

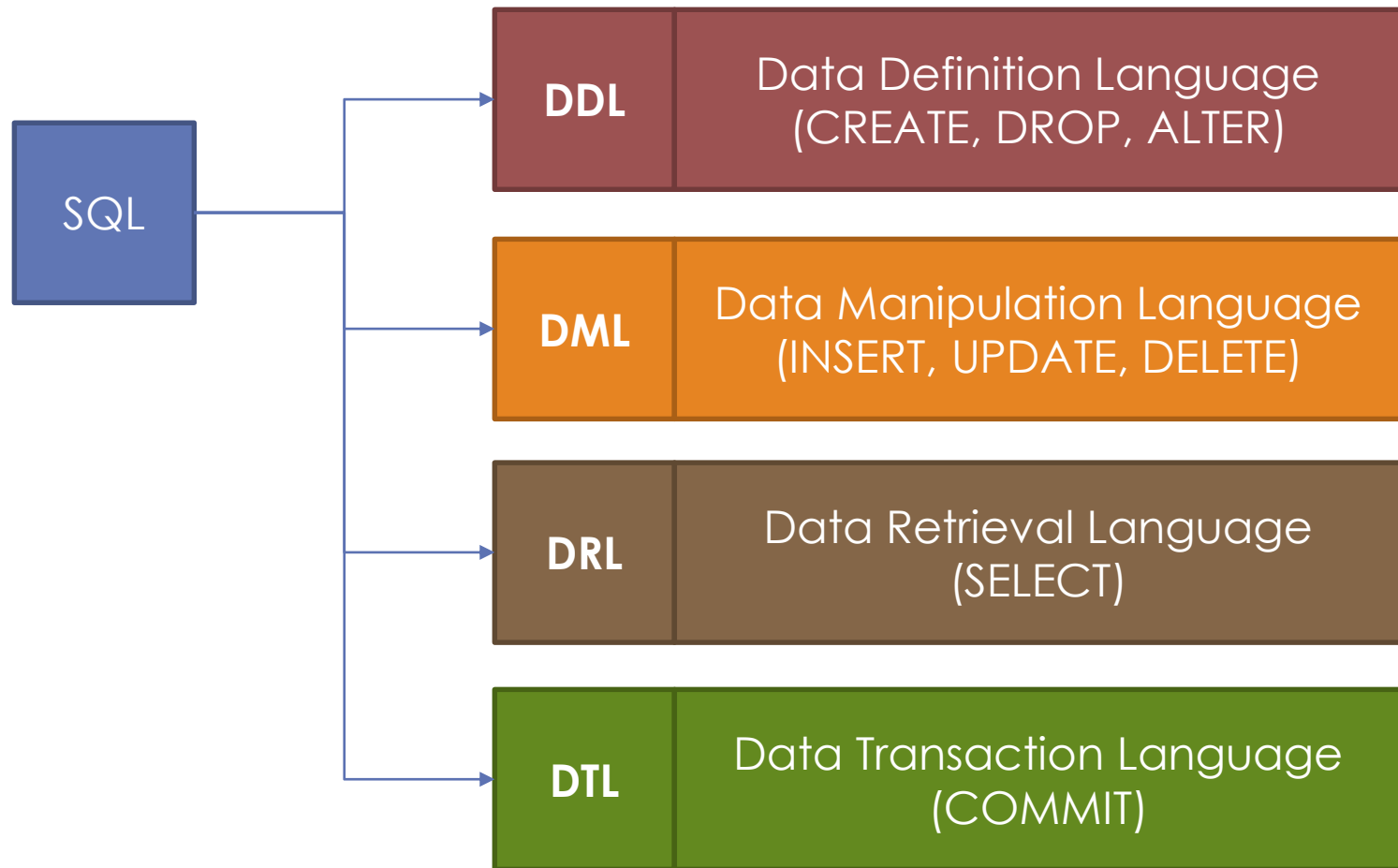


BASIS DATA

# DML (DATA MANIPULATION LANGUAGE)

*TIM AJAR BASIS DATA JTI-POLINEMA*

# STRUCTURED QUERY LANGUAGE

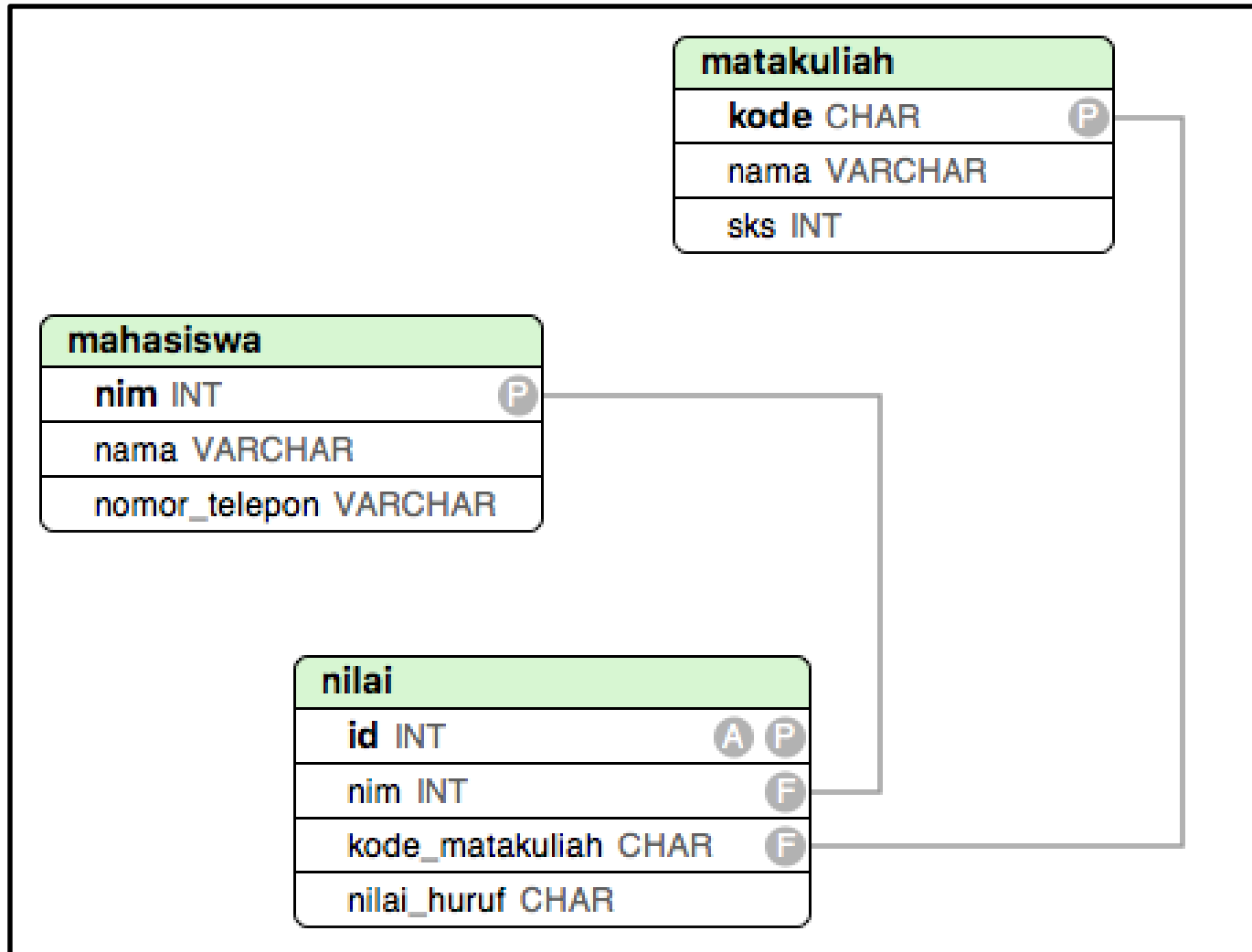


# DML (DATA MANIPULATION LANGUAGE)



- Adalah BAHASA yang digunakan untuk memerintahkan DBMS agar melakukan operasi-operasi yang sifatnya **MENGUBAH** nilai-nilai data pada (**ISI**) **tabel**.
- Yang diubah oleh:
  - DDL → **Struktur** tabel.
  - DML → **Isi** tabel.
- Perspektif:
  - DDL → **TABEL**.
  - DML → **BARIS**/Row/Record/Tuple.
- Ada 3 klausa utama:
  - INSERT : **Menambahkan** suatu BARIS baru.
  - UPDATE : **Mengganti** nilai pada suatu BARIS.
  - DELETE : **Menghapus** suatu BARIS.
- Dan 1 klausa syarat (filtering):
  - **WHERE**

