

BAB 2. SQL MULTI TABEL

2.1. KOMPETENSI DASAR

Setelah mempelajari bab ini mahasiswa:

- 1) Memahami penggunaan perintah query multi tabel menggunakan konsep WHERE-AND
- 2) Memahami penggunaan perintah query multi tabel menggunakan konsep JOIN

2.2. INDIKATOR

Setelah mempelajari bab ini mahasiswa:

- 1) Mampu menjelaskan dan mengimplementasikan perintah query WHERE-AND
- 2) Mampu menjelaskan dan mengimplementasikan perintah query JOIN

2.3. URAIAN MATERI

2.4. SQL menggunakan WHERE-AND

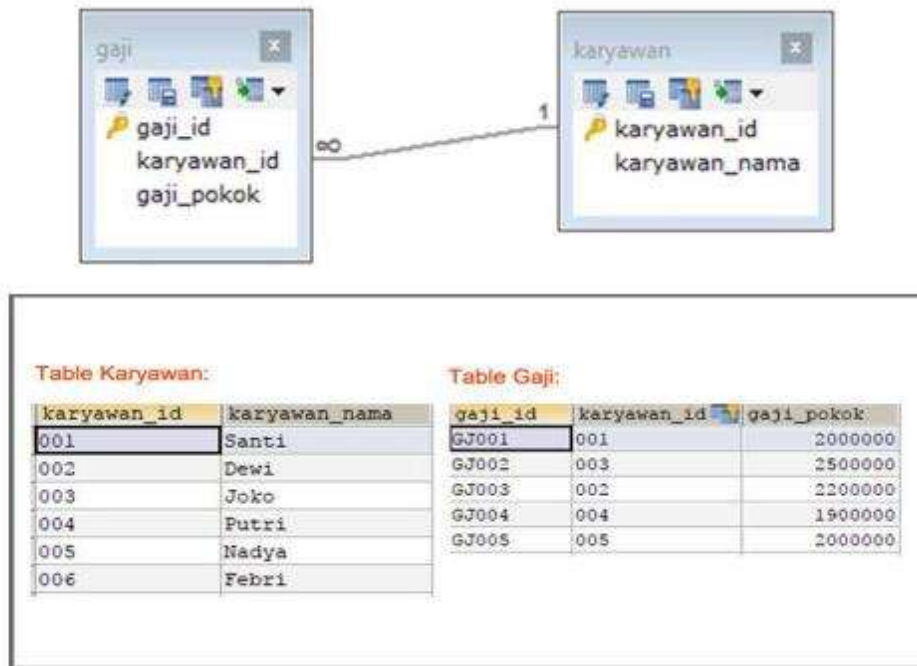
Didalam basis data relasional dimungkinkan untuk mengakses satu atau lebih tabel dalam suatu database dalam waktu bersamaan. Perintah query relasi bisa dilakukan dengan 2 macam perintah, yaitu WHERE-AND dan JOIN. Perintah JOIN sendiri mempunyai beberapa fungsi lagi antara lain INNER JOIN, OUTER JOIN, LEFT JOIN, RIGHT JOIN.

Contoh sintak dasar perintah query WHERE-AND:

```
SELECT nama_tabel1.nama_field1, nama_tabel2.nama_field2
FROM nama_tabel1, nama_tabel2, nama_tabel3
WHERE nama_tabel1.key1=nama_tabel2.key2 AND
nama_tabel3.key3=nama_tabel2.key2;
```

2.5. SQL menggunakan JOIN

Perintah query JOIN umumnya digunakan untuk menampilkan data dari beberapa tabel (relasi tabel). terdapat berbagai macam tipe JOIN antara lain Inner Join, Left Join, Right Join. Sebagai bahan pembelajaran, perhatikan relasi tabel dibawah ini.



2.5.1. Inner Join

Selain menggunakan perintah WHERE-AND, INNER JOIN merupakan jenis join yang paling umum digunakan untuk menampilkan record dari beberapa tabel. WHERE AND dan INNER JOIN menghasilkan record relasi yang sama. Perbedaan perintah tersebut terletak pada relasinya yaitu ON (untuk JOIN) dan AND (untuk WHERE-AND). Tipe join ini akan mengambil semua row dari table asal dan table tujuan dengan kondisi nilai key yang terkait saja, dan jika tidak maka row tersebut tidak akan muncul. Kalau tidak terdapat kondisi key terkait antar table, maka semua row dari kedua table dikombinasikan.

