

MODUL 4

JOIN

1. Join

Join merupakan sebuah konsep di dalam pengolahan data pada *database*. Konsep ini menggabungkan dua buah tabel atau lebih sehingga menghasilkan sebuah tabel baru yang bersifat *temporary* atau sementara. Melalui tabel baru ini akan dapat diperoleh hubungan tiap data pada tabel-tabel yang digabungkan. Tabel baru ini disebut sebagai *joined table*.

2. Macam-Macam Join

Pada MySQL dikenal beberapa macam *join* yang dapat dikelompokkan menjadi dua, yaitu *inner join* dan *outer join*. *Inner join* dapat dibagi kembali menjadi *natural join* dan *cross join*. Sedangkan *outer join* dapat dibedakan menjadi *left outer join*, *right outer join*, dan *full outer join*.

a. Inner Join dan Natural Join

Inner join dan *natural join* merupakan *join* yang digunakan untuk menemukan persimpangan atau perpotongan antara dua buah tabel yang di-*join*-kan. *Join* ini akan mengembalikan atau menampilkan data-data yang saling berpasangan di antara kedua buah tabel. *Syntax* untuk *inner join* dan *natural join* adalah sebagai berikut:

```
SELECT nama_kolom
FROM tabel_1 INNER JOIN tabel_2
ON tabel_1.nama_kolom = tabel_2.nama_kolom;
```

```
SELECT nama_kolom
FROM tabel_1 NATURAL JOIN tabel_2;
```

b. Cross Join

Cross join merupakan *inner join* dengan seluruh kondisi *join* (tidak hanya data yang berpasangan) dianggap bernilai *true*. Sesuai dengan namanya, *join* ini akan mengembalikan semua kemungkinan pasangan atau persilangan data pada tabel yang satu dengan data pada tabel yang lainnya. *Syntax* untuk *cross join* adalah sebagai berikut:

```
SELECT nama_kolom
FROM tabel_1 CROSS JOIN tabel_2;
```

c. Left Outer Join dan Right Outer Join

Left outer join merupakan *join* yang akan mengembalikan seluruh data pada tabel sebelah kiri (*left table*) yang memiliki pasangan pada tabel sebelah kanan (*right table*) ditambah data-data pada *left table* yang tidak memiliki pasangan pada tabel sebelah ka-

nan. Sedangkan *right outer join* sebaliknya. Untuk data-data yang tidak memiliki pasangan, pada *joined table* yang dihasilkan data-data tersebut akan dipasangkan dengan data *null*. *Syntax* untuk *left outer join* dan *right outer join* adalah sebagai berikut:

```
SELECT nama_kolom
FROM tabel_kiri LEFT OUTER JOIN tabel_kanan
ON tabel_kiri.nama_kolom = tabel_kanan.nama_kolom;
```

```
SELECT nama_kolom
FROM tabel_kiri RIGHT OUTER JOIN tabel_kanan
ON tabel_kiri.nama_kolom = tabel_kanan.nama_kolom;
```

d. Full Outer Join

Full outer join merupakan kombinasi dari hasil *left outer join* dengan *right outer join*. MySQL tidak mengenal klausa *full outer join*. Oleh karena itu, untuk mendapatkan *joined table* hasil dari *full outer join* digunakan klausa atau operator UNION untuk menggabungkan hasil *query* yang menggunakan *left outer join* dengan hasil *query* yang menggunakan *right outer join*. *Syntax* untuk *full outer join* adalah sebagai berikut:

```
SELECT nama_kolom
FROM tabel_kiri LEFT OUTER JOIN tabel_kanan
ON tabel_kiri.nama_kolom = tabel_kanan.nama_kolom
UNION
SELECT nama_kolom
FROM tabel_kiri RIGHT OUTER JOIN tabel_kanan
ON tabel_kiri.nama_kolom = tabel_kanan.nama_kolom;
```

3. Penggunaan Operator IS NULL

Seperti yang telah dijelaskan pada bagian *outer join*, data yang diperoleh dari suatu *query* dapat mengandung nilai *null*. Untuk mem-filter *query* agar hanya menampilkan data yang mengandung nilai *null* saja dapat menggunakan operator **IS NULL** pada klausa WHERE.