# DATABASE AKADEMIK



# DATABASE AKADEMIK: DDL

```
CREATE DATABASE akademik;
```

```
USE akademik;
```

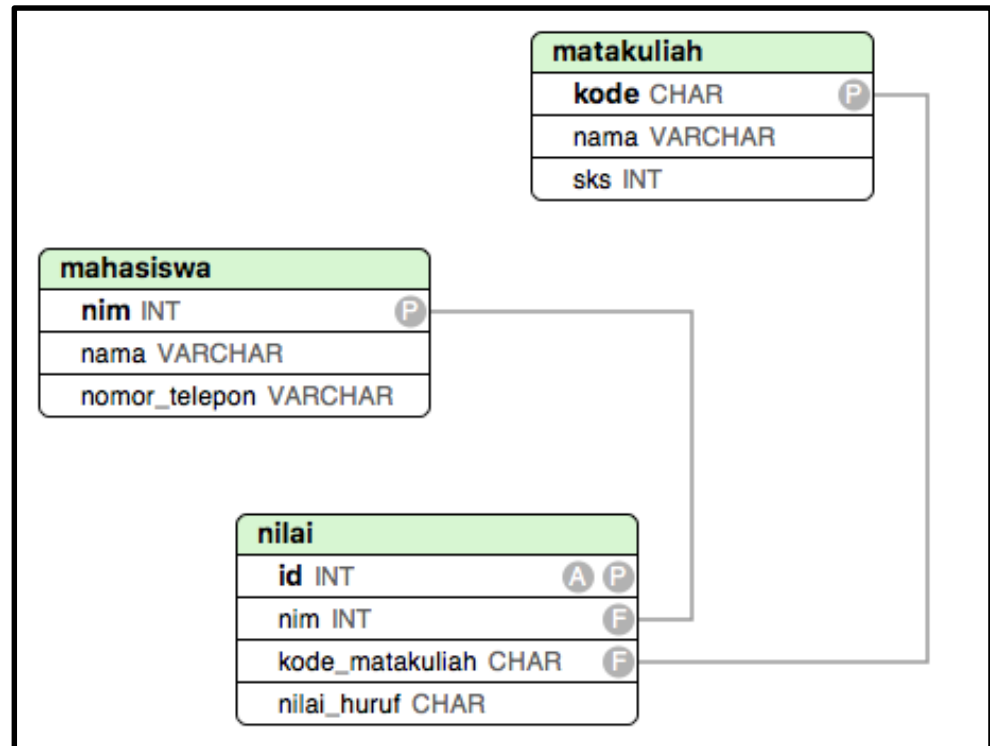
```
CREATE TABLE mahasiswa (  
  nim INT(2),  
  nama VARCHAR(255),  
  nomor_telepon VARCHAR(20),  
  PRIMARY KEY (nim)  
);
```

```
CREATE TABLE matakuliah (  
  kode CHAR(3),  
  nama VARCHAR(255),  
  sks INT(1),  
  PRIMARY KEY (kode)  
);
```

```
CREATE TABLE nilai (  
  id INT AUTO_INCREMENT,  
  nim INT(2),  
  kode_matakuliah CHAR(3),  
  nilai_huruf CHAR(2),  
  PRIMARY KEY (id)  
);
```

```
ALTER TABLE nilai  
  ADD FOREIGN KEY nim_idxfk (nim) REFERENCES mahasiswa (nim);
```

```
ALTER TABLE nilai  
  ADD FOREIGN KEY kode_matakuliah_idxfk (kode_matakuliah) REFERENCES matakuliah (kode);
```





# INSERT

- Digunakan untuk **menambahkan** RECORD/Baris baru pada suatu tabel.
- Klausa pembentuk:
  - **INSERT**
  - **INTO**
  - **VALUES**
- Format:
  1. **INSERT INTO** nama\_tabel (kolom1, kolom2, ...dst.) **VALUES** (nilai\_kolom1, nilai\_kolom2, ...dst.);
  2. **INSERT INTO** nama\_tabel **VALUES** (nilai\_kolom1, nilai\_kolom2, ...dst.);
  3. [Salah satu dari kedua format sebelumnya], (nilai\_kolom\_kolom\_baris1), (nilai\_kolom\_kolom\_baris2), ...dst.



# INSERT

- Contoh Format #1:
  - **INSERT INTO** nama\_tabel (kolom1, kolom2, ...dst.) **VALUES** (nilai\_kolom1, nilai\_kolom2, ...dst.);
- Digunakan jika kita ingin menambahkan data pada **sebagian** kolom saja.

SQL:

```
INSERT INTO mahasiswa (nim, nama) VALUES (1, 'Ani Rahmawati');
```

Akan menghasilkan:

+	---	+	-----	+	-----	+
	nim		nama		nomor_telepon	
+	---	+	-----	+	-----	+
	1		Ani Rahmawati		NULL	
+	---	+	-----	+	-----	+



# INSERT

- Contoh Format **#2**:
  - **INSERT INTO** nama\_tabel **VALUES** (nilai\_kolom1, nilai\_kolom2, ...dst.);
- Digunakan jika kita ingin menambahkan baris baru dengan data pada **semua** kolom.

SQL:

```
INSERT INTO mahasiswa VALUES (2, 'Budi Raharjo', '0858776453');
```

Akan menghasilkan:

+	-----	+	-----	+	-----	+
	nim		nama		nomor_telepon	
+	-----	+	-----	+	-----	+
	1		Ani Rahmawati		NULL	
	2		Budi Raharjo		0858776453	
+	-----	+	-----	+	-----	+



# INSERT

- Contoh Format **#3**:
  - [Salah satu dari kedua format sebelumnya], (nilai\_kolom\_kolom\_baris1), (nilai\_kolom\_kolom\_baris2), ...dst.
- Digunakan jika kita ingin menambahkan **beberapa baris** baru **sekaligus** dalam 1 SQL.

**SQL:**

```
INSERT INTO mahasiswa VALUES  
  (3, 'Charlie Setiabudi', '0859767553'),  
  (4, 'Diandra Paramita', '0858998745');
```

**Akan menghasilkan:**

nim	nama	nomor_telepon
1	Ani Rahmawati	NULL
2	Budi Raharjo	0858776453
3	Charlie Setiabudi	0859767553
4	Diandra Paramita	0858998745

# INSERT

- [Salah satu dari kedua format sebelumnya], (nilai\_semua\_kolom\_baris1), (nilai\_semua\_kolom\_baris2), ...dst.

