

Model-Model Data

Khabib Mustofa
khabib@ugm.ac.id

Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas
Gadjah Mada

August 30, 2007



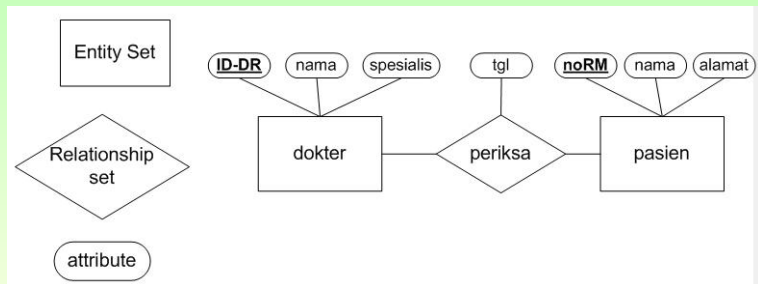
Outline

- 1 Model Entitas-Hubungan (Entity Relationship Model)
 - Pendahuluan
 - Terminologi
 - Instance suatu Himpunan Relasi
 - Constraint (Batasan) dan Cardinality
- 2 Model Relasional (*Relational Model*)
 - Pengantar
 - Karakteristik
- 3 Bahasa Query (*Query Language*)
 - Aljabar Relasional
 - Structured Query Language (SQL)
 - Korespondensi antara Aljabar Relasional dengan SQL
SELECT



E-R Model

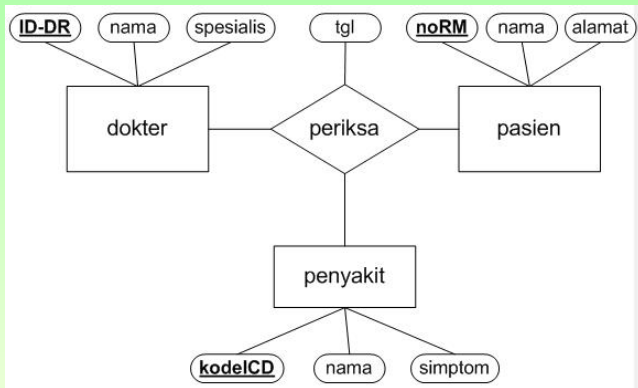
Model Entity-relationship memungkinkan penggambaran data yang terlibat dalam suatu organisasi sebagai kumpulan obyek-obyek dan hubungan antar mereka. → tahap awal desain / perancangan basis data.



Gambar: Simbol Dasar dan Contoh E-R diagram



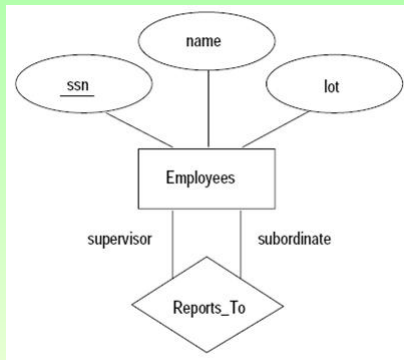
Relasi non-biner



Gambar: Relasi terner (memiliki tiga hubungan)



Roles / Peran



Gambar: Seorang pegawai (Employee) bisa memiliki peran yang berbeda pada suatu relasi



Pengertian Istilah

entity an object in the real world that is distinguishable from other objects

entity set a collection of similar entities

relationship an association among two or more entities

relationship set a collection of similar relationships. Himpunan relasi dapat dipandang sebagai sekumpulan *n-tuple* (n pasangan nilai) $f(e_1; \dots; e_n) | e_1 \in E_1; \dots; e_n \in E_n$

attribute a characteristic which describes an entity

domain a set of permitted or possible values for an attribute

instance a 'snapshot' (view) at an instant in time



Pengertian Istilah

entity an object in the real world that is distinguishable from other objects

entity set a collection of similar entities

relationship an association among two or more entities

relationship set a collection of similar relationships. Himpunan relasi dapat dipandang sebagai sekumpulan *n-tuple* (n pasangan nilai) $f(e_1; \dots; e_n) | e_1 \in E_1; \dots; e_n \in E_n$

