



# **PENGANTAR TEKNOLOGI SISTEM INFORMASI MANAGEMENT 2**

**PERTEMUAN 3 & 4**

**MODEL DATA RELASIONAL**

# SAP

- Pengertian model relasional, contoh tabel–tabel dan keterhubungannya
- Keuntungan model relasional
- Istilah–istilah dalam model relasional (Relasi, Atribut, Tupel, Domain, Derajat dan Cardinality)
- Relasional keys (Super key, Candidate Key, Primary Key, Alternatif.
- Relasional Integrity rules (Null, Entity, Referential Integrity)
- Bahasa pada model relasional
- Overview bahasa query formal (Aljabar relasional dan Kalkulus relasional )
- Overview bahasa query komersial (QUEL, QBE dan SQL)
- QBE



# PENGERTIAN MODEL RELASIONAL

- Basis Data relasional menggunakan tabel dua dimensi .
- yang terdiri atas baris dan kolom untuk memberi gambaran sebuah berkas data.



# CONTOH TABEL DAN KETERHUBUNGANNYA :

Nilai

<b>NPM</b>	<b>KDMK</b>	<b>MID</b>	<b>FINAL</b>
<b>10296832</b>	<b>KK021</b>	<b>60</b>	<b>75</b>
<b>10296126</b>	<b>KD132</b>	<b>70</b>	<b>90</b>
<b>31296500</b>	<b>KK021</b>	<b>55</b>	<b>40</b>
<b>41296525</b>	<b>KU122</b>	<b>90</b>	<b>80</b>
<b>21196353</b>	<b>KU122</b>	<b>75</b>	<b>75</b>
<b>50095487</b>	<b>KD132</b>	<b>80</b>	<b>0</b>
<b>10296832</b>	<b>KD132</b>	<b>40</b>	<b>30</b>

MAHASISWA

Mata Kuliah

<b>KDMK</b>	<b>MTKULIAH</b>	<b>SKS</b>
<b>KK021</b>	<b>P. Basis Data</b>	<b>2</b>
<b>KD132</b>	<b>SIM</b>	<b>3</b>
<b>KU122</b>	<b>Pancasila</b>	<b>2</b>

<b>NPM</b>	<b>Nama</b>	<b>Alamat</b>
<b>10296832</b>	<b>Nurhayati</b>	<b>Jakarta</b>
<b>10296126</b>	<b>Astuti</b>	<b>Jakarta</b>
<b>31296500</b>	<b>Budi</b>	<b>Depok</b>
<b>41296525</b>	<b>Prananingrum</b>	<b>Bogor</b>
<b>50096487</b>	<b>Pipit</b>	<b>Bekasi</b>
<b>21196353</b>	<b>Quraish</b>	<b>Bogor</b>

# KEUNTUNGAN BASIS DATA RELASIONAL

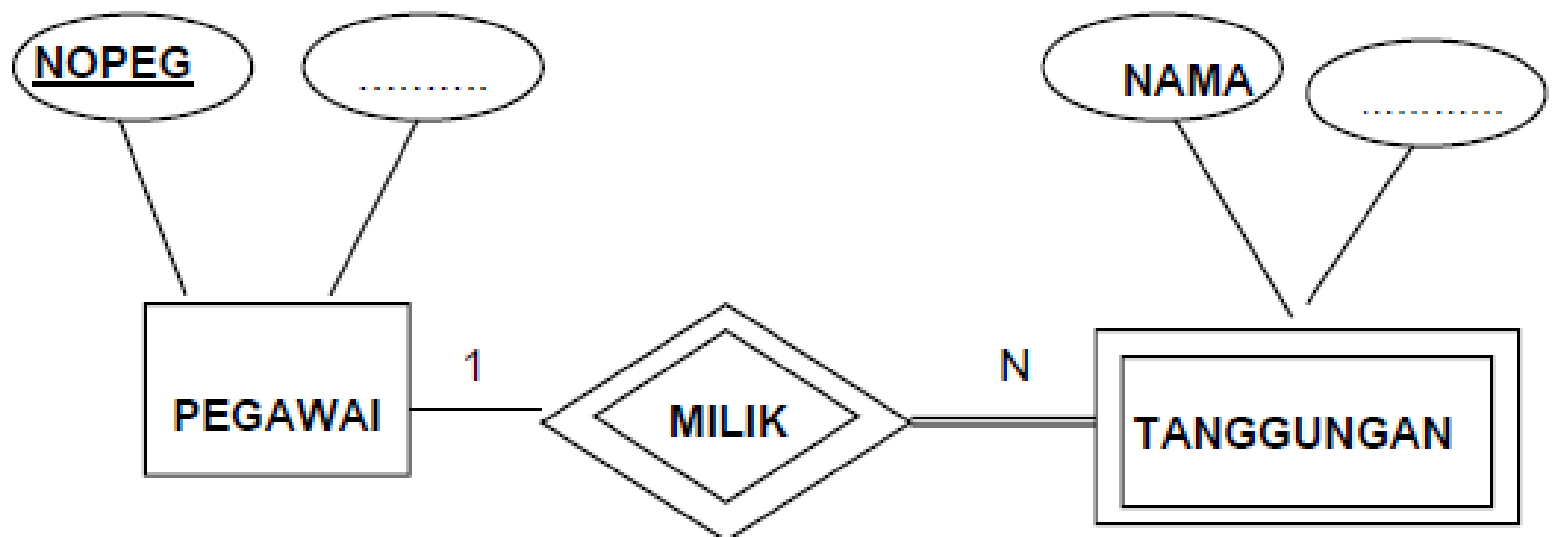
- Bentuknya sederhana
- Mudah melakukan berbagai operasi data
- Pekerjaan Programmer lebih mudah



# ISTILAH DALAM BASIS DATA RELASIONAL

## o *Entitas*

Weak Entity adalah suatu Entity dimana keberadaan dari entity tersebut tergantung dari keberadaan entity lain. Entity yang merupakan induknya disebut *Identifying Owner* dan *relationshipnya* disebut *Identifying Relationship*. *Weak Entity* selalu mempunyai *Total Participation constraint* dengan *Identifying Owner*.



- *Relasi:*

Sebuah tabel yang terdiri dari beberapa kolom dan beberapa baris.

- *Atribut:*

Kolom pada sebuah relasi.

