

MODUL 8 : PENGGUNAAN JOIN**Tujuan :**

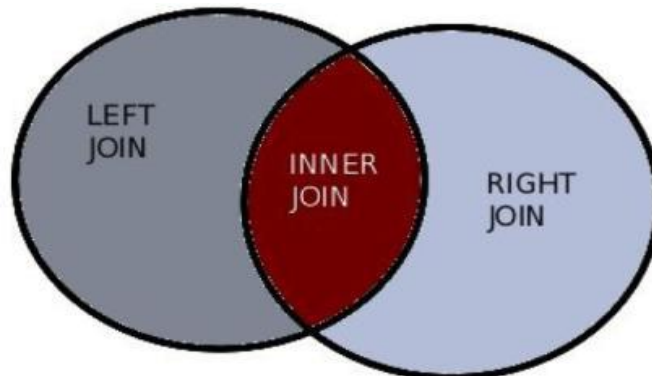
Setelah menyelesaikan modul ini, mahasiswa diharapkan dapat:

1. Mampu menggunakan perintah *join*, *inner join*, *left join*, *right join* dan *natural join*
2. Mampu melakukan operasi join pada dua tabel atau lebih

PENGUNAAN JOIN

Operasi *join* merupakan operasi yang digunakan untuk menggabungkan dua tabel atau lebih dengan hasil berupa gabungan dari kolom-kolom yang berasal dari tabel-tabel tersebut. *Join* yang sederhana, tabel-tabel digabungkan dan didasarkan pada pencocokan antara kolom pada tabel yang berbeda.

Join memperbolehkan pengguna untuk mengambil data dari beberapa tabel melalui satu *query*. Hanya menggunakan sebuah tabel artinya pengguna hanya dapat menyimpan atau memperoleh data yang terbatas atau justru menyimpan atau memperoleh data yang terlalu banyak sehingga tabelnya menjadi kurang baik. *Join* menghubungkan satu tabel dengan tabel yang lain (inilah yang dimaksud dengan *relational* dari istilah *relational database*).



INNER JOIN

Inner join adalah jenis *join* yang paling umum yang dapat digunakan pada semua *database*. Jenis ini dapat digunakan bila ingin merelasikan dua set data yang ada di tabel, letak relasinya setelah pada perintah **ON** pada *join*. Hasil dari *inner join* adalah gabungan kedua tabel yang memiliki data *join* yang sama.

Query yang digunakan :

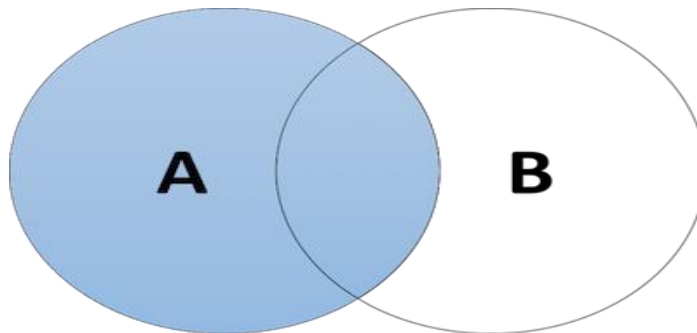
```
SELECT <field1>, <field2>, <fieldn>
FROM <tabel1>
INNER JOIN <tabel2> ON <key.tabel1> = <key.tabel2>
```

Atau

```
SELECT columns
FROM TableA
INNER JOIN TableB
ON A.columnName = B.columnName;
```

OUTER JOIN

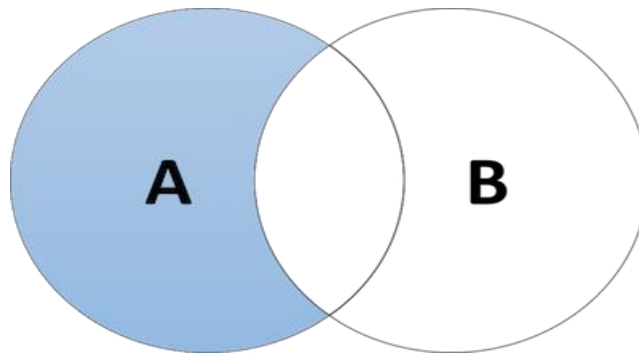
Left outer join (sering disingkat *left join*) akan mengembalikan seluruh baris dari tabel disebelah kiri yang dikenai kondisi **ON** dan hanya baris dari tabel disebelah kanan yang memenuhi kondisi *join*.



```
SELECT columns
FROM TableA
LEFT OUTER JOIN TableB
ON A.columnName = B.columnName
```

