

**TUGAS MODUL 4**  
**PRAKTIKUM BASIS DATA**  
**JOIN PART I**

Oleh:

Nama : Fitra Ilyasa

Nim : 120140048

Kelas : Basis Data RB



**PROGRAM STUDI TEKNIK INFORMATIKA**  
**JURUSAN TEKNIK ELEKTRO INFORMATIKA DAN SISTEM FISIS**  
**INSTITUT TEKNOLOGI SUMATERA**  
**LAMPUNG SELATAN**

**2022**

# BAB I

## TEORI DASAR

---

### Relationship

*Relationship* adalah suatu hubungan antara beberapa entitas. Konsep ini sangat penting sekali di dalam basis data, di mana memungkinkan entitas-entitas untuk saling berhubungan satu sama lain. Didalam sebuah *relationship*, *primary key* memiliki peran penting untuk mengaitkan entitas. Selain itu, *primary key* juga digunakan untuk mendefinisikan batasan keterhubungan.

### Join

*Join* merupakan salah satu konstruksi dasar dari SQL dan basis data. *Join* dapat didefinisikan sebagai kombinasi record dari dua atau lebih tabel di dalam basis data relasional dan menghasilkan sebuah tabel (*temporary*) baru yang disebut sebagai *joined tabel*. *Join* digunakan untuk menghubungkan data yang diambil dari tabel-tabel melalui sebuah kolom yang menghubungkan mereka.

Ada beberapa Jenis JOIN yang dapat digunakan seperti CROSS JOIN, INNER JOIN, STRAIGHT JOIN, OUTER JOIN (RIGHT dan LEFT JOIN), NATURAL JOIN.

#### a) Cross Join

*Cross join* merupakan operasi *join* yang paling dasar. Menghasilkan kombinasi semua baris yang terdapat pada tabel-tabel yang digabungkan, baik yang memiliki pasangan maupun yang tidak. *Join* jenis ini juga disebut dengan istilah *cartesian product*.

Bila menggunakan *cross join*, maka hasil dari *cross join* akan menciptakan hasil yang didasarkan pada semua kemungkinan kombinasi baris dalam kedua set data. Dengan demikian, jumlah baris yang dikembalikan adalah  $N \times M$ , dimana N adalah jumlah baris dalam kumpulan data A dan M jumlah baris dalam kumpulan data B. Jelas, jumlah baris dalam *cross join* dapat menjadi sampah.

Atau :

```
SELECT <nama_kolom> FROM <tabel_1>,<tabel_2>;
```

```
SELECT <nama_kolom> FROM <tabel_1> CROSS JOIN <tabel_2>
```

## b) Inner Join

*Inner join* Mengembalikan baris-baris dari dua tabel atau lebih yang memenuhi syarat. Umumnya penggunaanya didasarkan pada kunci tamu. *Inner join* pada dasarnya adalah menemukan persimpangan (*intersection*) antara dua buah tabel.

Jika menggunakan *inner join* antara tabel A dan B maka hasilnya hanya akan menampilkan data yang memiliki pasangan saja atau yang memiliki kesamaan saja.

```
SELECT <nama_kolom> FROM <tabel_1> INNER JOIN <tabel_2> ON 1.NamaKolom = 2.NamaKolom;
```

atau

```
SELECT <nama_kolom> FROM <tabel_1> INNER JOIN <tabel_2>  
WHERE tabel_1.kolom_kunci = tabel_2.kolom_kunci;
```

## c) Natural Join

Operasi INNER JOIN yang dilakukan pada semua atribut yang sama dari kedua tabel (nama dan domainnya). Pada tabel hasil, atribut-atribut yang sama hanya akan muncul sekali. Dalam menggabungkan tabel, operasi NATURAL JOIN tidak memerlukan penyebutan atribut penghubung. Penggabungan tabel dilakukan otomatis pada atribut yang sama (nama dan domainnya).

