

## Modul 4 : Aljabar dan Kalkulus Relasional

### 4.1 Waktu Pelaksanaan Praktikum

Durasi kegiatan praktikum = **170 menit**, dengan rincian sebagai berikut (misalkan):

1. 10 menit untuk penjelasan singkat tentang modul
2. 100 menit untuk pengayaan
3. 60 menit pembahasan

### 4.2 Tujuan

Setelah mengikuti praktikum ini mahasiswa diharapkan dapat:

1. Memahami dan mampu membuat Entity Relationship Diagram)

### 4.3 Alat & Bahan

1. Komputer

### 4.4 Dasar Teori

Bahasa Formal adalah bahasa *query* yang diterjemahkan dengan menggunakan simbol-simbol matematis. Bahasa formal ini terbagi dalam dua kelompok besar, yaitu prosedural dan non prosedural. Bahasa *query* prosedural adalah bahasa yang mengharuskan penggunaanya menspesifikasikan data apa yang dibutuhkan dan bagaimana untuk mendapatkannya. Bahasa *query* non prosedural adalah bahasa yang mengharuskan penggunaanya menspesifikasikan data apa yang dibutuhkan tanpa menspesifikasikan bagaimana untuk mendapatkannya.

Pada suatu aplikasi *database query* yang digunakan sangat kompleks, sehingga diperlukan pengetahuan yang mendalam mengenai bahasa dan semantiknya. Bahasa formal ini digunakan untuk memperoleh *query* paling efisien. Terdapat dua jenis bahasa formal, yaitu:

1. Aljabar relasional.
2. Kalkulus relasional.

Operasi dasar pada relational model adalah *Relational Algebra* atau aljabar relasional. Aljabar relasional merupakan kumpulan operasi terhadap relasi dan setiap operasi menggunakan satu atau lebih relasi untuk menghasilkan satu relasi yang baru. Aljabar relasional ini berfungsi untuk:

- Aljabar relasional memberikan fondasi formal untuk operasi model relasional.
- Aljabar relasional digunakan sebagai basis untuk implementasi dan optimasi *query* pada RDMSs (Relational Database Management Systems).
- Aljabar relasional digunakan bersama dengan SQL untuk RDMSs.

Aljabar relasional merupakan kunci pemahaman kerja internal RDBMs yang digunakan untuk mengolah *query* SQL secara lebih efisien. Pemrogram menggunakan *query* SQL, sedangkan DBMS menggunakan aljabar relasional sebagai bahasa perantara untuk menspesifikasikan algoritma *query*. Langkah-langkah dalam DBMS untuk pengolahan *query* adalah:

1. DBMS melakukan parsing terhadap string dari *query* SQL dan menerjemahkannya menjadi ekspresi aljabar relasional.
2. Ekspresi aljabar relasional ini akan dikonversi oleh *query optimizer* menjadi ekspresi lain yang ekuivalen namun lebih efisien untuk dieksekusi.
3. Berdasarkan ekspresi aljabar relasional yang telah dioptimasi ini, *query execution plan* disiapkan

oleh *query optimizer* yang kemudian akan ditransformasikan menjadi kode yang dapat dieksekusi pembangkit kode di DBMS. Karena ekspresi aljabar mempunyai semantik matematika yang presisi maka sistem dapat memverifikasi ekivalensi ekspresi yang dioptimasi.

Aljabar relasional tergolong kategori procedural yang menyediakan seperangkat operator untuk memanipulasi data. Terdapat tiga kelompok operasi dasar dalam aljabar relasional, yaitu:

- Unary relational operations
  - Selection
  - Projection
- Set theory operations
  - Union
  - Intersection
  - Minus
  - Cartesian product
- Binary relational operations
  - Natural join
  - Theta join
  - Division

Semua operasi tersebut menghasilkan relasi baru. Bahasa formal ini disebut sebagai aljabar relasional karena bahasa ini berdasar pada sejumlah operator yang beroperasi pada relasi-relasi (tabel-tabel). Masing-masing operator beroperasi pada satu relasi atau lebih atau menghasilkan relasi-relasi lain sebagai hasil. *Query* adalah ekspresi yang melibatkan operator-operator itu. Hasil ekspresi adalah relasi yang merupakan jawaban terhadap *query*.