Left Outer Join without Intersection

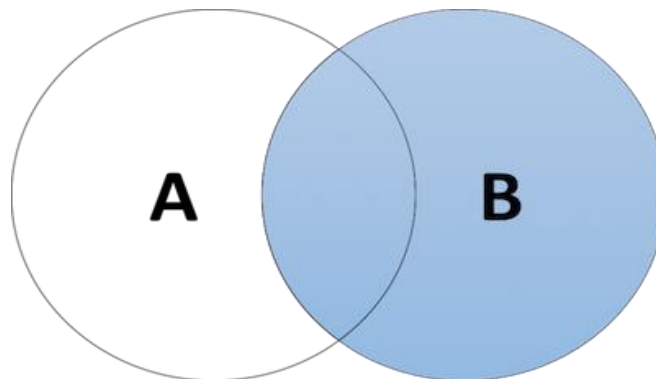
Join ini merupakan variasi dari *left outer join*. Pada *join* ini pengguna hanya akan mengambil data dari tabel sebelah kiri yang dikenai kondisi **ON** yang juga memenuhi kondisi *join* tanpa data dari tabel sebelah kanan yang memenuhi kondisi *join*.



```
SELECT columns  
FROM TableA  
LEFT OUTER JOIN TableB  
ON A.columnName = B.columnName  
WHERE B.columnName IS NULL
```

Right Outer Join

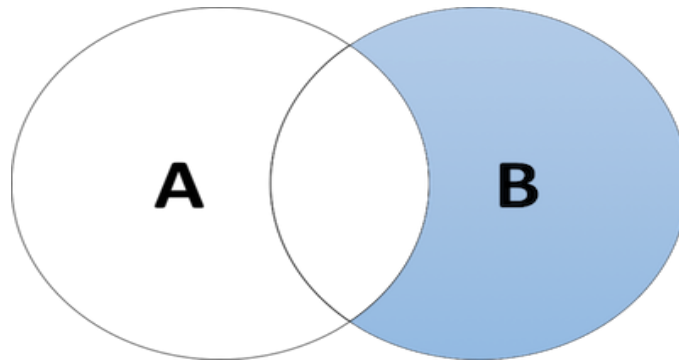
Right outer join (sering disingkat **right join**) akan mengembalikan semua baris dari tabel sebelah kanan yang dikenai kondisi **ON** dengan data dari tabel sebelah kiri yang memenuhi kondisi *join*. Teknik ini merupakan kebalikan dari *left outer join*.



```
SELECT columns  
FROM TableA  
RIGHT OUTER JOIN TableB  
ON A.columnName = B.columnName
```

Right Outer Join without Intersection

Teknik ini merupakan variasi dari *right outer join*. Pada **join** ini pengguna hanya akan mengambil data dari tabel sebelah kanan yang dikenai kondisi **ON** yang juga memenuhi kondisi *join* tanpa data dari tabel sebelah kanan yang memenuhi kondisi *join*.



```
SELECT columns
FROM TableA
RIGHT OUTER JOIN TableB
ON A.columnName = B.columnName
WHERE A.columnName IS NULL
```

NATURAL JOIN

Natural join digunakan untuk menampilkan semua data dari dua tabel dimana jika terdapat kolom yang sama, maka yang akan ditampilkan hanya salah satunya saja, yaitu kolom dari tabel sebelah kiri perintah *natural join*.

```
SELECT * FROM Tabel1 NATURAL JOIN Tabel2;
```

Terdapat penggabungan *Natural Join* dengan *Left* dan *Right Join* :