## BAB II

### PEMBAHASAN & ANALISIS

---

1. Tambahkan data berikut pada tabel Produk

**Command Line :** **CREATE TABLE Produk (Produk\_id varchar(10), Produk\_nama varchar(20), Jumlah\_Stok varchar(10), Supplier\_id varchar(10), PRIMARY KEY(Produk\_id)) ENGINE = InnoDB;**

**Command Line :** **INSERT INTO Produk value ('P552', 'Aqua 1 L', '300 pcs', 'S001'), ('P882', 'Indomilk 25 ml', '200 pcs', 'S004'), ('P333', 'Sari Roti 100 gram', '30 pcs', 'S005'), ('P123', 'Gulaku 1 Kg', '100 pcs', 'S005'), ('P453', 'Garam 30 gram', '20 pcs', 'S006'), ('P109', 'The Kotak 300 ml', '40 pcs', 'S002');**

**Command Line :** **SELECT \* FROM Produk;**

```
MariaDB [galeri_itera]> INSERT INTO Produk value ('P552', 'Aqua 1 L', '300 pcs', 'S001'), ('P882', 'Indomilk 25 ml', '200 pcs', 'S004'), ('P333', 'Sari Roti 100 gram', '30 pcs', 'S005'), ('P123', 'Gulaku 1 Kg', '100 pcs', 'S005'), ('P453', 'Garam 30 gram', '20 pcs', 'S006'), ('P109', 'The kotak 300 ml', '40 pcs', 'S002');
Query OK, 6 rows affected (0.069 sec)
Records: 6 Duplicates: 0 Warnings: 0
```

```
MariaDB [galeri_itera]> SELECT * FROM Produk;
+-----+-----+-----+-----+
| Produk_id | Produk_nama | Jumlah_Stok | Supplier_id |
+-----+-----+-----+-----+
| P109      | The Kotak 300 ml | 40 pcs      | S002        |
| P114      | Milo 100 ml      | 800 pcs     | S001        |
| P115      | Milo 150 ml      | 50 pcs      | S003        |
| P123      | Gulaku 1 Kg      | 100 pcs     | S005        |
| P235      | Aqua 250 ml      | 300 pcs     | S001        |
| P311      | Grand 320 ml     | 400 pcs     | S003        |
| P333      | Sari Roti 100 gram | 30 pcs      | S005        |
| P441      | Rojo Lele 5 kg   | 60 pcs      | S002        |
| P453      | Garam 30 gram    | 20 pcs      | S006        |
| P552      | Aqua 1 L         | 300 pcs     | S001        |
| P882      | Indomilk 25 ml   | 200 pcs     | S004        |
+-----+-----+-----+-----+
11 rows in set (0.001 sec)
```

2. Tambahkan data berikut pada tabel Supplier

**Command Line :** **CREATE TABLE Supplier (Supplier\_id varchar(10), Company\_nama varchar(20), Nama\_Kontak varchar(20), PRIMARY KEY(Supplier\_id)) ENGINE = InnoDB;**

**Command Line :** **INSERT INTO Supplier value ('S004', 'Pelita Baru', 'Puspa'), ('S005', 'Surya Kun', 'Siti'), ('S006', 'Ceria Kasih', 'Topan');**

*Command Line* : **SELECT \* FROM Supplier;**

```
XAMPP for Windows - mysql -u root -p
MariaDB [galeri_itera]> CREATE TABLE Supplier (Supplier_id varchar(10), Company_nama varchar(20), Nama_Kontak varchar(20), PRIMARY KEY (Supplier_id)) ENGINE = InnoDB;
Query OK, 0 rows affected (0.294 sec)

MariaDB [galeri_itera]> INSERT INTO Supplier value ('S004', 'Pelita Baru', 'Puspa'), ('S005', 'Surya Kun', 'Siti'), ('S006', 'Ceria Ka
sih', 'Topan');
Query OK, 3 rows affected (0.072 sec)
Records: 3 Duplicates: 0 Warnings: 0
```

```
MariaDB [galeri_itera]> SELECT * FROM Supplier;
+-----+-----+-----+
| Supplier_id | Company_nama | Nama_Kontak |
+-----+-----+-----+
| S001        | Semua Terang | Sugiono     |
| S002        | Suka Maju    | Rahmat      |
| S003        | Maju Terus   | Dayono      |
| S004        | Pelita Baru  | Puspa       |
| S005        | Surya Kun    | Siti        |
| S006        | Ceria Kasih  | Topan       |
+-----+-----+-----+
6 rows in set (0.001 sec)
```

3. Tambahkan data pada tabel Pegawai

*Command Line* : **CREATE TABLE Pegawai (Id\_pegawai varchar(10), Pegawai\_nama varchar(20), Jabatan varchar(10), PRIMARY KEY(Id\_pegawai)) ENGINE = InnoDB;**