- Jenis-jenis atribut :

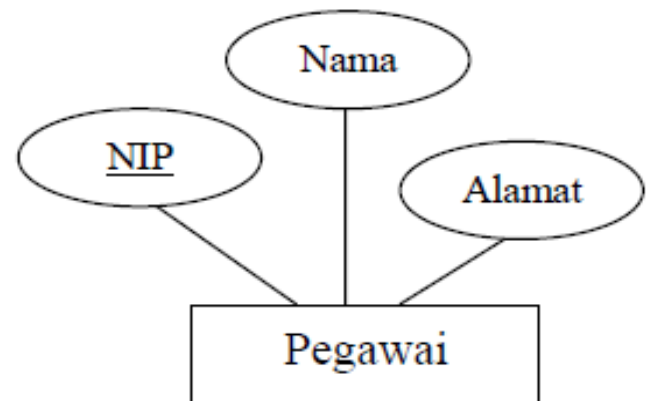
- Atribut key □ digunakan untuk mengidentifikasi suatu entity

secara unik

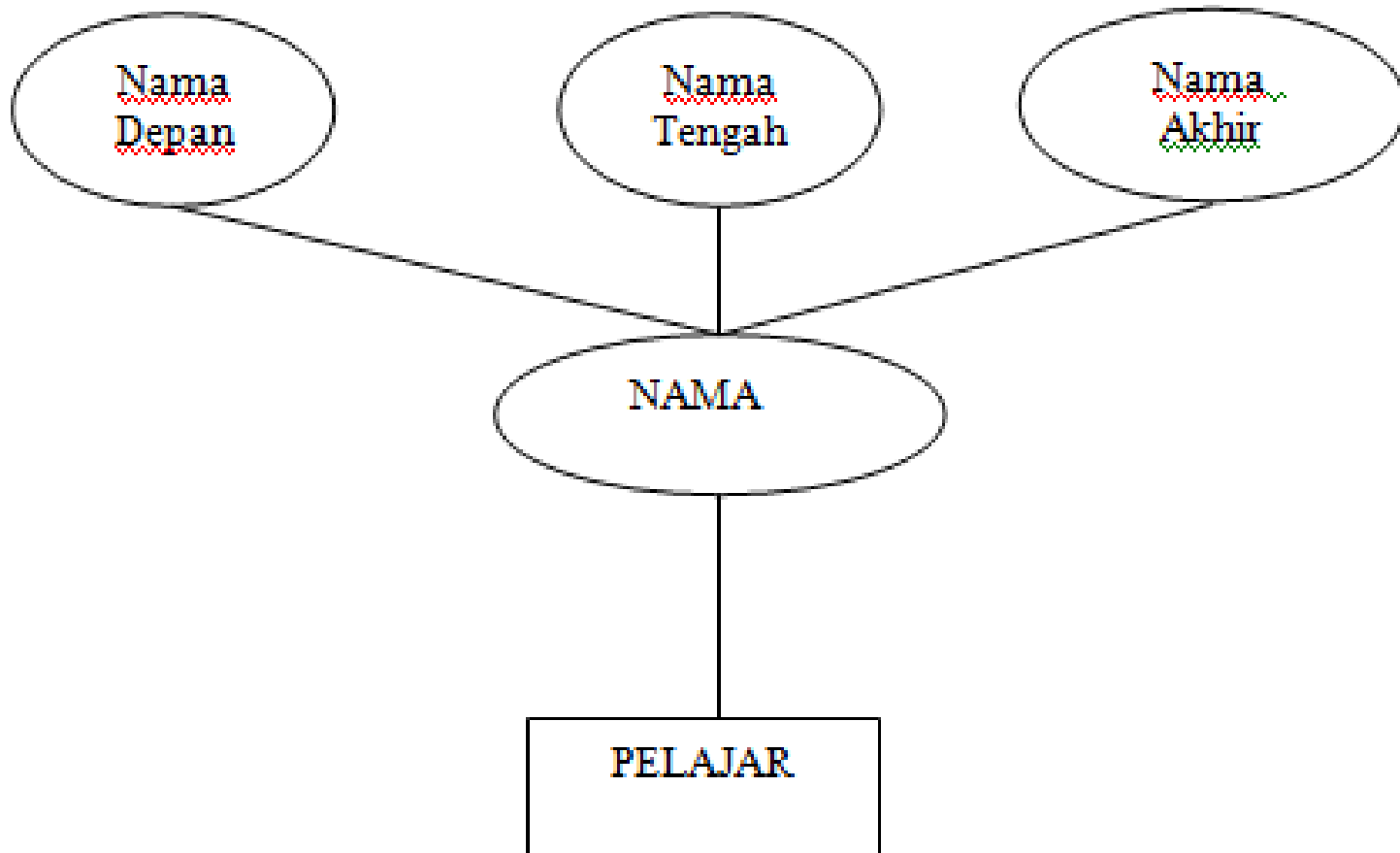
- Atribut tunggal memiliki nilai tunggal

---

Atribut tunggal/ Atribut biasa

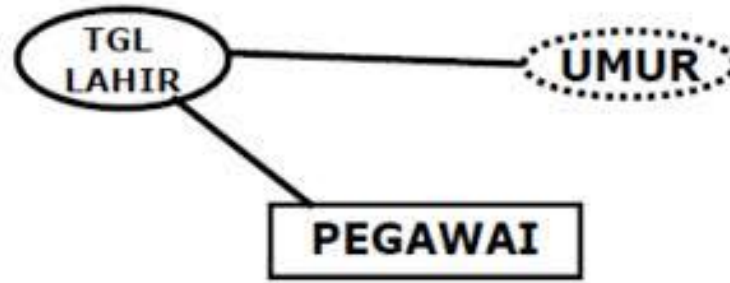


- Atribut composite □ dapat didekomposisi menjadi beberapa atribut lain

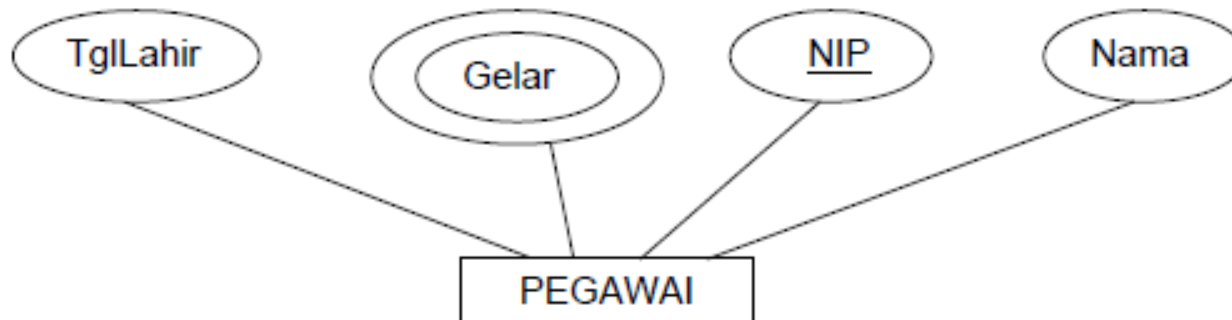




- Atribut derivative □ dihasilkan dari atribut yang lain



- Atribut multivalued □ memiliki sekelompok nilai untuk setiap instant entity



- *Tupel / Record*

Baris pada sebuah relasi

- *Domain*

Kumpulan nilai yang valid untuk satu atau lebih atribut

- *Derajat (degree)*

Jumlah atribut dalam sebuah relasi. Menjelaskan jumlah entity yang terlibat dalam suatu relationship

