



MODUL
DATABASE
(CCS120)

MODUL SESI IV

ALJABAR DAN KALKULUS RELASIONAL

DISUSUN OLEH

NOVIANDI, S.Kom, M.Kom

UNIVERSITAS ESA UNGGUL

2020

BAB IV

Aljabar dan Kalkulus Relasional

Tujuan

1. Arti dari istilah "kelengkapan relasional."
2. Bagaimana membentuk query dalam aljabar relasional.
3. Bagaimana membentuk query dalam tupel kalkulus relasional.
4. Bagaimana membentuk query dalam domain kalkulus relasional.
5. Kategori Relasional Data Manipulation Languages (DMLs).

Teori

Aljabar dan Kalkulus Relasional

Secara Informal

- Aljabar relasional dapat dikatakan sebagai bahasa procedural tingkat tinggi, karena dapat digunakan untuk memberi tahu DBMS bagaimana membangun relasi baru dari satu atau lebih relasi dalam database.
- Kalkulus relasional dapat dikatakan sebagai bahasa non-prosedural, karena dapat digunakan untuk merumuskan definisi relasi yang berkaitan dengan satu atau lebih relasi dalam database.

Secara Formal

- Aljabar dan kalkulus relasional setara satu sama lain: untuk setiap ekspresi dalam aljabar, ada ekspresi yang setara dalam kalkulus (dan sebaliknya).
- Aljabar dan kalkulus relasional *non-user-friendly languages*.
- Aljabar dan kalkulus relasional digunakan sebagai dasar untuk *Data Manipulation Languages* (DML) tingkat tinggi untuk database relasional, karena menggambarkan operasi dasar yang diperlukan dari setiap DML dan berfungsi sebagai standar perbandingan untuk bahasa relasional lainnya.
- Kalkulus relasional digunakan untuk mengukur kekuatan selektif bahasa relasional.

Aljabar Relasional

Aljabar relasional adalah bahasa teoritis dengan operasi yang bekerja pada satu atau lebih relasi untuk mendefinisikan relasi lain tanpa mengubah relasi aslinya.

Enam operasi dasar dalam aljabar relasional, yaitu:

1. Selection (σ)
2. Projection (π)
3. Cartesian product (\times)
4. Union (\cup)
5. Set Difference ($-$)
6. Rename (ρ)

Unary Operation

1. Selection atau Restriction (σ)



- Berguna untuk menampilkan data-data tertentu dari sebuah relasi
- Operasi Seleksi bekerja pada satu relasi R dan mendefinisikan relasi yang hanya berisi tupel R yang memenuhi kondisi yang ditentukan (predikat).

$\sigma_{\text{predicate}}(R)$

Keterangan:

- σ : Operator Seleksi
 predicate : Kondisi yang ingin di pilih/filter dalam menampilkan data dari suatu relasi
 R : relasi

Contoh 1:

Tampilkan semua **staff** dengan gaji lebih dari £10000.

$\sigma_{\text{gaji} > 1000}(\text{Staff})$

Keterangan:

- Staff merupakan relasi inputannya
- Gaji adalah predikatnya

Hasil dari perintah diatas adalah:

staffNo	fName	lName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	1-Oct-45	30000	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000	B003
SG14	David	Ford	Supervisor	M	24-Mar-58	18000	B003
SG5	Susan	Brand	Manager	F	3-Jun-40	24000	B003

Hasil dari *Selecting Gaji > 10000 dari Staff relation*

Proses ini akan menampilkan **seluruh data baris** yang memiliki gaji > 10000

Predikat yang lebih kompleks dapat dibuat dengan menggunakan operator logika (AND), (OR) dan ~(NOT).

Contoh 2:

Relation r

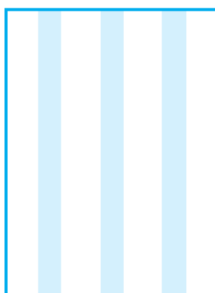
A	B	C	D
α	α	1	7
α	β	5	7
β	β	12	3
β	β	23	10

Tampilkan data nilai A=B **dan** D > 5

$$\sigma_{A=B \wedge D > 5}(r)$$

A	B	C	D
α	α	1	7
β	β	23	10

2. Projection (π)



- Operasi Proyeksi bekerja pada satu relasi R dan mendefinisikan relasi yang berisi subset vertical/kolom R, mengekstrak nilai atribut tertentu dan menghilangkan duplikat/redudansi.

$$\pi_{a_1, \dots, a_n}(R)$$

Contoh 1:

Buat daftar gaji untuk semua staf, yang hanya menampilkan detail staffNo, fName, lName, dan gaji.