## SQL:

```
INSERT INTO matakuliah (kode, nama) VALUES
('BDD', 'Basis Data Dasar'),
('PBO', 'Pemrograman Berorientasi Objek'),
('MMT', 'Multimedia Terapan'),
('SPK', 'Sistem Pendukung Keputusan'),
('KCB', 'Kecerdasan Buatan'),
('ASD', 'Algoritma dan Struktur Data');
```

## Akan menghasilkan:

kode	nama	sks
ASD	Algoritma dan Struktur Data	NULL
BDD	Basis Data Dasar	NULL
KCB	Kecerdasan Buatan	NULL
MMT	Multimedia Terapan	NULL
PBO	Pemrograman Berorientasi Objek	NULL
SPK	Sistem Pendukung Keputusan	NULL



# KLAUSA 'WHERE'

- **WHERE** digunakan pada statement-statement UPDATE, DELETE, dan SELECT sebagai filter/pembatas terhadap hasil yang dikembalikan.
- Format:
  - [Statement Utama] **WHERE** kolom\_patokan [operator\_perbandingan] nilai\_patokan;
  - [Statement Utama] **WHERE** kolom\_patokan1 [operator\_perbandingan1] nilai\_patokan1 [operator\_logika1] kolom\_patokan2 [operator\_perbandingan2] nilai\_patokan2 [operator\_logika2] ...dst.;
- Operator perbandingan/comparison operator dapat berupa:
  - =, <, >, <=, >=, <>
- Operator logika dapat berupa:
  - **AND, OR**
- Contoh:
  - **SELECT \* FROM** matakuliah **WHERE** kode = 'ASD' ;
  - **UPDATE** matakuliah **SET** sks = 2 **WHERE** nama = 'Kecerdasan Buatan';
  - **DELETE FROM** matakuliah **WHERE** kode = 'SPK';
  - **DELETE FROM** matakuliah **WHERE** kode = 'SPK' **OR** kode = 'ASD';



# KLAUSA 'WHERE'

**SELECT \* FROM** matakuliah **WHERE** kode = 'ASD' ;

Statement utama

Kolom patokan

Operator perbandingan

Nilai patokan

```
[mysql> SELECT * FROM mahasiswa;
```

nim	nama	nomor_telepon
1	Ani Rahmawati	NULL
2	Budi Raharjo	NULL
3	Charlie Setiabudi	0859767553
4	Diandra Paramita	0858998745

```
[mysql> SELECT * FROM mahasiswa WHERE nim < 4;
```

nim	nama	nomor_telepon
1	Ani Rahmawati	NULL
2	Budi Raharjo	NULL
3	Charlie Setiabudi	0859767553

\*Klausula WHERE **membatasi** hasil query SELECT.



# UPDATE

- Digunakan untuk **mengubah/mengganti** nilai RECORD/Baris yang sudah ada pada suatu tabel.
- Klausa pembentuk:
  - **UPDATE**
  - **SET**
  - **WHERE**
- Format:
  1. **UPDATE** nama\_tabel **SET** nama\_kolom = nilai\_baru **WHERE** nama\_kolom\_patokan **[operator\_perbandingan]** nilai\_patokan;
  2. **UPDATE** nama\_tabel **SET** nama\_kolom1 = nilai\_baru1, nama\_kolom2 = nilai\_baru2, ...dst. **WHERE** nama\_kolom\_patokan **[operator\_perbandingan]** nilai\_patokan;
- Operator perbandingan/comparison operator dapat berupa:
  - **=, <, >, <=, >=, <>**



# UPDATE

- Contoh Format #1:
  - **UPDATE** nama\_tabel **SET** nama\_kolom = nilai\_baru **WHERE** nama\_kolom\_patokan [operator\_perbandingan] nilai\_patokan;
- Digunakan ketika kita ingin mengganti nilai suatu baris untuk **1 kolom** tertentu saja.

SQL:

```
UPDATE mahasiswa SET nomor_telepon = '0857550234'  
WHERE nim = 1;
```

Akan menghasilkan:

nim	nama	nomor_telepon
1	Ani Rahmawati	0857550234
2	Budi Raharjo	0858776453
3	Charlie Setiabudi	0859767553
4	Diandra Paramita	0858998745

# UPDATE

- Contoh Format #2:
  - **UPDATE** nama\_tabel **SET** nama\_kolom1 = nilai\_baru1, nama\_kolom2 = nilai\_baru2, ...dst. **WHERE** nama\_kolom\_patokan **[operator\_perbandingan]** nilai\_patokan;
- Digunakan ketika kita ingin mengganti nilai suatu baris untuk **beberapa** kolom sekaligus.

```
UPDATE matakuliah SET
  nama = 'Multimedia Terapan Tingkat Lanjut',
  sks = 3
WHERE kode = 'MMT';
```

Akan menghasilkan:

kode	nama	sks
ASD	Algoritma dan Struktur Data	NULL
BDD	Basis Data Dasar	NULL
KCB	Kecerdasan Buatan	NULL
MMT	Multimedia Terapan Tingkat Lanjut	3
PBO	Pemrograman Berorientasi Objek	NULL
SPK	Sistem Pendukung Keputusan	NULL



# DELETE

- Digunakan untuk **menghapus** suatu **RECORD/Baris** yang sebelumnya ada pada suatu tabel.
- Klausa pembentuk:
  - **DELETE**
  - **FROM**
  - **WHERE**
- Format:
  1. **DELETE FROM** nama\_tabel **WHERE** nama\_kolom\_patokan **[operator\_perbandingan]** nilai\_patokan;
  2. **DELETE \* FROM** nama\_tabel; atau **DELETE FROM** nama\_tabel;
- Operator perbandingan/comparison operator dapat berupa:
  - **=, <, >, <=, >=, <>**





# DELETE

- Contoh Format #1:
  - **DELETE FROM** nama\_tabel **WHERE** nama\_kolom\_patokan [operator\_perbandingan] nilai\_patokan;
- Digunakan ketika kita ingin menghapus suatu baris dengan **syarat** tertentu.