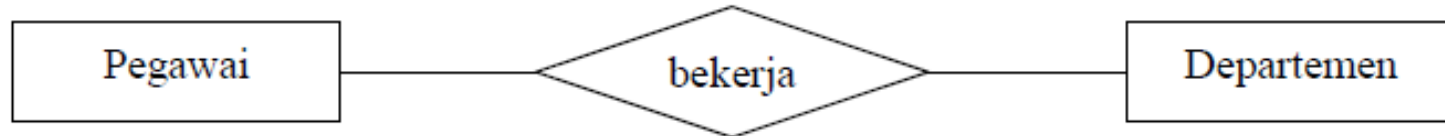


o Menjelaskan jumlah entity yang terlibat dalam suatu relationship

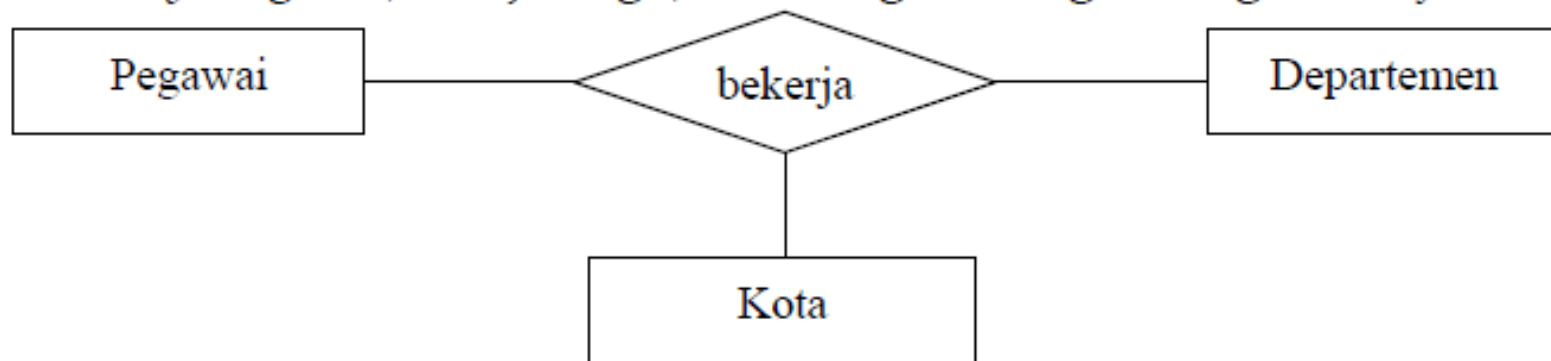
- *Unary Degree* (Derajat satu) → hanya satu entity yang terlibat



- *Binary Degree* (Derajat dua) → menghubungkan dua entity



- *Ternary Degree* (Derajat tiga) → menghubungkan tiga entity

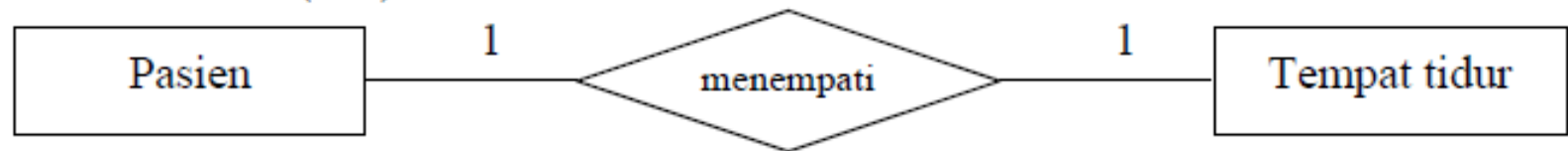


## ○ *Cardinality*

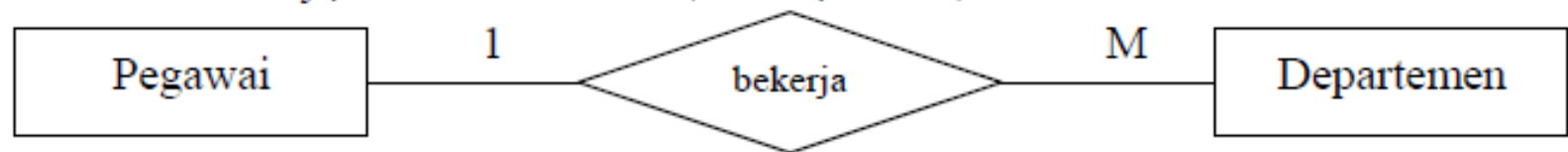
Jumlah tupel dalam sebuah relasi

### ○ Jenis rasio kardinalitas :

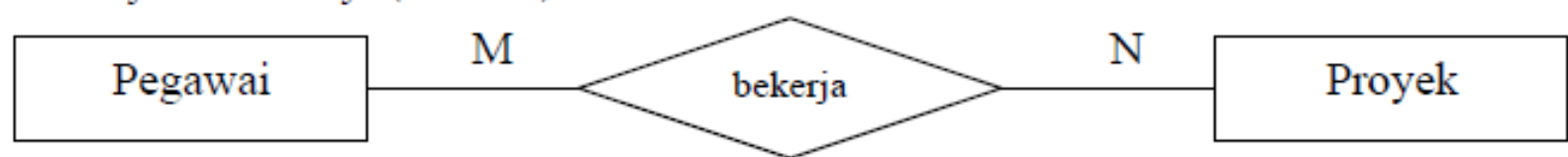
- One to one (1:1)