$\pi_{staffNo, fName, lName, Gaji}(Staff)$

Hasil:

staffNo	fName	lName	salary
SL21	John	White	30000
SG37	Ann	Beech	12000
SG14	David	Ford	18000
SA9	Mary	Howe	9000
SG5	Susan	Brand	24000
SL41	Julie	Lee	9000

Contoh 2:

Relation r

A	B	C
α	10	1
α	20	1
β	30	1
β	40	2

Tampilkan nilai A dan C dari relation r

$\pi_{A,C}(r)$

A	C
α	1
α	1
β	1
β	2

=

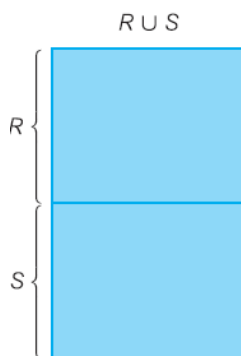
A	C
α	1
β	1
β	2

Karena pada operator ini akan menampilkan nilai kolom dan menghilangkan redundansi data.

Set Operations

Pada bagian ini memeriksa operasi biner dari aljabar relasional, yang dimulai dengan operasi *Union*, *Set difference*, *Intersection* dan *Cartesian product*

1. Union (\cup)



- Gabungan dua relasi R dan S mendefinisikan relasi yang berisi semua tupel R, atau S, atau keduanya R dan S, tupel duplikat dihilangkan. R dan S harus kompatibel dengan *Union-Compatible*.

Note:

- Perhatikan bahwa nama atribut tidak digunakan dalam mendefinisikan *Union-Compatible*.
- Dalam beberapa kasus, operasi Proyeksi dapat digunakan untuk membuat dua relasi *Union-Compatible*.

Contoh:

Tampilkan semua kota antara *branch office* atau *property for rent*

$$\pi_{kota}(\mathbf{Branch}) \cup \pi_{kota}(\mathbf{PropertyForRent})$$

Hasil:

Kota
London
Aberdeen
Glasgow
Bristol

2. Cartesian product (x)

P	Q	P × Q	
A	B	A	B
a	1	a	1
b	2	a	2
	3	a	3
		b	1
		b	2
		b	3

- Operasi produk Cartesian mendefinisikan relasi yang merupakan rangkaian dari setiap tupel relasi R dengan setiap tupel relasi S.

Contoh:

Buat daftar nama dan komentar dari semua klien yang telah melihat *property for rent*.

$$\left(\pi_{clientNo, fName, IName}(Client) \right) \times \left(\pi_{clientNo, propertyNo, comment}(Viewing) \right)$$

Hasil:

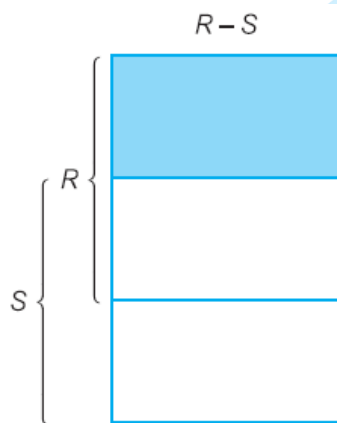
client,clientNo	fName	IName	Viewing.clientNo	propertyNo	comment
CR76	John	Kay	CR56	PA14	too small
CR76	John	Kay	CR76	PG4	too remote
CR76	John	Kay	CR56	PG4	
CR76	John	Kay	CR62	PA14	no dining room
CR76	John	Kay	CR56	PG36	
CR56	Aline	Stewart	CR56	PA14	too small
CR56	Aline	Stewart	CR76	PG4	too remote
CR56	Aline	Stewart	CR56	PG4	
CR56	Aline	Stewart	CR62	PA14	no dining room
CR56	Aline	Stewart	CR56	PG36	
CR74	Mike	Ritchie	CR56	PA14	too small
CR74	Mike	Ritchie	CR76	PG4	too remote
CR74	Mike	Ritchie	CR56	PG4	
CR74	Mike	Ritchie	CR62	PA14	no dining room
CR74	Mike	Ritchie	CR56	PG36	
CR62	Mary	Tregear	CR56	PA14	too small
CR62	Mary	Tregear	CR76	PG4	too remote
CR62	Mary	Tregear	CR56	PG4	
CR62	Mary	Tregear	CR62	PA14	no dining room
CR62	Mary	Tregear	CR56	PG36	

$$\sigma_{Client.clientNo=Viewing.clientNo} \left(\left(\pi_{clientNo, fName, IName}(Client) \right) \times \left(\pi_{clientNo, propertyNo, coment}(Viewing) \right) \right)$$

