

LAPORAN PRAKTIKUM
MATA KULIAH PRAKTIKUM BASIS DATA

Dosen Pengampu : Farid Angga Pribadi, S.Kom., M.Kom.

PERTEMUAN 11 : DATA MANIPULATION LANGUAGE (DML)



Nama : Yonanda Mayla Rusdiaty

NIM : 2341760184

Prodi : D-IV Sistem Informasi Bisnis

JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG

2024

PRAKTIKUM 1 : Membuat Database untuk Percobaan

1. Membuat database jadwal_perkuliahan

```
▷ Run | New Tab | 🔒 Active Connection  
DROP DATABASE IF EXISTS jadwal_perkuliahan;  
  
▷ Run | New Tab  
CREATE DATABASE jadwal_perkuliahan;  
  
▷ Run | New Tab  
USE jadwal_perkuliahan;
```

2. Membuat tabel dosen

```
▷ Run | New Tab  
CREATE TABLE dosen (  
    kode_dosen VARCHAR (4) NOT NULL,  
    nama_dosen VARCHAR (100) DEFAULT NULL  
);
```

3. Membuat tabel hari

```
▷ Run | New Tab  
CREATE TABLE HARI (  
    kode_hari VARCHAR (3) NOT NULL,  
    nama_hari VARCHAR (10) DEFAULT NULL  
);
```

4. Membuat tabel jadwal

```
▷ Run | New Tab | Copy
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
);
```

5. Membuat tabel jp

```
▷ Run | New Tab | Copy
CREATE TABLE jp (
    kode_jp INT (3) NOT NULL,
    jp_mulai TIME DEFAULT NULL,
    jp_selesai TIME DEFAULT NULL
);
```

6. Membuat tabel kelas

```
▷ Run | New Tab | Copy
CREATE TABLE kelas (
    kode_kelas VARCHAR (10) NOT NULL,
    kode_prodi VARCHAR (3) DEFAULT NULL,
    nama_kelas VARCHAR (5) DEFAULT NULL
);
```

7. Membuat tabel mk

```
▷ Run | New Tab  
CREATE TABLE mk (  
    kode_mk VARCHAR (5) NOT NULL,  
    nama_mk VARCHAR (100) DEFAULT NULL  
);
```

8. Membuat tabel prodi

```
▷ Run | New Tab  
CREATE TABLE prodi (  
    kode_prodi VARCHAR (3) NOT NULL,  
    nama_prodi VARCHAR (100) DEFAULT NULL  
);
```

9. Membuat tabel ruang

```
▷ Run | New Tab | Copy  
CREATE TABLE ruang (  
    kode_ruang VARCHAR (5) NOT NULL,  
    nama_ruang VARCHAR (20) DEFAULT NULL,  
    deskripsi_ruang VARCHAR (100) DEFAULT NULL  
);
```

10. Menambahkan Primary key ke tabel dosen, hari, jadwal, jp, kelas, mk, prodi, ruang

▷ Run | New Tab

```
ALTER TABLE dosen  
ADD PRIMARY KEY (kode_dosen);
```

▷ Run | New Tab

```
ALTER TABLE hari  
ADD PRIMARY KEY (kode_hari);
```

▷ Run | New Tab

```
ALTER TABLE jadwal  
ADD PRIMARY KEY (kode_jadwal);
```

▷ Run | New Tab

```
ALTER TABLE jp  
ADD PRIMARY KEY (kode_jp);
```

▷ Run | New Tab

```
ALTER TABLE kelas  
ADD PRIMARY KEY (kode_kelas);
```

▷ Run | New Tab

```
ALTER TABLE mk  
ADD PRIMARY KEY (kode_mk);
```

▷ Run | New Tab

```
ALTER TABLE prodi  
ADD PRIMARY KEY (kode_prodi);
```

▷ Run | New Tab

```
ALTER TABLE ruang  
ADD PRIMARY KEY (kode_ruang);
```

11. Modifikasi kode_jadwal pada tabel jadwal

```
▷ Run | New Tab  
ALTER TABLE jadwal  
MODIFY kode_jadwal INT (10) NOT NULL AUTO_INCREMENT;
```

12. Menambahkan foreign key ke tabel jadwal

```
▷ Run | New Tab | Copy  
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);
```

13. Menambahkan foreign key ke tabel kelas

```
▷ Run | New Tab  
ALTER TABLE kelas  
ADD FOREIGN KEY (kode_prodi) REFERENCES prodi (kode_prodi);
```

14. Menampilkan seluruh table dalam database

Tables_in_jadwal_perkul
dosen
hari
jadwal
jp
kelas
mk
prodi
ruang

PRAKTIKUM 2 : Percobaan Statement INSERT

1. Menambahkan data menggunakan INSERT

```
Run | New Tab  
INSERT INTO mk (kode_mk, nama_mk) VALUES ('02010', 'Basis Data');
```