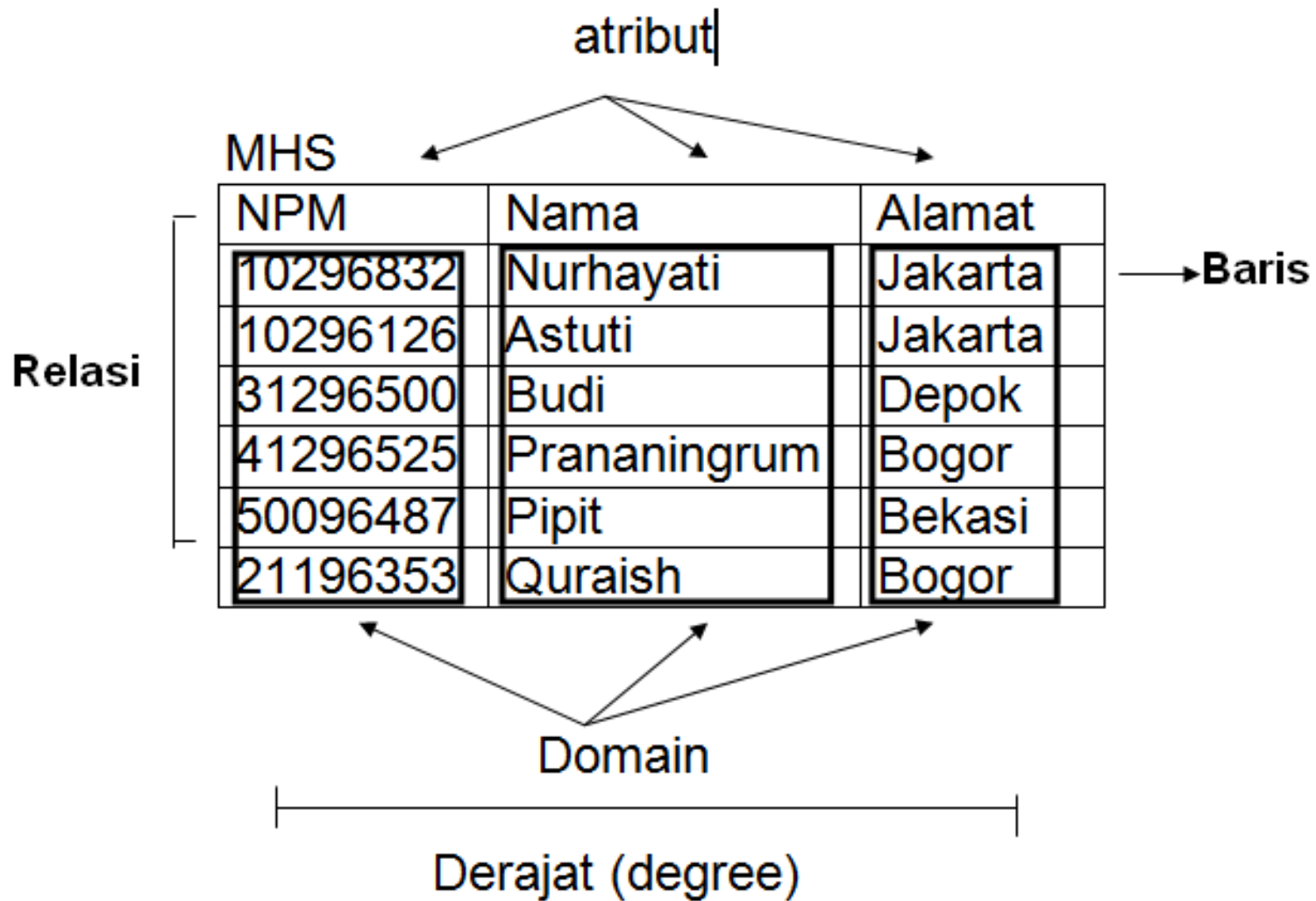


- One to many/many to one (1: M / M:1)



- Many to many (M : N)





# RELATIONAL KEY

- *Super key*

Satu atribut/kumpulan atribut yang secara unik mengidentifikasi sebuah tupel di dalam relasi

- *Candidate key*

Atribut di dalam relasi yang biasanya mempunyai nilai unik

- *Primary key*

Candidate key yang dipilih untuk mengidentifikasikan tupel secara unik dalam relasi

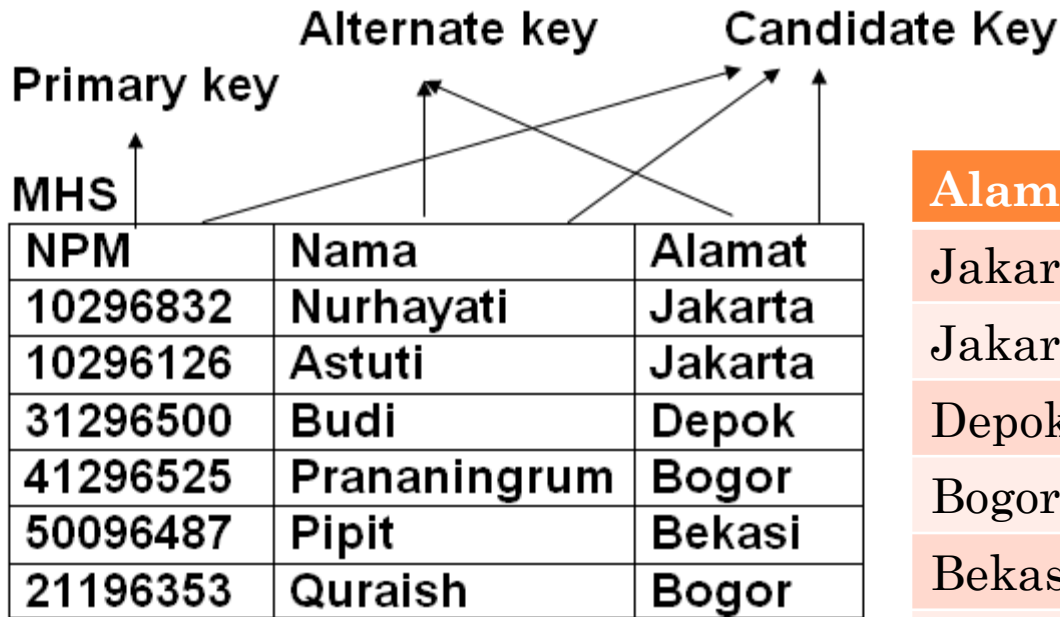
- *Alternate key*

Candidate key yang tidak dipilih sebagai primary key

- *Foreign key*

Atribut dengan domain yang sama yang menjadi kunci utama pada sebuah relasi tetapi pada relasi lain atribut tersebut hanya sebagai atribut biasa





Alamat	Provinsi
Jakarta	DKI Jakarta
Jakarta	DKI Jakarta
Depok	Jawa Barat
Bogor	Jawa Barat
Bekasi	Jawa Barat
Bogor	Jawa Barat

- Super Key : NPM, Nama, Alamat, Provinsi
- Candidate Key : NPM, Nama, Alamat
- Primary Key : NPM
- Alternate Key : Nama
- Foreign Key : NPM



# PARTICIPATION CONSTRAINT

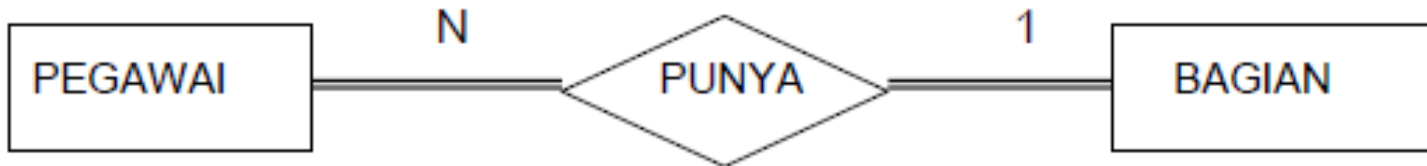
- Participation constraint merupakan batasan yang menjelaskan apakah keberadaan suatu entity tergantung pada hubungannya dengan entity lain .Terdapat 2 macam participation constraint , total participation constraints dan partial participation constraints





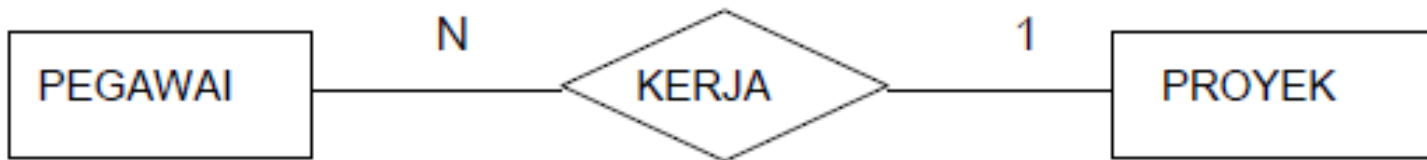
- **Total Participation**

Keberadaan suatu entity tergantung pada hubungannya dengan entity lain.