#### 4.4.1 Unary Relational Operational

##### 4.4.1.1 Selection ( $\sigma$ )

Operasi ini bertujuan untuk memilih tuple-tuple yang sesuai dengan kondisi atau predikat P tertentu yang diinginkan pada suatu relasi R. Kondisi atau predikat P pada operasi selection ini adalah ekspresi logika yang terdiri dari:

- Operand : konstanta/atribut/relasi
- Operator aritmatika :  $<$ ,  $=$ ,  $>$ ,  $\geq$ ,  $\neq$ ,  $\leq$
- Operator logika :  $\text{and}(\wedge)$ ,  $\text{or}(\vee)$  dan  $\text{not}(\sim)$

Contoh sintaksis pada operasi selection ini adalah:

Bila diketahui R1 :

A	B	C
a	b	c
d	e	f
e	b	f

Maka  $\sigma_{B='b'}(R1) =$

A	B	C
a	b	c
e	b	f

#### Contoh Query:

Skema relasi MAHASISWA (nim, nama, tgl\_lhr, alamat, email, telpon)

Dicari informasi mengenai mahasiswa yang mempunyai nim = '0641345'

$\delta_{nim='0641345'}$  (MAHASISWA)

#### 4.4.1.2 Projection ( $\pi$ )

Operasi ini bertujuan untuk memilih atribut-atribut yang sesuai dengan kondisi atau predikat P tertentu yang diinginkan pada suatu relasi R. Contoh sintaksis pada operasi projection ini adalah:

Bila diketahui R1 :

A	B	C
a	b	c
d	e	f
g	h	i

Maka  $\Pi_{A,C}(R1) =$

A	C
a	c
d	f
g	i

#### Contoh Query:

Skema relasi MAHASISWA (nim, nama, tgl\_lhr, alamat, kota, email, telpon) Dicari informasi mengenai nama dan kota mahasiswa

$\Pi_{nama,kota}(MAHASISWA)$

#### 4.4.2 Set Teori Operational

##### 4.4.2.1 Union ( $\cup$ )

Operasi union ini bertujuan untuk membentuk suatu relasi yang terdiri dari tupel – tupel yang berada pada salah satu relasi atau pada kedua relasi, dengan syarat :

- Memiliki aritas yang sama
- Memiliki domain atribut yang sama

Contoh sintaksis pada operasi union ini adalah:

Bila diketahui R1 dan R2:

R1

A	B	C
a	b	c
d	e	f

R2

A	B	C
c	f	A
a	b	c

Maka  $R1 \cup R2 =$

A	B	C
a	b	c
d	e	f
c	f	a

#### Contoh Query:

Skema relasi MAHASISWA (nim, nama, tgl\_lahir, alamat, email, telpon)

Skema relasi MATA KULIAH (kode\_mk, nama\_mk, sks)

Gabungan data dari relasi MAHASISWA dengan data dari relasi MATA KULIAH

$MAHASISWA \cup MATA KULIAH$

#### 4.4.2.2 Intersection ( $\cap$ )

Operasi union ini bertujuan untuk membentuk suatu relasi yang terdiri atas tuple –tuple yang sama dari dua relasi, dengan syarat:

- Memiliki aritas yang sama
  - Memiliki domain atribut yang sama
- Contoh sintaksis pada operasi intersection ini adalah:

Bila diketahui R1 dan R2:

R1

A	B	C
a	b	c
d	e	f

R2

A	B	C
c	f	a
a	b	c

Maka  $R1 \cap R2 =$

A	B	C
a	b	c

#### Contoh Query:

Skema relasi MATA KULIAH ( kode\_mk, nama\_mk, sks )

Skema relasi NILAI( nim, kode\_mk, n\_uts, n\_uas )

Dicari kode mata kuliah yang mempunyai sks = 2 yang diambil oleh mahasiswa dengan NIM "0641345"

$\Pi_{\text{kode\_mk}} ( \sigma_{\text{sks} = 2} (\text{MATAKULIAH}) ) \cap \Pi_{\text{kode\_mk}} ( \sigma_{\text{nim} = '0641345'} (\text{NILAI}) )$

#### 4.4.2.3 Cartesian Product ( $\times$ )

Operasi cartesian product ini bertujuan untuk membentuk suatu relasi dari dua relasi yang terdiri dari kombinasi tuple-tuple yang mungkin. Jika aritas R1 adalah k1 dan aritas R2 adalah k2 maka  $R1 \times R2$  adalah kumpulan kombinasi semua tuple-tuple dengan aritas (k1+k2) dengan komponen k1 pertama adalah tuple-tuple dari R1 dan komponen berikutnya dari R2. Simbol ini adalah  $\times$  bukan huruf x.