4. Penggunaan Operator DISTINCT

Pada saat melakukan pengambilan data di dalam *database* terkadang dijumpai suatu kondisi dimana data yang sama diperoleh atau muncul lebih dari satu baris. Untuk menghilangkan kemunculan data yang berulang-ulang ini dapat menggunakan operator **DISTINCT** pada klausa SELECT. Dengan menggunakan operator ini, data yang sama hanya akan muncul satu kali atau hanya pada satu baris.

5. Penggunaan Operator AS

Operator **AS** berfungsi untuk mengubah nama kolom atau *field* pada suatu hasil *query*. Perubahan nama kolom ini hanya berlaku pada hasil *query* tersebut, tidak mengubah nama kolom asalnya.

A. Latihan

Buatlah tabel baru dengan struktur dan data-data sebagai berikut:

Nama Tabel	Nama Kolom	Tipe Data	Keterangan
siswa	nis	CHAR(5)	PRIMARY KEY
	nama	VARCHAR(50)	NOT NULL
	jenis_kelamin	ENUM('L', 'P')	NOT NULL
	tempat_lahir	VARCHAR(20)	NOT NULL
	tanggal_lahir	DATE	NOT NULL
	alamat	VARCHAR(100)	NOT NULL
nilai_siswa	nis	CHAR(5)	PRIMARY KEY
	kode_pelajaran	VARCHAR(8)	NOT NULL
	nilai	INT	NOT NULL

Tabel: siswa

Nis	nama	jenis_kelamin	tempat_lahir	tanggal_lahir	alamat
1000	Abdul Rizal	L	Banyuwangi	1996-06-04	Jln. Ikan Cupang
1000	Bramono Hadi	L	Surabaya	1996-05-29	Jln. Ikan Wader
1000	Denalia Putri	P	Banyuwangi	1996-02-15	Jln. Ikan Cupang
1000	Erwin Nanda Fadli	L	Malang	1997-01-14	Jln. Kalimantan

Tabel: nilai_siswa

nis	kode_pelajaran	nilai
10005	MTK2	90
10002	MTK3	86
10005	BIO2	64
10001	BIND1	90
10004	BIO2	86
10004	FIS2	89
10005	IPS	71
10004	BIND1	68
10002	FIS1	67
10002	MTK1	74
10002	PKN	73
10001	MTK3	81
10001	IPS	62
10004	MTK2	71
10001	BIO2	83
10001	BIND2	83
10005	BIO1	88
10002	MTK2	95

nis	kode_pelajaran	nilai
10001	MTK2	87
10001	BIO1	81
10005	BIND1	95
10004	MTK1	64
10002	IPS	84

1. Inner Join dan Natural Join

Inner join maupun *natural join* dapat digunakan untuk mengetahui data siswa sekaligus nilai yang diperolehnya. Jalankan *query* berikut dan perhatikan hasilnya:

Query 1

```
SELECT *  
FROM siswa NATURAL JOIN nilai_siswa;
```

Query 2

```
SELECT *  
FROM siswa INNER JOIN nilai_siswa  
ON siswa.nis = nilai_siswa.nis;
```

Perbedaan apakah yang Anda temukan antara hasil dari *query 1* dibandingkan dengan hasil dari *query 2*?

2. Cross Join

Jalankan *query* berikut:

```
SELECT *  
FROM siswa CROSS JOIN nilai_siswa;
```

Bandingkan hasil *query* tersebut dengan *query* yang menggunakan INNER JOIN dan NATURAL JOIN! Perbedaan apa yang Anda temukan?

3. Left Outer Join dan Right Outer Join

Jalankan *query* berikut dan perhatikan hasilnya:

```
SELECT *  
FROM siswa LEFT OUTER JOIN nilai_siswa  
ON siswa.nis = nilai_siswa.nis;
```

Jalankan pula *query* berikut dan perhatikan kembali hasilnya:

```
SELECT *  
FROM siswa RIGHT OUTER JOIN nilai_siswa  
ON siswa.nis = nilai_siswa.nis;
```

Perbedaan dan persamaan apakah yang Anda temukan dari hasil kedua *query*

tersebut? Adakah data dengan nilai *null* yang muncul di sana? Apakah yang menyebabkan data dengan nilai *null* tersebut muncul?