**Partial Participation**

Keberadaan suatu entity tidak tergantung pada hubungannya dengan entity lain.



# RELATIONAL INTEGRITY RULES

- **Null**

**Nilai suatu atribut yang tidak diketahui dan tidak cocok untuk baris (tuple) tersebut**

- **Entity Integrity**

**Tidak ada satu komponen primary key yang bernilai null.**

- **Referential Integrity**

**Suatu domain dapat dipakai sebagai kunci primer bila merupakan atribut tunggal pada domain yang bersangkutan.**



# BAHASA PADA BASIS DATA RELATIONAL

- Menggunakan bahasa query → pernyataan yang diajukan untuk mengambil informasi

Terbagi 2 :

- Bahasa Formal

Bahasa query yang diterjemahkan dengan menggunakan simbol-simbol matematis.

Contoh:      Aljabar relasional  
                  Kalkulus relasional



## ○ Aljabar Relasional (Bahasa Prosedural)

Bahasa query prosedural → pemakai menspesifikasikan data apa yang dibutuhkan dan bagaimana untuk mendapatkannya.

- Lebih bersifat operasional, sangat berguna untuk merepresentasikan eksekusi perencanaan.
- Operasi-operasi Dasar :
  - selection
  - projection
  - Union
  - intersection
  - set difference
- Operasi lainnya :
  - cross/cartesian product
  - join
  - division
  - sum, average, min, max



- Jenis Operasi :

- Operasi *Unary* :

*selection & projection.*

Disebut operasi *unary*, karena dapat digunakan hanya pada satu tabel.

- Operasi *Binary* :

*union, intersection, set difference, cartesian product, join* dan *division.*

Disebut operasi *binary*, karena memerlukan sepasang tabel.



# OPERASI SELECTION ( $\sigma$ )

- Operasi yang digunakan untuk memilih subset dari tuple-tuple pada suatu tabel.
- Notasi :  $\sigma_p(t)$
- $p$  disebut *selection predicate* (yang menentukan pemilihan)
- $p$  merupakan formula dari kalkulus perbandingan (*proportional calculus*) yang berisi **terms** yang dihubungkan oleh :  $\wedge$  (and),  $\vee$  (or),  $\sim$  (not)

setiap **term** salah satu dari :

$\langle \text{atribut} \rangle \text{ op } \langle \text{atribut} \rangle$  or  $\langle \text{konstanta} \rangle$

atau

$\langle \text{atribut} \rangle \text{ op } \langle \text{atribut} \rangle$

dengan  $op$  salah satu dari :  $=, \neq, >, \geq, <, \leq$



# OPERASI SELECTION LANJ.

- Contoh : dilakukan operasi  $\sigma_{\text{jenis}=\text{"action"}}(\text{film})$  pada tabel film sebagai berikut :

<b>kode_film</b>	<b>jenis</b>	<b>judul</b>	<b>jml_keping</b>	<b>jml_film</b>
A01	action	Spiderman	2	3
A02	action	Spiderman 2	2	5
D01	drama	Love Story	1	3
H01	horor	Evil Death	3	2

- Hasilnya : suatu relasi dengan tuple-tuple yang memiliki jenis sama dengan action.

<b>kode_film</b>	<b>jenis</b>	<b>judul</b>	<b>jml_keping</b>	<b>jml_film</b>
A01	action	Spiderman	2	3
A02	action	Spiderman 2	2	5



# OPERASI PROJECTION ( $\pi$ )

- Operasi yang digunakan untuk memilih subset dari kolom-kolom pada suatu tabel
- Notasi :  $\pi_{A_1, A_2, \dots, A_n}(t)$ , dimana  $A_1, A_2, \dots, A_n$  adalah daftar nama atribut dan  $t$  adalah nama suatu tabel.
- Hasilnya : suatu tabel dengan atribut yang tercantum pada daftar nama atribut pada operasi.





# OPERASI PROJECTION (LANJ)

- Contoh : dilakukan operasi  $\pi_{\text{kode\_film, jenis, judul}}(\text{film})$  pada tabel film sebagai berikut :

<b>kode_film</b>	<b>jenis</b>	<b>judul</b>	<b>jml_keping</b>	<b>jml_film</b>
A01	action	Spiderman	2	3
A02	action	Spiderman 2	2	5
D01	drama	Love Story	2	3
H01	horor	Evil Death	2	2

- Hasilnya :

<b>kode_film</b>	<b>jenis</b>	<b>judul</b>
A01	action	Spiderman
A02	action	Spiderman 2
D01	drama	Love Story
H01	horor	Evil Death



# OPERASI UNION/GABUNGAN ( $\cup$ )

- Operasi yang digunakan untuk menggabungkan hasil dari beberapa query.
- Notasi :  $r \cup s$ , menghasilkan suatu tabel baru yang elemen barisnya merupakan elemen dari  $r$  dan  $s$ , tidak ada duplikasi data.
- Untuk  $r \cup s$  harus valid :  $r$ ,  $s$  harus memiliki jumlah atribut yang sama.



# OPERASI UNION/GABUNGAN (LANJ)

o tabel r

A	B	C
aa	1	7
bb	5	7
bb	12	7
dd	23	10

tabel s

A	B	C
aa	5	7
cc	4	4
dd	10	7
aa	6	10

o  $\pi_A(r) \cup \pi_A(s)$

A
aa
bb
dd
cc



# OPERASI INTERSECTION/IRISAN ( $\cap$ )

- Notasi :  $r \cap s$ , menghasilkan suatu tabel baru dengan elemen barisnya merupakan elemen yang terdapat pada tabel  $r$  dan elemen tersebut juga terdapat pada tabel  $s$ , tidak ada duplikasi data.  $r$  dan  $s$  harus memiliki jumlah atribut yang sama.

- tabel  $r$

A	B
a	1
a	2
b	1

- tabel  $s$

A	B
a	2
b	3

- $r \cap s$

A	B
a	2



# OPERASI SET DIFFERENCE (−)

- Notasi  $r - s$ , hasilnya tabel yang elemen barisnya terdapat pada  $r$  tetapi tidak terdapat di  $s$ .  $r$  dan  $s$  harus memiliki jumlah atribut yang sama.

