

Modul 5

Join Antar Tabel

5.1 Join

Join merupakan sebuah operasi yang digunakan untuk mendapatkan data gabungan dari dua tabel atau lebih. Operasi ini digunakan dalam perintah SELECT dan biasanya dipakai untuk memperoleh data secara detail dari tabel-tabel yang saling terkait (memiliki relasi). Misalnya karena tabel `jadwal_dokter` hanya memuat kode dokter saja tanpa nama dokter, maka kita harus melakukan join dengan tabel dokter yang memuat nama dokter.

Terdapat 3 jenis join yaitu :

1. Cross Join (*Cartesian Join*)

Cross join menghasilkan output berupa kombinasi dari semua baris yang terdapat dalam tabel-tabel yang digabungkan baik yang berpasangan maupun yang tidak berpasangan. Pada kenyataannya join jenis ini jarang bahkan tidak pernah dipakai. Meskipun begitu, jenis join inilah yang merupakan dasar dari join antartabel.

2. Inner Join

Inner join menghasilkan output berupa kombinasi baris-baris yang memiliki pasangan saja. Kombinasi baris yang tidak berpasangan akan dieliminasi atau tidak digunakan. Baris-baris yang tidak memiliki pasangan pada tabel lainnya juga tidak dimunculkan.

3. Outer Join

Outer join hampir sama dengan inner join, hanya saja baris yang tak memiliki pasangan tetap akan ditampilkan. Outer join dibagi menjadi tiga, yaitu:

- Left join
- Right join
- Full outerjoin

Sintaks umum yang sering dipakai:

Untuk Cross Join

```
SELECT tabel_1.kolom, tabel_2.kolom
FROM tabel_1, tabel_2

atau

SELECT tabel_1.kolom, tabel_2.kolom
FROM tabel_1 CROSS JOIN tabel_2
```

Untuk Inner Join

```
SELECT tabel_1.kolom, tabel_2.kolom
FROM tabel_1, tabel_2
WHERE tabel_1.kolom_kunci_1=tabel_2.kolom_kunci_2

atau

SELECT tabel_1.kolom, tabel_2.kolom
FROM tabel_1 INNER JOIN tabel_2
ON tabel_1.kolom_kunci_1=tabel_2.kolom_kunci_2
```

Untuk Outer Join

```
SELECT tabel_1.kolom, tabel_2.kolom
FROM tabel_1 [LEFT|RIGHT|FULL OUTER JOIN] tabel_2
ON tabel_1.kolom_kunci_1=tabel_2.kolom_kunci_2
```

Contoh:

Perintah untuk menggabungkan data tabel jadwal_dokter dengan tabel dokter

```
SELECT * FROM jadwal_dokter INNER JOIN dokter
ON jadwal_dokter.kd_dokter=dokter.kd_dokter
```

5.2 Outer Join

Operasi outer join terbagi menjadi tiga jenis yaitu :

5.2.1 Left outer join

Data yang ada pada tabel sebelah kiri akan ditampilkan semua sesuai pasangannya, jika ada data yang tidak memiliki pasangan, pada tabel sebelah kanan akan diisi dengan NULL.

5.2.2 Right outerjoin

Data yang ada pada tabel sebelah kanan akan ditampilkan semua sesuai pasangannya, jika ada data yang tidak memiliki pasangan, pada tabel sebelah kiri akan diisikan NULL.

5.2.3 Full outerjoin

Semua data dari tabel-tabel yang di join ditampilkan, baik itu yang secara left outer join, right outer join maupun inner join.

Aktifkan database anda kemudian jalankan perintah berikut:

```
SELECT * FROM dokter LEFT JOIN jadwal_dokter
ON  dokter.kd_dokter=jadwal_dokter.kd_dokter

SELECT * FROM dokter RIGHT JOIN jadwal_dokter
ON  dokter.kd_dokter=jadwal_dokter.kd_dokter

SELECT * FROM dokter FULL JOIN jadwal_dokter
ON  dokter.kd_dokter=jadwal_dokter.kd_dokter
```

Coba bandingkan perintah di atas dengan perintah di bawah ini:

```
SELECT * FROM jadwal_dokter LEFT JOIN dokter
ON  dokter.kd_dokter=jadwal_dokter.kd_dokter

SELECT * FROM jadwal_dokter RIGHT JOIN dokter
ON  dokter.kd_dokter=jadwal_dokter.kd_dokter

SELECT * FROM jadwal_dokter FULL JOIN dokter
ON  dokter.kd_dokter=jadwal_dokter.kd_dokter
```

Untuk lebih memahami perbedaan antara berbagai jenis JOIN, coba jalankan perintah-perintah di bawah ini:

```
SELECT dokter.nama, jadwal_dokter.hari, jadwal_dokter.shift
FROM dokter INNER JOIN jadwal_dokter
ON  dokter.kd_dokter=jadwal_dokter.kd_dokter

SELECT dokter.nama, jadwal_dokter.hari, jadwal_dokter.shift
FROM dokter LEFT JOIN jadwal_dokter
ON  dokter.kd_dokter=jadwal_dokter.kd_dokter

SELECT dokter.nama, jadwal_dokter.hari, jadwal_dokter.shift
FROM dokter RIGHT JOIN jadwal_dokter
ON  dokter.kd_dokter=jadwal_dokter.kd_dokter

SELECT dokter.nama, jadwal_dokter.hari, jadwal_dokter.shift
FROM dokter FULL JOIN jadwal_dokter
ON  dokter.kd_dokter=jadwal_dokter.kd_dokter
```