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

```
Run | New Tab  
INSERT INTO mk VALUES ('02041', 'Teknologi Data');  
  
Run | New Tab | JSON  
SELECT * FROM mk; 3ms
```

kode_mk varchar	nama_mk varchar
02010	Basis Data
02041	Teknologi Data

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

```
Run | New Tab  
INSERT INTO mk VALUES ('02041', 'Teknologi Data'); 13ms  
  
Run | New Tab | JSON  
SELECT * FROM mk;
```


kode_mk varchar	nama_mk varchar
02010	Basis Data
02041	Teknologi Data

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

```

Run | New Tab
INSERT INTO mk VALUES
('02004', 'Aljabar Linier'),
('02005', 'Analisis dan Desain Beroorientasi Objek'),
('02006', 'Bahasa Indonesia');
💡
Run | New Tab | JSON
SELECT * FROM mk; 25ms

```

kode_mk varchar	nama_mk varchar
02004	Aljabar Linier
02005	Analisis dan Desain Beroor
02006	Bahasa Indonesia
02010	Basis Data
02041	Teknologi Data

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

```
Run | New Tab
INSERT INTO mk (kode_mk, nama_mk) VALUES
('02001', 'Agama'),
('02002', 'Aljabar Linier'),
('02003', 'Algoritma dan Struktur Data');
Run | New Tab | JSON
SELECT * FROM mk; 27ms
```

kode_mk varchar	nama_mk varchar
02001	Agama
02002	Aljabar Linier
02003	Algoritma dan Struktur Data
02004	Aljabar Linier
02005	Analisis dan Desain Berorientasi Objek
02006	Bahasa Indonesia
02010	Basis Data
02041	Teknologi Data

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

```
Run | New Tab
INSERT INTO mk SET
kode_mk = '02011',
nama_mk = 'Desain Pemrograman Web';

Run | New Tab | JSON
SELECT * FROM mk; 9ms
```

kode_mk varchar	nama_mk varchar
02001	Agama
02002	Aljabar Linier
02003	Algoritma dan Struktur Data
02004	Aljabar Linier
02005	Analisis dan Desain Beroori
02006	Bahasa Indonesia
02010	Basis Data
02011	Desain Pemrograman Web
02041	Teknologi Data

7. 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”)

```
Run | New Tab
CREATE TABLE mk_backup (
  kode_mk VARCHAR(5) NOT NULL,
  nama_mk VARCHAR(100) DEFAULT NULL
);

Run | New Tab
INSERT INTO mk_backup SELECT * FROM mk;

Run | New Tab | JSON
SELECT * FROM mk_backup; 7ms
```

kode_mk varchar	nama_mk varchar
02001	Agama
02002	Aljabar Linier
02003	Algoritma dan Struktur Data
02004	Aljabar Linier
02005	Analisis dan Desain Beroorientasi Objek
02006	Bahasa Indonesia
02010	Basis Data
02011	Desain Pemrograman Web
02041	Teknologi Data
02001	Agama
02002	Aljabar Linier
02003	Algoritma dan Struktur Data
02004	Aljabar Linier
02005	Analisis dan Desain Beroorientasi Objek
02006	Bahasa Indonesia

PRAKTIKUM 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:

```
Run | New Tab
UPDATE jadwal SET kode_dosen = 'D010'

Run | New Tab | JSON
SELECT * FROM jadwal;
```

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. Tampilkan isi data menggunakan statement SELECT berikut

kode_jadwal int	kode_kelas varchar	kode_dose varchar	kode_mk varchar	kode_ruang varchar	kode_hari varchar	jp_mulai int	jp_selesai int
1	2021020101	D001	02001	0504	001	7	9
2	2021010103	D001	02001	0506	002	9	11
3	2021010105	D001	02001	0806	003	10	12
4	2021010102	D001	02001	0506	004	1	3
5	2021010106	D001	02001	0806	004	4	6
6	2021010101	D001	02001	0506	004	7	9
7	2021010104	D001	02001	0506	005	10	12
8	2021010206	D002	02037	0702	001	7	12
9	2021020202	D002	02036	0708	003	2	4
10	2021010205	D002	02037	0713	004	1	6
11	2021020209	D002	02025	0719	004	7	12
12	2021020301	D003	02012	0508	002	1	4
13	2021020302	D003	02012	0508	002	1	4
14	2021010201	D003	02017	0719	003	2	5
15	2021010202	D003	02017	0719	003	2	5
16	2021010203	D003	02017	0507	005	2	5