Hasil:

client.clientNo	fName	IName	Viewing.clientNo	propertyNo	comment
CR76	John	Kay	CR76	PG4	too remote
CR56	Aline	Stewart	CR56	PA14	too small
CR56	Aline	Stewart	CR56	PG4	
CR56	Aline	Stewart	CR56	PG36	
CR62	Mary	Tregear	CR62	PA14	no dining room

3. Set Difference (-)



- Operasi Perbedaan himpunan mendefinisikan relasi yang terdiri dari tupel-tupel yang ada dalam relasi R, tetapi tidak di S.

- R dan S harus kompatibel dengan union.

R - S

Contoh:

Buat daftar semua kota yang memiliki *branch office* tetapi tidak ada *property for rent*

$$\pi_{kota}(Branch) - \pi_{kota}(PropertyForRent)$$

Hasil:

Kota
Bristol

4. Intersection (∩)

- Operasi Intersection mendefinisikan relasi yang terdiri dari himpunan semua tupel yang ada di R dan S.
- R dan S harus kompatibel dengan union

Contoh:

Buat daftar semua kota di mana terdapat *branch office* dan setidaknya satu *property for rent*.

$$\pi_{kota}(Branch) \cap \pi_{kota}(PropertyForRent)$$

Hasil:

Kota
London
Aberdeen
Glasgow

Note:

Kita dapat mengekspresikan operasi Intersection dalam hal operasi Set difference:

Join Operations

Ada beberapa bentuk operasi join, yaitu:

1. Theta join
2. Equijoin (jenis tertentu dari Theta join)
3. Natural join
4. Outer join
5. Semi join

Theta join (θ – join)

- Mendefinisikan relasi yang berisi tupel yang memenuhi predikat F dari produk Cartesian dari R dan S.
- Predikat F adalah dari bentuk $R.a_i \theta S.b_i$, dimana θ dapat menjadi salah satu operator pembandingan ($<$, \leq , $>$, \geq , $=$, \neq).

$$R \bowtie_{\theta} S = \sigma_{\theta}(R \times S)$$

Contoh:

Buat daftar nama dan komentar dari semua klien yang telah melihat properti yang disewakan.

$$\left(\pi_{clientNo, fName, lName}(Client) \right) \bowtie_{Client.clientNo=Viewing.clientNo} \left(\pi_{clientNo, propertyNo, comment}(Viewing) \right)$$

Atau

$Result \leftarrow TempClient \bowtie_{TempClient.clientNo=TempViewing.clientNo} TempViewing$

Hasil:

client,clientNo	fName	lName	Viewing,clientNo	propertyNo	comment
CR76	John	Kay	CR56	PA14	too small
CR76	John	Kay	CR76	PG4	too remote
CR76	John	Kay	CR56	PG4	
CR76	John	Kay	CR62	PA14	no dining room
CR76	John	Kay	CR56	PG36	
CR56	Aline	Stewart	CR56	PA14	too small
CR56	Aline	Stewart	CR76	PG4	too remote
CR56	Aline	Stewart	CR56	PG4	
CR56	Aline	Stewart	CR62	PA14	no dining room
CR56	Aline	Stewart	CR56	PG36	
CR74	Mike	Ritchie	CR56	PA14	too small
CR74	Mike	Ritchie	CR76	PG4	too remote
CR74	Mike	Ritchie	CR56	PG4	
CR74	Mike	Ritchie	CR62	PA14	no dining room
CR74	Mike	Ritchie	CR56	PG36	
CR62	Mary	Tregear	CR56	PA14	too small
CR62	Mary	Tregear	CR76	PG4	too remote
CR62	Mary	Tregear	CR56	PG4	
CR62	Mary	Tregear	CR62	PA14	no dining room
CR62	Mary	Tregear	CR56	PG36	

Natural Join

- Natural join adalah Equijoin dari dua relasi R dan S pada semua atribut umum x.
- Satu kemunculan setiap atribut umum dieliminasi dari hasil.