Setelah menjalankan query di atas, ada terdapat beberapa persamaan dan perbedaan antara operasi join yang satu dengan lainnya. Yang menjadi kesamaannya ialah bahwa semua operasi join pasti menggabungkan beberapa tabel untuk mendapatkan data gabungan, sedangkan perbedaan antara cross join dengan jenis join lainnya adalah dengan ada tidaknya persyaratan atau pengecekan kondisi join. Cross join merupakan jenis join yang tidak memakai persyaratan, sementara jenis join lainnya memerlukan persyaratan. Persyaratan yang dimaksud adalah data pada kolom foreign key di tabel yang merujuk harus sama dengan data pada kolom primary key di tabel yang dirujuk.

Secara umum persyaratan atau kondisi tersebut dapat dituliskan sebagai berikut:

```
tabel_1.kolom_primary_key = tabel_2.kolom_foreign_key
```

Sintaks tersebut dapat ditulis secara terbalik dan tetap memiliki arti yang sama, sebagai berikut:

```
tabel_2.kolom_foreign_key = tabel_1.kolom_primary_key
```

Catatan:

Kolom-kolom_foreign_key pada tabel_2 harus mereferensi ke kolom_primary_key pada tabel_1.

Perancangan Basis Data

Karena keharusan adanya foreign key dan primary key yang saling berhubungan itu, maka inner join dan outer join hanya dapat dilakukan pada tabel yang memiliki relasi saja. Sementara itu, cross join dapat dilakukan pada sembarang tabel, baik tabel yang saling berhubungan maupun tidak.

Untuk INNER JOIN terdapat 2 cara penulisan, yaitu:

1. Cara pertama adalah dengan memisahkan daftar tabel yang akan dijoin dengan tanda koma dan melakukan pengecekan kondisi menggunakan WHERE.
2. Cara kedua adalah menggunakan *keyword* INNER JOIN atau disingkat dengan JOIN saja di antara dua tabel yang akan di-join-kan dan melakukan pengecekan kondisi dengan *keyword* ON.

Cara pertama lebih singkat dan akan lebih terasa jika yang di-joinkan lebih dari dua tabel. Hal ini terjadi karena *keyword* JOIN dan ON merupakan satu kesatuan yang hanya dapat diterapkan pada dua buah tabel saja. Untuk lebih dari dua tabel kita harus menuliskan JOIN dan ON berulang kali. Perhatikan contoh berikut:

```
SELECT resep.kd_resep, pasien.nama, penyakit.penyakit
FROM resep, pasien, penyakit
WHERE resep.kd_pasien=pasien.kd_pasien
AND resep.kd_penyakit=penyakit.kd_penyakit

SELECT resep.kd_resep, pasien.nama, penyakit.penyakit
FROM resep JOIN pasien
ON resep.kd_pasien=pasien.kd_pasien
JOIN penyakit
ON resep.kd_penyakit=penyakit.kd_penyakit
```

Walaupun penulisannya lebih panjang tapi bentuk yang kedua lebih dianjurkan karena membutuhkan memory yang lebih sedikit dan prosesnya lebih cepat. Hal ini akan terasa sekali jika melibatkan tabel-tabel dengan data yang banyak. Tidak seperti inner join, penulisannya hanya ada satu cara. Walaupun demikian kita dapat menyingkat dengan menghilangkan *keyword* OUTER, sehingga LEFT OUTER JOIN dapat disingkat menjadi LEFT JOIN dan seterusnya.

Perancangan Basis Data

5.2 Praktikum

- 1) Buatlah perintah untuk menggabungkan data tabel resep dengan tabel dokter
- 2) Buatlah perintah untuk menggabungkan data tabel resep dengan tabel pasien
- 3) Buatlah perintah untuk menggabungkan data tabel resep dengan tabel penyakit
- 4) Buatlah perintah untuk menggabungkan data tabel resep dengan tabel obat
- 5) Buatlah perintah untuk menampilkan jadwal dokter (nama dokter, hari, shift)
- 6) Buatlah perintah untuk menampilkan nama dokter yang praktiknya hari selasa
- 7) Buatlah perintah untuk menampilkan nama dokter dengan tanggal pemeriksaan 3 Agustus 2006
- 8) Buatlah perintah untuk menampilkan nama pasien dan penyakitnya yang diperiksa hari senin
- 9) Buatlah perintah untuk menggabungkan data tabel resep dengan tabel dokter (pasien, penyakit, dan obat).
- 10) Buatlah perintah untuk menampilkan data resep (kode resep, hari, tanggal, nama pasien, nama penyakit, nama obat, nama dokter pemeriksaan)