2. 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:

```
Run | New Tab
UPDATE jadwal
SET kode_dosen = 'D022'
WHERE kode_mk = '02010';

Run | New Tab | JSON
SELECT * FROM jadwal WHERE kode_mk = '02010';
```

kode jadwal int	kode kelas varchar	kode dose varchar	kode mk varchar	kode ruan varchar	kode hari varchar	jp_mulai int	jp_selesai int
29	2021010105	D022	02010	0701	002	1	3
104	2021010103	D022	02010	0615	001	10	12
105	2021010103	D022	02010	0619	003	1	6
113	2021020103	D022	02010	0501	002	4	6
114	2021020104	D022	02010	0501	002	4	6
116	2021020103	D022	02010	0615	004	7	12
117	2021020104	D022	02010	0617	005	1	6
129	2021020109	D022	02010	0716	001	4	6
130	2021020108	D022	02010	0715	002	2	4
198	2021020107	D022	02010	0702	001	9	11
202	2021020101	D022	02010	0617	002	3	5
215	2021010106	D022	02010	0716	001	7	9
231	2021010104	D022	02010	0701	001	4	6
235	2021020105	D022	02010	0708	001	7	9
285	2021020102	D022	02010	0708	002	1	3
352	2021010101	D022	02010	0707	002	4	6

3. 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.

```
Run | New Tab
UPDATE jadwal
SET kode_dosen = NULL
WHERE kode_dosen = 'D010';
Run | New Tab | JSON
SELECT * FROM jadwal; 16ms
```

kode_jadwal int	kode_kelas varchar	kode_dose varchar	kode_mk varchar	kode_ruan varchar	kode_hari varchar
48	2021010103	(NULL)	02026	0806	001
49	2021010202	(NULL)	02023	0704	001
50	2021010202	(NULL)	02023	0713	002
51	2021010201	(NULL)	02023	0705	003
52	2021010201	(NULL)	02023	0716	005

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

```
▷ Run | New Tab
UPDATE jadwal
SET kode_dosen = 'D012',
    kode_ruang = '0702'
WHERE kode_kelas = '2021020204';

▷ Run | New Tab | JSON
SELECT * FROM jadwal WHERE kode_kelas = '2021020204'; 3ms
```

kode_jadwal int	kode_kelas varchar	kode_dose varchar	kode_mk varchar	kode_ruan varchar	kode_hari varchar	jp_mulai int	jp_selesai int
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

5. 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 Yuniarto SKom., MKom.', maka dapat digunakan SQL dengan format berikut.

```
▷ Run | New Tab
UPDATE jadwal
SET kode_dosen = (SELECT kode_dosen FROM dosen WHERE nama_dosen = 'Dika Rizky Yuniarto S.Kom., M.Kom.')
WHERE kode_mk = '02010';

▷ Run | New Tab | JSON
SELECT * FROM jadwal WHERE kode_mk = '02010'; 3ms
```


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

PRAKTIKUM 4 : Percobaan Stetement 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:

```
Run | New Tab
DELETE FROM jadwal WHERE kode_dosen IS NOT NULL;
Run | New Tab | JSON
SELECT * FROM jadwal; 3ms
```

kode_jadwal int	kode_kelas varchar	kode_dose varchar	kode_mk varchar	kode_ruang varchar	kode_hari varchar	jp_mulai int	jp_selesai int
29	2021010105	(NULL)	02010	0701	002	1	3
48	2021010103	(NULL)	02026	0806	001	3	4
49	2021010202	(NULL)	02023	0704	001	8	11
50	2021010202	(NULL)	02023	0713	002	7	10
51	2021010201	(NULL)	02023	0705	003	7	10
52	2021010201	(NULL)	02023	0716	005	8	11
104	2021010103	(NULL)	02010	0615	001	10	12
105	2021010103	(NULL)	02010	0619	003	1	6
113	2021020103	(NULL)	02010	0501	002	4	6
114	2021020104	(NULL)	02010	0501	002	4	6
116	2021020103	(NULL)	02010	0615	004	7	12
117	2021020104	(NULL)	02010	0617	005	1	6

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

▷ Run | New Tab

```
DELETE FROM jadwal;
```



▷ Run | New Tab | JSON

```
SELECT * FROM jadwal;
```

kode_jadwal
int

kode_kelas
varchar

kode_dose
varchar

kode_mk
varchar

kode_ruang
varchar

kode_hari
varchar


jp_mulai
int

jp_selesai
int

TUGAS

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

```
UPDATE mk
SET nama_mk = 'Basis Data Dasar'
WHERE nama_mk = 'Basis Data'
```



Run | New Tab | JSON

```
SELECT * FROM mk; 6ms
```

kode_mk varchar	nama_mk varchar
02010	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!
➔ Dalam database tersebut, untuk tabel jadwal kita perlu merubah nama matkul menggunakan kode matkul, dimana kode matkul untuk matkul basis data dasar yaitu ‘02010’, dan kode untuk hari senin yaitu ‘001’

