

## MODUL 8 – SQL JOIN

### 1.1. CAPAIAN PEMBELAJARAN

1. Mahasiswa mampu menjelaskan SQL untuk mengakses multi table
2. Mahasiswa mampu mengimplementasikan SQL untuk mengakses multi table

### 1.2. ALAT DAN BAHAN

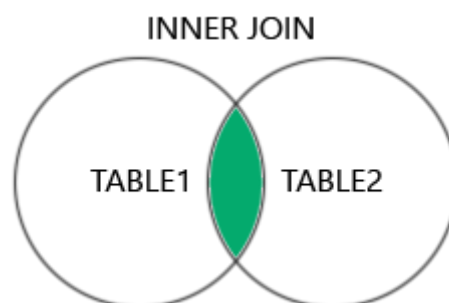
1. Seperangkat komputer lengkap/Laptop dengan koneksi internet
2. Sistem Operasi Windows/Mac/Linux
3. Aplikasi Paket Web server XAMPP
4. Aplikasi Kantor (Microsoft Office/Libre Office/WPS Office/etc)

### 1.3. DASAR TEORI

Join merupakan salah satu konsep penting dalam SQL yang memungkinkan pengguna untuk menggabungkan data dari dua atau lebih tabel berdasarkan kondisi yang ditentukan. Dengan menggunakan join, pengguna dapat menghasilkan hasil query yang lebih kaya dan lebih terperinci dengan mengambil informasi dari beberapa tabel sekaligus.

Terdapat 4 jenis SQL join berdasarkan hasil yang ditampilkan. Jenis yang pertama adalah INNER JOIN yaitu join yang menggabungkan baris dari dua tabel berdasarkan kondisi yang sesuai di kedua tabel. Ini menghasilkan hasil yang hanya berisi baris-baris di mana kondisi join terpenuhi. INNER JOIN menghasilkan hasil yang terdiri dari kolom-kolom yang ditentukan dari kedua tabel. Bentuk umum dari perintah INNER JOIN adalah sebagai berikut :

```
SELECT *  
FROM table1 A  
INNER JOIN table2 B  
ON A.columnName = B.columnName;
```

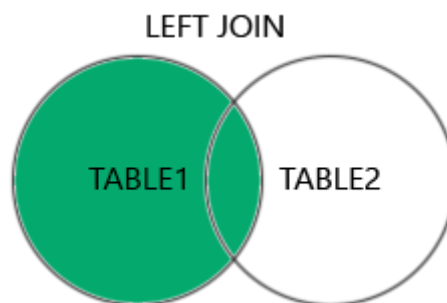


*Gambar 1 SQL Inner Join*

Jenis yang kedua yaitu *LEFT JOIN* yang menggabungkan semua baris dari tabel kiri (tabel pertama dalam klausa JOIN) dengan baris-baris yang cocok dari tabel kanan (tabel kedua dalam klausa JOIN) berdasarkan kondisi yang ditentukan. Jika tidak ada cocokan, nilai NULL akan

digunakan untuk kolom-kolom dari tabel kanan. *LEFT JOIN* memastikan bahwa semua baris dari tabel kiri tetap termasuk dalam hasil, bahkan jika tidak ada cocokan dalam tabel kanan. Bentuk umum dari perintah *LEFT JOIN* adalah sebagai berikut :

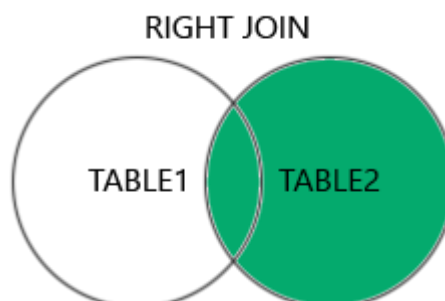
```
SELECT *  
FROM table1 A  
LEFT JOIN table2 B  
ON A.columnName = B.columnName;
```