SQL:

```
DELETE FROM matakuliah WHERE kode = 'BDD';
```

Akan menghasilkan:

kode	nama	sks
ASD	Algoritma dan Struktur Data	NULL
KCB	Kecerdasan Buatan	NULL
MMT	Multimedia Terapan Tingkat Lanjut	3
PBO	Pemrograman Berorientasi Objek	NULL
SPK	Sistem Pendukung Keputusan	NULL



# DELETE

- Contoh Format #2:
  - **DELETE \* FROM** nama\_tabel; atau **DELETE FROM** nama\_tabel;
- Digunakan ketika kita ingin menghapus **semua** baris/records pada suatu tabel.
- **WARNING: Tidak dapat di-undo!**

SQL:

```
SET SQL_SAFE_UPDATES = 0;  
DELETE FROM matakuliah;  
SET SQL_SAFE_UPDATES = 1;
```

Akan menghasilkan:

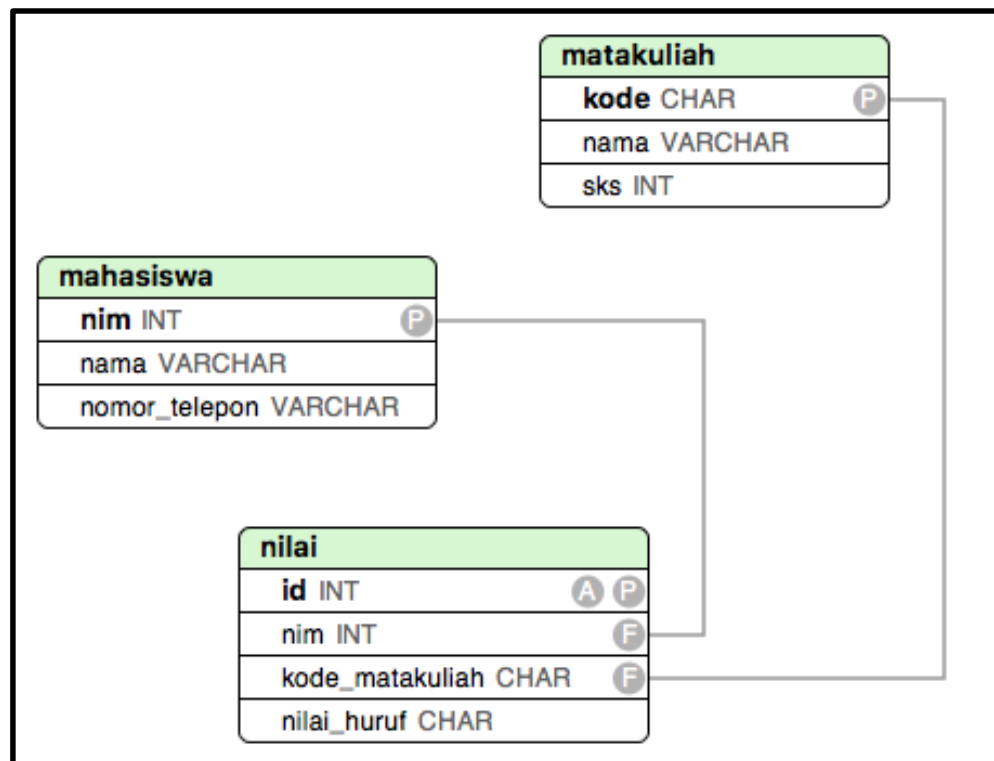
```
[mysql> SELECT * FROM matakuliah;  
Empty set (0.00 sec)
```

```
mysql> █
```