Contoh:

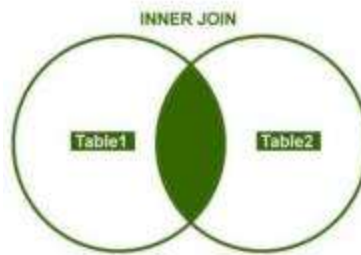
```
SELECT * FROM karyawan INNER JOIN gaji ON
karyawan.karyawan_id=gaji.karyawan_id;
```

Hasil eksekusi query diatas adalah

gaji_id	karyawan_id	gaji_pokok	karyawan_id	karyawan_nama
GJ001	001	2000000	001	Santi
GJ002	003	2500000	003	Joko
GJ003	002	2200000	002	Dewi
GJ004	004	1900000	004	Putri
GJ005	005	2000000	005	Nadya

Baris data yang ditampilkan sebanyak 5 baris data, karena INNER JOIN hanya memperhitungkan kondisi key yang terkait antara tabel karyawan dengan tabel gaji. Sedangkan

karyawan dengan karyawan_id='006' tidak ditampilkan karena tidak terkait dengan tabel gaji. Jika dibuat diagram venn-nya akan terlihat seperti gambar berikut:



2.5.2. Left Join

LEFT JOIN atau biasa juga dikenal dengan LEFT OUTER JOIN merupakan perintah join untuk menampilkan semua data sebelah kiri dari table yang di joinkan dan menampilkan data sebelah kanan yang cocok dengan kondisi join. Jika tidak ditemukan kecocokan, maka akan di set NULL secara otomatis.

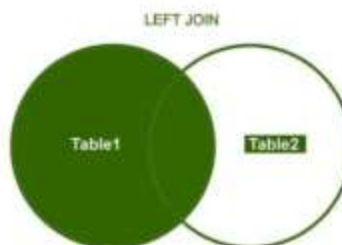
Contoh:

```
SELECT * FROM karyawan LEFT JOIN gaji ON
karyawan.karyawan_id=gaji.karyawan_id;
```

Perintah query di atas akan menghasilkan output sebagai berikut:

karyawan_id	karyawan_nama	gaji_id	karyawan_id	gaji_pokok
001	Santi	GJ001	001	2000000
003	Joko	GJ002	003	2500000
002	Dewi	GJ003	002	2200000
004	Putri	GJ004	004	1900000
005	Nadya	GJ005	005	2000000
006	Febri	(NULL)	(NULL)	(NULL)

Banyaknya record yang ditampilkan sebanyak 6 record. Karena LEFT JOIN akan menampilkan semua table sebelah kiri dari kondisi join yaitu table karyawan. Semua data pada table karyawan akan ditampilkan, meskipun tidak ada kecocokan key pada table gaji. Jika dibuat diagram venn-nya akan terlihat seperti gambar berikut:



Selain kondisi diatas, LEFT JOIN juga bisa menampilkan data yang hanya kondisi key pada table tamu (*foreign key*) kosong (*NULL*).

Contoh:

```
SELECT * FROM karyawan LEFT JOIN gaji ON
karyawan.karyawan_id=gaji.karyawan_id
WHERE gaji.karyawan IS NULL;
```

Perintah query di atas akan menghasilkan output sebagai berikut:

karyawan_id	karyawan_nama	gaji_id	karyawan_id	gaji_pokok
006	Febri	(NULL)	(NULL)	(NULL)

Data yang ditampilkan hanya 1 record. Hal ini dikarenakan, hanya ada satu data yang belum memiliki kecocokan key pada table tamu. Untuk mempermudah anda memahami perbedaan antara kedua LEFT JOIN ini coba perhatikan diagram venn berikut:



Dengan melihat perbedaan dari diagram venn tersebut, maka Anda akan mudah memahami bagaimana left join ini bekerja. Ingat LEFT JOIN ini sangat penting untuk anda pahami, karena disaat Anda mulai mengerjakan project yang cukup kompleks, maka anda akan banyak berkuat dengan left join ini.