*Gambar 2 SQL Left Join*

Jenis yang ketiga yaitu *RIGHT JOIN*. *RIGHT JOIN* mirip dengan *LEFT JOIN*, tetapi menggabungkan semua baris dari tabel kanan dengan baris-baris yang cocok dari tabel kiri. Jika tidak ada cocokan, nilai NULL akan digunakan untuk kolom-kolom dari tabel kiri. *RIGHT JOIN* memastikan bahwa semua baris dari tabel kanan tetap termasuk dalam hasil, bahkan jika tidak ada cocokan dalam tabel kiri. Bentuk umum dari perintah *RIGHT JOIN* adalah sebagai berikut :

```
SELECT *  
FROM table1 A  
RIGHT JOIN table2 B  
ON A.columnName = B.columnName;
```

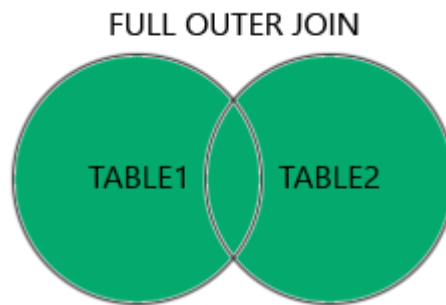


*Gambar 3 SQL Right Join*

Jenis yang keempat yaitu *FULL JOIN* yang menggabungkan semua baris dari kedua tabel, mencocokkan baris yang sesuai dari setiap tabel dan menyertakan baris yang tidak memiliki cocokan di tabel lain. Jika tidak ada cocokan, nilai NULL akan digunakan untuk kolom-kolom dari tabel yang

tidak memiliki cocokan. FULL JOIN menghasilkan gabungan lengkap dari baris-baris dari kedua tabel. Bentuk umum dari perintah *RIGHT JOIN* adalah sebagai berikut :

```
SELECT *  
FROM table1 A  
FULL JOIN table2 B  
ON A.columnName = B.columnName;
```



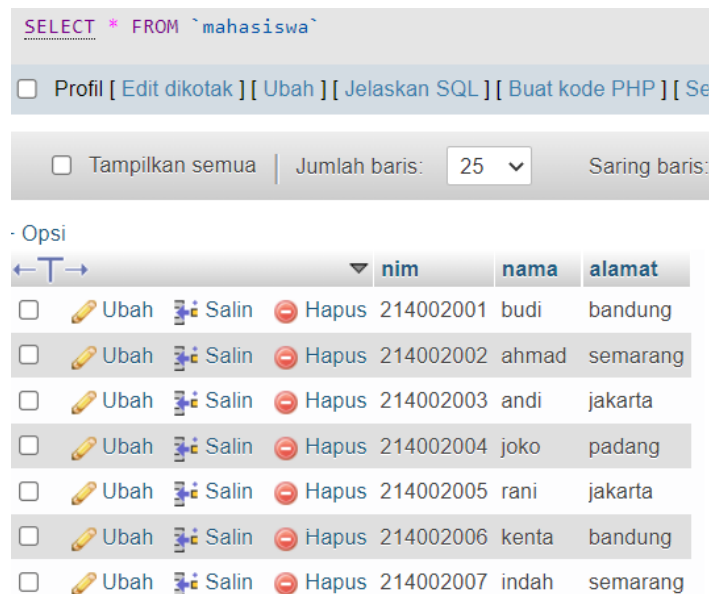
*Gambar 4 SQL Full Join*

## 1.4. PRAKTIKUM

Langkah praktikum yang pertama adalah siapkan 2 tabel yang akan kita gunakan untuk operasi join. Tabel-tabel tersebut adalah tabel sebagai berikut :

### a. Tabel mahasiswa

Tabel mahasiswa berisi data mahasiswa yang meliputi NIM dengan tipe data char(9) sebagai primary key, nama varchar(30) NOT NULL dan alamat dengan tipe data varchar(30) NOT NULL. Setelah tabel berhasil dibuat lalu masukkan data seperti pada gambar berikut :



```
SELECT * FROM `mahasiswa`
```

☐ Profil [ Edit dikotak ] [ Ubah ] [ Jelaskan SQL ] [ Buat kode PHP ] [ Se...