Notasi Natural join adalah:

$$R \bowtie S$$

Contoh:

$$\left(\pi_{clientNo, fName, IName}(Client)\right) \bowtie \left(\pi_{clientNo, propertyNo, comment}(Viewing)\right)$$

Atau

$$Result \leftarrow TempClient \bowtie TempViewing$$

Hasil:

clientNo	fName	IName	propertyNo	comment
CR76	John	Kay	PG4	too remote
CR56	Aline	Stewart	PA14	too small
CR56	Aline	Stewart	PG4	
CR56	Aline	Stewart	PG36	
CR62	Mary	Tregear	PA14	no dining room

Aggregation dan Grouping Operations

Aggregate operations

- Menerapkan daftar fungsi agregat, AL, ke relasi R untuk menentukan relasi di atas daftar agregat.
- AL berisi satu atau lebih pasangan (<aggregate function>, <attribute>).

AL(R)

Fungsi utama aggregate adalah:

- COUNT : Mengembalikan jumlah nilai dalam atribut terkait
- SUM : Mengembalikan jumlah nilai dalam atribut terkait
- AVG : Mengembalikan rata-rata nilai dalam atribut terkait
- MIN : Mengembalikan nilai terkecil dalam atribut terkait
- MAX : Mengembalikan nilai terbesar dalam atribut terkait

Grouping operation

- Mengelompokkan tupel relasi R menurut atribut pengelompokan, GA, dan kemudian menerapkan daftar fungsi agregat AL untuk menentukan relasi baru.
- AL berisi satu atau lebih pasangan (<aggregate_function>, <attribute>). Relasi yang dihasilkan berisi atribut pengelompokan, GA, bersama dengan hasil dari masing-masing fungsi agregat.

$$GA\ AL(R)$$

Kalkulus Relasional

- Dalam kalkulus relasional, tidak ada penjelasan tentang cara mengevaluasi query.
- Query kalkulus relasional menentukan apa yang akan diambil alih dan bagaimana mengambilnya.
- Kalkulus relasional tidak terkait dengan kalkulus diferensial dan integral dalam matematika, tetapi mengambil namanya dari cabang logika simbolik yang disebut kalkulus predikat.
- Ketika diterapkan pada basis data, ia ditemukan dalam dua bentuk: kalkulus **relasional tupel**, seperti yang semula dikemukakan oleh Codd (1972a), dan kalkulus **relasional domain**, seperti yang dikemukakan oleh Lacroix dan Pirotte (1977).

Kalkulus Relasional Tupel

- Kalkulus relasional tupel adalah bahasa query yang non prosedural. Bahasa ini mendeskripsikan informasi yang diinginkan tanpa memberi prosedurnya secara detil untuk mendapatkan informasi tersebut.
- Kalkulus relasional tupel merupakan basis untuk bahasa query QUEL.

Query pada kalkulus relasional tupel di ekspresikan dengan:

$$\{T \mid P(t)\}$$

Dimana,

T : Merupakan nilai yang ingin dicari

P : Merupakan representasi table/entitas yang berkaitan

(t) : Merupakan query yang ingin dieksekusi untuk dihasilkan

$$\{ T \mid \exists E \in \text{Entitas} (T.\text{Tuple1} = E.\text{Tuple1} \wedge E.\text{Tuple1} = \text{'String'}) \}$$

Untuk kondisi seperti diatas kita tau bahwa, t merupakan nilai yang di cari, E merupakan ekspresi dari sebuah entitas (sehingga diwakilkan dalam bentuk char E), dan isi query yang di eksekusi berisi keterangan-keterangan termasuk persilangan jodoh **t yang di cari** dengan **e sebuah entitas** dan sisanya di wakilkkan oleh operator and (^) untuk memberikan kondisi yang ingin dicari.