*Command Line* : **INSERT INTO Pegawai value ('Pg\_002', 'Siska', 'Casier'), ('Pg\_003', 'Nuri', 'Casier'), ('Pg\_004', 'Jamal', 'Casier');**

*Command Line* : **SELECT \* FROM Pegawai;**

```
XAMPP for Windows - mysql -u root -p
MariaDB [galeri_itera]> CREATE TABLE Pegawai (Id_pegawai varchar(10), Pegawai_nama varchar(20), Jabatan varchar(10), PRIMARY KEY(Id_peg
awai)) ENGINE = InnoDB;
Query OK, 0 rows affected (0.290 sec)

MariaDB [galeri_itera]> INSERT INTO Pegawai value ('Pg_002', 'Siska', 'Casier'), ('Pg_003', 'Nuri', 'Casier'), ('Pg_004', 'Jamal', 'Ca
sier');
Query OK, 3 rows affected (0.071 sec)
Records: 3 Duplicates: 0 Warnings: 0
```

```
MariaDB [galeri_itera]> SELECT * FROM Pegawai;
+-----+-----+-----+
| Id_pegawai | Pegawai_nama | Jabatan |
+-----+-----+-----+
| pg_001     | Santi        | Cashier |
| Pg_002     | Siska        | Casier  |
| Pg_003     | Nuri         | Casier  |
| Pg_004     | Jamal        | Casier  |
| pg_011     | Sinta        | Manager |
| pg_201     | Santo        | cashier |
+-----+-----+-----+
6 rows in set (0.001 sec)
```

4. Tabel Pembeli tetap seperti minggu sebelumnya

**Command Line : CREATE TABLE Pembeli (Id\_pembeli varchar(10), pembeli\_nama varchar(10), pembeli\_kontak varchar(10), PRIMARY KEY(Id\_pembeli)) ENGINE = InnoDB;**

**Command Line : INSERT INTO Pembeli value ('C\_901', 'Rudi', '081231511'), ('C\_991', 'Andi', '085212021111'), ('C\_800', 'Egi', '0812521221'), ('C\_810', 'Ardi', '0862145121');**

**Command Line : SELECT \* FROM Pembeli;**

```
XAMPP for Windows - mysql -u root -p
MariaDB [galeri_itera]> CREATE TABLE Pembeli (Id_pembeli varchar(10), pembeli_nama varchar(10), pembeli_kontak varchar(10), PRIMARY KEY(Id_pembeli)) ENGINE = InnoDB;
Query OK, 0 rows affected (0.302 sec)

MariaDB [galeri_itera]> INSERT INTO Pembeli value ('C_901', 'Rudi', '081231511'), ('C_991', 'Andi', '085212021111'), ('C_800', 'Egi', '0812521221'), ('C_810', 'Ardi', '0862145121');
Query OK, 4 rows affected, 1 warning (0.052 sec)
Records: 4 Duplicates: 0 Warnings: 1

MariaDB [galeri_itera]> SELECT * FROM Pembeli;
+-----+-----+-----+
| Id_pembeli | pembeli_nama | pembeli_kontak |
+-----+-----+-----+
| C_800      | Egi          | 0812521221     |
| C_810      | Ardi         | 0862145121     |
| C_901      | Rudi         | 081231511      |
| C_991      | Andi         | 0852120211     |
+-----+-----+-----+
4 rows in set (0.001 sec)
```

5. Buatlah tabel Transaksi sesuai keterangan pada tabel berikut dan isilah data ke dalam tabel transaksi.

**Command Line : CREATE TABLE Transaksi (Id\_transaksi int(10), Id\_pembeli varchar(10), Id\_pegawai varchar(10), Produk\_Id varchar(10), Tgl\_transaksi varchar(20), Jumlah\_beli varchar(10), PRIMARY KEY(Id\_transaksi)) ENGINE = InnoDB;**

**Command Line : ALTER TABLE Transaksi ADD FOREIGN KEY(Id\_pembeli) REFERENCES pembeli(Id\_pembeli) ON DELETE RESTRICT ON UPDATE CASCADE;**

**Command Line : ALTER TABLE Transaksi ADD FOREIGN KEY(Id\_pegawai) REFERENCES pegawai(Id\_pegawai) ON DELETE RESTRICT ON UPDATE CASCADE;**