☐ Tampilkan semua | Jumlah baris: 25 ▾ Saring baris:

+ Opsi

				nim	nama	alamat
<input type="checkbox"/>	Ubah	Salin	Hapus	214002001	budi	bandung
<input type="checkbox"/>	Ubah	Salin	Hapus	214002002	ahmad	semarang
<input type="checkbox"/>	Ubah	Salin	Hapus	214002003	andi	jakarta
<input type="checkbox"/>	Ubah	Salin	Hapus	214002004	joko	padang
<input type="checkbox"/>	Ubah	Salin	Hapus	214002005	rani	jakarta
<input type="checkbox"/>	Ubah	Salin	Hapus	214002006	kenta	bandung
<input type="checkbox"/>	Ubah	Salin	Hapus	214002007	indah	semarang

### b. Tabel peminjaman

Tabel peminjaman berisi data peminjaman buku oleh mahasiswa yaitu id\_peminjaman dengan tipe data integer primary key auto increment, nim char(9) NOT NULL, dan buku varchar(100) NOT NULL. Setelah tabel berhasil dibuat lalu isikan data seperti pada gambar berikut :



```
SELECT * FROM `peminjaman`
```

☐ Profil [ Edit dikotak ] [ Ubah ] [ Jelaskan SQL ] [ Buat kode PHP ] [ Segarkan ]

☐ Tampilkan semua | Jumlah baris: 25 ▾ Saring baris: Cari di tabel ini

+ Opsi

				id_peminjaman	nim	buku
<input type="checkbox"/>	Ubah	Salin	Hapus	1	214002001	Buku Psikologi Klinis
<input type="checkbox"/>	Ubah	Salin	Hapus	2	214002002	Buku Biologi
<input type="checkbox"/>	Ubah	Salin	Hapus	3	214002003	Buku Kimia
<input type="checkbox"/>	Ubah	Salin	Hapus	4	214002005	Buku Teknik Kelistrikan
<input type="checkbox"/>	Ubah	Salin	Hapus	5	214002006	Buku Teknik Bangunan
<input type="checkbox"/>	Ubah	Salin	Hapus	6	214002010	Buku Dasar-dasar Pemrograman
<input type="checkbox"/>	Ubah	Salin	Hapus	7	214002011	Buku Fisika

### 1.4.1. INNER JOIN

Lakukan perintah SELECT dengan menggunakan INNER JOIN untuk mengambil data pada kedua tabel tersebut. Tuliskan perintah seperti berikut ini :

```
SELECT * FROM mahasiswa INNER JOIN peminjaman ON mahasiswa.nim = peminjaman.nim;
```

Lalu eksekusi perintah tersebut dan perhatikan hasil yang ditampilkan oleh perintah tersebut!

#### 1.4.2. LEFT JOIN

Lakukan perintah SELECT dengan menggunakan LEFT JOIN untuk mengambil data pada kedua tabel tersebut. Tuliskan perintah seperti berikut ini :

```
SELECT* FROM mahasiswa LEFT JOIN peminjaman ON mahasiswa.nim = peminjaman.nim;
```

Lalu eksekusi perintah tersebut dan perhatikan hasil yang ditampilkan oleh perintah tersebut!

#### 1.4.3. RIGHT JOIN

Lakukan perintah SELECT dengan menggunakan RIGHT JOIN untuk mengambil data pada kedua tabel tersebut. Tuliskan perintah seperti berikut ini :

```
SELECT* FROM mahasiswa RIGHT JOIN peminjaman ON mahasiswa.nim = peminjaman.nim;
```

Lalu eksekusi perintah tersebut dan perhatikan hasil yang ditampilkan oleh perintah tersebut!

