

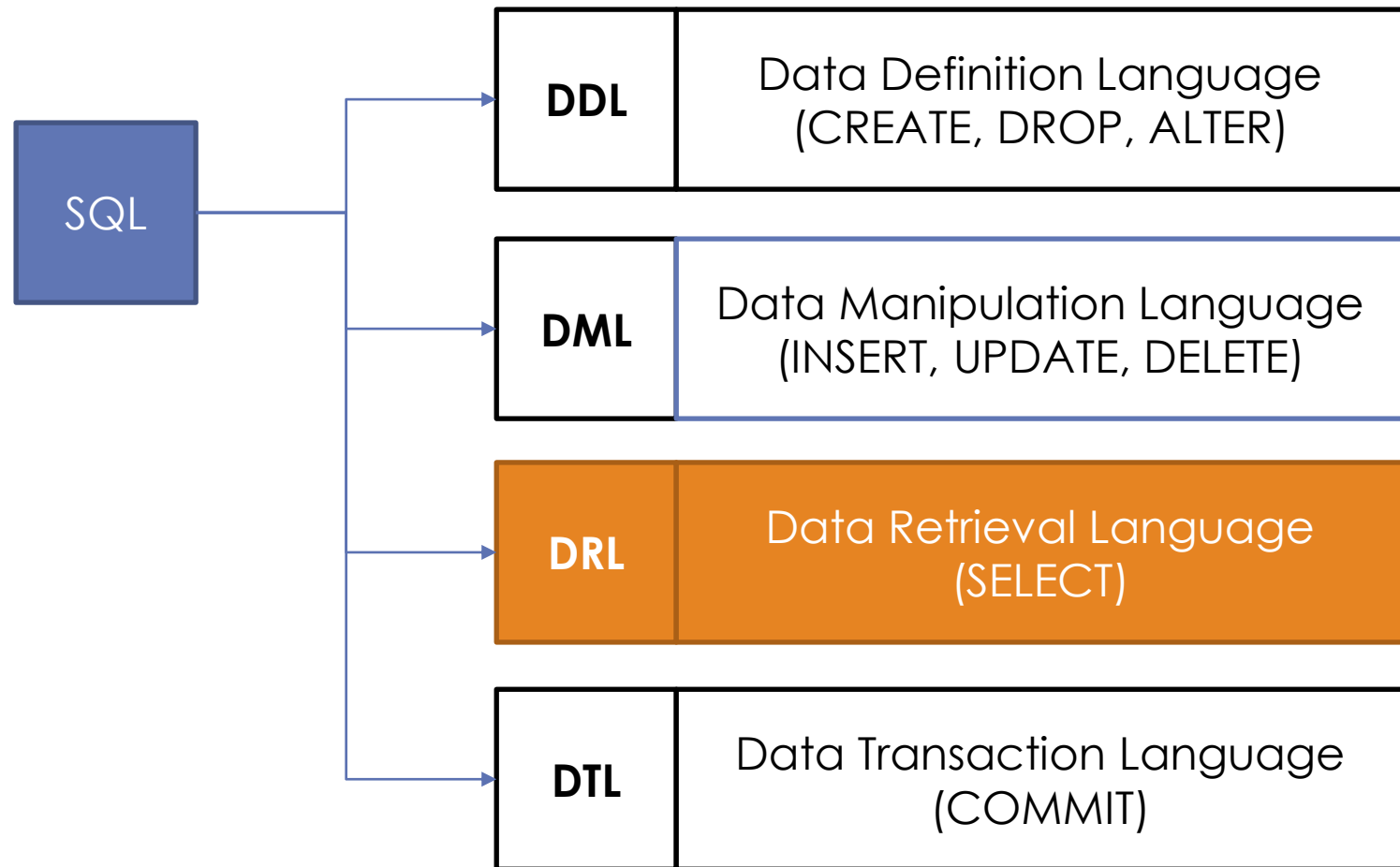


## BASIS DATA

# DRL(DATA RETRIEVAL LANGUAGE) **SELECT**

*TIM AJAR BASIS DATA JTI-POLINEMA*

# STRUCTURED QUERY LANGUAGE

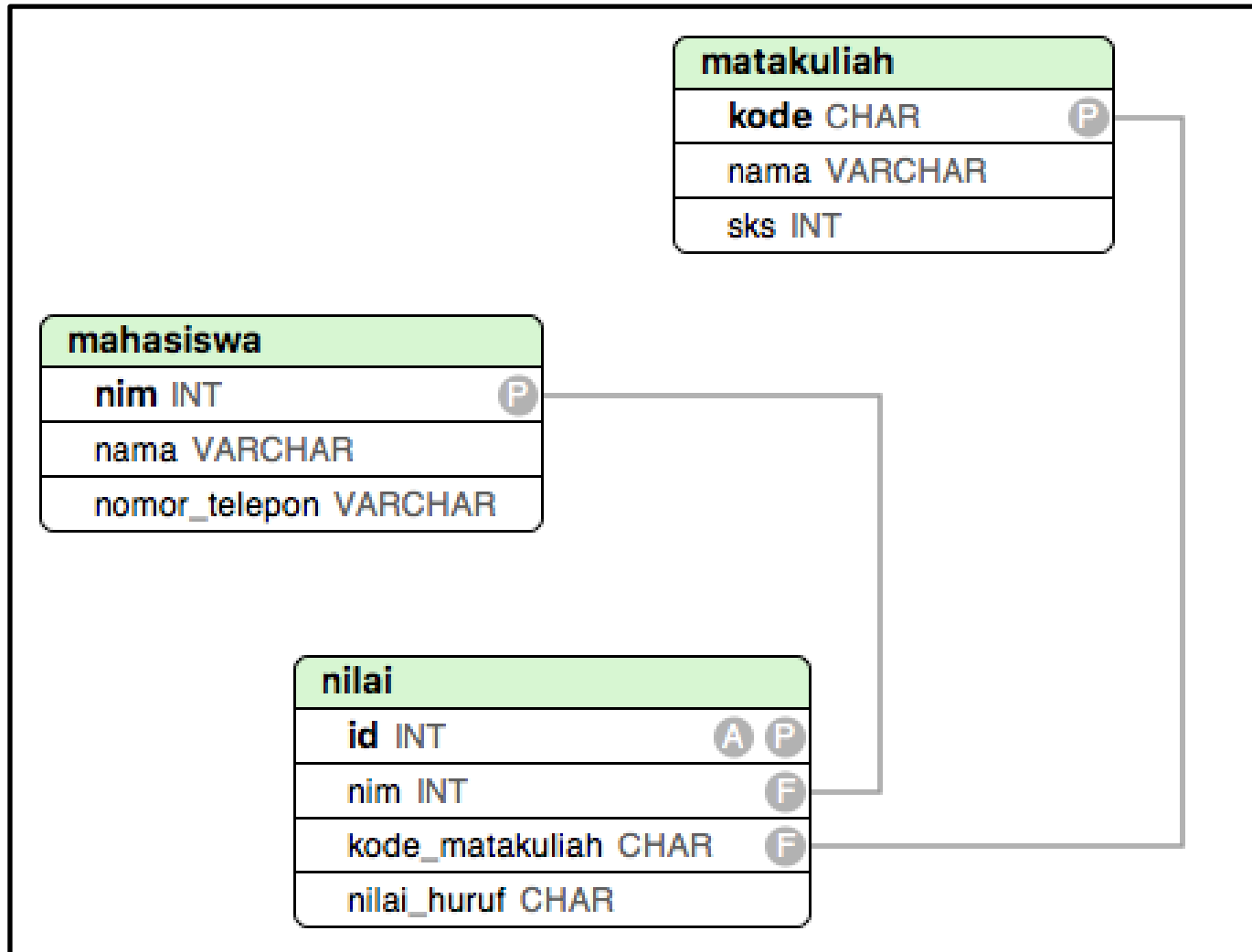


# DRL (DATA RETRIEVAL LANGUAGE)



- 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)
```

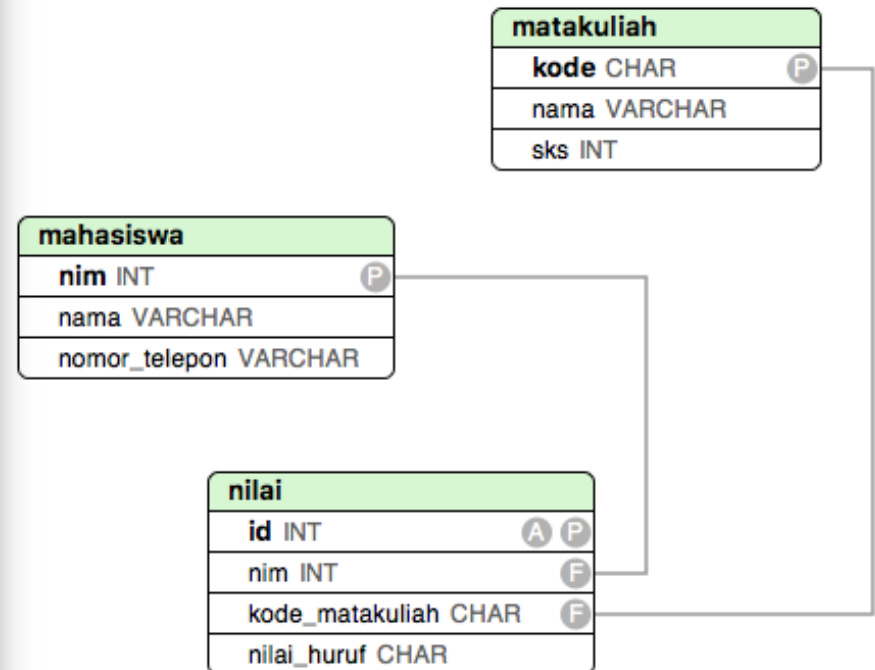
```
MariaDB [akademik]> select * from matakuliah;
```

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 beberapa cara 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 GROUP BY untuk menampilkan susunan data dalam bentuk grup

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

SELECT dengan AND, OR and NOT untuk menampilkan data dengan kondisi dan, atau, tidak