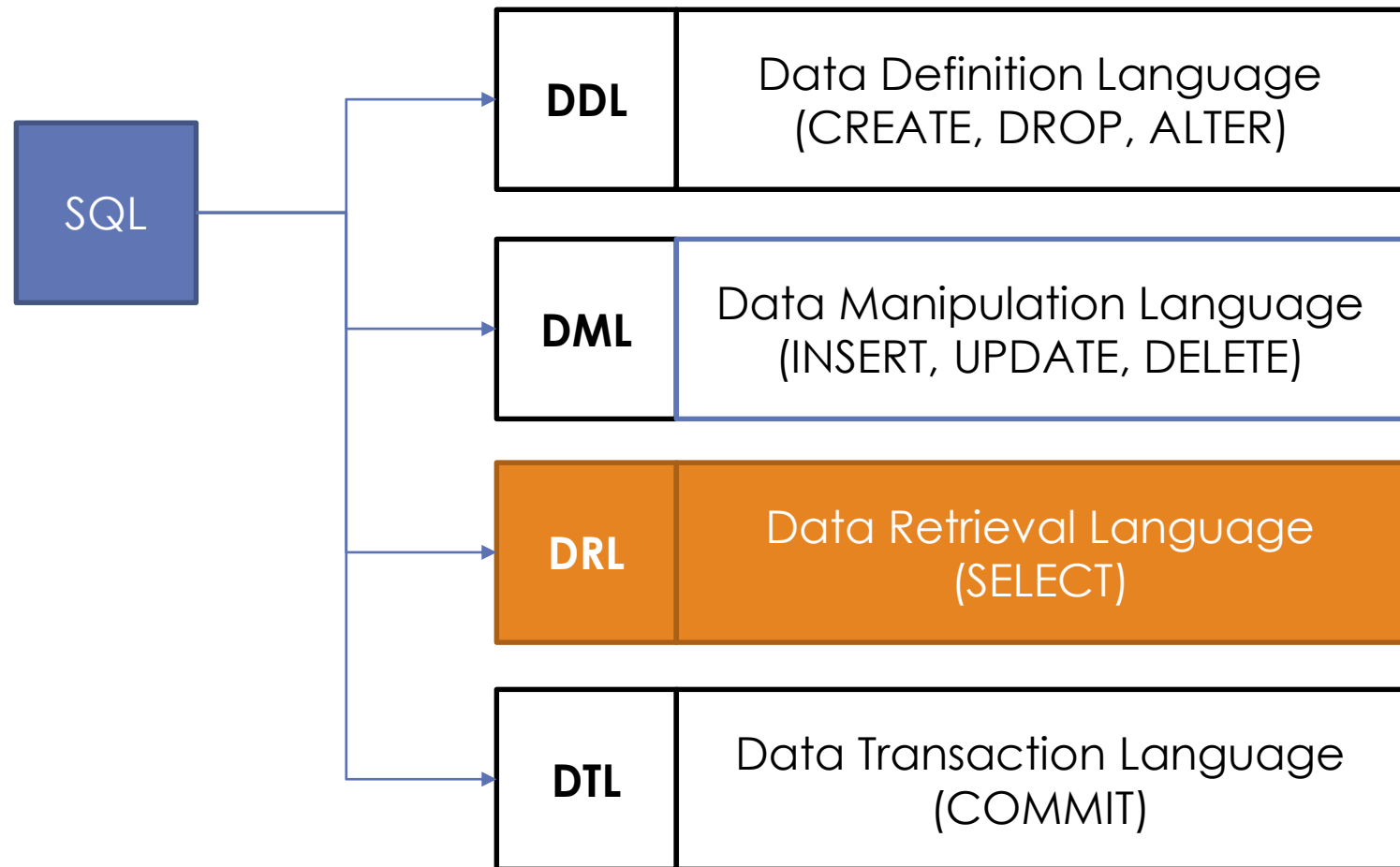


# LATIHAN

- Buatlah SQL untuk mengisi tabel nilai dengan 5 data yang valid!
- Buatlah SQL untuk mengisi SKS yang kosong pada tabel matakuliah sehingga semua matakuliah tersebut SKS-nya menjadi = 2.
- Buatlah SQL untuk menghapus data matakuliah dengan nama Sistem Pendukung Keputusan.