Jalankan kembali *query* yang menggunakan operator IS NULL berikut:

```
SELECT *  
FROM siswa LEFT OUTER JOIN nilai_siswa  
ON siswa.nis = nilai_siswa.nis  
WHERE nilai_siswa.nis IS NULL;
```

Apakah perbedaan yang Anda temukan antara hasil *query* LEFT OUTER JOIN yang mengandung operator IS NULL ini dengan *query* yang LEFT OUTER JOIN tanpa operator IS NULL?

4. Full Outer Join

Untuk melihat bentuk *joined table* dari hasil *query* yang menggunakan klausa FULL OUTER JOIN jalankanlah *query* berikut dan perhatikan hasilnya:

```
SELECT *  
FROM siswa LEFT OUTER JOIN nilai_siswa  
ON siswa.nis = nilai_siswa.nis  
UNION  
SELECT *  
FROM siswa RIGHT OUTER JOIN nilai_siswa  
ON siswa.nis = nilai_siswa.nis;
```

5. Penggunaan Operator DISTINCT dan AS

Contoh penggunaan operator DISTINCT adalah pada saat pengguna ingin mengetahui siswa mana saja yang nilainya sudah tercatat di dalam *database*. Untuk mendapatkan data tersebut dapat menggunakan *query* berikut ini:

```
SELECT siswa.nis, siswa.nama  
FROM siswa NATURAL JOIN nilai_siswa;
```

Jalankan *query* tersebut kemudian perhatikan apakah ada lebih dari satu *record* dengan data yang sama? Bandingkanlah dengan hasil dari *query* yang mengandung operator DISTINCT berikut:

```
SELECT DISTINCT siswa.nis, siswa.nama  
FROM siswa NATURAL JOIN nilai_siswa;
```

Sedangkan contoh penggunaan klausa AS adalah pada saat pengguna ingin menampilkan data rata-rata nilai yang diperoleh tiap siswa. Jika tanpa menggunakan operator AS, maka *query* yang digunakan untuk mendapatkan data tersebut adalah sebagai berikut:

```
SELECT siswa.nis, siswa.nama,
AVG(nilai) FROM siswa NATURAL JOIN
nilai_siswa GROUP BY siswa.nis;
```

Perhatikan hasil yang diperoleh, terutama nama kolom ketiga. Kemudian jalankan *query* berikut dan perhatikan hasilnya:

```
SELECT siswa.nis, siswa.nama, AVG(nilai) AS "Rata-Rata"
FROM siswa NATURAL JOIN nilai_siswa
GROUP BY siswa.nis;
```

Perhatikan kolom ketiga dari hasil yang diperoleh. Apakah terjadi perubahan nama kolom jika dibandingkan dengan *query* sebelumnya?

B. Tugas Praktikum

Sebuah perpustakaan sekolah memiliki suatu *database* yang digunakan untuk menyimpan data buku serta mencatat data peminjaman dan pengembalian buku oleh siswa. Struktur tabel dan data-data di dalam *database* milik perpustakaan tersebut adalah sebagai berikut:

Nama Tabel	Kolom	Tipe Data	Keterangan
siswa	nis	CHAR(5)	PRIMARY KEY
	nama	VARCHAR(50)	NOT NULL
	alamat	VARCHAR(30)	NOT NULL
	gender	ENUM('L', 'P')	NOT NULL
	jurusan	VARCHAR(3)	NOT NULL
	kelas	ENUM('X', 'XI', 'XII')	NOT NULL
buku	kode_buku	CHAR(7)	PRIMARY KEY
	judul	VARCHAR(50)	NOT NULL
	pengarang	VARCHAR(50)	NOT NULL
	penerbit	VARCHAR(20)	NOT NULL
	tahun_terbit	YEAR	NOT NULL
	stok_awal	INT(2)	NOT NULL DEFAULT 0
peminjaman	id_pinjam	INT	PRIMARY KEY
	nis	CHAR(5)	NOT NULL
	kode_buku	CHAR(7)	NOT NULL
	tgl_pinjam	DATE	NOT NULL
pengembalian	id_pinjam	INT	NOT NULL
	tgl_kembali	DATE	NOT NULL