**Command Line : ALTER TABLE Transaksi ADD FOREIGN KEY(Produk\_id) REFERENCES produk(Produk\_id) ON DELETE RESTRICT ON UPDATE CASCADE;**

**Command Line : DESC Transaksi;**

```

XAMPP for Windows - mysql -u root -p
MariaDB [galeri_itera]> CREATE TABLE Transaksi (Id_transaksi int(10), Id_pembeli varchar(10), Id_pegawai varchar(10), Produk_Id varchar(10), Tgl_transaksi varchar(20), Jumlah_beli varchar(10), PRIMARY KEY(Id_transaksi)) ENGINE = InnoDB;
Query OK, 0 rows affected (0.271 sec)

MariaDB [galeri_itera]> ALTER TABLE Transaksi ADD FOREIGN KEY(Id_pembeli) REFERENCES pembeli(Id_pembeli) ON DELETE RESTRICT ON UPDATE CASCADE;
Query OK, 0 rows affected (0.924 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [galeri_itera]> ALTER TABLE Transaksi ADD FOREIGN KEY(Id_pegawai) REFERENCES pegawai(Id_pegawai) ON DELETE RESTRICT ON UPDATE CASCADE;
Query OK, 0 rows affected (0.979 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [galeri_itera]> ALTER TABLE Transaksi ADD FOREIGN KEY(Produk_id) REFERENCES produk(Produk_id) ON DELETE RESTRICT ON UPDATE CASCADE;
Query OK, 0 rows affected (0.782 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [galeri_itera]> DESC Transaksi;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Id_transaksi | int(10)   | NO   | PRI | NULL    |       |
| Id_pembeli   | varchar(10) | YES  | MUL | NULL    |       |
| Id_pegawai   | varchar(10) | YES  | MUL | NULL    |       |
| Produk_Id    | varchar(10) | YES  | MUL | NULL    |       |
| Tgl_transaksi | varchar(20) | YES  |     | NULL    |       |
| Jumlah_beli  | varchar(10) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.021 sec)

```

**Command Line :** INSERT INTO Transaksi value (1, 'C\_901','Pg\_002', 'P552', '2022-09-12', '3 pcs'), (2, 'C\_901', 'Pg\_003', 'P109', '2022-09-15', '10 pcs'), (3, 'C\_810', 'Pg\_001', 'P114', '2022-09-15', '2 pcs'), (4, 'C\_991', 'Pg\_004', 'P333', '2022-09-18', '3 pcs'), (5, 'C\_800', 'Pg\_001', 'P552', '2022-09-18', '1 pcs'), (6, 'C\_810', 'Pg\_002', 'P552', '2022-09-20', '10 pcs');

**Command Line :** SELECT \* FROM Transaksi;

```

MariaDB [galeri_itera]> INSERT INTO Transaksi value (1, 'C_901','Pg_002', 'P552', '2022-09-12', '3 pcs'), (2, 'C_901', 'Pg_003', 'P109', '2022-09-15', '10 pcs'), (3, 'C_810', 'Pg_001', 'P114', '2022-09-15', '2 pcs'), (4, 'C_991', 'Pg_004', 'P333', '2022-09-18', '3 pcs'), (5, 'C_800', 'Pg_001', 'P552', '2022-09-18', '1 pcs'), (6, 'C_810', 'Pg_002', 'P552', '2022-09-20', '10 pcs');
Query OK, 6 rows affected (0.065 sec)
Records: 6 Duplicates: 0 Warnings: 0

MariaDB [galeri_itera]> SELECT * FROM Transaksi;
+-----+-----+-----+-----+-----+-----+
| Id_transaksi | Id_pembeli | Id_pegawai | Produk_Id | Tgl_transaksi | Jumlah_beli |
+-----+-----+-----+-----+-----+-----+
| 1 | C_901 | Pg_002 | P552 | 2022-09-12 | 3 pcs |
| 2 | C_901 | Pg_003 | P109 | 2022-09-15 | 10 pcs |
| 3 | C_810 | Pg_001 | P114 | 2022-09-15 | 2 pcs |
| 4 | C_991 | Pg_004 | P333 | 2022-09-18 | 3 pcs |
| 5 | C_800 | Pg_001 | P552 | 2022-09-18 | 1 pcs |
| 6 | C_810 | Pg_002 | P552 | 2022-09-20 | 10 pcs |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.001 sec)