Tabel artikel aljabar relational

MAHASISWA			
NPM	NamaMhsw	Jurusan	NoTelp
07001	Arham Answar	Sistem Informasi	087833537766
07002	Dimas Sanjaya	Sistem Informasi	085367830998
07003	Roy Suryo	Teknik Informatika	081354387294
07004	Alfa Farhan	Teknik Informatika	081355289467
07005	Sintia Amelia	Akuntansi	085699670343
07006	Siswo Kusnomo	Management	087888947205

MATAKULIAH			
KodeMatkul	NamaMatkul	SKS	KodeDosen
TSI200	Sistem Informasi Manajemen	3	D101
TSI240	SMBD	3	D220
TSI300	Jaringan Komputer	3	D224
AKT100	Dasar Akutansi	3	D315
AKT200	Management Keuangan	3	D315

NILAI				
NPM	KodeMatkul	Semester	Tahun	Nilai
07001	TSI200	Ganjil	2018	A
07001	TSI240	Ganjil	2018	B
07003	TSI300	Ganjil	2018	B
07003	TSI240	Ganjil	2018	C
07005	TSI200	Genap	2018	A
07005	AKT100	Genap	2018	C
07005	AKT200	Genap	2018	C

Contoh 1:

Tampilkan NPM mahasiswa yang mendapat nilai A. Maka kurang lebih query kalkulus relational tuple seperti berikut ini.

$$\{ T \mid \exists N \in \text{Nilai} (T.NPM = N.NPM \wedge N.Nilai = 'A') \}$$

NPM
07001
07005

Contoh 2:

Tampilkan NPM, KodeMatkul, dan Nilai dari mahasiswa yang memperoleh nilai A. Maka kurang lebih query kalkulus relational tuplenya seperti berikut ini.

$$\{ T \mid \exists N \in \text{Nilai} (T.NPM = N.NPM \wedge T.KodeMatkul = N.KodeMatkul \wedge T.Nilai = N.Nilai \wedge N.Nilai = 'A') \}$$

NPM	KodeMatkul	Nilai
07001	AKT100	A
07005	TSI240	A

Contoh 3:

Tampilkan NPM, Nama, KodeMatkul, NamaMatkul yang memperoleh nilai A. Maka kurang lebih query kalkulus relational tuplenya akan seperti berikut ini.

$$\{ T \mid \exists M \in \text{Mahasiswa} \exists K \in \text{Matakuliah} \exists N \in \text{Nilai} (M.NPM = N.NPM \wedge K.KodeMatkul = N.KodeMatkul \wedge T.NPM = M.NPM \wedge T.Nama = M.Nama \wedge T.KodeMatkul = K.KodeMatkul \wedge T.NamaMatkul = K.NamaMatkul \wedge N.Nilai = 'A') \}$$

NPM	Nama	KodeMatkul	NamaMatkul
07001	Arham Ansvar	AKT100	Dasar Akuntansi
07005	Sintia Amelia	TSI240	SMBD

Kalkulus Relasional Domain

- Kalkulus relasional domain juga adalah bahasa query yang non prosedural dan karenanya berhubungan dekat dengan kalkulus relasional tupel.
- Berbeda dengan kalkulus relasional tupel, bahasa ini menggunakan variabel domain yang mengambil nilai dari domain atribut, bukan dari nilai seluruh tupel.
- Kalkulus relasional domain merupakan basis untuk bahasa query QBE

Latihan

1. Apa perbedaan antara bahasa prosedural dan nonprocedural?
Bagaimana Anda mengklasifikasikan aljabar relasional dan kalkulus relasional?
2. Jelaskan istilah-istilah berikut:
 - a. Kalkulus relasional tupel
 - b. Kalkulus relasional domain
3. Definisikan lima operasi aljabar relasional dasar. Tentukan operasi Join dan Intersection.
4. Diskusikan perbedaan antara lima operasi join: Theta join, Equijoin, Natural join, Outer join, dan Semijoin.
Berikan contoh untuk menggambarkan jawaban Anda.

5. Ada berbagai jenis operasi join yang dapat digunakan untuk mengambil data, berdasarkan relasi yang berbeda. Jelaskan hubungan antara theta dan equal join.

6. Apa perbedaan antara bilangan eksistensial dan universal dalam kalkulus relasional?

Berikan contoh untuk menjelaskan bagaimana keduanya diterapkan dalam pernyataan.

7. Jelaskan notasi-notasi berikut ini

- $\pi_{kota}(Branch) \cup \pi_{kota}(PropertyForRent)$
- $(\pi_{clientNo, fName, IName}(Client)) \times (\pi_{clientNo, propertyNo, comment}(Viewing))$
- $\pi_{kota}(Branch) - \pi_{kota}(PropertyForRent)$
- $(\pi_{clientNo, fName, IName}(Client)) \bowtie (\pi_{clientNo, propertyNo, comment}(Viewing))$