# STRUCTURED QUERY LANGUAGE

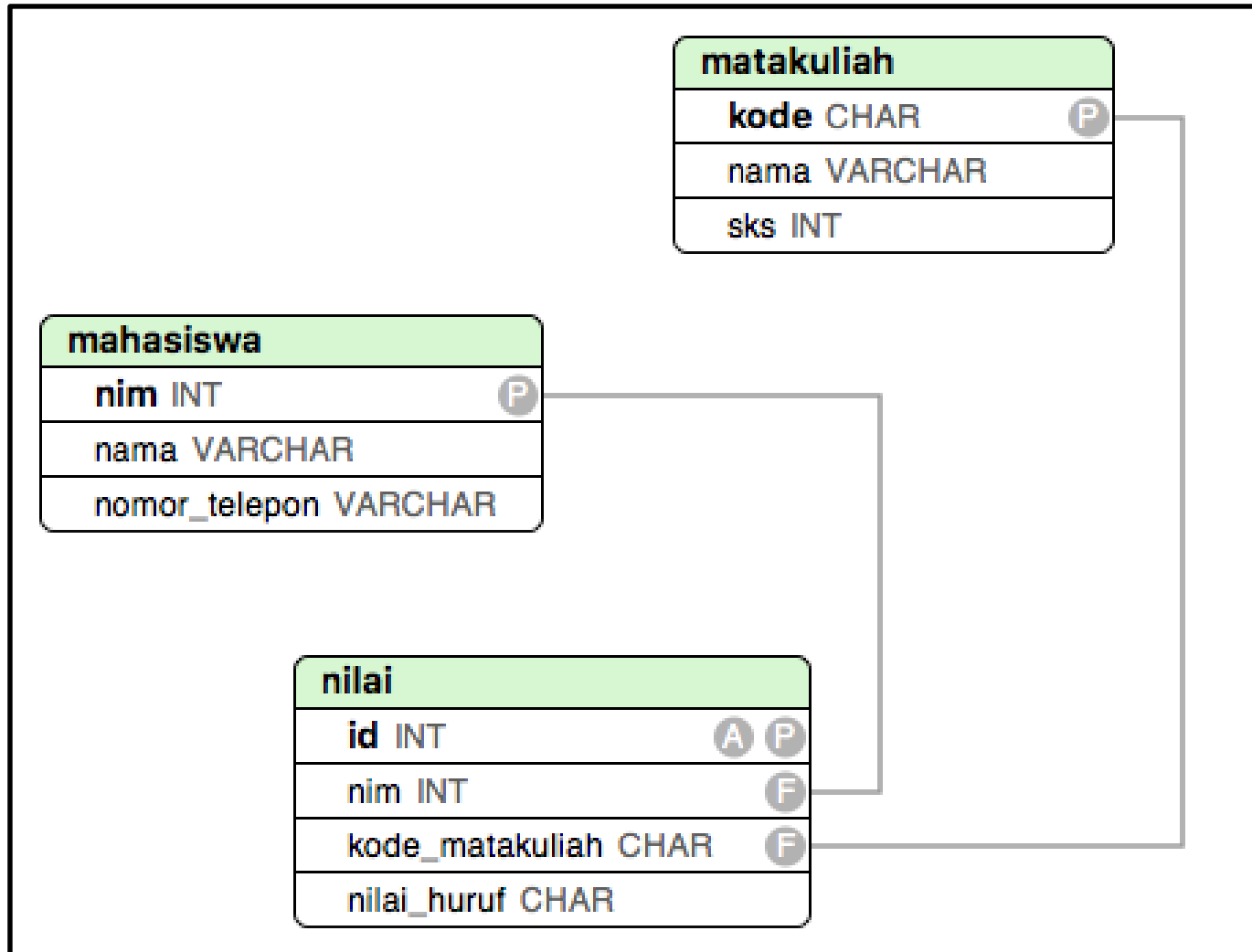


# DRL (DATA RETRIEVAL LANGUAGE)



- DRL atau DSL (Data Selection Language) adalah set perintah yang digunakan untuk mengambil data dari server basis data.
- DRL dapat digunakan pada query, sub-query dan fungsi agregasi
- **Beberapa klausa DRL :**
  - **FROM** : Memilih nama sebuah tabel
  - **WHERE** : Menunjukkan secara spesifik suatu row/baris yang akan dicari
  - **GROUP BY** : Menyusun data ke dalam grup
  - **HAVING** : Memilih diantara kelompok-kelompok data yang didefinisikan klausa GROUP BY
  - **ORDER BY** : Secara spesifik digunakan untuk menentukan suatu baris pada kolom tertentu
  - **AS** : Memberikan nama alias sementara untuk tabel atau kolom

# DATABASE AKADEMIK





# DATABASE AKADEMIK: DDL

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

nim	nama	nomor_telepon
171401	Ani Rahmawati	0858776453
171402	Budi Raharjo	NULL
171403	Charlie Setiabudi	0813543226
171404	Diandra Paramita	NULL

```
4 rows in set (0.00 sec)
```

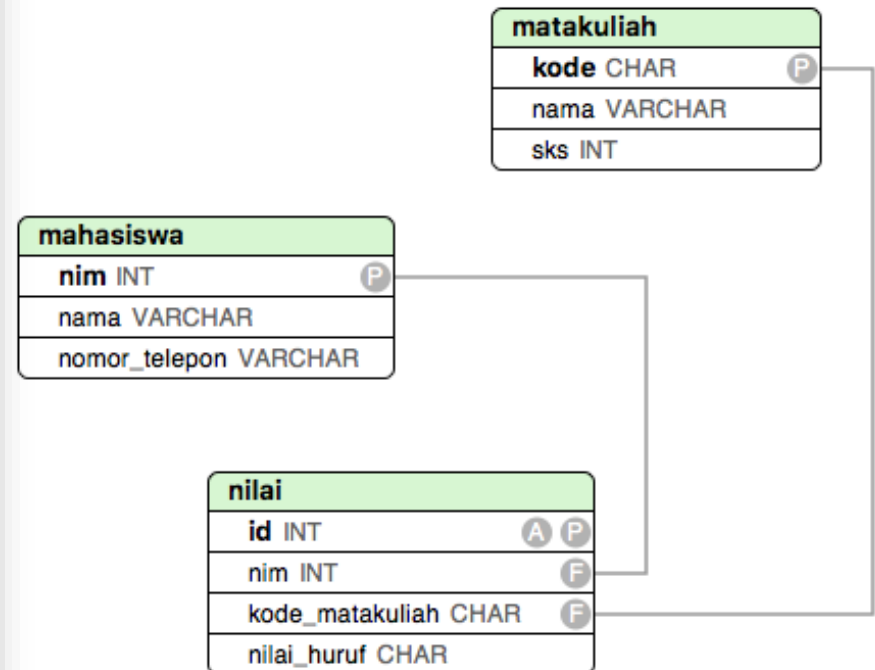
```
MariaDB [akademik]> select * from matakuliah;
```

kode	nama	sks
ASD	Algoritma dan Struktur Data	NULL
BDD	Basis Data Dasar	NULL
KCB	Kecerdasan Buatan	NULL
MMT	Multimedia Terapan	NULL
PBO	Pemrograman Berorientasi Objek	NULL
SPK	Sistem Pendukung Keputusan	NULL

```
6 rows in set (0.00 sec)
```

```
MariaDB [akademik]> select * from nilai;
```

id	nim	kode_matakuliah	nilai_huruf
1	171402	ASD	A
2	171402	SPK	C
3	171401	ASD	B
4	171401	SPK	B
5	171403	ASD	A
6	171403	SPK	A
7	171403	BDD	A