- tabel  $r$

A	B	C
aa	1	7
bb	5	7
bb	12	7
dd	23	10

- tabel  $s$

A	B	C
aa	5	7
cc	4	4
dd	10	7
aa	6	10

- $\pi_A(r) - \pi_A(s)$

A
bb



# OPERASI CROSS/CARTESIAN-PRODUCT

- Notasi  $r \times s$ , menghasilkan semua kombinasi baris/tuple dari 2 buah tabel, yaitu tabel  $r$  dan  $s$ . Jumlah atribut tidak harus sama. Contoh :

- tabel  $r$  :

A	B
a	1
b	3

- maka  $r \times s$  :

A	B	C	D
a	1	a	6
a	1	b	5
a	1	c	12
b	3	a	6
b	3	b	5
b	3	c	12

- tabel  $s$  :

C	D
a	6
b	5
c	12



# KOMPOSISI OPERASI-OPERASI RELASIONAL

- Operasi-operasi aljabar relasional dapat digabungkan dengan ekspresi aljabar relasional lainnya.
- Contoh : terdapat tabel film sebagai berikut :

<b>kode_film</b>	<b>jenis</b>	<b>judul</b>	<b>jml_keping</b>	<b>jml_film</b>
A01	action	Spiderman	2	3
A02	action	Spiderman 2	2	5
D01	drama	Kabayan	2	3
H01	horor	Scream	2	2



# KOMPOSISI OPERASI-OPERASI RELASIONAL

- Tampilkan kode\_film, judul dan jml\_film dengan  $jml\_film > 3$ , maka operasi komposisinya sebagai berikut :

$\pi_{kode\_film, judul, jml\_film}(\sigma_{jml\_film > 3}(film))$

hasilnya :

<b>kode_film</b>	<b>judul</b>	<b>jml_film</b>
A02	Spiderman 2	5





# OPERASI JOIN

- Digunakan untuk menggabungkan dua tabel atau lebih dengan hasil berupa gabungan dari kolom-kolom yang berasal dari tabel-tabel tersebut.

- tabel R

<b>sid</b>	<b>sname</b>	<b>rating</b>	<b>age</b>
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0

- tabel S

<b>sid</b>	<b>bid</b>	<b>day</b>
22	101	10/10/96
58	103	11/12/96



## OPERASI JOIN (LANJ)

- Operasinya : R      S sama dengan operasi  $\sigma(R \times S)$ , disebut juga *Full Join*.
- Hasilnya :

sid	sname	rating	age	sid	bid	day
22	dustin	7	45.0	22	101	10/10/96
22	dustin	7	45.0	58	103	11/12/96
31	lubber	8	55.5	22	101	10/10/96
31	lubber	8	55.5	58	103	11/12/96
58	rusty	10	35.0	22	101	10/10/96
58	rusty	10	35.0	58	103	11/12/96



# OPERASI DIVISION (/)

- Notasi :  $R/S$ .
- Menghasilkan suatu tabel dari dua buah tabel yang terdiri dari atribut dari tabel R yang tidak terdapat pada tabel S dengan tupel-tupel dari tabel R yang memiliki kesamaan dengan tupel-tupel yang ada pada tabel A secara keseluruhan dan tidak terdapat duplikasi data.



# OPERASI DIVISION (LANJ)

o tabel R

sno	pno
s1	p1
s1	p2
s1	p3
s1	p4
s2	p1
s2	p2
s3	p2
s4	p2
s4	p4

tabel S1

pno
p2

R/S1

sno
s1
s2
s3
s4

tabel S2

pno
p2
p4

R/S2

sno
s1
s4



# OPERASI SUM, AVERAGE, MIN, MAX, COUNT

- SUM – digunakan untuk menghitung jumlah nilai pada suatu kolom.
- AVERAGE – digunakan untuk menghitung rata-rata dari suatu kolom.
- MIN – digunakan untuk mencari nilai minimal pada suatu kolom.
- MAX – digunakan untuk mencari nilai maksimal pada suatu kolom.
- COUNT – digunakan untuk menghitung jumlah baris pada suatu kolom.



# KALKULUS RELASIONAL

**Bahasa query non-prosedural → pemakai menspesifikasikan data apa yang dibutuhkan tanpa menspesifikasikan bagaimana untuk mendapatkannya.**

**Terbagi 2 :**

- Kalkulus Relasional Tupel**
- Kalkulus Relasional Domain**



# KALKULUS RELASIONAL BERDASARKAN TUPEL

- Dalam kalkulus relasional tupel digunakan variabel dari tupelnya .variabel dari suatu tupel adalah daerah yang terdefinisi sebagai nama dari suatu relasi. contoh :
- manampilkan nama\_propinsi pada kode\_propinsi 'pp2'  $\{t \mid t \in \text{ms\_propinsi} \wedge t(\text{kode\_propinsi}) = \text{'pp2'}\}$  .
- jika ditampilkan dalam SQL maka perintahnya adalah :

```
SELECT nama_propinsi FROM ms_propinsi  
WHERE kode_propinsi = 'pp2';
```



# KALKULUS RELASIONAL BERDASARKAN DOMAIN

- Contoh Kalkulus domain;
- Mencari data mengenai nama\_cabang pada tabel ms\_cabang. {<kk,nk,kp> | <kk,nk,kp> ?  
ms\_cabang }
- Jika ditampilkan dalam SQL, maka perintahnya adalah :

**SELECT nama\_cabang FROM ms\_cabang;**





# CONTOH-CONTOH BASIS DATA RELASIONAL

- **DB2 → IBM**
- **ORACLE → Oracle**
- **SYBASE → Powersoft**
- **INFORMIX → Informix**
- **Microsoft Access → Microsoft**



- **Bahasa Komersial**

**Bahasa Query yang dirancang sendiri oleh programmer menjadi suatu program aplikasi agar pemakai lebih mudah menggunakannya (user friendly).**

**Contoh :**

- **QUEL**

**Berbasis pada bahasa kalkulus relasional**

- **QBE**

**Berbasis pada bahasa kalkulus relasional**

- **SQL**

**Berbasis pada bahasa kalkulus relasional dan aljabar relasional**



# QUEL

- QUEL adalah query language dari DBMS relasional INGRES.
- Dikembangkan di University of California di Berkeley dengan menggunakan sistem operasi UNIX.
- Dipergunakan secara berdiri sendiri dengan menuliskan instruksi ke prosesor QUEL
- Atau disisipkan ke dalam bahasa pemrograman C.
- Di dalam bahasa C, instruksi QUEL dimulai dengan tanda ##



# STRUKTUR DASAR QUEL

1. Setiap variable tuple ditulis dalam range yaitu :  
**Range of t is r**
2. Kalimat retrieve memiliki fungsi sama dengan SELECT pada SQL
3. Kalimat WHERE berisi predikat yang ditentukan
4. Bentuk query QUEL

Range of t1 is r1

Range of t2 is r2

⋮

Range of tm is rn

Retrieve(t1.Aj1, t2.Aj2, ....., tm.Ajm

Where P

5. Tidak dapat melaksanakan operasi aljabar seperti intersection, union dan minus



# QUEL DIBAGI ATAS :

## 1. DDL (Data Definition Language)

- CREATE : membuat tabel
- INDEX : membuat indeks
- DEFINE VIEW : membuat view
- DESTROY : menghapus tabel, indeks atau view
- MODIFY : merubah struktur tabel/indeks