attribute a characteristic which describes an entity

domain a set of permitted or possible values for an attribute

instance a 'snapshot' (view) at an instant in time



Pengertian Istilah

entity an object in the real world that is distinguishable from other objects

entity set a collection of similar entities

relationship an association among two or more entities

relationship set a collection of similar relationships. Himpunan relasi dapat dipandang sebagai sekumpulan *n-tuple* (n pasangan nilai) $f(e_1; \dots; e_n) | e_1 \in E_1; \dots; e_n \in E_n$

attribute a characteristic which describes an entity

domain a set of permitted or possible values for an attribute

instance a 'snapshot' (view) at an instant in time



Pengertian Istilah

entity an object in the real world that is distinguishable from other objects

entity set a collection of similar entities

relationship an association among two or more entities

relationship set a collection of similar relationships. Himpunan relasi dapat dipandang sebagai sekumpulan *n-tuple* (n pasangan nilai) $f(e_1; \dots; e_n) | e_1 \in E_1; \dots; e_n \in E_n$

attribute a characteristic which describes an entity

domain a set of permitted or possible values for an attribute

instance a 'snapshot' (view) at an instant in time



Pengertian Istilah

entity an object in the real world that is distinguishable from other objects

entity set a collection of similar entities

relationship an association among two or more entities

relationship set a collection of similar relationships. Himpunan relasi dapat dipandang sebagai sekumpulan *n-tuple* (n pasangan nilai) $f(e_1; \dots; e_n) | e_1 \in E_1; \dots; e_n \in E_n$

attribute a characteristic which describes an entity

domain a set of permitted or possible values for an attribute

instance a 'snapshot' (view) at an instant in time



Pengertian Istilah

entity an object in the real world that is distinguishable from other objects

entity set a collection of similar entities

relationship an association among two or more entities

relationship set a collection of similar relationships. Himpunan relasi dapat dipandang sebagai sekumpulan *n-tuple* (n pasangan nilai) $f(e_1; \dots; e_n) | e_1 \in E_1; \dots; e_n \in E_n$

attribute a characteristic which describes an entity

domain a set of permitted or possible values for an attribute

instance a 'snapshot' (view) at an instant in time



Kunci (*key*)

- 1 Kunci Super (*Super key*) \Rightarrow (kumpulan) atribut yang dapat mengidentifikasi unik entitas-entitas tapi tak harus minimal
- 2 Kunci Calon (*Candidate Key*) \Rightarrow (kumpulan) atribut yang dapat mengidentifikasi unik entitas-entitas dan harus minimal
- 3 Kunci Primer (*Primary Key*) \Rightarrow salah satu kunci calon yang dipilih sebagai *identifying attribute*
- 4 Kunci Tamu (*Foreign Key*) \Rightarrow (kumpulan) atribut yang mengacu ke (kumpulan) atribut kunci pada tabel yang lain. Ini digunakan untuk kunci perelasian.
- 5 Kunci Alternatif (*Alternate Key*) \Rightarrow kunci calon yang tidak dipilih sebagai kunci primer

Semua atribut yang berperan sebagai kunci tidak boleh kosong (*null*). Khusus untuk kunci tamu, selain tak boleh kosong, nilainya juga harus sudah ada dalam atribut kunci tabel yang diacu.



Kunci (*key*)

- 1 Kunci Super (*Super key*) \Rightarrow (kumpulan) atribut yang dapat mengidentifikasi unik entitas-entitas tapi tak harus minimal
- 2 Kunci Calon (*Candidate Key*) \Rightarrow (kumpulan) atribut yang dapat mengidentifikasi unik entitas-entitas dan harus minimal
- 3 Kunci Primer (*Primary Key*) \Rightarrow salah satu kunci calon yang dipilih sebagai *identifying attribute*
- 4 Kunci Tamu (*Foreign Key*) \Rightarrow (kumpulan) atribut yang mengacu ke (kumpulan) atribut kunci pada tabel yang lain. Ini digunakan untuk kunci perelasian.
- 5 Kunci Alternatif (*Alternate Key*) \Rightarrow kunci calon yang tidak dipilih sebagai kunci primer

Semua atribut yang berperan sebagai kunci tidak boleh kosong (*null*). Khusus untuk kunci tamu, selain tak boleh kosong, nilainya juga harus sudah ada dalam atribut kunci tabel yang diacu.