Contoh pada kasus diatas, hanya dengan memanfaatkan left join kita bisa menampilkan semua data karyawan yang sudah ada gajinya dan siapa yang belum ada gajinya. Selain itu anda juga dapat menampilkan semua data karyawan yang belum ada gajinya dengan fungsi LEFT JOIN WHERE NULL.

2.5.3. Right Join

Kebalikan dari LEFT JOIN adalah RIGHT JOIN, atau biasa juga dikenal dengan RIGHT OUTER JOIN. RIGHT JOIN akan menampilkan semua data yang ada di table sebelah kanan dan

mencari kecocokan key pada table sebelah kiri. Jika tidak ditemukan kecocokan, maka akan di set NULL secara otomatis pada table sebelah kiri.

Contoh:

```
SELECT * FROM karyawan RIGHT JOIN gaji ON
karyawan.karyawan_id=gaji.karyawan_id;
```

Query di atas akan menampilkan output seperti gambar berikut:

gaji_id	karyawan_id	gaji_pokok	karyawan_id	karyawan_nama
GJ001	001	2000000	001	Santi
GJ002	003	2500000	003	Joko
GJ003	002	2200000	002	Dewi
GJ004	004	1900000	004	Putri
GJ005	005	2000000	005	Nadya
(NULL)	(NULL)	(NULL)	006	Febri

Pada output diatas, anda dapat melihat bahwa terdapat NULL pada table sebelah kiri. Hal ini dikarenakan tidak ditemukan kecocokan key diantara kedua table. Untuk lebih mudah memahaminya, perhatikan diagram venn berikut:



Selain kondisi diatas, RIGHT JOIN juga bisa menampilkan data yang hanya kondisi key pada table tamu (*foreign key*) kosong (NULL).

Contoh:

```
SELECT * FROM karyawan RIGHT JOIN gaji
ON karyawan.karyawan_id=gaji.karyawan_id
WHERE gaji.karyawan IS NULL;
```

Query di atas akan menghasilkan output seperti gambar berikut:

gaji_id	karyawan_id	gaji_pokok	karyawan_id	karyawan_nama
(NULL)	(NULL)	(NULL)	006	Febri

Data yang ditampilkan hanya 1 record. Hal ini dikarenakan, hanya ada satu data yang belum memiliki kecocokan key pada table tamu. Untuk mempermudah anda memahami

perbedaan antara kedua RIGHT JOIN ini coba perhatikan diagram venn berikut dan bandingkan dengan diagram venn sebelumnya:



2.6. SOAL LATIHAN

Berdasarkan soal latihan 1.4, tambahkan kembali 2 tabel untuk membentuk relasi tabel. Tabel tersebut antara lain:

Tabel Pengarang

KD_KARANG	PENGARANG	ALAMAT
12	Adityo PW	Sleman
10	Suparman	Sleman
11	Irwanto	JOgja
row(s) 1 - 3 of 3		

Tabel Penerbit

KD_TERBIT	PENERBIT	ALAMAT
1	GaYa Media	Jl. Aspal
2	Ditanam Press	Jl. Aspal Apik
3	Ty0 Publisher	Jl. Brongkalan Boto
row(s) 1 - 3 of 3		

Berdasarkan 3 tabel yang terbentuk (Tabel Buku, Tabel Pengarang, Tabel Penerbit), jawab pertanyaan berikut menggunakan perintah query.

- 1) Tampilkan judul buku, nama pengarang dan nama penerbit dari masing-masing buku menggunakan query WHERE-AND, INNER JOIN, LEFT JOIN, dan RIGHT JOIN. Kemudian perhatikan perbedaan dari masing-masing hasilnya selanjutnya jelaskan perbedaan hasil tersebut.
- 2) Tampilkan nama pengarang dan judul buku yang jumlah bukunya lebih dari 30 buah.
- 3) Tampilkan judul buku yang ditulis oleh pengarang bernama AdityoPW
- 4) Tampilkan jumlah buku yang di terbitkan oleh penerbit Ty0 Publisher.