Contoh :

```
CREATE S(SN = TEXT(5),SNAME=TEXT(20),  
STATUS=I2, CITY=TEXT(15))
```



# QUEL DIBAGI ATAS (LANJ):

## 2. DML (Data Manipulation Language)

- RETRIEVE : menampilkan data
- REPLACE : mengupdate data
- DELETE : menghapus record
- APPEND : memasukkan record baru

*Contoh :*

- a. Dapatkan nama supplier yang tinggal di Paris dan mempunyai status lebih besar dari 20

**RETRIEVE(S.SNAME) WHERE CITY='Paris' AND STATUS > 20**

- b. Menampilkan data semua supplier

**RETRIEVE (S.ALL)**

- c. Dapatkan nama supplier yang mensupply part P2

**RETRIEVE (S.SNAME) WHERE S.SN=SP.SN AND SP.PN='P2'**



# QBE (QUERY BY EXAMPLE)

- QBE mempunyai beberapa kelebihan yang tidak dipunyai oleh bahasa-bahasa lain, antara lain pengguna dapat langsung mempergunakan editor layar khusus dan langsung membuat query di terminal tersebut.
- QBE tidak mempergunakan pernyataan-pernyataan dalam bahasa Inggris.
- Untuk mendapatkan tuple-tuple yang akan ditayangkan di layar, semua instruksi diakhiri dengan titik.
- Pengoperasian tersebut seperti PRINT, UPDATE, INSERT dan DELETE.
- Pada beberapa kasus pengguna membutuhkan 2 atau lebih tabel yang ditampilkan di layar. Untuk menghubungkan, caranya dengan memasukkan elemen ke dalam struktur tabel dan diawali dengan garis bawah ( ) dan terdiri sampai dengan 18 karakter alphabet atau numeric.

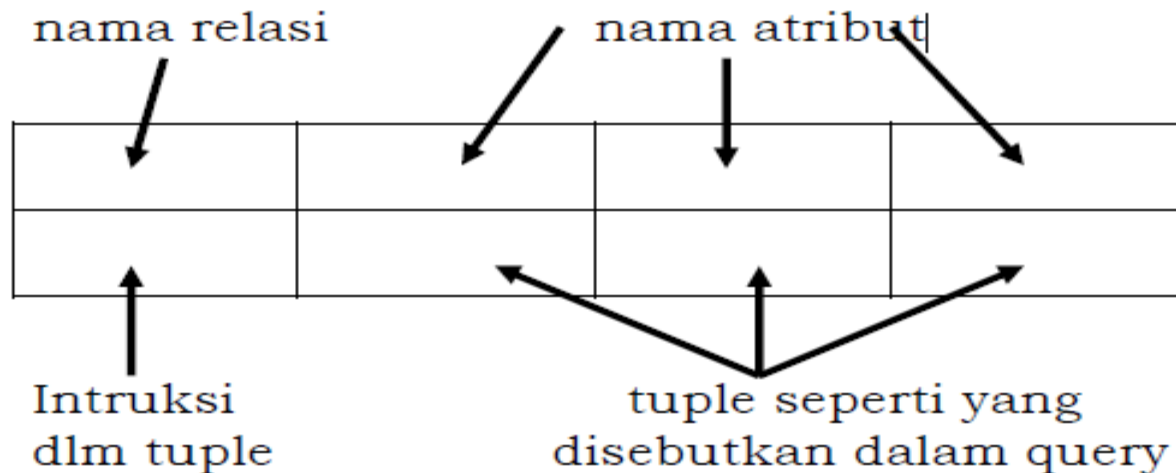


# PERINTAH DAN BAGAN UMUM

Perintah yang digunakan adalah :

- P.(print) : menampilkan data
- I. (insert) : memasukkan record baru
- D. (delete) : menghapus record
- U. (update) : merubah record

Bagan Umum tabel dari QBE





# CONTOH PRINT QBE

1. Dapatkan nomor supplier yang tinggal di Paris dan mempunyai status lebih besar dari 20

<b>S</b>	<b>SN</b>	<b>SNAME</b>	<b>STATUS</b>	<b>CITY</b>
	P.		>20	Paris

atau dapat juga dengan menggunakan condition box

<b>S</b>	<b>SN</b>	<b>SNAME</b>	<b>STATUS</b>	<b>CITY</b>
	P.		_ST	_SC
<b>CONDITION</b>				
_ST>20 AND _SC=Paris				



# CONTOH INSERT DAN UPDATE QBE

- Masukkan data pengiriman barang yang dilakukan oleh supplier S5 untuk barang P1 sebanyak 200

<b>SP</b>	<b>SN</b>	<b>PN</b>	<b>QTY</b>
I.	S5	P1	200

- Ubah data untuk barang P2, warnanya diganti dengan Yellow dan weightnya ditambah dengan 5

<b>P</b>	<b>PN</b>	<b>PNAME</b>	<b>COLOR</b>	<b>WEIGHT</b>	<b>WEIGHT</b>	<b>CITY</b>
	P2		U.Yellow	_WT	U._WT+5	



## CONTOH DELETE QBE

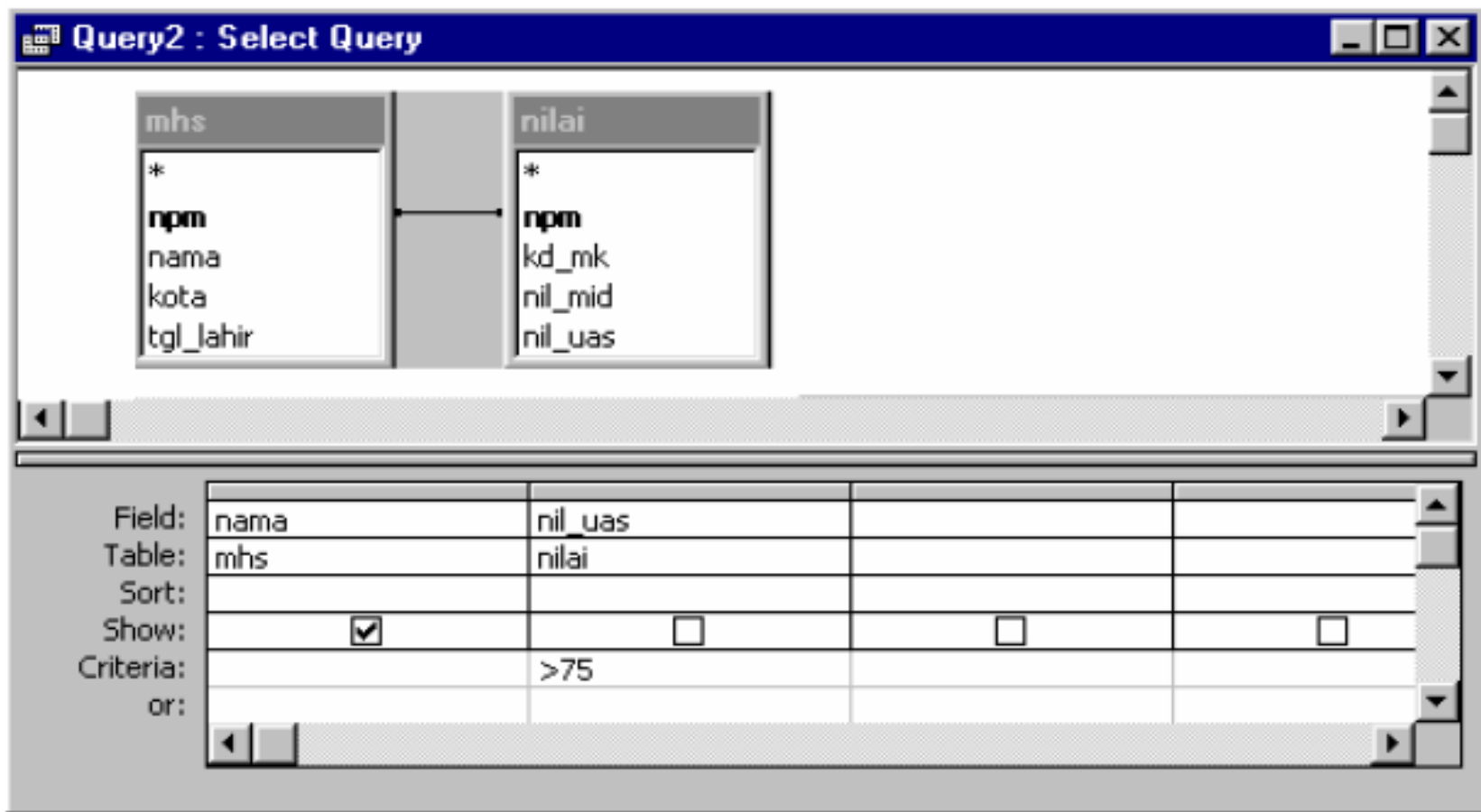
- Hapus data supplier yang mempunyai nomor supplier S5