#### 1.4.4. FULL JOIN

Perintah FULL join pada MySQL tidak bisa langsung kita eksekusi dengan perintah FULL JOIN karena pada MySQL tidak menyediakan syntax tersebut. Jika ingin menggunakan FULL JOIN pada MySQL maka kita butuh satu perintah untuk menggabungkan 2 perintah sekaligus yaitu menggunakan perintah UNION. Sebagai contoh silahkan tuliskan perintah berikut ini :

```
1 SELECT * FROM mahasiswa LEFT JOIN peminjaman ON mahasiswa.nim = peminjaman.nim
2 UNION ALL
3 SELECT * FROM mahasiswa RIGHT JOIN peminjaman ON mahasiswa.nim = peminjaman.nim
```

Lalu eksekusi perintah tersebut dan perhatikan hasil yang ditampilkan oleh perintah tersebut! Setelah anda selesai mempraktikkan tipe-tipe dari join tabel tersebut silahkan perhatikan beberapa perbedaan pada keluaran masing-masing perintah, lalu tuliskan pada laporan hasil pengamatan anda.

### 1.5. TUGAS MODUL 7

#### 1.5.1. Soal

1. Buatlah sebuah database baru dengan nama DBMS11-NIM (NIM diisi NIM masing-masing), kemudian membuat sebuah tabel dengan nama TbPegawai dengan struktur table sbb:

nip	varchar(5)	PK
nama	varchar(25)	
tgl_lahir	date	
Kota	varchar(15)	
thn_masuk	year(4)	
gol	int(1)	

2. Isikan data pada TbPegawai sebagai berikut :

001	Ahmad Burhanuddin	1990-12-12	Bandung	2010	3
002	Amin Imsyorry	1989-12-02	Ciamis	2009	2
003	Budhy Bungaox	1989-01-20	Tasikmalaya	2011	4
004	Zulkarnaen	1991-02-20	Bandung	2009	1
005	Dewi Sudewa	1990-12-02	Bogor	2009	1
006	Ina Nurlian	1993-08-09	Sumedang	2011	4
007	Cheppy Chardut	1992-07-09	Garut	2011	4
008	Dodong M	1990-07-10	Bandung	2010	3
009	Gandung P	1990-06-25	Bogor	2009	2
010	Agus Wisman	1993-10-17	Garut	2010	3

3. Buatlah Tabel baru dengan nama TbGaji dengan struktur Gol Int(1) dan Gapok int(12), kemudian isikan data masing-masing golongan dan gaji sebagai berikut :

Gol	Gapok
1	1500000
2	2000000
3	3500000
4	5000000

4. Relasikan table TbPegawai dengan TbGaji.

5. Tampilkan data dengan memanfaatkan perintah JOIN (INNER, LEFT, RIGHT, dan FULL) lalu amatilah hasilnya!

### 1.5.2. Petunjuk Pengerjaan

a) Tugas:

- Tugas membuat perintah SQL sesuai pada soal
- Tuliskan perintah dan screenshot hasil perintah.

b) Laporan:

- Buatlah laporan akhir berdasarkan diagram yang Anda buat.
- Laporan dibuat sesuai format dan **diketik**.

- Masukkan langkah-langkah pengerjaan tugas ke dalam laporan dalam bentuk screenshot dan penjelasan.
- **Cantumkan tanda tangan** Anda di setiap halaman dokumen laporan.
- Jika di dalam laporan ada gambar atau *screenshot* yang ingin ditampilkan, bisa ditempelkan pada halaman(menyesuaikan)
- Laporan disimpan dalam bentuk pdf
- Penamaan *file* pdf : " **LaporanModul8\_DBMS\_NPM.pdf**"

c) Pengumpulan:

- *File* yang dikumpulkan yaitu:
  - Laporan : **TugasModul8\_DBMS\_NPM.pdf**
- **Batas Pengumpulan:** Sebelum Pertemuan Praktik Ke 12.