Tabel: siswa

nis	nama	alamat	gender	jurusan	kelas
10001	Alif Hanan	Muncar	L	RPL	XII
10002	Minawati Sulistia	Purwoharjo	P	RPL	XII

nis	nama	alamat	gender	jurusan	kelas
10003	Citra Hadi Rozikin	Banyuwangi	P	RPL	XII
10004	Wanda Putri	Srono	P	APK	XII
10005	Yunita Nurjanah	Purwoharjo	P	APK	XII
10006	Yuyun Suci Anggarista	Genteng	P	APK	XII
11001	Rusdiana	Banyuwangi	P	RPL	XI
11002	Maulidina Wati	Banyuwangi	P	RPL	XI
11003	Ahmad Ruslan	Wongsorejo	L	RPL	XI
11004	Lutfi Hasan	Banyuwangi	L	PM	XI
11005	Malik Ibrahim	Banyuwangi	L	PM	XI
11006	Maulana Indra	Muncar	L	PM	XI
12001	Ni Nyoman Sukma Yanuarti	Gilimanuk	P	RPL	X
12002	Axcel Bramono	Banyuwangi	L	RPL	X
12003	Jhonatan Putra	Genteng	L	RPL	X
12004	Nanda Nirmala	Banyuwangi	P	APH	X
12005	Muslimah	Banyuwangi	P	APH	X
12006	Ina Pratiwi	Genteng	P	APH	X

Tabel: buku

kode_buku	judul	pengarang	penerbit	tahun_terbit	stok_awal
NOR0001	English for Business	Mc. Steward	Graha Pustaka	2009	20
NOR0002	Pendidikan Kewarganegaraan	Ahmad Harits	Aksara Pelita	2010	40
NOR0003	Matematika untuk SMK	Ridwan Anam	Aksara Pelita	2010	50
PSR0001	Manajemen Penjualan	Siti Nasuha	Bumi Merdeka	2011	10
PSR0002	Strategi Menarik Pelanggan	Siti Nasuha	Bumi Merdeka	2011	10
TIK0001	Belajar MySQL	Abdul Kadir	Andi Publisher	2008	10
TIK0002	Handal Pemrograman PHP	Lukmanul Hakim	Lokomedia	2009	20
TIK0003	Pemrograman	Abdul Kadir	Andi Publisher	2009	10
TIK0004	Pemrograman Berbasis Objek	Abdul Kadir	Andi Publisher	2010	8

Tabel: peminjaman

id_pinjam	nis	kode_buku	tgl_pinjam
1	11003	TIK0004	2012-09-24
2	11002	TIK0001	2012-09-24
3	11004	PSR0001	2012-09-25
4	11004	PSR0002	2012-09-25
5	11005	PSR0001	2012-09-25
6	11005	PSR0002	2012-09-25
7	12004	NOR0001	2012-09-27
8	10004	NOR0003	2012-09-28
9	12003	TIK0001	2012-09-28
10	12001	TIK0001	2012-09-28

Tabel: pengembalian

id_pinjam	tgl_kembali
2	2012-09-26
6	2012-09-27
3	2012-09-28
4	2012-09-28
10	2012-10-03

Berdasarkan data-data tersebut, tuliskanlah *query* SQL untuk menyelesaikan permasalahan-permasalahan berikut dengan memanfaatkan prinsip JOIN:

1. Menampilkan **nama**, **kelas**, dan **jurusan** siswa yang **pernah** melakukan peminjaman buku.

nama	kelas	jurusan
Ahmad Ruslan	XI	RPL
Maulidina Wati	XI	RPL
Lutfi Hasan	XI	PM
Malik Ibrahim	XI	PM
Nanda Nirmala	X	APH
Wanda Putri	XII	APK
Jhonatan Putra	X	RPL
Ni Nyoman Sukma Yanuarti	X	RPL