```

7. Tampilkan data Produk dan supplier menggunakan Natural Join

**Command Line :** SELECT \* FROM produk NATURAL JOIN supplier;

```

XAMPP for Windows - mysql -u root -p
MariaDB [galeri_itera]> SELECT * FROM produk NATURAL JOIN supplier;
+-----+-----+-----+-----+-----+-----+
| Supplier_id | Produk_id | Produk_nama | Jumlah_Stok | Company_nama | Nama_Kontak |
+-----+-----+-----+-----+-----+-----+
| S002 | P109 | The Kotak 300 ml | 40 pcs | Suka Maju | Rahmat |
| S001 | P114 | Milo 100 ml | 800 pcs | Semua Terang | Sugiono |
| S003 | P115 | Milo 150 ml | 50 pcs | Maju Terus | Dayono |
| S005 | P123 | Gulaku 1 Kg | 100 pcs | Surya Kun | Siti |
| S001 | P235 | Aqua 250 ml | 300 pcs | Semua Terang | Sugiono |
| S003 | P311 | Grand 320 ml | 400 pcs | Maju Terus | Dayono |
| S005 | P333 | Sari Roti 100 gram | 30 pcs | Surya Kun | Siti |
| S002 | P441 | Rojo Lele 5 kg | 60 pcs | Suka Maju | Rahmat |
| S006 | P453 | Garam 30 gram | 20 pcs | Ceria Kasih | Topan |
| S001 | P552 | Aqua 1 L | 300 pcs | Semua Terang | Sugiono |
| S004 | P882 | Indomilk 25 ml | 200 pcs | Pelita Baru | Puspa |
+-----+-----+-----+-----+-----+-----+
11 rows in set (0.022 sec)

```

8. Tampilkan data Produk dan supplier yang sesuai (sama) menggunakan Join.

**Command Line : SELECT \* FROM produk INNER JOIN Supplier WHERE produk.Supplier\_id = supplier.Supplier\_id;**

```

XAMPP for Windows - mysql -u root -p
MariaDB [galeri_itera]> SELECT * FROM produk INNER JOIN Supplier WHERE produk.Supplier_id = supplier.Supplier_id;
+-----+-----+-----+-----+-----+-----+
| Produk_id | Produk_nama | Jumlah_Stok | Supplier_id | Supplier_id | Company_nama | Nama_Kontak |
+-----+-----+-----+-----+-----+-----+
| P109 | The Kotak 300 ml | 40 pcs | S002 | S002 | Suka Maju | Rahmat |
| P114 | Milo 100 ml | 800 pcs | S001 | S001 | Semua Terang | Sugiono |
| P115 | Milo 150 ml | 50 pcs | S003 | S003 | Maju Terus | Dayono |
| P123 | Gulaku 1 Kg | 100 pcs | S005 | S005 | Surya Kun | Siti |
| P235 | Aqua 250 ml | 300 pcs | S001 | S001 | Semua Terang | Sugiono |
| P311 | Grand 320 ml | 400 pcs | S003 | S003 | Maju Terus | Dayono |
| P333 | Sari Roti 100 gram | 30 pcs | S005 | S005 | Surya Kun | Siti |
| P441 | Rojo Lele 5 kg | 60 pcs | S002 | S002 | Suka Maju | Rahmat |
| P453 | Garam 30 gram | 20 pcs | S006 | S006 | Ceria Kasih | Topan |
| P552 | Aqua 1 L | 300 pcs | S001 | S001 | Semua Terang | Sugiono |
| P882 | Indomilk 25 ml | 200 pcs | S004 | S004 | Pelita Baru | Puspa |
+-----+-----+-----+-----+-----+-----+
11 rows in set (0.001 sec)

```

9. Dengan melakukan salah satu dari jenis Join tabel, tampilkan Jabatan pegawai, id pegawai, tanggal transaksi untuk pegawai dengan id “Pg\_001”.

