

# JOIN

# Pemrograman SQL Lanjut

# PMI 1413

$$4(2/2)$$


Eko Win Kenali, S.Kom., M.Cs.

# Pendahuluan

Bab ini akan membahas mengenai :

- Dasar QUERY antar Tabel.
- Pengenalan dan Penggunaan Operator JOIN
- Pengenalan Sub-Query
- Pengenalan Pandangan (VIEW)

# Bahan

Untuk dapat menggunakan perintah-perintah SQL, di perlukan perubahan field pada tabel yang telah ada dan penambahan tabel-tabel baru.

## pegawai

	Column Name	Data Type	Length	Allow Nulls
?	nip	nvarchar	10	
	nama	nvarchar	50	✓
	sex	char	1	✓
	tempat_lahir	nvarchar	30	✓
	tgl_lahir	datetime	8	✓
	tmt	datetime	8	✓
	kd_golongan	nvarchar	2	✓
	kd_formasi	nvarchar	2	✓
→	kd_departemen	nvarchar	2	✓

## sex

	Column Name	Data Type	Length	Allow Nulls
?	sex	char	1	
	jenis_kelamin	nvarchar	15	✓
	sebutan	nvarchar	10	✓

## golongan

	Column Name	Data Type	Length	Allow Nulls
?	kd_golongan	nvarchar	2	
	nm_golongan	nvarchar	50	✓
	gapok	money	8	✓

## Departemen

	Column Name	Data Type	Length	Allow Nulls
?	kd_departemen	nvarchar	2	
	nm_departemen	nvarchar	50	✓

## Formasi

	Column Name	Data Type	Length	Allow Nulls
?	kd_formasi	nvarchar	2	
	nm_formasi	nvarchar	25	✓

## pegawai

nip	nama	...
132310741	Budiman	...
132310742	Eko Win Kenali	...
132310743	Shinta Rahma	...
132310744	Indah Muniarti	...
132310745	Heri Herlambang	...
132310739	Riko Hendrawan	...
132310737	Kurniawan	...
132310740	Sri Karmila	...
132310746	Dwi Arini	...
132310747	Kamyono	...

Kd_formasi	Kd_departemen
01	01
02	03
02	02
02	02
03	04
02	04
02	02
03	02
01	04
01	01

## Departemen

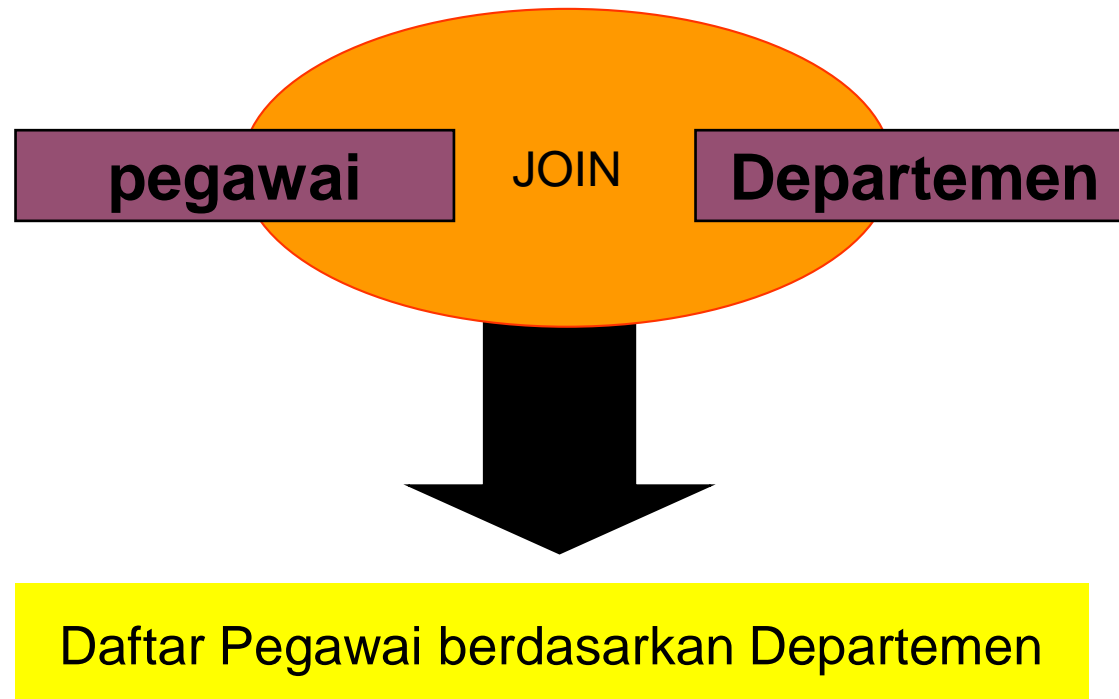
Kd_departemen	Nm_departemen
01	BAAKPSI
02	PS. AGRIBISNIS
03	PS. MANAJEMEN INFORMATIKA
04	KEPEGAWAIAN DAN UMUM