2. Menampilkan **nama**, **kelas**, dan **jurusan** siswa yang **tidak pernah** melakukan peminjaman buku.

nama	kelas	jurusan
Alif Hanan	XII	RPL
Minawati Sulistia	XII	RPL
Citra Hadi Rozikin	XII	RPL
Yunita Nurjanah	XII	APK
Yuyun Suci Anggarista	XII	APK
Rusdiana	XI	RPL
Maulana Indra	XI	PM
Axcel Bramono	X	RPL
Muslimah	X	APH
Ina Pratiwi	X	APH

3. Menampilkan **judul** dan **pengarang** buku yang pernah dipinjam oleh siswa.

judul	pengarang
Pemrograman Berbasis Objek	Abdul Kadir
Belajar MySQL	Abdul Kadir
Manajemen Penjualan	Siti Nasuha
Strategi Menarik Pelanggan	Siti Nasuha
English for Business	Mc. Steward
Matematika untuk SMK	Ridwan Anam

4. Menampilkan **judul** dan **pengarang** buku yang belum pernah dipinjam oleh siswa.

judul	pengarang
Pendidikan Kewarganegaraan	Ahmad Harits
Handal Pemrograman PHP	Lukmanul Hakim
Pemrograman JavaScript	Abdul Kadir

5. Menampilkan **nama**, **kelas**, dan **jurusan** siswa yang pernah melakukan peminjaman buku beserta **judul** dan **tanggal peminjamannya**.

nama	kelas	jurusan	judul	tgl_pinjam
Ahmad Ruslan	XI	RPL	Pemrograman Berbasis Objek	2012-09-24
Maulidina Wati	XI	RPL	Belajar MySQL	2012-09-24
Lutfi Hasan	XI	PM	Manajemen Penjualan	2012-09-25
Lutfi Hasan	XI	PM	Strategi Menarik Pelanggan	2012-09-25
Malik Ibrahim	XI	PM	Manajemen Penjualan	2012-09-25
Malik Ibrahim	XI	PM	Strategi Menarik Pelanggan	2012-09-25
Nanda Nirmala	X	APH	English for Business	2012-09-27
Wanda Putri	XII	APK	Matematika untuk SMK	2012-09-28
Jhonatan Putra	X	RPL	Belajar MySQL	2012-09-28
Ni Nyoman Sukma Yanuarti	X	RPL	Belajar MySQL	2012-09-28

6. Menampilkan data **nama**, **kelas**, dan **jurusan** siswa yang melakukan peminjaman buku beserta **jumlah peminjamannya**.

nama	kelas	jurusan	jumlah_peminjaman
Ahmad Ruslan	XI	RPL	1
Jhonatan Putra	X	RPL	1
Lutfi Hasan	XI	PM	2
Malik Ibrahim	XI	PM	2
Maulidina Wati	XI	RPL	1
Nanda Nirmala	X	APH	1
Ni Nyoman Sukm	X	RPL	1
Wanda Putri	XII	APK	1

7. Menampilkan **judul** buku yang sudah dikembalikan beserta **nama**, **kelas**, dan **jurusan** siswa yang meminjamnya juga **tanggal peminjaman** dan **tanggal pengembaliannya**.

judul	nama	kelas	jurusan	tgl_pinjam	tgl_kembali
Belajar MySQL	Maulidina Wati	XI	RPL	2012-09-24	2012-09-26
Strategi Menarik Pelanggan	Malik Ibrahim	XI	PM	2012-09-25	2012-09-27
Manajemen Penjualan	Lutfi Hasan	XI	PM	2012-09-25	2012-09-28
Strategi Menarik Pelanggan	Lutfi Hasan	XI	PM	2012-09-25	2012-09-28
Belajar MySQL	Ni Nyoman Sukma Yanuarti	X	RPL	2012-09-28	2012-10-03

8. Menampilkan **nama**, **kelas**, dan **jurusan** siswa yang belum mengembalikan buku.

nama	kelas	jurusan
Ahmad Ruslan	XI	RPL
Malik Ibrahim	XI	PM
Nanda Nirmala	X	APH
Wanda Putri	XII	APK
Jhonatan Putra	X	RPL