Kunci (*key*)

- 1 Kunci Super (*Super key*) \Rightarrow (kumpulan) atribut yang dapat mengidentifikasi unik entitas-entitas tapi tak harus minimal
- 2 Kunci Calon (*Candidate Key*) \Rightarrow (kumpulan) atribut yang dapat mengidentifikasi unik entitas-entitas dan harus minimal
- 3 Kunci Primer (*Primary Key*) \Rightarrow salah satu kunci calon yang dipilih sebagai *identifying attribute*
- 4 Kunci Tamu (*Foreign Key*) \Rightarrow (kumpulan) atribut yang mengacu ke (kumpulan) atribut kunci pada tabel yang lain. Ini digunakan untuk kunci pereliasan.
- 5 Kunci Alternatif (*Alternate Key*) \Rightarrow kunci calon yang tidak dipilih sebagai kunci primer

Semua atribut yang berperan sebagai kunci tidak boleh kosong (*null*). Khusus untuk kunci tamu, selain tak boleh kosong, nilainya juga harus sudah ada dalam atribut kunci tabel yang diacu.



Kunci (*key*)

- 1 Kunci Super (*Super key*) \Rightarrow (kumpulan) atribut yang dapat mengidentifikasi unik entitas-entitas tapi tak harus minimal
- 2 Kunci Calon (*Candidate Key*) \Rightarrow (kumpulan) atribut yang dapat mengidentifikasi unik entitas-entitas dan harus minimal
- 3 Kunci Primer (*Primary Key*) \Rightarrow salah satu kunci calon yang dipilih sebagai *identifying attribute*
- 4 Kunci Tamu (*Foreign Key*) \Rightarrow (kumpulan) atribut yang mengacu ke (kumpulan) atribut kunci pada tabel yang lain. Ini digunakan untuk kunci perelasian.
- 5 Kunci Alternatif (*Alternate Key*) \Rightarrow kunci calon yang tidak dipilih sebagai kunci primer

Semua atribut yang berperan sebagai kunci tidak boleh kosong (*null*). Khusus untuk kunci tamu, selain tak boleh kosong, nilainya juga harus sudah ada dalam atribut kunci tabel yang diacu.



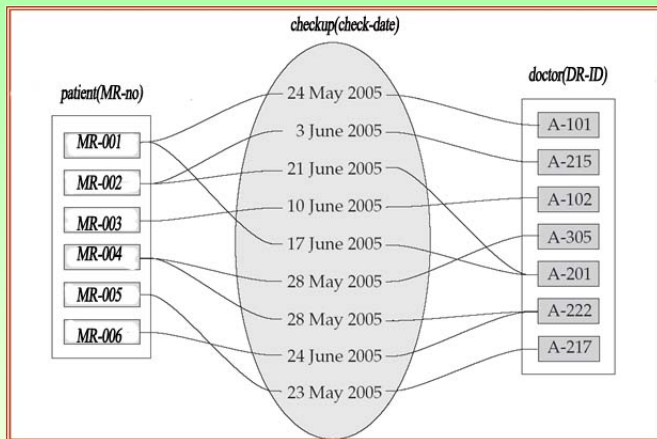
Kunci (*key*)

- 1 Kunci Super (*Super key*) \Rightarrow (kumpulan) atribut yang dapat mengidentifikasi unik entitas-entitas tapi tak harus minimal
- 2 Kunci Calon (*Candidate Key*) \Rightarrow (kumpulan) atribut yang dapat mengidentifikasi unik entitas-entitas dan harus minimal
- 3 Kunci Primer (*Primary Key*) \Rightarrow salah satu kunci calon yang dipilih sebagai *identifying attribute*
- 4 Kunci Tamu (*Foreign Key*) \Rightarrow (kumpulan) atribut yang mengacu ke (kumpulan) atribut kunci pada tabel yang lain. Ini digunakan untuk kunci perelasian.
- 5 Kunci Alternatif (*Alternate Key*) \Rightarrow kunci calon yang tidak dipilih sebagai kunci primer

Semua atribut yang berperan sebagai kunci tidak boleh kosong (*null*). Khusus untuk kunci tamu, selain tak boleh kosong, nilainya juga harus sudah ada dalam atribut kunci tabel yang diacu.