Run | New Tab | Copy

```
UPDATE jadwal
SET kode_hari = '001',
    jp_mulai = 5,
    jp_selesai = 10
WHERE kode_mk = '02010';
```

Run | New Tab | JSON

```
SELECT * FROM jadwal WHERE kode_mk = '02010'; 5ms
```

kode_jadwal int	kode_kelas varchar	kode_dose varchar	kode_mk varchar	kode_ruang varchar	kode_hari varchar	jp_mulai int	jp_selesai int
29	2021010105	(NULL)	02010	0701	001	5	10
104	2021010103	(NULL)	02010	0615	001	5	10
105	2021010103	(NULL)	02010	0619	001	5	10
113	2021020103	(NULL)	02010	0501	001	5	10
114	2021020104	(NULL)	02010	0501	001	5	10
116	2021020103	(NULL)	02010	0615	001	5	10
117	2021020104	(NULL)	02010	0617	001	5	10
129	2021020109	(NULL)	02010	0716	001	5	10
130	2021020108	(NULL)	02010	0715	001	5	10
198	2021020107	(NULL)	02010	0702	001	5	10
202	2021020101	(NULL)	02010	0617	001	5	10
215	2021010106	(NULL)	02010	0716	001	5	10

- Hapus jadwal perkuliahan “Kewarganegaraan” pada tabel jadwal!
Sebelumnya seperti ini :

	kode_mk varchar	nama_mk varchar
> 16	02016	Kewarganegaraan

Setelah dihapus akan menjadi seperti ini :

```

Run | New Tab
DELETE FROM mk
WHERE kode_mk = (SELECT kode_mk FROM mk WHERE nama_mk = 'Kewarganegaraan');

Run | New Tab | JSON
SELECT * FROM mk; 2ms

```

kode_mk varchar	nama_mk varchar
02015	Internet Of Things
02017	Komputasi Multimedia
02018	Machine Learning
02019	Manajemen Jaringan Komp
02020	Manajemen Proyek
02021	Manajemen Proyek

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

```
Run | New Tab
INSERT INTO mk (kode_mk, nama_mk) VALUES
('02042', 'Pancasila'),
('02043', 'Bela Negara'),
('02044', 'Wawasan Nusantara');

Run | New Tab | JSON
SELECT * FROM mk; 3ms
```


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

▶ Run | New Tab

```
INSERT INTO mk_backup  
SELECT * FROM mk;
```



▶ Run | New Tab | JSON

```
SELECT * FROM mk_backup; 6m
```

kode_mk varchar	nama_mk varchar
02001	Agama
02002	Aljabar Linier
02003	Algoritma dan Struktur Data
02004	Aljabar Linier
02005	Analisis Dan Desain Berorien

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 !

```
▷ Run | New Tab | Copy
CREATE TABLE mahasiswa (
  nim INT NOT NULL,
  nama_mahasiswa VARCHAR(100) DEFAULT NULL,
  kode_kelas VARCHAR(10) DEFAULT NULL,
  PRIMARY KEY (nim),
  FOREIGN KEY (kode_kelas) REFERENCES kelas (kode_kelas)
);

▷ Run | New Tab
INSERT INTO kelas (kode_kelas) VALUES ('SIB1E');

▷ Run | New Tab | Copy
INSERT INTO mahasiswa (nim, nama_mahasiswa, kode_kelas) VALUES
(1, 'Afril', 'SIB1E'),
(2, 'Aida', 'SIB1E'),
(3, 'Alfin', 'SIB1E'),
(4, 'Annisa', 'SIB1E'),
(5, 'Aqil', 'SIB1E'),
(6, 'Athallah', 'SIB1E'),
(7, 'Bayu', 'SIB1E'),
(8, 'Claudya', 'SIB1E'),
(9, 'Devita', 'SIB1E'),
(10, 'Dwi Septa', 'SIB1E');

▷ Run | New Tab | JSON
SELECT * FROM mahasiswa;
```

nim int	nama_mahasiswa varchar	kode_kelas varchar
1	Afril	SIB1E
2	Aida	SIB1E
3	Alfin	SIB1E
4	Annisa	SIB1E
5	Aqil	SIB1E
6	Athallah	SIB1E
7	Bayu	SIB1E
8	Claudya	SIB1E
9	Devita	SIB1E
10	Dwi Septa	SIB1E

```

Run | New Tab
UPDATE jadwal
SET kode_dosen = 'D024'
WHERE kode_mk = (SELECT kode_mk FROM mk WHERE nama_mk = 'Basis Data Dasar') AND kode_kelas = 'SIB1E';

Run | New Tab | JSON
SELECT * FROM jadwal;

```