# QUERY

- Query merupakan operasi yang melibatkan satu atau lebih tabel untuk melakukan retrieval data.
- Retrieval data dilakukan dengan beberapacara berikut :

SELECT \* untuk memilih semua kolom

SELECT dengan WHERE untuk menampilkan baris dengan suatu kondisi

SELECT dengan DISTINCT untuk menampilkan data dengan eliminasi data yang sama (duplicate)

SELECT dengan IN untuk menampilkan data yang spesifik

SELECT dengan BETWEEN untuk menampilkan data pada jarak (range) tertentu

SELECT dengan LIKE untuk menampilkan data yang memiliki kemiripan dengan keyword yang diinginkan

SELECT dengan ORDER BY untuk menampilkan baris secara spesifik dan terurut maju atau mundur





# QUERY (SELECT-WHERE)

Format :

```
SELECT column1, column2,...  
FROM table_name  
WHERE condition;
```

**SQL:**

```
SELECT nim, kode_matakuliah  
FROM nilai  
WHERE nilai_huruf = 'A'
```

**Akan menghasilkan:**

nim	kode_matakuliah
171402	ASD
171403	ASD
171403	SPK
171403	BDD



# QUERY (SELECT-DISTINCT)

Format :

```
SELECT DISTINCT column FROM table_name;
```

SQL:

```
SELECT DISTINCT nilai_huruf  
FROM nilai;
```

Akan menghasilkan:

nilai_huruf
A
C
B



# QUERY (SELECT-IN)

Format :

```
SELECT column_name(s)
FROM table_name
WHERE column_name IN (value1, value2, ...);
```

SQL:

```
SELECT *
FROM nilai
WHERE nilai_huruf IN ('A','C');
```

Akan menghasilkan:

id	nim	kode_matakuliah	nilai_huruf
1	171402	ASD	A
2	171402	SPK	C
5	171403	ASD	A
6	171403	SPK	A
7	171403	BDD	A



# QUERY (SELECT-BETWEEN)

Format :

```
SELECT column_name(s)
FROM table_name
WHERE column_name BETWEEN value1 AND value2;
```

SQL:

```
SELECT *
FROM nilai
WHERE nilai_huruf
BETWEEN 'B' AND 'D';
```

Akan menghasilkan:

id	nim	kode_matakuliah	nilai_huruf
2	171402	SPK	C
3	171401	ASD	B
4	171401	SPK	B



# QUERY (SELECT-LIKE)

Format :

```
SELECT column1,column2,...  
FROM table_name  
WHERE columnN LIKE pattern;
```

SQL:

```
SELECT *  
FROM mahasiswa  
WHERE nama like '%ra%';
```

Akan menghasilkan:

nim	nama	nomor_telepon
171401	Ani Rahmawati	0858776453
171402	Budi Raharjo	NULL
171404	Diandra Paramita	NULL



# QUERY (SELECT-ORDER BY)

Format :

```
SELECT column1, column2, ...  
FROM table_name  
ORDER BY column1, column2, ... ASC|DESC;
```

SQL:

```
SELECT kode, nama  
FROM matakuliah  
ORDER BY nama DESC
```

Akan menghasilkan:

kode	nama
SPK	Sistem Pendukung Keputusan
PBO	Pemrograman Berorientasi Objek
MMT	Multimedia Terapan
KCB	Kecerdasan Buatan
BDD	Basis Data Dasar
ASD	Algoritma dan Struktur Data



# LATIHAN

1. Buatlah SQL untuk mengisi kolom sks pada tabel matakuliah! (Pastikan jumlah sks ada yang berbeda)
2. Buatlah SQL untuk mengisi kolom nilai \_huruf dengan 3 record baru untuk mahasiswa yang mendapatkan nilai 'B+' pada tabel nilai!
3. Buatlah SQL untuk mencari kode matakuliah yang sudah diisi pada tabel nilai dan tabel matakuliah!
4. Buatlah SQL untuk mencari data yang tidak bernilai diantara C dan E!
5. Tampilkan semua kolom dalam tabel matakuliah yang mempunyai jumlah sks lebih dari jumlah sks yang dimiliki oleh nama mata kuliah Kecerdasan Buatan!

# PERTANYAAN??







# REFERENSI

- Dwi Puspitasari, S.Kom, “**Buku Ajar Dasar Basis Data**”, *Program Studi Manajemen Informatika Politeknik Negeri Malang*, 2012.
- Fathansyah, “**Basisdata Revisi Kedua**”, Bandung: Informatika, 2015.
- <http://www.tutorialride.com>
- <https://www.w3schools.com/>



TERIMAKASIH