

Pertemuan 11

Bahasa Query Terapan Lanjutan

Query Terapan lanjutan

Tabel dibawah ini untuk mengerjakan perintah **JOIN**

Tabel Nilai

| NIM | KD_MK | MID | FINAL |
|----------|-------|-----|-------|
| 10296832 | KK021 | 60 | 75 |
| 10296126 | KD132 | 70 | 90 |
| 31296500 | KK021 | 55 | 40 |
| 41296525 | KU122 | 90 | 80 |
| 21196353 | KU122 | 75 | 75 |
| 50095487 | KD132 | 80 | 0 |

Tabel Mahasiswa

| NIM | NAMA | ALAMAT |
|----------|-------------|---------|
| 10296832 | Nurhayati | Jakarta |
| 10296126 | Astuti | Jakarta |
| 31296500 | Budi | Depok |
| 41296525 | Prananigrum | Bogor |
| 50096487 | Pipit | Bekasi |
| 21196353 | Quraish | Bogor |
| 10296001 | Fintri | Depok |
| 21198002 | Julizar | Jakarta |

Tabel MataKuliah

| KD_MK | NAMA_MK | SKS |
|-------|----------------------------|-----|
| KK021 | Sistem Basis Data | 2 |
| KD132 | Sistem Informasi Manajemen | 3 |
| KU122 | Pancasila | 2 |

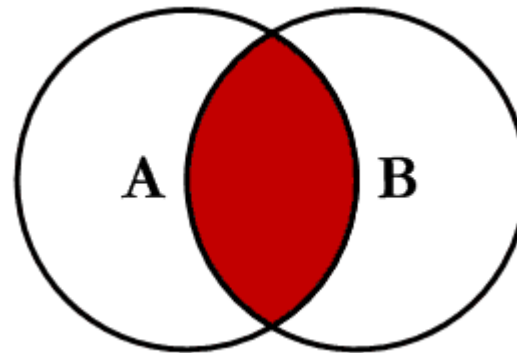
JOIN

JOIN digunakan untuk memilih data dari dua tabel atau lebih.

1. INNER JOIN

Menggabungkan dua tabel dimana diantara dua tabel datanya bersesuaian.

```
SELECT select_list  
FROM Table_A A  
JOIN Table_B B  
ON A.Key = B.Key
```



Source : https://pojokprogrammer.net/id/content/representasi-sql-join-secara-visual?language_content_entity=id

Contoh INNER JOIN

```
SELECT Nilai.NIM, Mahasiswa.NAMA, Nilai.KD_MK, Nilai.MID  
FROM Nilai INNER JOIN Mahasiswa  
ON Nilai.NIM = Mahasiswa.NIM
```

Hasil :

| NIM | NAMA | KD_MK | MID |
|----------|-------------|-------|-----|
| 10296832 | Nurhayati | KK021 | 60 |
| 10296126 | Astuti | KD132 | 70 |
| 31296500 | Budi | KK021 | 55 |
| 41296525 | Prananigrum | KU122 | 90 |
| 21196353 | Quraish | KU122 | 75 |
| 50095487 | Pipit | KD132 | 80 |

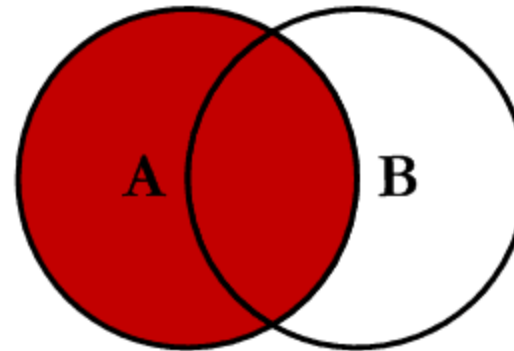
JOIN

JOIN digunakan untuk memilih data dari dua tabel atau lebih.