R1

A	B	C
1	d	f
3	c	g
4	b	a

R2

E	F
n	15
m	25

Maka  $R1 \times R2 =$

A	B	C	E	F
1	d	f	n	15
3	c	g	n	15
4	b	a	n	15
1	d	f	m	25
3	c	g	m	25
4	b	a	m	25

#### Contoh Query:

Skema relasi MAHASISWA (nim, nama, tgl\_lahir, alamat, email, telpon)

Skema relasi MATA KULIAH (kode\_mk, nama\_mk, sks)

MAHASISWA X MATA KULIAH

#### 4.4.2.4 Set Difference(-)

Operasi set difference ini bertujuan untuk membentuk suatu relasi yang terdiri dari tupel-tupel yang berada pada relasi pertama dan tidak berada pada relasi kedua atau kedua-duanya.

Contoh sintaksis pada operasi set difference ini adalah:

Bila diketahui R1 dan R2:

R1

A	B	C
a	b	c
d	e	f
c	b	g

R2

A	B	C
c	f	a
a	b	c
a	d	e

Maka  $R1 - R2 =$

A	B	C
d	e	f
c	b	g

#### Contoh Query :

Skema relasi MATA KULIAH (kode\_mk, nama\_mk, sks)

Skema relasi NILAI (nim, kode\_mk, n\_uts, n\_uas)

Dicari kode mata kuliah yang bersks 3 dan nilai uts untuk kode mata kuliah tersebut di bawah 50

$$\pi_{\text{kode\_mk}} (\delta_{\text{sks} = 3} (\text{MATA KULIAH})) - \pi_{\text{kode\_mk}} (\delta_{\text{n\_uts} > 50} (\text{NILAI}))$$

### 4.4.3 Binary Relational Operational

#### 4.4.3.1 Natural Join ( $\bowtie$ )

Operasi natural join ini bertujuan untuk membentuk suatu relasi dari dua relasi yang terdiri dari kombinasi yang mungkin dari relasi – relasi dengan syarat bahwa operasi ini dilakukan jika kedua relasi memiliki satu atau lebih atribut yang sama.

Bila diketahui R1 dan R2:

R1

A	B	C
a	b	c
d	b	c
c	e	f
d	b	h

R2

B	C	D
b	c	d
b	c	z
b	d	x

Maka  $R1 \bowtie R2 =$

A	B	C	D	A
a	b	c	d	a
a	b	c	z	a
d	b	c	d	d
d	b	c	z	d

#### Contoh Query:

Skema relasi MAHASISWA (nim, nama, tgl\_lahir, alamat, email, telpon)

Skema relasi NILAI ( nim, kode\_mk, n\_uts, n\_uas )

Dicari nama mahasiswa yang mengambil mata kuliah dengan kode mata kuliah “TIF123”

$$\pi_{\text{nama}} (\delta_{\text{kode\_mk} = \text{'TIF123'}} (\text{MAHASISWA} \bowtie \text{NILAI}))$$

#### 4.4.3.2 Theta Join( $\theta$ )

Operasi theta join ini bertujuan untuk membentuk suatu relasi dari dua relasi yang terdiri dari kombinasi yang mungkin dari relasi – relasi dengan kondisi tertentu. Contoh sintaksis pada operasi theta join ini adalah:

Bila diketahui R1 dan R2:

R1

A	B	C
1	3	a
4	7	b
8	2	c

R2

D	E
3	1
5	2

Maka  $R1 \theta R2 =$

A	B	C	D	E
1	3	a	3	1
1	3	a	5	2

#### Contoh Query:

Skema relasi MAHASISWA (nim, nama, tgl\_lahir, alamat, email, telpon)

Skema relasi NILAI ( nim, kode\_mk, n\_uts, n\_uas )

Dicari nama mata kuliah yang diambil oleh mahasiswa dengan NIM “0641345” dengan kode mata kuliah pada relasi nilai harus sama dengan relasi mata kuliah

$\Pi_{\text{nama}} ( \delta_{\text{kode\_mk} = \text{'TIF123'}} \wedge (\text{mata\_kuliah.kode\_mk} = \text{nilai.kode\_mk}) (\text{MATA KULIAH X NILAI}) )$

#### 4.4.3.3 Division ( $\div$ )

Operasi division ini bertujuan untuk mendapatkan nilai yang ada pada salah satu atribut dari relasi pembilang yang nilai atributnya sama dengan nilai atribut relasi penyebut, atau dengan kata lain semua tuple-tuple t dengan aritas k1 - k2, jika R1 mengandung semua tuple dengan aritas k1 maka t adalah anggota R1 dengan syarat jika k1 aritas R1 dan k2 aritas R2, maka  $k1 > k2$  dan  $k2 \neq 0$ .

Bila diketahui R1 dan R2:

R1

A	B	A
x	y	x
z	y	z
f	m	f
d	s	d

R2

B
y
s

Maka  $R1 \div R2 =$

A
x
z
d