

# Aljabar Relasional

Badie'ah, ST, M.Kom

# Apa itu Aljabar Relational ?

- Merupakan Bahasa yang mudah dipahami oleh mesin DBMS
- SQL yang biasa kita gunakan nantinya akan diterjemahkan ke dalam bentuk aljabar relasional sehingga query yang kita buat akan dipahami oleh DBMS, diproses dan ditampilkan hasilnya

# OPERASI ALJABAR RELASIONAL

1. Selection ( $\sigma$ )
2. Projection ( $\pi$ )
3. Union ( $\cup$ )
4. Set-Difference ( $-$ )
5. Cartesian-Product ( $\times$ , biasa disebut dengan cross product)
6. Rename ( $\rho$ )

# Selection ( $\sigma$ )

- Operasi ini digunakan untuk menyeleksi record/baris yang memenuhi predikat (syarat atau kondisi) yang sudah ditentukan
- Kalau di SQL adalah seleksi menggunakan kondisi WHERE
- Operator perbandingan ( $=$ ,  $\neq$ ,  $<$ ,  $\leq$ ,  $>$ ,  $\geq$ ) biasanya digunakan pada operasi ini
- Beberapa predikat data digabungkan menggunakan penghubung AND ( $\wedge$ ) dan OR ( $\vee$ ) atau negasi ( $\sim$ )

# Selection ( $\sigma$ )

- Sintaksnya :

$$\sigma_P (E1)$$

- $P \rightarrow$  predikat/kondisi/syarat dari atribut E1
- $E1 \rightarrow$  tabel atau relasi

# Selection ( $\sigma$ )

- Contoh :
  - Tampilkan seluruh data mahasiswa yang kota asalnya adalah “Semarang”

SQL

```
SELECT * FROM mahasiswa WHERE kota_asal="Semarang"
```



Aljabar  
Relasional

```
 $\sigma$  kota_asal="Semarang" (mahasiswa)
```

# Selection ( $\sigma$ )

- Contoh : Tampilkan seluruh data mahasiswa yang kota asalnya “Semarang” dan tahun lahirnya “1999”

SQL

```
SELECT * FROM mahasiswa WHERE kota_asal="Semarang" AND  
tahun_lahir=1999
```



Aljabar  
Relasional

$\sigma$  kota\_asal="Semarang"  $\wedge$  tahun\_lahir=1999 (mahasiswa)

# Projection ( $\pi$ )

- Merupakan operasi yang digunakan untuk memperoleh kolom-kolom tertentu untuk ditampilkan
- Sintaks :

**$\pi$  S (E1)**

atau

**$\pi$  column<sub>1</sub>, ..., column<sub>n</sub> (tabel)**



# Projection ( $\pi$ )

- Contoh : Tampilkan kolom nim, nama, alamat dari tabel mahasiswa

SQL

**SELECT nim, nama, alamat FROM mahasiswa**



Aljabar  
Relasional

**$\pi$  nim, nama, alamat** (mahasiswa)

# Projection ( $\pi$ )

- Contoh : Tampilkan kolom nim, nama, alamat dari tabel mahasiswa yang kota asalnya “Semarang” dan tahun lahirnya 1999

SQL

```
SELECT nim, nama, alamat FROM mahasiswa WHERE  
kota_asal="Semarang" AND tahun_lahir=1999
```



Aljabar  
Relasional

```
 $\pi$  nim, nama, alamat ( $\sigma$  kota_asal="Semarang"  $\wedge$  tahun_lahir=1999 (mahasiswa))
```

# Union (U)

- Union merupakan operasi untuk menggabungkan tabel dengan syarat tabel memiliki atribut dan struktur yang sama
- Sintaks :

E1 U E2

# Union (U)

- Contoh : gabungkan data kota dari tabel customer dan tabel supplier

SQL

```
SELECT kota FROM customer  
UNION  
SELECT kota FROM supplier
```



Aljabar  
Relasional

$$\pi \text{ kota } (\text{customer}) \cup \pi \text{ kota } (\text{supplier})$$

# Union (U)

- Contoh : gabungkan data kota dari tabel customer dan tabel supplier

SQL

```
SELECT kota FROM customer WHERE kota="Semarang"  
UNION  
SELECT kota FROM supplier WHERE kota="Jakarta"
```



Aljabar  
Relasional

$$\pi \text{ kota } (\sigma \text{ kota="Semarang" } (\text{customer})) \cup \pi \text{ kota } (\sigma \text{ kota="Jakarta" } (\text{supplier}))$$

# Set-Difference (-)

- Digunakan untuk mendapatkan record yang berada pada suatu tabel namun tidak ada di tabel lainnya
- Mysql tidak memiliki perintah SQL spesifik terhadap operasi ini
- Sebaliknya, Oracle memiliki perintah MINUS untuk menjalankan operasi ini
- Sintaks :

E1-E2

# Set-Difference (-)

Tabel Customer

Kota
Semarang
Pekalongan
Solo

Customer - Supplier

Kota
Semarang

Menampilkan kota yang ada di tabel Customer tapi tidak ada di tabel supplier

Tabel Supplier

Kota
Solo
Jakarta
Pekalongan

Supplier - Customer

Kota
Jakarta

Menampilkan kota yang ada di tabel supplier tapi tidak ada di tabel customer

# Set-Difference (-)

ORACLE

```
(SELECT kota FROM customer)  
MINUS  
(SELECT kota FROM supplier)
```

MySQL

```
SELECT kota FROM customer WHERE kota NOT IN  
(SELECT kota FROM supplier)
```