## formasi

Kd_formasi	Nm_formasi
01	Struktural
02	Fungsional
03	Honorar

## 1. Dasar Query Antar Tabel

SQL mempunyai kemampuan untuk menggabungkan dua tabel atau lebih guna membentuk sebuah informasi yang lebih informatif/lengkap/komplek. Prosesnya disebut dengan JOIN. Begitu pula dengan tabel hasilnya.



Beberapa Hal yang perlu di perhatikan dalam query antar tabel.

1. Setiap Kolom disebutkan dengan bentuk

**nama\_tabel.nama\_kolom**, contoh : *pegawai.nip*,  
*pegawai.nama*, *departemen.nm\_departemen*.

2. Nama-nama Tabel yang dilibatkan dalam query perlu disebutkan dalam/setelah klausa FROM. Contoh : *FROM pegawai, departemen*.

3. Kondisi dalam klausa WHERE berfungsi untuk menentukan macam atau jenis JOIN yang terbentuk. (*akan dijelaskan dalam materi pengenalan operator JOIN*).

# Studi Kasus-1

Menggabungkan kolom pada tabel pegawai dengan kolom pada tabel departemen.

**Pegawai** (nip,nama,kd\_departemen)

**Departemen** (kd\_departemen, nm\_departemen).

Dan tabel keluaran/hasil querynya adalah :

**Tabel baru**(nip,nama,nm\_departemen).

```
SELECT pegawai.nip, pegawai.nama, departemen.nm_departemen
FROM pegawai, departemen
WHERE pegawai.kd_departemen = departemen.kd_departemen
```

	nip	nama	nm_departemen
1	132310737	Kurniawan	PS. AGRIBISNIS
2	132310739	Riko Hendrawan	KEPEGAWAIAN DAN UMUM
3	132310740	Sri Karmila	PS. AGRIBISNIS
4	132310741	Budiman	BAAKPSI
5	132310742	Eko Win Kenali	PS. MANAJEMEN INFORMATIKA
6	132310743	Shinta Rahma	PS. AGRIBISNIS
7	132310744	Indah Muniarti	PS. AGRIBISNIS
8	132310745	Heri Herlambang	KEPEGAWAIAN DAN UMUM
9	132310746	Dwi Arini	KEPEGAWAIAN DAN UMUM
10	132310747	Kamyono	BAAKPSI

# Penggunaan Alias

```
SELECT A.nip, A.nama, B.nm_departemen  
FROM pegawai as A, departemen as B  
WHERE A.kd_departemen = B.kd_departemen
```



# Perkalian Kartesian

Bentuk Paling Sederhana dari penggabungan dua buah tabel akan membentuk perkalian kartesian (cartesian product), atau biasa disebut dengan **cross join** atau **full join**. Hal ini akan di peroleh jika klausa **WHERE** tidak disebutkan.

```
SELECT pegawai.nip, pegawai.nama, departemen.nm_departemen
FROM pegawai, departemen
```