Instance of relationship set *CHECKUP*



Gambar: Tabel Contoh nilai-nilai dalam himpunan relasi CHECKUP



Instance of relationship set *CHECKUP*

MR-no	DR-ID	checkup-date
MR-001	A-101	24-May- 2005
MR-001	A-201	17-June- 2005
MR-002	A-215	03-June- 2005
MR-002	A-201	21-June- 2005
MR-003	A-102	10-June- 2005
MR-004	A-305	28-May- 2005
MR-004	A-222	28-May- 2005
MR-005	A-217	23-May- 2005
MR-006	A-222	24-June- 2005

Gambar: Diagram Contoh nilai-nilai dalam himpunan relasi CHECKUP



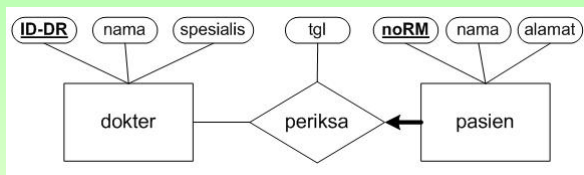
Key Constraint batasan yang berlaku pada suatu atribut kunci, terutama dalam kaitannya dengan relasi. Contoh: seorang pasien harus sudah pernah memeriksakan diri. Dengan demikian dalam himpunan relasi PERIKSA, harus ada catatan mengenai pasien, yang diwakili oleh keberadaan atribut No-RM. Jadi setiap nilai No-RM harus muncul pada himpunan relasi PERIKSA

Participation Constraint batasan yang menggambarkan partisipasi suatu himpunan entitas pada suatu himpunan relasi.

- **total participation** bila setiap entitas harus muncul pada himpunan relasi (minimal sekali). Contoh : entitas PASIEN pada relasi PERIKSA
- **partial participation** bila tidak setiap entitas harus muncul. Contoh : entitas DOKTER pada himpunan PERIKSA. Artinya tidak setiap dokter pernah memeriksa pasien.



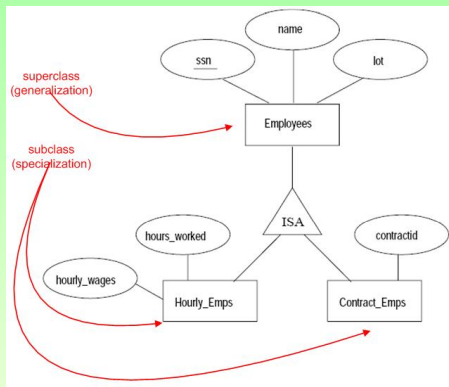
Partisipasi Total



Gambar: Partisipasi Total himpunan entitas PASIEN pada himpunan relasi PERIKSA



Hirarki Kelas



Gambar: Contoh gambaran hirarki klas

- Covering Constraint
- Overlap Constraint

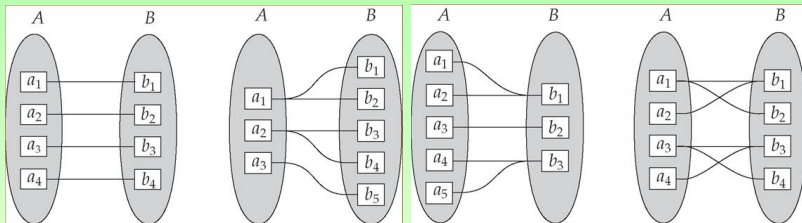


Kardinalitas Pemetaan (*Mapping Cardinality*)

- express the number of entities to which another entity can be associated via a relationship set
- Most useful in describing binary relationship sets.
- For a binary relationship set the mapping cardinality must be one of the following types:
 - One to one
 - One to many
 - Many to one
 - Many to many



Ilustrasi Kardinalitas Pemetaan



(a) One-to-one dan one-to-many

(b) many-to-one dan many-to-many

Gambar: Mapping Cardinality

Note: Some elements in A and B may not be mapped to any elements in the other set



Outline

- 1 Model Entitas-Hubungan (*Entity Relationship Model*)
 - Pendahuluan
 - Terminologi
 - Instance suatu Himpunan Relasi
 - Constraint (Batasan) dan Cardinality
- 2 Model Relasional (*Relational Model*)
 - Pengantar
 - Karakteristik
- 3 Bahasa Query (*Query Language*)
 - Aljabar Relasional
 - Structured Query Language (SQL)
 - Korespondensi antara Aljabar Relasional dengan SQL SELECT



Pengertian Dasar

Dalam relational model suatu **database** dipandang sebagai kumpulan satu atau lebih **relasi**¹ / *tabel*, dimana setiap relasi adalah berupa sebuah tabel dua dimensi yang terdiri atas *baris-baris* dan *kolom-kolom*. Kelebihan utama model relasional dibanding model data lainnya adalah kesederhanaan representasi dan kemudahannya dimana dengan kesederhanaannya ini queri yang kompleks pun bisa diformulasikan.