2. LEFT JOIN atau LEFT OUTER JOIN

Menggabungkan dua tabel dimana diantara dua tabel datanya bersesuaian dan juga semua record pada tabel sebelah kiri.

```
SELECT select_list  
FROM Table_A A  
LEFT JOIN Table_B B  
ON A.Key = B.Key
```



Source : https://pojokprogrammer.net/id/content/representasi-sql-join-secara-visual?language_content_entity=id

Contoh LEFT JOIN

```
SELECT Mahasiswa.NIM, Mahasiswa.NAMA, Nilai.KD_MK,  
Nilai.MID  
FROM Mahasiswa LEFT OUTER JOIN Nilai  
ON Nilai.NIM = Mahasiswa.NIM
```

Hasil:

| NIM | NAMA | KD_MK | MID |
|----------|-------------|-------|-----|
| 10296832 | Nurhayati | KK021 | 60 |
| 10296126 | Astuti | KD132 | 70 |
| 31296500 | Budi | KK021 | 55 |
| 41296525 | Prananigrum | KU122 | 90 |
| 21196353 | Quraish | KU122 | 75 |
| 50095487 | Pipit | KD132 | 80 |
| 10296001 | Fintri | - | - |
| 21198002 | Julizar | - | - |

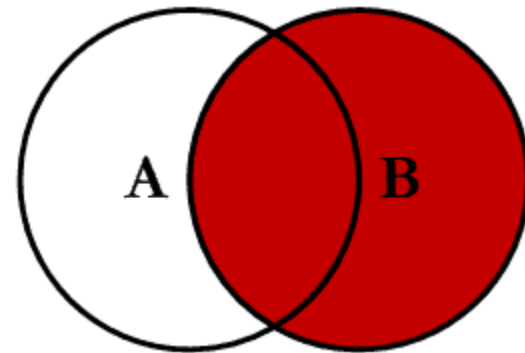
JOIN

JOIN digunakan untuk memilih data dari dua tabel atau lebih.

3. RIGHT JOIN atau RIGHT OUTER JOIN

Menggabungkan dua tabel dimana diantara dua tabel datanya bersesuaian dan juga semua record pada tabel sebelah kanan.

```
SELECT select_list  
FROM Table_A A  
RIGHT JOIN Table_B B  
ON A.Key = B.Key
```



Source : https://pojokprogrammer.net/id/content/representasi-sql-join-secara-visual?language_content_entity=id

Query Terapan lanjutan

Tabel dibawah ini untuk mengerjakan perintah **JOIN**

Tabel Nilai

| NIM | KD_MK | MID | FINAL |
|----------|-------|-----|-------|
| 10296832 | KK021 | 60 | 75 |
| 10296126 | KD132 | 70 | 90 |
| 31296500 | KK021 | 55 | 40 |
| 41296525 | KU122 | 90 | 80 |
| 21196353 | KU122 | 75 | 75 |
| 50095487 | KD132 | 80 | 0 |

Tabel Mahasiswa

| NIM | NAMA | ALAMAT |
|----------|-------------|---------|
| 10296832 | Nurhayati | Jakarta |
| 10296126 | Astuti | Jakarta |
| 31296500 | Budi | Depok |
| 41296525 | Prananigrum | Bogor |
| 50096487 | Pipit | Bekasi |
| 21196353 | Quraish | Bogor |
| 10296001 | Fintri | Depok |
| 21198002 | Julizar | Jakarta |

Tabel MataKuliah

| KD_MK | NAMA_MK | SKS |
|-------|----------------------------|-----|
| KK021 | Sistem Basis Data | 2 |
| KD132 | Sistem Informasi Manajemen | 3 |
| KU122 | Pancasila | 2 |

Contoh RIGHT JOIN

```
SELECT Mahasiswa.NIM, Mahasiswa.NAMA, Nilai.KD_MK,  
Nilai.MID  
FROM Nilai RIGHT OUTER JOIN Mahasiswa  
ON Nilai.NIM = Mahasiswa.NIM
```