132310737	Kurniawan	BAAKPSI
132310739	Riko Hendrawan	BAAKPSI
132310740	Sri Karmila	BAAKPSI
132310741	Budiman	BAAKPSI
132310742	Eko Win Kenali	BAAKPSI
132310743	Shinta Rahma	BAAKPSI
132310744	Indah Muniarti	BAAKPSI
132310745	Heri Herlambang	BAAKPSI
132310746	Dwi Arini	BAAKPSI
132310747	Kamyono	BAAKPSI
132310737	Kurniawan	PS. AGRIBISNIS
132310739	Riko Hendrawan	PS. AGRIBISNIS
132310740	Sri Karmila	PS. AGRIBISNIS
132310741	Budiman	PS. AGRIBISNIS
132310742	Eko Win Kenali	PS. AGRIBISNIS
132310743	Shinta Rahma	PS. AGRIBISNIS
132310744	Indah Muniarti	PS. AGRIBISNIS
132310745	Heri Herlambang	PS. AGRIBISNIS
132310746	Dwi Arini	PS. AGRIBISNIS
132310747	Kamyono	PS. AGRIBISNIS
132310737	Kurniawan	PS. MANAJEMEN INFORMATIKA
132310739	Riko Hendrawan	PS. MANAJEMEN INFORMATIKA
132310740	Sri Karmila	PS. MANAJEMEN INFORMATIKA
132310741	Budiman	PS. MANAJEMEN INFORMATIKA
132310742	Eko Win Kenali	PS. MANAJEMEN INFORMATIKA
132310743	Shinta Rahma	PS. MANAJEMEN INFORMATIKA
132310744	Indah Muniarti	PS. MANAJEMEN INFORMATIKA
132310745	Heri Herlambang	PS. MANAJEMEN INFORMATIKA
132310746	Dwi Arini	PS. MANAJEMEN INFORMATIKA
132310747	Kamyono	PS. MANAJEMEN INFORMATIKA
132310737	Kurniawan	KEPEGAWAIAN DAN UMUM
132310739	Riko Hendrawan	KEPEGAWAIAN DAN UMUM
132310740	Sri Karmila	KEPEGAWAIAN DAN UMUM
132310741	Budiman	KEPEGAWAIAN DAN UMUM
132310742	Eko Win Kenali	KEPEGAWAIAN DAN UMUM
132310743	Shinta Rahma	KEPEGAWAIAN DAN UMUM
132310744	Indah Muniarti	KEPEGAWAIAN DAN UMUM
132310745	Heri Herlambang	KEPEGAWAIAN DAN UMUM
132310746	Dwi Arini	KEPEGAWAIAN DAN UMUM
132310747	Kamyono	KEPEGAWAIAN DAN UMUM

Tabel 1		Tabel 2	
A	B	X	Y
1	2	5	6
3	4	7	8

Tabel 3			
A	B	X	Y
1	2	5	6
3	4	5	6
1	2	7	8
3	4	7	8

# 1. Dasar Query Antar Tabel

## 1.1 Equaljoin dan Non Equaljoin

**Equaljoin** = Penggabungan antartabel yang menggunakan operator sama dengan (=) pada kondisi dalam klausa WHERE untuk membandingkan satu kolom dengan kolom lainnya.

```
SELECT pegawai.nama, departemen.nm_departemen, golongan.kd_golongan
FROM pegawai, departemen, golongan
WHERE pegawai.kd_golongan = golongan.kd_golongan
AND pegawai.kd_departemen = departemen.kd_departemen
```

	nama	nm_departemen	kd_golongan
1	Kurniawan	PS. AGRIBISNIS	02
2	Riko Hendrawan	KEPEGAWAIAN DAN UMUM	02
3	Sri Karmila	PS. AGRIBISNIS	02
4	Budiman	BAAKPSI	01
5	Eko Win Kenali	PS. MANAJEMEN INFORMATIKA	03
6	Shinta Rahma	PS. AGRIBISNIS	02
7	Indah Muniarti	PS. AGRIBISNIS	02
8	Heri Herlambang	KEPEGAWAIAN DAN UMUM	01
9	Dwi Arini	KEPEGAWAIAN DAN UMUM	01
10	Kamyono	BAAKPSI	01

Jika operator yang digunakan untuk Menghubungkan tabel satu dengan Tabel lainnya tidak berupa (=). Maka Di sebut sebagai NON EQUALJOIN

# 1. Dasar Query Antar Tabel

## 1. 2. SelfJoin

**Selfjoin** = merupakan upaya penggabungan dari tabel yang sama.

Memperoleh pasangan pegawai dari departemen '02'=PS. AGRIBISNIS dengan departemen '03'=PS. MANAJEMEN INFORMATIKA.

```
SELECT A.nama, B.nama  
FROM pegawai as A, pegawai as B  
WHERE A.kd_departemen ='02' AND B.kd_departemen ='03'
```

	nama	nama
1	Kurniawan	Eko Win Kenali
2	Sri Karmila	Eko Win Kenali
3	Shinta Rahma	Eko Win Kenali
4	Indah Muniarti	Eko Win Kenali

# 1. Dasar Query Antar Tabel

## 1. 3. UNION

**UNION** berguna untuk menggabungkan hasil dari dua buah query dari dua buah tabel.

Sebagai contoh terdapat dua buah tabel. Yaitu : **Siswa\_SD** dan **Siswa\_SMP**

Siswa_SD		Siswa_SMP	
NIS.	Nama	NIS.	Nama
0120	Erika Wardani	0220	Susiana
0121	M. Taufik	0221	Rustamaji
0122	Saidah	0222	Indah Ahliani
0123	Dwi Arini	0223	Gatot

```
SELECT nis, nama  
FROM Siswa_SD
```