¹beda dengan pengertian RELASI pada E-R model

Pengertian Dasar

Relasi (dalam pengertian model relasional) terdiri atas dua elemen:

- Instance : sebuah tabel tersusun atas baris dan kolom.
cacah baris = cardinality, cacah atribut = derajat (*degree, arity*).
- Schema : menyatakan nama relasi, plus nama dan tipe masing-masing kolom.
Contoh Patients(sid: string, name: string, login: string, age: integer, address: string).

Sebuah domain diacu dalam suatu skema relasi dengan menyebutkan nama domainnya.



Pengertian

Sebuah *instance* dari suatu relasi adalah suatu himpunan tuple-tuple (*records*), dimana setiap tuple memiliki cacah atribut yang sama seperti pada skema relasinya.

Sebuah *instance* dari relasi dapat dipandang sebagai suatu tabel dimana setiap tuple adalah suatu baris, dan setiap baris memiliki cacah field yang sama.



Pengertian

Secara formal, bila diberikan himpunan D_1, D_2, \dots, D_n , suatu relasi r adalah himpunan bagian dari $D_1 \times D_2 \times \dots \times D_n$. Dengan demikian, suatu relasi adalah himpunan yang terdiri atas n -tuple (a_1, a_2, \dots, a_n) dimana setiap $a_i \in D_i$

Jika $\text{nama_pelanggan} = \{\text{Arjo, Joni, Budi, Linda}\}$

$\text{jalan_pelanggan} = \{\text{Utama, Utara, Taman}\}$

$\text{kota_pelanggan} = \{\text{Yogyakarta, Tanjungpinang, Bima}\}$

Maka $r = \{(\text{Arjo, Utama, Bima}),$

$(\text{Budi, Utara, Yogyakarta}),$

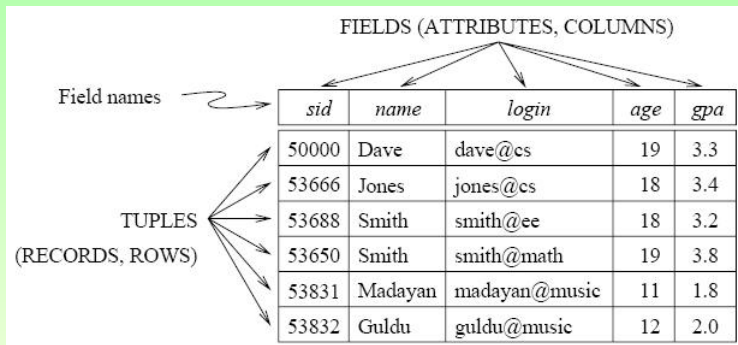
$(\text{Joni, utara, Tanjungpinang}),$

$(\text{Linda, Taman, Yogyakarta})\}$ adalah sebuah relasi atas

$\text{nama_pelanggan} \times \text{jalan_pelanggan} \times \text{kota_pelanggan}$



Representasi Relasi sebagai tabel dua dimensi



Gambar: Salah satu 'instance' dari relasi STUDENT



Karakteristik suatu relasi/tabel

- secara teoritis tidak ada baris yang persis sama (karena dipandang sebagai himpunan tuple-tuple)
- urutan baris/record tidak membedakan suatu relasi
- bila setiap kolom memiliki nama, urutan kemunculan kolom juga tidak membedakan suatu relasi



Outline

- 1 Model Entitas-Hubungan (*Entity Relationship Model*)
 - Pendahuluan
 - Terminologi
 - Instance suatu Himpunan Relasi
 - Constraint (Batasan) dan Cardinality

- 2 Model Relasional (*Relational Model*)
 - Pengantar
 - Karakteristik

- 3 Bahasa Query (*Query Language*)
 - Aljabar Relasional
 - Structured Query Language (SQL)
 - Korespondensi antara Aljabar Relasional dengan SQL SELECT