1. Natural Left Join

Natural left join digunakan untuk menampilkan semua data dari tabel sebelah kiri perintah *natural left join* beserta pasangannya dari tabel sebelah kanan. Meskipun terdapat data dari sebelah kiri tidak memiliki pasangan, tetap akan ditampilkan dengan pasangannya berupa nilai NULL.

```
SELECT * FROM Tabel1 NATURAL LEFT JOIN Tabel2;
```

2. *Natural Right Join*

Natural right join digunakan untuk menampilkan semua data dari tabel sebelah kanan perintah *natural right join* beserta pasangannya dari tabel sebelah kiri. Meskipun terdapat data dari sebelah kanan tidak memiliki pasangan, tetap akan ditampilkan dengan pasangannya berupa nilai NULL.

```
SELECT * FROM Tabel1 NATURAL RIGHT JOIN Tabel2;
```

LANGKAH PRAKTIKUM

1. Buat sebuah tabel dengan nama tabel pegawai dan isi data tabel seperti berikut :

IdPegawai	NamaPegawai
P01	Ardila
P02	Budianto
P03	Cindy Putri
P04	Erwin
P05	Deriansyah
P06	Saputra

2. Buat tabel gaji dan isi tabel gaji dengan data seperti berikut:

IdGaji	IdPegawai	GajiPokok
G-01	P01	2500000
G-02	P02	3000000
G-03	P04	2000000
G-04	P03	2800000
G-05	P05	3200000

3. Gunakan perintah *inner join* pada tabel pegawai dan gaji :

```
mysql> select * from Pegawai
-> INNER JOIN Gaji
-> ON Pegawai.IdPegawai=Gaji.IdPegawai;
```

IdPegawai	NamaPegawai	IdGaji	IdPegawai	GajiPokok
P01	Ardila	G-01	P01	2500000
P02	Budianto	G-02	P02	3000000
P04	Erwin	G-03	P04	2000000
P03	Cindy Putri	G-04	P03	2800000
P05	Deriansyah	G-05	P05	3200000

5 rows in set (0.00 sec)

```
mysql> select * from Gaji
-> INNER JOIN Pegawai
-> ON Pegawai.IdPegawai=Gaji.IdPegawai;
```

IdGaji	IdPegawai	GajiPokok	IdPegawai	NamaPegawai
G-01	P01	2500000	P01	Ardila
G-02	P02	3000000	P02	Budianto
G-03	P04	2000000	P04	Erwin
G-04	P03	2800000	P03	Cindy Putri
G-05	P05	3200000	P05	Deriansyah

5 rows in set (0.00 sec)

4. Gunakan perintah *left join* pada tabel pegawai dan gaji.

```
mysql> select * from Pegawai
-> LEFT JOIN Gaji
-> ON Pegawai.IdPegawai=Gaji.IdPegawai;
```

IdPegawai	NamaPegawai	IdGaji	IdPegawai	GajiPokok
P01	Ardila	G-01	P01	2500000
P02	Budianto	G-02	P02	3000000
P03	Cindy Putri	G-04	P03	2800000
P04	Erwin	G-03	P04	2000000
P05	Deriansyah	G-05	P05	3200000
P06	Saputra	NULL	NULL	NULL

6 rows in set (0.00 sec)

```
mysql> select * from Gaji
-> LEFT JOIN Pegawai
-> ON Pegawai.IdPegawai=Gaji.IdPegawai;
```

IdGaji	IdPegawai	GajiPokok	IdPegawai	NamaPegawai
G-01	P01	2500000	P01	Ardila
G-02	P02	3000000	P02	Budianto
G-03	P04	2000000	P04	Erwin
G-04	P03	2800000	P03	Cindy Putri
G-05	P05	3200000	P05	Deriansyah

5 rows in set (0.00 sec)

5. Gunakan perintah *right join* pada tabel pegawai dan gaji.

```
mysql> select * from Pegawai
-> RIGHT JOIN Gaji
-> ON Pegawai.IdPegawai=Gaji.IdPegawai;
```

IdPegawai	NamaPegawai	IdGaji	IdPegawai	GajiPokok
P01	Ardila	G-01	P01	2500000
P02	Budianto	G-02	P02	3000000
P04	Erwin	G-03	P04	2000000
P03	Cindy Putri	G-04	P03	2800000
P05	Deriansyah	G-05	P05	3200000