<b>S</b>	<b>SN</b>	<b>SNAME</b>	<b>STATUS</b>	<b>CITY</b>
D.	S5			



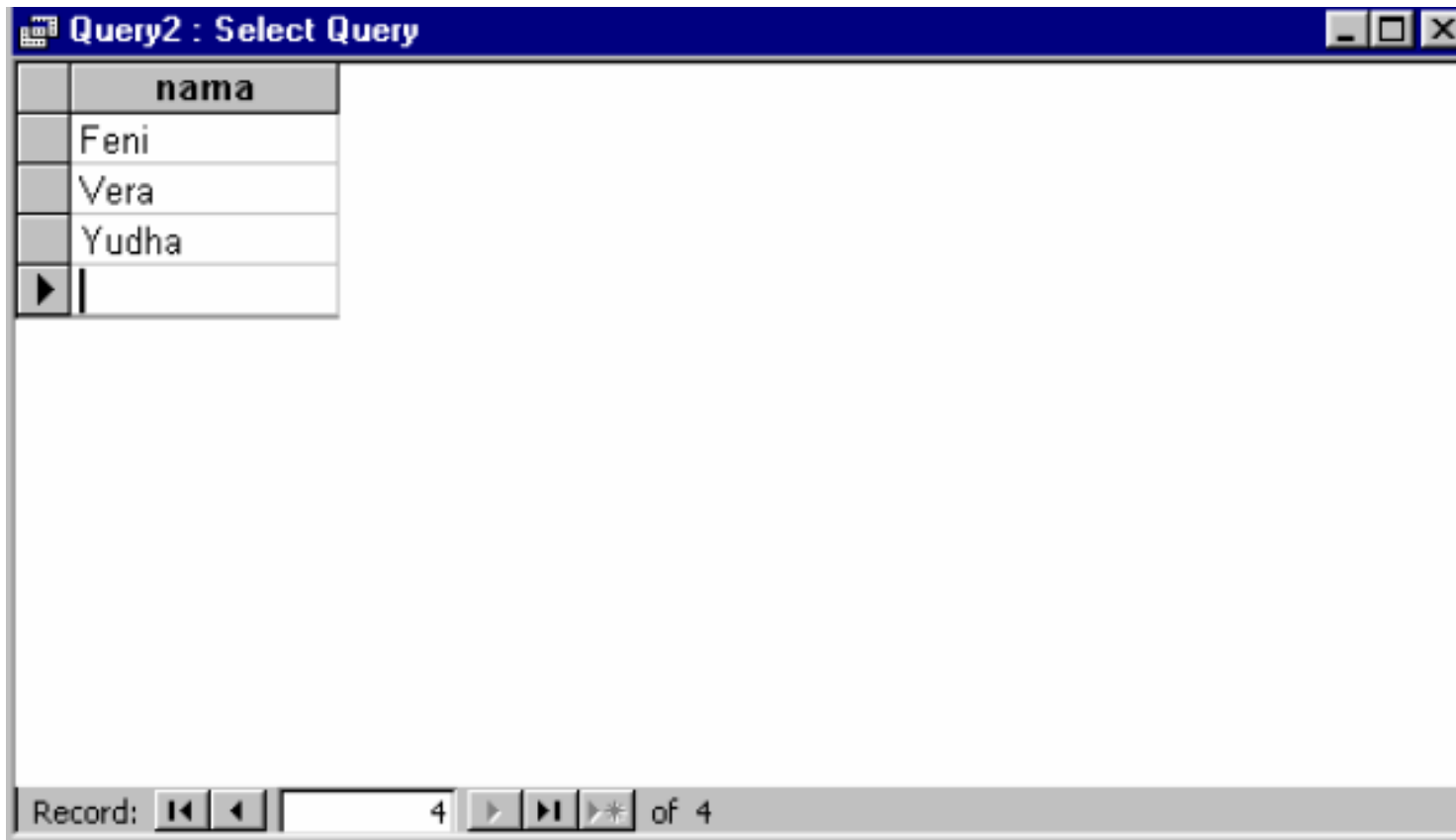
# DENGAN MICROSOFT ACCESS

- Dengan



# DENGAN MICROSOFT ACCESS LANJ

- Hasil



The screenshot shows a Microsoft Access window titled "Query2 : Select Query". The window displays a table with a single column named "nama". The table contains three rows of data: "Feni", "Vera", and "Yudha". The table is currently empty, with a cursor visible in the first cell of the first row. The status bar at the bottom indicates "Record: 4 of 4".

nama
Feni
Vera
Yudha



# OPERATOR TAMBAHAN DALAM QBE

Beberapa fungsi agregat operator, yaitu :

- SUM ( Menjumlahkan bilangan )
- AVG ( Mencari rata-rata )
- MAX ( Mengambil nilai tertinggi )
- MIN ( Mengambil nilai terendah )
- CNT ( Menghitung jumlah record )
- ALL ( Menghitung semua data )
- VN ( Bila ada yang sama hanya ditulis sekali )

Penggunaan berbagai kondisi, seperti :

- OR
- AND
- NOT
- BETWEEN
- LIKE
- ORDER BY
- GROUP BY