**UNION**

```
SELECT nis, nama  
FROM Siswa_SMP
```

# 1. Dasar Query Antar Tabel

## 1. 4. INNER JOIN dan OUTER JOIN

**Equijoin** Seringkali dibedakan menjadi dua kategori yakni inner equijoin (innerjoin) dan outer equijoin (outerjoin).

**Namun sebelumnya perlu dibuat tabel baru (matakuliahdosen).**

### Struktur Tabel

New Table in 'AFTERMID' on 'IISMUSER1'				
	Column Name	Data Type	Length	Allow Nulls
▶	kd_mk	nvarchar	4	✓
	nm_mk	nvarchar	50	✓
	sks	int	4	✓
	nip	nvarchar	10	✓

### Data

	kd_mk	nm_mk	sks	nip
▶	P001	Basis Data 1	2	132310742
	P002	Basis Data 2	2	132310742
	P001	Basis Data 1	2	132310745
	P003	Jaringan Komputer	2	132310745
*				

**Buatlah query sebagai berikut :**

```
SELECT pegawai.nama, matakuliahdosen.kd_mk,matakuliahdosen.nm_mk  
FROM pegawai, matakuliahdosen  
WHERE pegawai.nip = matakuliahdosen.nip
```

**Hasil query sebagai berikut :**

	nama	kd_mk	nm_mk
1	Eko Win Kenali	P001	Basis Data 1
2	Eko Win Kenali	P002	Basis Data 2
3	Heri Herlambang	P001	Basis Data 1
4	Heri Herlambang	P003	Jaringan Komputer

## **INNER JOIN**

Hanya record yang memiliki data pada tabel yang akan di tampilkan.

## **TUGAS**

Bagaimana Jika akan menampilkan keseluruhan data dosen yang telah memiliki mata kuliah dan yang belum memiliki mata kuliah juga ikut ditampilkan.?

## **OUTER JOIN**

# 1. Dasar Query Antar Tabel

## 1. 5. NATURAL JOIN dan UNNATURAL JOIN

**Natural Join** Penggabungan dua buah tabel atau lebih yang menggunakan kolom yang berkedudukan sebagai kunci tamu Dan kunci primer (sekali pun hanya dalam desain basis data) sebagai penghubung.

pegawai

	Column Name	Data Type	Length	Allow Nulls
🔑	nip	nvarchar	10	
	nama	nvarchar	50	✓
	sex	char	1	✓
	tempat_lahir	nvarchar	30	✓
	tgl_lahir	datetime	8	✓
	tmt	datetime	8	✓
	kd_golongan	nvarchar	2	✓
	kd_formasi	nvarchar	2	✓
	kd_departemen	nvarchar	2	✓

**Kd\_golongan**  
Kunci tamu di tabel pegawai

	Column Name	Data Type	Length	Allow Nulls
🔑	kd_golongan	nvarchar	2	
	nama_golongan	nvarchar	50	✓
	gaji	money	8	✓

**Kd\_golongan**  
Kunci primer di tabel golongan

**Unnatural Join** Mengungkapkan penggabungan dengan kondisi penghubung antar tabel tidak mencerminkan hubungan yang lazim.

Kondisi yang di bandingkan bukan merupakan kunci primer ataupun kunci tamu.

```
SELECT A.nama, B.nama  
FROM pegawai A, pegawai B  
WHERE A.tempat_lahir ='Lampung'  
      AND B.tempat_lahir ='Lampung'
```



## 1. Dasar Query Antar Tabel

### 1. 6. Operator INTERSECT

**Intersect** berguna untuk memperoleh baris-baris (record) yang terdapat di dalam dua buah tabel atau lebih.

#### Sintak Query

```
SELECT kolom_1  
FROM nama_tabel1  
INTERSECT  
SELECT kolom_1  
FROM nama_tabel2
```

## 1. Dasar Query Antar Tabel

### 1. 7. Operator EXCEPT/MINUS

**EXCEPT/MINUS**, Jika terdapat tabel A dan Tabel B, operasi A EXCEPT B akan menampilkan semua yang ada pada tabel A, tetapi tidak terdapat pada tabel B.

#### Sintak Query

```
SELECT kolom_1
FROM tabel_A
EXCEPT
SELECT kolom_1
FROM tabel_B
```

## 2. Pengenalan dan Penggunaan Operator JOIN

Bab ini akan membahas mengenai sejumlah operator JOIN yang disediakan pada sistem yang mengikuti SQL92, Seperti CROSS JOIN, NATURAL JOIN, LEFT OUTER JOIN dan RIGHT OUTER JOIN.