Pengertian dan Pembagian bahasa query

Bahasa query (*query language*) adalah bahasa yang dapat dipergunakan oleh pengguna untuk mendapatkan atau merubah informasi dari suatu basis data

Bahasa query:

- procedural : user harus tahu urutan langkah-langkah dalam mendapatkan informasi \Rightarrow Aljabar Relasional (*Relational Algebra*)
- non procedural : user tak perlu tahu urutan langkah dalam pengeksekusian queri \Rightarrow Structured Query Language (SQL)



Operasi-operasi pada relasi

Operasi pada relasi selalu menghasilkan relasi juga, artinya bila dua relasi/tabel dioperasikan dengan operator aljabar relasional maka hasilnya juga berupa relasi/tabel

- 1 Cartesian Product (\times)
- 2 select (σ)
- 3 project (π)
- 4 union (\cup)
- 5 set difference ($-$)
- 6 set intersection (\cap)
- 7 join dan *natural join* (\bowtie)



Operasi CARTESIAN PRODUCT

- Notasi ; rs , dimana $rs = \{pq | p \in r \text{ dan } q \in s\}$, yaitu hasilnya berupa relasi/tabel yang diperoleh dengan memasang setiap tuple pada relasi pertama dengan setiap tuple pada relasi kedua.
- banyaknya kolom sama dengan jumlah kolom kedua relasi / tabel
- banyaknya baris / record sama dengan perkalian banyaknya record pada relasi pertama dan record pada relasi kedua

A	B
α	1
β	2

r

C	D	E
α	10	a
β	10	a
β	20	b
γ	10	b

s

A	B	C	D	E
α	1	α	10	a
α	1	β	10	a
α	1	β	20	b
α	1	γ	10	b
β	2	α	10	a
β	2	β	10	a
β	2	β	20	b
β	2	γ	10	b

(a) Relasi r dan s (b) Relasi rs

Gambar: Cartesian Product



Operasi SELECT

yaitu operasi mengambil baris-baris (tuple-tuple) yang memenuhi kriteria tertentu.

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

Tabel: Relasi r

Bagaimana kalau hanya ingin yang memenuhi kriteria $A = B$ dan $D > 5$?

A	B	C	D
α	α	1	7
β	β	23	10

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



Operasi SELECT

yaitu operasi mengambil baris-baris (tuple-tuple) yang memenuhi kriteria tertentu.

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

Tabel: Relasi r

Bagaimana kalau hanya ingin yang memenuhi kriteria $A = B$ dan $D > 5$?

A	B	C	D
α	α	1	7
β	β	23	10

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



operasi SELECT (lanjutan)

- Notasinya $\sigma_{\rho}(r)$
- ρ disebut predikat seleksi (*selection predicate*)
- Definisi $\sigma_{\rho}(r) = \{t | t \in r \text{ and } p(t)\}$, yaitu berupa himpunan tuple-tuple t dimana t anggota r dan bila predikat p dievaluasi terhadap t menghasilkan nilai benar
- p adalah suatu pernyataan yang dapat terdiri atas suku-suku yang dihubungkan dengan tanda \wedge (and), \vee (or) atau \neg (not). Setiap pernyataan berbentuk
 $\langle \textit{attribute} \rangle \textit{op} \langle \textit{attribute} \rangle$ atau
 $\langle \textit{attribute} \rangle \textit{op} \langle \textit{constant} \rangle$
dan \textit{op} adalah salah satu dari simbol $=, <, >, \geq, \leq, \neq$



Review tentang nilai kebenaran dalam matematika dan logika komputer

X	Y	X AND Y	X OR Y
F	F	F	F
F	T	F	T
T	F	F	T
T	T	T	T

Tabel: Nilai kebenaran untuk operator AND dan OR

X	$\neg X$
F	T
T	F

Tabel: Nilai kebenaran untuk operator NOT



Operasi PROJECT

- untuk mengambil kolom-kolom tertentu dari suatu relasi/tabel
- Notasi : $\pi_{daftar_atribut}(r)$ dimana *daftar_atribut* adalah atribut-atribut yang ada pada relasi/tabel *r* yang akan ditampilkan
- Hasil operasi berupa kolom-kolom yang ada pada *r* dengan menghilangkan semua kolom yang tidak ada dalam *daftar_atribut*
- bila ada baris yang sama (terduplikasi), cukup dituliskan sekali (hilangkan sisanya)