Hasil :

| NIM | NAMA | KD_MK | MID |
|----------|-------------|-------|-----|
| 10296832 | Nurhayati | KK021 | 60 |
| 10296126 | Astuti | KD132 | 70 |
| 31296500 | Budi | KK021 | 55 |
| 41296525 | Prananigrum | KU122 | 90 |
| 21196353 | Quraish | KU122 | 75 |
| 50095487 | Pipit | KD132 | 80 |
| 10296001 | Fintri | - | - |
| 21198002 | Julizar | - | - |

Contoh Lain Join

tb_kota

| id_kota | nama_kota | id_provinsi |
|---------|-----------|-------------|
| 1 | Jakarta | 1 |
| 2 | Semarang | 2 |
| 3 | Pati | 2 |
| 4 | Bandung | 4 |
| 5 | Surabaya | 5 |
| 6 | Medan | 6 |

tb_provinsi

| id_provinsi | nama_provinsi |
|-------------|---------------|
| 1 | DKI Jakarta |
| 2 | Jawa Tengah |
| 3 | Papua Barat |
| 4 | Jawa Barat |
| 5 | Jawa Timur |

Tampilkan data-data yang mempunyai nilai sama antara id_provinsi pada table tb_kota dan id_provinsi pada table tb_provinsi

SELECT * FROM tb_kota INNER JOIN tb_provinsi ON tb_kota.id_provinsi = tb_provinsi.id_provinsi;

| id_kota | nama_kota | id_provinsi | id_provinsi | nama_provinsi |
|---------|-----------|-------------|-------------|---------------|
| 1 | Jakarta | 1 | 1 | DKI Jakarta |
| 2 | Semarang | 2 | 2 | Jawa Tengah |
| 3 | Pati | 2 | 2 | Jawa Tengah |
| 4 | Bandung | 4 | 4 | Jawa Barat |
| 5 | Surabaya | 5 | 5 | Jawa Timur |

Tampilkan hanya nama_kota dan nama_provinsi saja

SELECT nama_kota, nama_provinsi FROM tb_kota INNER JOIN tb_provinsi ON tb_kota.id_provinsi = tb_provinsi.id_provinsi;

| nama_kota | nama_provinsi |
|-----------|---------------|
| Jakarta | DKI Jakarta |
| Semarang | Jawa Tengah |
| Pati | Jawa Tengah |
| Bandung | Jawa Barat |
| Surabaya | Jawa Timur |

<https://yukcoding.id/belajar-inner-left-right-join-pada-sql/>

Contoh Lain Join

tb_kota

| id_kota | nama_kota | id_provinsi |
|---------|-----------|-------------|
| 1 | Jakarta | 1 |
| 2 | Semarang | 2 |
| 3 | Pati | 2 |
| 4 | Bandung | 4 |
| 5 | Surabaya | 5 |
| 6 | Medan | 6 |

tb_provinsi

| id_provinsi | nama_provinsi |
|-------------|---------------|
| 1 | DKI Jakarta |
| 2 | Jawa Tengah |
| 3 | Papua Barat |
| 4 | Jawa Barat |
| 5 | Jawa Timur |

Tampilkan data-data Pada table *tb_provinsi* (kanan).

`SELECT * FROM tb_kota LEFT JOIN tb_provinsi ON
tb_kota.id_provinsi = tb_provinsi.id_provinsi;`

| id_kota | nama_kota | id_provinsi | id_provinsi | nama_provinsi |
|---------|-----------|-------------|-------------|---------------|
| 1 | Jakarta | 1 | 1 | DKI Jakarta |
| 2 | Semarang | 2 | 2 | Jawa Tengah |
| 3 | Pati | 2 | 2 | Jawa Tengah |
| 4 | Bandung | 4 | 4 | Jawa Barat |
| 5 | Surabaya | 5 | 5 | Jawa Timur |
| 6 | Medan | 6 | NULL | NULL |