Aljabar  
Relasional

$$\pi \text{ kota}_{(\text{customer})} - \pi \text{ kota}_{(\text{supplier})}$$



# Cartesian-Product (X)

- Operasi ini digunakan merelasikan semua record-record yang berasal dari dua tabel
- Operasi cartesian product umumnya tidak berdiri sendiri karena dapat digunakan bersama dengan operasi selection ( $\sigma$ ) dan projection ( $\pi$ )
- Sintaks :

E1 X E2

# Masih ingat ini ?

Tabel Dokter

kd_dokter	nama_dokter	kd_spesialis	telepon	gender
D01	dr.Arief	UMM	0814262728299	L
D04	dr. Sarah	JTG	083383373772	P
D02	dr.Agung	DLM	085363738339	L
D03	dr.Bambang	BDH	082628282626	L

Tabel Spesialis

kd_spesialis	spesialis
ANK	Anak
BDH	Bedah
DLM	Penyakit Dalam
GIG	Gigi
JTG	Jantung
KDG	Kandungan
KLT	Kulit
MAT	Mata
SRF	Saraf
THT	THT
UMM	Dokter Umum

## Tujuannya :

Ingin mengetahui Dokter X adalah spesialis apa ?

## Query yang dilakukan :

```
select dokter.nama_dokter, spesialis.spesialis
from dokter, spesialis
```

## HASILNYA :

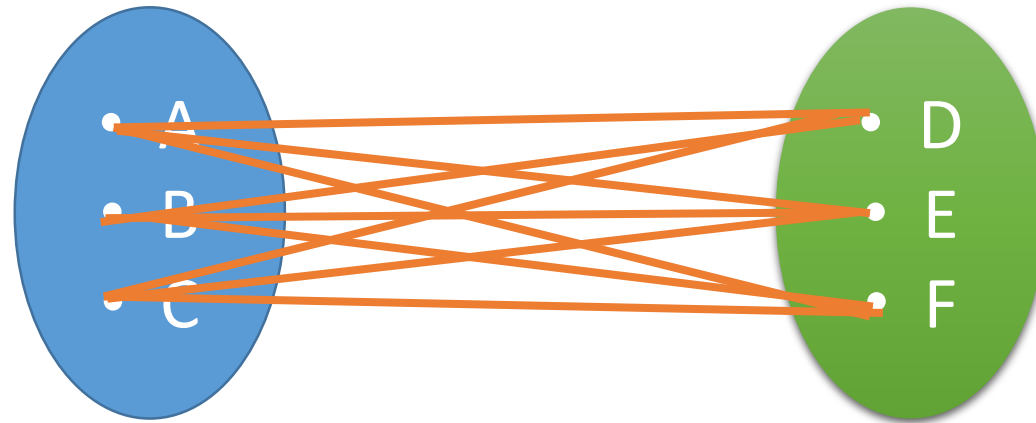
nama_dokter	spesialis
dr.Arief	Anak
dr.Agung	Anak
dr.Bambang	Anak
dr. Sarah	Anak
dr.Arief	Bedah
dr.Agung	Bedah
dr.Bambang	Bedah
dr. Sarah	Bedah
dr.Arief	Penyakit Dalam
dr.Agung	Penyakit Dalam
dr.Bambang	Penyakit Dalam
dr. Sarah	Penyakit Dalam
dr.Arief	Gigi
dr.Agung	Gigi
dr.Bambang	Gigi

Ada 44 Rows

WHY  
??? !!

# Apa yang terjadi ?

- Data pada tabel dokter akan dipetakan dengan data yang mana pada kolom spesialis di tabel spesialis
- Logikanya :



Semua elemen pada himpunan BIRU **akan dipetakan semua** ke elemen pada himpunan HIJAU  
Jumlah total pemetaannya adalah sejumlah = Jumlah elemen himpunan BIRU X jumlah elemen pada himpunan HIJAU

$$= 3 \times 3$$

$$= 9$$

# Cartesian-Product (X)

- Kasus tadi menggambarkan operasi pemetaan atau yang disebut Cartesian-product (cross product) pada DBMS

# Cartesian-Product (X)

Contoh : Tampilkan seluruh data yang ada di tabel dokter dan tabel spesialis yang spesialisnya "BEDAH"

SQL

```
SELECT * FROM dokter, spesialis WHERE  
dokter.kd_spesialis=spesialis.kd_spesialis AND  
spesialis.spesialis="BEDAH"
```



Aljabar  
Relasional

$\sigma \text{ dokter.kd\_spesialis=spesialis.kd\_spesialis} \wedge \text{spesialis.spesialis="BEDAH"} \text{ (dokterXspesialis)}$

# Rename ( $\rho$ )

- Operasi ini digunakan untuk memberikan nama alias terhadap :
  - Tabel
  - Kolom
  - Keduanya
- Sintaks :

$$\rho_x(R)$$

# Rename ( $\rho$ )

Contoh 1: Memberikan nama alias mhs pada **tabel** mahasiswa

SQL

```
SELECT * FROM mahasiswa AS mhs
```



Aljabar  
Relasional

$\rho$  mhs (mahasiswa)

# Rename ( $\rho$ )

Contoh 2: Memberikan nama alias **kolom** nama, alamat pada tabel mahasiswa dengan A dan B

SQL

```
SELECT nama AS A, alamat AS B FROM mahasiswa
```



Aljabar  
Relasional

$$\rho_{A,B} (\pi_{\text{nama, alamat}} (\text{mahasiswa}))$$



# Rename ( $\rho$ )

Contoh 3: Memberikan nama alias pada **tabel** mahasiswa menjadi mhs & **kolom** nama, alamat menjadi A dan B

SQL

SELECT nama AS A, alamat AS B FROM mahasiswa AS mhs



Aljabar  
Relasional

$\rho$  mhs(A,B) ( $\pi$  nama, alamat (mahasiswa))