Contoh Operasi PROJECT

A	B	C
α	10	1
α	20	1
β	30	1
β	40	2

Tabel: Relasi r

A	C		A	C
α	1		α	1
α	1	=	β	1
β	1		β	2
β	2			

Gambar: Hasil $\pi_{A,C}(r)$



Operasi UNION

- Notasi : $r \cup s$, dimana $r \cup s = \{t | t \in r \text{ atau } t \in s\}$, yaitu berupa relasi yang anggota-anggotanya adalah tuple-tuple yang berasal dari r atau s . Bila ada yang terduplikasi, cukup satu saja yang dipertahankan
- Syarat operasi dapat dilangsungkan
 - 1 r dan s harus memiliki cacah kolom (*arity*) yang sama
 - 2 masing-masing kolom dari kedua relasi harus kompatibel, artinya nilai-nilai di dalamnya harus berasal dari domain yang sama.



Operasi Selisih Himpunan (SET DIFFERENCE)

- Notasi : $r - s$, dimana $r - s = \{t | t \in r \text{ dan } t \notin s\}$, yaitu berupa relasi yang anggota-anggotanya adalah tuple-tuple yang berasal dari r tetapi tidak di dalam s .
- Syarat operasi dapat dilangsungkan
 - 1 r dan s harus memiliki cacah kolom (*arity*) yang sama
 - 2 masing-masing kolom dari kedua relasi harus kompatibel, artinya nilai-nilai di dalamnya harus berasal dari domain yang sama.



Operasi Irisan Himpunan (SET INTERSECTION)

- Notasi : $r \cap s$, dimana $r \cap s = \{t | t \in r \text{ dan } t \in s\}$, yaitu berupa relasi yang anggota-anggotanya adalah tuple-tuple yang ada dalam r dan ada dalam s .
- Syarat operasi dapat dilangsungkan
 - 1 r dan s harus memiliki cacah kolom (*arity*) yang sama
 - 2 masing-masing kolom dari kedua relasi harus kompatibel, artinya nilai-nilai di dalamnya harus berasal dari domain yang sama.



Contoh Operasi UNION, SET DIFFERENCE dan INTERSECTION

A	B
α	1
α	2
β	1

r

A	B
α	2
β	3

s

Gambar: Relasi *r* dan *s*

A	B
α	1
α	2
β	1
β	3

(a) $r \cup s$

A	B
α	1
β	1

(b) $r - s$

A	B
α	2

(c) $r \cap s$



Gabungan Beberapa Operasi

rxs

A	B	C	D	E
α	1	α	10	a
α	1	β	10	a
α	1	β	20	b
α	1	γ	10	b
β	2	α	10	a
β	2	β	10	a
β	2	β	20	b
β	2	γ	10	b

$\sigma_{A=C}(rxs)$

A	B	C	D	E
α	1	α	10	a
β	2	β	10	a
β	2	β	20	b

Gambar: Hasil Operasi $\sigma_{A=C}(rxs)$



Join

adalah operasi gabungan antara Cartesian Product dengan SELECT berdasar kriteria tertentu.

Contoh Operasi komposit sebelumnya adalah contoh operasi JOIN.

conditional join adalah join yang berdasar pada kriteria tertentu yang bukan operasi persamaan nilai atribut

inner join adalah join yang berdasar pada kriteria tertentu berupa persamaan nilai atribut

natural join adalah operasi inner join yang dilakukan pada semua atribut yang sama dari kedua relasi (nama dan domainnya). Pada tabel/relasi hasil, kolom-kolom yang sama hanya muncul sekali