SELECT dengan UNION, INTERSECT dan EXCEPT untuk menampilkan data dengan operasi himpunan yang melibatkan lebih dari satu tabel



# 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
E





# 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-GROUP BY)

Format :

```
SELECT column1, column2, ...  
condition  
FROM table_name  
GROUP BY column1, column2, ...;
```

SQL:

```
SELECT nim, kode_matakuliah, nilai_huruf  
FROM nilai  
GROUP BY nilai_huruf
```

Akan menghasilkan:

nim	kode_matakuliah	nilai_huruf
171402	ASD	A
171401	ASD	B
171402	SPK	C



# 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

# QUERY (SELECT-AND | OR | NOT)



Format :

```
SELECT column1, column2, ...  
FROM table_name  
WHERE condition1 AND|OR|NOT condition2 AND|OR|NOT  
condition3 ...;
```

SQL:

```
SELECT *  
FROM nilai  
WHERE nim = 171401  
AND kode_matakuliah = 'ASD';
```

Akan menghasilkan:

id	nim	kode_matakuliah	nilai_huruf
3	171401	ASD	B



# QUERY

## (SELECT-UNION | INTERSECT | EXCEPT )

Format :

```
SELECT column_name(s) FROM table1  
UNION|INTERSECT|EXCEPT  
SELECT column_name(s) FROM table2;
```

SQL:

```
SELECT kode  
FROM matakuliah  
UNION  
SELECT kode_matakuliah  
FROM nilai;
```

Akan menghasilkan:

+	-----	+
	<b>kode</b>	
+	-----	+
	<b>ASD</b>	
	<b>BDD</b>	
	<b>KCB</b>	
	<b>MMT</b>	
	<b>PBO</b>	
	<b>SPK</b>	
+	-----	+



# QUERY

## (SELECT-UNION | INTERSECT | EXCEPT )-ALL

Format :

```
SELECT column_name(s) FROM table1  
UNION|INTERSECT|EXCEPT-ALL  
SELECT column_name(s) FROM table2;
```

SQL:

```
SELECT kode  
FROM matakuliah  
UNION ALL  
SELECT kode_matakuliah  
FROM nilai;
```

Akan menghasilkan:

kode
ASD
BDD
KCB
MMT
PBO
SPK
ASD
ASD
ASD
BDD
SPK
SPK
SPK





# SUB-QUERY

- Sub-query adalah adanya query di dalam query lain. Sub-query juga disebut dengan perintah SELECT bersarang (nested SELECT).
- Retrival data dalam sub-query dilakukan oleh SELECT untuk menemukan data pada SELECT utama.
- Kegunaan utama :
  - Test Keaggotaan
  - Perbandingan Himpunan

# SUB-QUERY (TES KEANGGOTAAN)



Format :

```
SELECT column_name(s)
FROM table_name
WHERE column_name IN (SELECT STATEMENT);
```

**SQL:**

```
SELECT nim, kode_matakuliah, nilai_huruf
FROM nilai
WHERE nilai_huruf
in (SELECT MIN(nilai_huruf) FROM nilai);
```

**Akan menghasilkan:**

nim	kode_matakuliah	nilai_huruf
171402	ASD	A
171403	ASD	A
171403	SPK	A
171403	BDD	A

# SUB-QUERY (PERBANDINGAN HIMPUNAN)



Format :

```
SELECT column_name(s)
FROM table_name
operator
(SELECT STATEMENT);
```

**SQL:**

```
SELECT nim, kode_matakuliah, nilai_huruf
FROM nilai
WHERE nilai_huruf > ALL
(SELECT nilai_huruf FROM nilai
WHERE nilai_huruf = 'A');
```

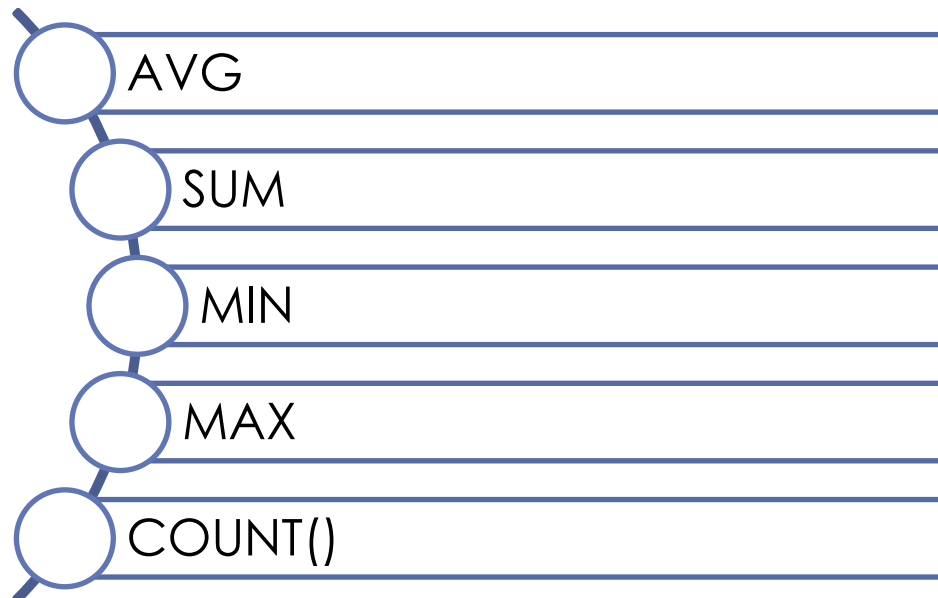
**Akan menghasilkan:**

nim	kode_matakuliah	nilai_huruf
171402	SPK	C
171401	ASD	B
171401	SPK	B



# AGREGASI

- Fungsi agregasi akan melakukan perhitungan kepada sekumpulan nilai dan menghasilkan suatu nilai tunggal.
- Agregasi terdiri dari beberapa statement yaitu :





# AGREGASI

(SELECT AVG | MAX | MIN | SUM | COUNT())

Format :

```
SELECT column_name(s)
FROM table_name
operator
(SELECT STATEMENT);
```

SQL:

```
SELECT MAX(nilai_huruf) FROM nilai;
```

Akan menghasilkan:

max(nilai_huruf)
C



PERTANYAAN??



# 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!



TERIMAKASIH





# 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/>