_mk = '02016';
Query OK, 15 rows affected (0.003 sec)

```

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

```

MariaDB [jadwal_perkuliahan]> INSERT INTO mk VALUES
-> ('02042', 'Pancasila'),
-> ('02043', 'Bela Negara'),
-> ('02044', 'Wawasan Nusantara');
Query OK, 3 rows affected (0.002 sec)
Records: 3  Duplicates: 0  Warnings: 0

```

5. Hapus semua isi data pada tabel mk_backup!

```

MariaDB [jadwal_perkuliahan]> DELETE FROM mk_backup;
Query OK, 9 rows affected (0.003 sec)

```

6. Isi data tabel mk_backup dengan isi dari tabel mk!

```

MariaDB [jadwal_perkuliahan]> insert into mk_backup select * from mk;
Query OK, 44 rows affected (0.008 sec)
Records: 44  Duplicates: 0  Warnings: 0

```

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 berturut-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:

```
MariaDB [jadwal_perkuliahan]> CREATE TABLE mahasiswa (nim char(10) PRIMARY KEY, nama_mahasiswa varchar(50), kode_kelas varchar(10));
Query OK, 0 rows affected (0.008 sec)

MariaDB [jadwal_perkuliahan]> ALTER TABLE mahasiswa ADD FOREIGN KEY (kode_kelas) REFERENCES kelas(kode_kelas);
Query OK, 0 rows affected (0.059 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [jadwal_perkuliahan]>
MariaDB [jadwal_perkuliahan]> INSERT INTO mahasiswa VALUES
  -> ('2341720057', 'KHOIROTUN NISA', '2021020105'),
  -> ('2341720160', 'MAULANA RENGGA RAMADAN', '2021020105'),
  -> ('2341720013', 'MUHAMAD SYAIFULLAH', '2021020105'),
  -> ('2341720074', 'MUHAMMAD AMMAR HAFIZH', '2021020105'),
  -> ('2341720133', 'MUHAMMAD DHIA RAMADHANIL HAQ', '2021020105'),
  -> ('2341720075', 'MUHAMMAD NAUFAL PRATOMO', '2021020105'),
  -> ('2341720167', 'NECHA SYIFA SYAFITRI', '2021020105'),
  -> ('2341720082', 'NOKLENT FARDIAN ERIX', '2021020105'),
  -> ('2341720180', 'RAFI ODY PRASETYO', '2021020105'),
  -> ('2341720164', 'RAUDHIL FIRDAUS NAUFAL', '2021020105');
Query OK, 10 rows affected (0.004 sec)
Records: 10 Duplicates: 0 Warnings: 0
```

```
MariaDB [jadwal_perkuliahan]> select * from mahasiswa;
```

nim	nama_mahasiswa	kode_kelas
2341720013	MUHAMAD SYAIFULLAH	2021020105
2341720057	KHOIROTUN NISA	2021020105
2341720074	MUHAMMAD AMMAR HAFIZH	2021020105
2341720075	MUHAMMAD NAUFAL PRATOMO	2021020105
2341720082	NOKLENT FARDIAN ERIX	2021020105
2341720133	MUHAMMAD DHIA RAMADHANIL HAQ	2021020105
2341720160	MAULANA RENGGA RAMADAN	2021020105
2341720164	RAUDHIL FIRDAUS NAUFAL	2021020105
2341720167	NECHA SYIFA SYAFITRI	2021020105
2341720180	RAFI ODY PRASETYO	2021020105

10 rows in set (0.001 sec)

Dosen:

```
MariaDB [jadwal_perkuliahan]> INSERT INTO dosen VALUES
  -> ('D086','Yan Watequlis Syaifuddin, ST., M.MT.');
```

```
Query OK, 1 row affected (0.002 sec)
```

```
MariaDB [jadwal_perkuliahan]>
MariaDB [jadwal_perkuliahan]> UPDATE jadwal
  -> SET kode_dosen = 'D086'
  -> WHERE (kode_kelas = '2021020105' AND kode_mk = '02010');
```

```
Query OK, 1 row affected (0.012 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

```
MariaDB [jadwal_perkuliahan]>
MariaDB [jadwal_perkuliahan]> SELECT * FROM jadwal WHERE (kode_kelas = '2021020105' AND kode_mk = '02010');
```

kode_jadwal	kode_kelas	kode_dosen	kode_mk	kode_ruang	kode_hari	jp_mulai	jp_selesai
235	2021020105	D086	02010	0708	001	5	10

1 row in set (0.002 sec)