Contoh Operasi Conditional Join

A	B
α	1
α	2
β	1

r

A	B
α	2
β	3

s

Gambar: Relasi *r* dan *s*

r.A	r.B	s.A	s.B
α	1	α	2
α	1	β	3
α	2	β	3
β	1	α	2
β	1	β	3

Tabel: $\sigma_{r.B < s.B}(rxs)$



Contoh Operasi Inner Join

A	B
α	1
α	2
β	1

r

A	B
α	2
β	3

s

Gambar: Relasi *r* dan *s*

r.A	r.B	s.A	s.B
α	1	α	2
α	2	α	2
β	1	β	3

Tabel: $\sigma_{r.A=s.A}(r \times s)$



Contoh Operasi Natural Join

A	B
α	1
α	2
β	1

r

A	B
α	2
β	3

s

Gambar: Relasi *r* dan *s*

A	B
α	2

Tabel: Hasil Operasi $r \bowtie s = \pi_{r.A, r.B}(\sigma_{r.A=s.A \wedge r.B=s.B}(r \times s))$



Contoh Natural Join

A	B	C	D
α	1	α	a
β	2	γ	a
γ	4	β	b
α	1	γ	a
δ	2	β	b

r

B	D	E
1	a	α
3	a	β
1	a	γ
2	b	δ
3	b	ϵ

s

Gambar: Relasi *r* dan *s*

A	B	C	D	E
α	1	α	a	α
α	1	α	a	γ
α	1	γ	a	α
α	1	γ	a	γ
δ	2	β	b	δ

Gambar: Hasil operasi $r \bowtie s = \pi_{r.A,r.B,r.C,r.D,s.E}(\sigma_{r.B=s.B \wedge r.D=s.D}(r \times s))$



Komposisi SQL

- **Data Definition Language**, adalah perintah yang dipergunakan untuk melakukan manipulasi pada struktur basis data, baik pembuatan ataupun perubahan (domain, integrity constraints, security). Contoh: CREATE, DROP, ALTER, GRANT
- **Data Manipulation Language**, adalah perintah yang digunakan untuk memanipulasi data di dalam basis data. Contoh : INSERT, DELETE, SELECT, UPDATE



SQL SELECT

- adalah perintah untuk mengambil / membaca data yang ada dalam database.
- struktur umumnya

```
SELECT daftar_atribut
FROM daftar_tabel
WHERE predikat_penyaringan
```

dimana

daftar_atribut atribut yang nilai-nilainya akan ditampilkan sebagai hasil seleksi

daftar_tabel tabel-tabel yang (sebagian) datanya akan ditampilkan. Tabel yang disebutkan seharusnya merupakan tabel-tabel yang saling berelasi.

predikat_penyaringan predikat atau kondisi yang disertakan untuk "membuang" data-data yang tidak ingin ditampilkan. Kondisi ini biasanya berisi kondisi pereliasian antar tabel dan kondisi penyaringan record-record yang tidak ingin ditampilkan

- Struktur umum di atas ekuivalen dengan ekapresi Aljabar Relasional

$$\pi_{\text{daftar_atribut}}(\sigma_{\text{predikat_penyaringan}}(\text{daftar_tabel}))$$


Rangkuman Korespondensi

Operasi Alj. Rel.	SQL SELECT	Link
rxs	SELECT * FROM r,s	2
$\sigma_{predikat_seleksi}(r)$	SELECT * FROM r WHERE predikat_seleksi	3
$\pi_{A,B,C}(r)$	SELECT A,B,C FROM r	7
$\pi_{A,B,C}(\sigma_{predikat_seleksi}(r))$	SELECT A,B,C FROM r WHERE predikat_seleksi	
$r \cup s$	(SELECT * FROM r) UNION (SELECT * FROM s)	11
$r \cap s$	SELECT * FROM r WHERE (A,B) IN (SELECT * FROM s) → bila r dan s sama-sama hanya memiliki kolom A dan B	11
$r - s$	SELECT * FROM r WHERE (A,B) NOT IN (SELECT * FROM s) → bila r dan s sama-sama hanya memiliki kolom A dan B	11

Tabel: Ekuivalensi antara Perintah Aljabar Relasional dengan SQL SELECT



Contoh Kasus Sebelumnya

- pasien(noRM, namaPasien, Alamat, Telp, namaIbuPasien)
- dokter(NIP, namaDokter, spesialisasi)
- obat(kodeObat, namaObat, jenisObat)
- penyakit(kodePenyakit, namaPenyakit)
- pemeriksaan(noRM, tgl_periksa, dokter, kodePenyakit, noResep)
- apoteker(kodeApt, nama_Apt)
- resep(noResep, tglResep, penanggungjawab)
- obatpasien(noResep, obat, dosis)

