# FIRDAUS SOLIHIN UNIVERSITAS TRUNOJOYO

# Macam JOIN

- Cartesian Product
- Equi Join
- Self Join
- INNER JOIN
- LEFT JOIN atau LEFT OUTER JOIN
- RIGHT JOIN atau RIGHT OUTER JOIN
- FULL JOIN
- UNION

# Cartesian Product

- Operasi Join yang tidak melibatkan kondisi tertentu.
- SELECT mahasiswa.Nim, mahasiswa.Nama, mahasiswa.Alamat, matakuliah.No-Mk, matakuliah.Nama FROM mahasiswa, matakuliah;
- SELECT mhs.Nim, mhs.Nama, mhs.Alamat, mk.No-Mk, mk.Nama FROM mahasiswa mhs, matakuliah mk;

# Contoh Cartesian Product

Mahasiswa			
Nim	Nama	Alamat	
2005001		Bangkalan	
2005002	Edi	Surabaya	
2005003	Reni	Sumenep	

MataKuliah			
No-MK	Nama		
MKU-001	Matematika		
MKK-003	Pemrograman		
MKK-005	Basis Data		

Nim	Nama	Alamat	No-MK	Nama
2005001	Amin	Bangkalan	MKU-001	Matematika
2005001	Amin	Bangkalan	MKK-003	Pemrograman
2005001	Amin	Bangkalan	MKK-005	Basis Data
2005002	Edi	Surabaya	MKU-001	Matematika
2005002	Edi	Surabaya	MKK-003	Pemrograman
2005002	Edi	Surabaya	MKK-005	Basis Data
2005003	Reni	Sumenep	MKU-001	Matematika
2005003	Reni	Sumenep	MKK-003	Pemrograman
2005003	Reni	Sumenen	MKK-005	Basis Data

# Equi-Join

- operasi penggabungan pada table1 dan table2 dimana untuk operasi JOIN dilakukan dengan cara menyamakan nilai dari satu column pada table1 dengan satu column pada table2.
- SELECT mhs.Nim, mhs.Nama, mhs.Kd-Jur, jur.Nama
   FROM mahasiswa mhs, jurusan jur
   WHERE mhs.Kd-jur = jur.Kd-jur

# Contoh Equi-Join

#### Mahasiswa

Nim	Nama	Kd-Jur
2005001	Amin	J01
2005002		J02
2005003	Reni	J02

#### Jurusan

Kd-Jur	Nama
J01	Industri
J02	Informatika
J03	D3 MI

Nim	Nama	Kd-Jur	Nama
2005001	Amin	J01	Industri
2005002	Edi	J02	Informatika
2005003	Reni	J02	Informatika

## Self-Join

- Menggabungkan sebuah table dengan table itu sendiri. dgn asumsi table yang di gabungkan terdiri atas 2 table yang berbeda (menggunakan alias).
- SELECT p1.no, p1.nama, p1.alamat, p2.nama
   FROM pegawai p1, pegawai p2
   WHERE p1.no = p2.manager\_id;

### Contoh Tabel Self-Join

# Pegawai No Nama Alamat Manager\_id 1 Amin Bangkalan 0 2 Edi Surabaya 1 3 Reni Sumenep 1 4 Dani Sidoarjo 0 5 Yuli Mojokerto 4 6 Wawan Jombang 4 7 Arman Surabaya 4

No	Nama	Alamat	Manager
5	Yuli	Mojokerto	Dani
6	Wawan	Jombang	Dani
7	Arman	Surabaya	Dani

# Membuat Tabel contoh

#### Persons

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
2	Svendson	Tove	Borgvn 23	Sandnes
3	Pettersen	Kari	Storgt 20	Stavanger

#### Orders

O_Id	OrderNo	P_Id
1	77895	3
2	44678	3
3	22456	1
4	24562	1
5	34764	15

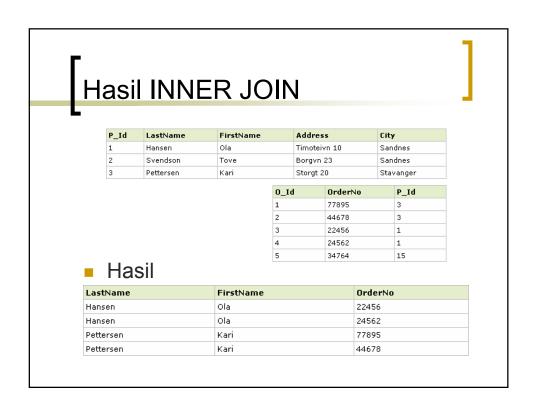
# INNER JOIN

- Syntax
- SELECT column\_name(s)FROM table\_name1INNER JOIN table\_name2ON

table\_name1.column\_name=table\_name2.column\_name

# Contoh INNER JOIN

SELECT Persons.LastName,
 Persons.FirstName, Orders.OrderNo
 FROM Persons
 INNER JOIN Orders
 ON Persons.P\_Id=Orders.P\_Id
 ORDER BY Persons.LastName