<https://yukcoding.id/belajar-inner-left-right-join-pada-sql/>

Contoh Lain Join

tb_kota

| id_kota | nama_kota | id_provinsi |
|---------|-----------|-------------|
| 1 | Jakarta | 1 |
| 2 | Semarang | 2 |
| 3 | Pati | 2 |
| 4 | Bandung | 4 |
| 5 | Surabaya | 5 |
| 6 | Medan | 6 |

tb_provinsi

| id_provinsi | nama_provinsi |
|-------------|---------------|
| 1 | DKI Jakarta |
| 2 | Jawa Tengah |
| 3 | Papua Barat |
| 4 | Jawa Barat |
| 5 | Jawa Timur |

SELECT * FROM tb_kota RIGHT JOIN tb_provinsi ON
tb_kota.id_provinsi = tb_provinsi.id_provinsi;

| id_kota | nama_kota | id_provinsi | id_provinsi | nama_provinsi |
|---------|-----------|-------------|-------------|---------------|
| 1 | Jakarta | 1 | 1 | DKI Jakarta |
| 2 | Semarang | 2 | 2 | Jawa Tengah |
| 3 | Pati | 2 | 2 | Jawa Tengah |
| 4 | Bandung | 4 | 4 | Jawa Barat |
| 5 | Surabaya | 5 | 5 | Jawa Timur |
| NULL | NULL | NULL | 3 | Papua Barat |

<https://yukcoding.id/belajar-inner-left-right-join-pada-sql/>

Data Access

1. GRANT → Memberikan hak akses / hak istimewa pengguna

Sintaks : GRANT hak_akses ON nama_db
TO nama_pemakai
[IDENTIFIED BY] [PASSWORD] 'Password'
[WITH GRANT OPTION] ;

GRANT hak_akses ON [nama_db]nama_tabel
TO nama_pemakai
[IDENTIFIED BY] [PASSWORD] 'Password'
[WITH GRANT OPTION];

Contoh :

Berikan hak akses kepada Adi untuk menampilkan nilai final test pada tabel Nilai.

GRANT SELECT (FINAL) ON NILAI TO ADI

Data Access lanjutan

2. **REVOKE** → Menarik hak akses pengguna yang diberikan lewat perintah GRANT

Sintaks : REVOKE hak_akses ON nama_db
FROM nama_pemakai ;

REVOKE hak_akses ON nama_tabel
FROM nama_pemakai ;

Contoh :

Tarik kembali dari Adi hak akses untuk menampilkan nilai final test

REVOKE SELECT (FINAL) ON NILAI FROM ADI

Data Integrity

RECOVER TABLE

Sintaks : `RECOVER TABLE nama_tabel`

Contoh :

Kembalikan keadaan data mahasiswa seperti pada saat sebelum terjadi kerusakan

`RECOVER TABLE MHS ;`

Auxiliary

1. **SELECT ... INTO OUTFILE** 'filename'

Sintaks ini digunakan untuk mengekspor data dari tabel ke file lain.

Sintaks : **SELECT ... INTO**
 OUTFILE 'Nama File'
 [FIELDS | COLUMNS]
 [TERMINATED BY 'string']
 [[OPTIONALLY] ENCLOSED BY 'char']
 [ESCAPED BY 'char']]

Contoh :

Ubah semua data mahasiswa ke bentuk ASCII dan disimpan ke file teks di directory/home/adi dengan pemisah antar kolom '|'

```
SELECT * FROM MHS
INTO OUTFILE "/home/adi/teks"
FIELDS TERMINATED BY "|";
```


Auxiliary lanjutan

2. LOAD

Sintaks query ini digunakan untuk mengimpor data dari file lain ke tabel.

Sintaks : LOAD DATA INFILE “ nama_path”
 INTO TABLE nama_tabel [nama_kolom] ;
 [FIELDS | COLUMNS]
 [TERMINATED BY '*string*']
 [[OPTIONALLY] ENCLOSED BY '*char*']
 [ESCAPED BY '*char*']]

Contoh :

Memasukkan data-data dari file teks yang berada pada direktori “/home/adi” ke dalam tabel MHS_2. Dimana pemisah antara kolom dalam file teks adalah tab (\t) :

```
LOAD FROM “/home/adi/teks”  
INTO MHS_2  
FIELDS TERMINATED BY ‘\t’;
```