### 2. 1. Operator CROSS JOIN

**Cross Join** berguna melakukan operasi cross join atau perkalian kartesian. Pemakaiannya di letakkan pada klausa FROM.

#### Sintak Query

```
SELECT *  
FROM nama_tabel1 CROSS JOIN nama_tabel2
```

#### Identik dengan Sintak :

```
SELECT *  
FROM nama_tabel1, nama_tabel2
```

## 2. Pengenalan dan Penggunaan Operator JOIN

### 2. 2. Operator NATURAL JOIN

**Natural Join** akan melakukan operasi equijoin dengan memperlakukan nama-nama kolom yang sama sebagai kolom penghubung.

**Sintak Query**

```
SELECT nm_tabelA.X, nm_tabelB.X  
FROM nm_tabelA NATURAL JOIN nm_tabelB
```

**Identik dengan Sintak :**

```
SELECT nm_tabelA.X, nm_tabelB.X  
FROM nama_tabelA, nama_tabelB  
WHERE nm_tabelA.X = nm_tabelB.X
```

## 2. Pengenalan dan Penggunaan Operator JOIN

### 2. 3. Operator LEFT OUTER JOIN

Bila baris pada tabel yang terletak di sebelah kiri operator ini ada yang tidak memiliki pasangan dengan tabel yang terletak di sebelah kanan operator ini, baris ini tetap disertakan dalam hasil penggabungan.

### 2. 4. Operator RIGHT OUTER JOIN

Merupakan kebalikan dari operator LEFT OUTER JOIN. Bila baris pada tabel yang terletak di sebelah kanan operator ini ada yang tidak memiliki pasangan dengan tabel yang terletak di kiri operator ini, baris ini tetap di sertakan dalam hasil penggabungan.

### 2. 5. Operator FULL OUTER JOIN

Gabungan dari LEFT OUTER JOIN dan RIGHT OUTER JOIN

### 2. 6. Operator UNION JOIN

Menggabungkan kedua tabel dengan kolom pasangannya akan di beri nilai NULL. Bila tabel pertama berisi N1 baris dan tabel kedua berisi N2 baris maka hasilnya berupa  $N1+N2$  baris.

## Soal Latihan:

1. Buatlah tabel-tabel di bawah ini di dalam database AFTERMID.

B_Film			
No.	Field.	Size.	Type
1.	<b>Kd_bfilm</b>	4	nvarchar
2.	nm_bfilm	70	nvarchar
3.	Sex	1	nvarchar

Film			
No.	Field.	Size.	Type
1.	<b>Kd_film</b>	3	nvarchar
2.	judul_film	70	nvarchar
3.	jenis	15	nvarchar

Filmdanpemeran			
No.	Field.	Size.	Type
1.	Kd_film	3	nvarchar
2.	Kd_bfilm	4	nvarchar

2. Isilah data pada tabel B\_Film dan Film dengan data seperti di bawah ini

B_Film			Filmdanpemeran	
Kd_bfilm	nm_bfilm	sex	Kd_film	kd_bfilm
0001	Keanu Reeves	L	001	0002
0002	Jhonny Deep	L	004	0001
0003	Jacky Chan	L	002	0004
0004	Jet Li	L	003	0009
0005	Sandra Bullock	P	005	0001
0006	Aliyah	P	006	0007
0007	Angelina Jolie	P	007	0010
0008	Sharone Stone	P	008	0003
0009	Demi Moore	P	004	0005
0010	Mel Gibson	L	010	0004

## Film

Kd_film	judul_film	jenis
001	Pirates of Carribean 1	action
002	Kiss of The Dragon	action
003	G. I. Jean	action
004	Speed	action
005	Matrix	action
006	Tom Rider	action
007	The Patriot	action
008	The Tuxedo	action
009	To Fast to Farious	action
010	Upon time in Chine 2	Drama



3. Buatlah perintah query dengan menggunakan operator-operator join yang anda ketahui, Untuk mengerjakan perintah berikut:
  - a. Menampilkan data pemeran film dan judul film yang dibintanginya dengan operator Inner join dan operator outer join.
  - b. Menampilkan data pemeran utama pria dan wanita dalam film "SPEED"
  - c. Menampilkan data pemeran film beserta pasangannya di semua judul film, untuk pemeran film yang tidak ada pasangannya juga ikut di tampilkan.
  - d. Menampilkan data film yang diperankan oleh pemeran pria saja
  - e. Menampilkan data seluruh pemeran film dengan judul filmnya dengan crossjoin.
  - f. Buktikan penggunaan operator 2.3, 2.4, 2.5, 2.6 untuk kasus tabel pemeran, film dan film yang diperankan.