5 rows in set (0.00 sec)

```
mysql> select * from Gaji
-> RIGHT JOIN Pegawai
-> ON Pegawai.IdPegawai=Gaji.IdPegawai;
```

IdGaji	IdPegawai	GajiPokok	IdPegawai	NamaPegawai
G-01	P01	2500000	P01	Ardila
G-02	P02	3000000	P02	Budianto
G-04	P03	2800000	P03	Cindy Putri
G-03	P04	2000000	P04	Erwin
G-05	P05	3200000	P05	Deriansyah
NULL	NULL	NULL	P06	Saputra

6 rows in set (0.00 sec)

6. Penggunaan *Left Outer Join without Intersection*.

```
mysql> select * from Pegawai
-> LEFT JOIN Gaji
-> ON Pegawai.IdPegawai=Gaji.IdPegawai
-> WHERE Gaji.IdPegawai is NULL;
```

IdPegawai	NamaPegawai	IdGaji	IdPegawai	GajiPokok
P06	Saputra	NULL	NULL	NULL

1 row in set (0.00 sec)

7. Penggunaan *Right Outer Join without Intersection*.

```
mysql> select * from Gaji
-> RIGHT JOIN Pegawai
-> ON Pegawai.IdPegawai=Gaji.IdPegawai
-> WHERE Gaji.IdPegawai is NULL;
```

IdGaji	IdPegawai	GajiPokok	IdPegawai	NamaPegawai
NULL	NULL	NULL	P06	Saputra

1 row in set (0.00 sec)

8. Penggunaan *Natural Join*.

```
mysql> select * from Pegawai NATURAL JOIN Gaji;
```

IdPegawai	NamaPegawai	IdGaji	GajiPokok
P01	Ardila	G-01	2500000
P02	Budianto	G-02	3000000
P04	Erwin	G-03	2000000
P03	Cindy Putri	G-04	2800000
P05	Deriansyah	G-05	3200000

5 rows in set (0.00 sec)

```
mysql> select * from Gaji NATURAL JOIN Pegawai;
```

IdPegawai	IdGaji	GajiPokok	NamaPegawai
P01	G-01	2500000	Ardila
P02	G-02	3000000	Budianto
P04	G-03	2000000	Erwin
P03	G-04	2800000	Cindy Putri
P05	G-05	3200000	Deriansyah

5 rows in set (0.00 sec)

9. Penggunaan *Natural Left Join*.

```
mysql> select * from Pegawai NATURAL LEFT JOIN Gaji;
```

IdPegawai	NamaPegawai	IdGaji	GajiPokok
P01	Ardila	G-01	2500000
P02	Budianto	G-02	3000000
P03	Cindy Putri	G-04	2800000
P04	Erwin	G-03	2000000
P05	Deriansyah	G-05	3200000
P06	Saputra	NULL	NULL

6 rows in set (0.00 sec)

```
mysql> select * from Gaji NATURAL LEFT JOIN Pegawai;
```

IdPegawai	IdGaji	GajiPokok	NamaPegawai
P01	G-01	2500000	Ardila
P02	G-02	3000000	Budianto
P04	G-03	2000000	Erwin
P03	G-04	2800000	Cindy Putri
P05	G-05	3200000	Deriansyah

5 rows in set (0.01 sec)

10. Penggunaan *Natural Right Join*.

```
mysql> select * from Pegawai NATURAL RIGHT JOIN Gaji;
```

IdPegawai	IdGaji	GajiPokok	NamaPegawai
P01	G-01	2500000	Ardila
P02	G-02	3000000	Budianto
P04	G-03	2000000	Erwin
P03	G-04	2800000	Cindy Putri
P05	G-05	3200000	Deriansyah

5 rows in set (0.00 sec)

```
mysql> select * from Gaji NATURAL RIGHT JOIN Pegawai;
```

IdPegawai	NamaPegawai	IdGaji	GajiPokok
P01	Ardila	G-01	2500000
P02	Budianto	G-02	3000000
P03	Cindy Putri	G-04	2800000
P04	Erwin	G-03	2000000
P05	Deriansyah	G-05	3200000
P06	Saputra	NULL	NULL

6 rows in set (0.00 sec)