Auxiliary lanjutan

3. RENAME TABLE

Sintaks :

```
RENAME TABLE OldnamaTabel  
TO NewNamaTabel
```

Contoh :

```
RENAME TABLE MHS  
TO MAHASISWA
```

Fungsi Aggregate

MENGGUNAKAN FUNGSI AGGREGATE :

1. COUNT digunakan untuk menghitung jumlah.
Menghitung jumlah record mahasiswa dari tabel MAHASISWA
`SELECT COUNT(*) FROM MAHASISWA`
2. SUM digunakan untuk menghitung total dari kolom yang mempunyai tipe data numerik.
`SELECT SUM(SKS) AS 'TOTAL SKS' FROM MATAKULIAH`

Fungsi Aggregate lanjutan

3. AVG digunakan untuk menghitung rata-rata dari data-data dalam sebuah kolom.

```
SELECT AVG(FINAL) AS 'FINAL' FROM Nilai
```

4. MIN digunakan untuk menghitung nilai minimal dalam sebuah kolom.

```
SELECT MIN(FINAL) FROM Nilai
```

5. MAX digunakan untuk menghitung nilai maksimum dalam sebuah kolom

```
SELECT MAX(MID) FROM Nilai
```

SUBQUERY

SUBQUERY

Adalah subselect yang dapat digunakan di klausa WHERE dan HAVING dipernyataan select luar untuk menghasilkan tabel akhir.

Aturan-aturan untuk membuat subquery, yaitu :

1. Klausa Order By tidak boleh digunakan di subquery, Order By hanya dapat digunakan di pernyataan Select luar.
2. Klausa subquery Select harus berisi satu nama kolom tunggal atau ekspresi kecuali untuk subquery-subquery menggunakan kata kunci EXIST
3. Secara default nama kolom di subquery mengacu ke nama tabel di klausa FROM dari subquery tersebut.
4. Saat subquery adalah salah satu dua operan dilibatkan di perbandingan, subquery harus muncul disisi kanan perbandingan

Subquery lanjutan

Penggunaanan ANY dan ALL

Jika subquery diawali kata kunci ALL, syarat hanya akan bernilai TRUE jika dipenuhi semua nilai yang dihasilkan subquery itu.

Jika subquery diawali kata kunci ANY, syaratnya akan bernilai TRUE jika dipenuhi sedikitnya satu nilai yang dihasilkan subquery tersebut.

Subquery lanjutan

Penggunaanan **EXIST** DAN **NOT EXIST**

EXIST akan mengirim nilai TRUE jika dan hanya jika terdapat sedikitnya satu baris di tabel hasil yang dikirim oleh subquery dan EXIST mengirim nilai FALSE jika subquery mengirim tabel kosong.

Untuk NOT EXIST kebalikan dari EXIST.

(Masing-masing dosen membuat contoh untuk subquery)

Subquery lanjutan

CONTOH SUBQUERY :

1. Ambil nilai mid dan final dari mahasiswa yang bernama Astuti.

```
SELECT MID, FINAL FROM NILAI WHERE NIM=(  
SELECT NIM FROM MAHASISWA WHERE  
NAMA='Astuti')
```

2. Ambil nilai kode matakuliah, mid dan final dari mahasiswa yang tinggal di jakarta.

```
SELECT KD_MK, MID, FINAL FROM NILAI WHERE NIM  
IN(SELECT NIM FROM MAHASISWA WHERE ALAMAT =  
'Jakarta')
```


Subquery lanjutan

3. Ambil nama-nama mahasiswa yang mengikuti ujian.

```
SELECT NAMA FROM MAHASISWA WHERE EXISTS  
(SELECT NIM FROM NILAI WHERE NILAI.NIM=  
MAHASISWA.NIM)
```

4. Ambil nama-nama mahasiswa yang tidak mengikuti ujian.

```
SELECT NAMA FROM MAHASISWA WHERE NOT  
EXISTS (SELECT NIM FROM NILAI WHERE NILAI.NIM=  
MAHASISWA.NIM)
```