# Simpulan INNER JOIN

- Menampilkan baris hasil jika ada setidaknya satu kesamaan pada dua nilai field yang dijoin.
- Jika ada baris record pada tabel pertama yang nilai fieldnya tidak terdapat pada tabel kedua maka baris record tersebut tidak akan ditempilkan

# **LEFT JOIN**

- Syntax
- SELECT column\_name(s)FROM table\_name1LEFT JOIN table\_name2ON

table\_name1.column\_name=table\_name2.column\_name

# Contoh LEFT JOIN

SELECT Persons.LastName,
 Persons.FirstName, Orders.OrderNo
 FROM Persons
 LEFT JOIN Orders
 ON Persons.P\_Id=Orders.P\_Id
 ORDER BY Persons.LastName

#### Hasil LEFT JOIN P\_Id LastName FirstName Address City Hansen Ola Timoteivn 10 Sandnes Borgvn 23 Svendson Tove Sandnes Pettersen Storgt 20 Stavanger O\_Id OrderNo P Id 77895 44678 3 3 22456 24562 34764 15 Hasil LastName OrderNo FirstName Hansen Ola 24562 Hansen Pettersen Kari 77895 44678 Pettersen Kari Svendson

# Simpulan LEFT JOIN

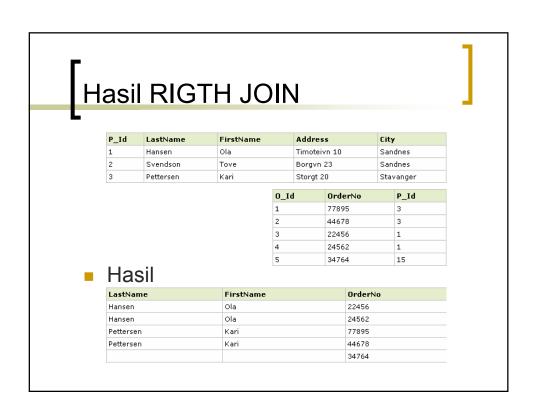
 Menampilkan baris hasil yang didapat dari semua baris dari tabel pertama walaupun ada ketidaksamaan nilai field yang di join.

# **RIGTH JOIN**

- Syntax
- SELECT column\_name(s)
   FROM table\_name1
   RIGHT JOIN table\_name2
   ON
   table\_name1.column\_name=table\_name2.column\_name

# Contoh RIGTH JOIN

 SELECT Persons.LastName, Persons.FirstName, Orders.OrderNo FROM Persons RIGHT JOIN Orders ON Persons.P\_Id=Orders.P\_Id ORDER BY Persons.LastName



# Simpulan RIGTH JOIN

Menampilkan baris hasil yang didapat dari semua baris dari tabel kedua walaupun ada ketidaksamaan nilai field yang di join.

# **FULL JOIN**

- Syntax
- SELECT column\_name(s)
   FROM table\_name1
   FULL JOIN table\_name2
   ON
   table\_name1.column\_name=table\_name2.column\_name
- FULL JOIN Tidak Berlaku di MySQL

# Contoh FULL JOIN

 SELECT Persons.LastName, Persons.FirstName, Orders.OrderNo FROM Persons FULL JOIN Orders ON Persons.P\_Id=Orders.P\_Id ORDER BY Persons.LastName



# Simpulan FULL JOIN

 Menampilkan baris hasil yang didapat dari semua baris dari tabel pertama dan kedua walaupun ada ketidaksamaan nilai field yang di join.

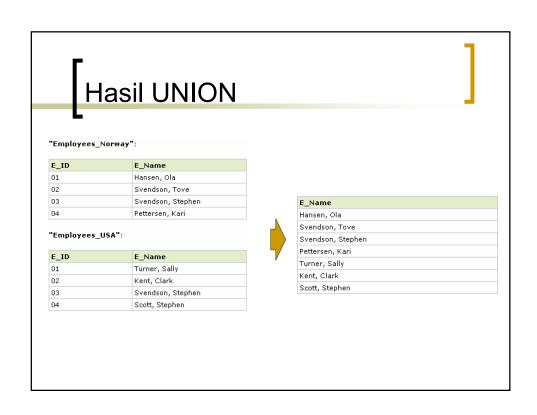
# UNION

- Syntax
- SELECT column\_name(s) FROM table\_name1 UNION

SELECT column\_name(s) FROM table\_name2

# Contoh UNION

 SELECT E\_Name FROM Employees\_Norway UNION SELECT E\_Name FROM Employees\_USA



# UNION ALL

- SELECT column\_name(s) FROM table\_name1 UNION ALL SELECT column\_name(s) FROM table\_name2
- UNION ALL tidak dikenal di MySQL

# Contoh UNION ALL

 SELECT E\_Name FROM Employees\_Norway UNION ALL SELECT E\_Name FROM Employees\_USA