**Command Line : SELECT Jabatan, Id\_pegawai, tgl\_transaksi FROM pegawai NATURAL JOIN transaksi WHERE Id\_Pegawai = "pg\_001";**

```

XAMPP for Windows - mysql -u root -p
MariaDB [galeri_itera]> SELECT Jabatan, Id_pegawai, tgl_transaksi FROM pegawai NATURAL JOIN transaksi WHERE Id_Pegawai = "pg_001";
+-----+-----+-----+
| Jabatan | Id_pegawai | tgl_transaksi |
+-----+-----+-----+
| Cashier | pg_001 | 2022-09-15 |
| Cashier | pg_001 | 2022-09-18 |
+-----+-----+-----+
2 rows in set (0.001 sec)

```

10. Dengan melakukan salah satu dari jenis Join tabel, tampilkan Company nama, ProdukId, Supplier id ketika Jumlah stok > 50pcs.



**Command Line : SELECT Company\_nama, Produk\_id, Supplier\_id FROM produk NATURAL JOIN Supplier WHERE Jumlah\_stok > 50;**

```
MariaDB [galeri_itera]> SELECT Company_nama, Produk_id, Supplier_id FROM produk NATURAL JOIN Supplier WHERE Jumlah_stok > 50;
```

Company_nama	Produk_id	Supplier_id
Semua Terang	P114	S001
Surya Kun	P123	S005
Semua Terang	P235	S001
Maju Terus	P311	S003
Suka Maju	P441	S002
Semua Terang	P552	S001
Pelita Baru	P882	S004

7 rows in set, 11 warnings (0.001 sec)

11. Dengan melakukan salah satu dari jenis Join tabel, tampilkan Produk id, id transaksi , jumlah dan tanggal transaksi ketika jumlah yang dibeli > 6pcs.

**Command Line : SELECT Produk\_id, id\_transaksi, jumlah\_beli, tgl\_transaksi FROM transaksi NATURAL JOIN produk WHERE jumlah\_beli > 6;**

```
MariaDB [galeri_itera]> SELECT Produk_id, id_transaksi, jumlah_beli, tgl_transaksi FROM transaksi NATURAL JOIN produk WHERE jumlah_beli > 6;
```

Produk_id	id_transaksi	jumlah_beli	tgl_transaksi
P109	2	10 pcs	2022-09-15
P552	6	10 pcs	2022-09-20

2 rows in set, 6 warnings (0.001 sec)

12. Tampilkan data dari tabel Transaksi dan pegawai yang sesuai(sama).

**Command Line : SELECT \* FROM transaksi INNER JOIN pegawai WHERE transaksi.Id\_pegawai = pegawai.Id\_pegawai;**

```
XAMPP for Windows - mysql -u root -p
MariaDB [galeri_itera]> SELECT * FROM transaksi INNER JOIN pegawai WHERE transaksi.Id_pegawai = pegawai.Id_pegawai;
```

Id_transaksi	Id_pembeli	Id_pegawai	Produk_Id	Tgl_transaksi	Jumlah_beli	Id_pegawai	Pegawai_nama	Jabatan
1	C_901	Pg_002	P552	2022-09-12	3 pcs	Pg_002	Siska	Casier
2	C_901	Pg_003	P109	2022-09-15	10 pcs	Pg_003	Nuri	Casier
3	C_810	Pg_001	P114	2022-09-15	2 pcs	Pg_001	Santi	Cashier
4	C_991	Pg_004	P333	2022-09-18	3 pcs	Pg_004	Jamal	Casier
5	C_800	Pg_001	P552	2022-09-18	1 pcs	Pg_001	Santi	Cashier
6	C_810	Pg_002	P552	2022-09-20	10 pcs	Pg_002	Siska	Casier

6 rows in set (0.001 sec)

13. Tampilkan id\_pembeli, Pembeli\_nama, tanggal transaksi, Produk\_id dan jumlah\_beli yang sama (sesuai) dengan melakukan Join pada tabel pembeli dan transaksi.

**Command Line : SELECT pembeli.Id\_pembeli, Pembeli\_nama, Tgl\_transaksi, produk\_id, jumlah\_beli FROM pembeli INNER JOIN transaksi WHERE pembeli.Id\_pembeli = transaksi.id\_pembeli;**

```
MariaDB [galeri_itera]> SELECT pembeli.Id_pembeli, Pembeli_nama, Tgl_transaksi, produk_id, jumlah_beli FROM pembeli INNER JOIN transaksi WHERE pembeli.Id_pembeli = transaksi.id_pembeli;
```

Id_pembeli	Pembeli_nama	Tgl_transaksi	produk_id	jumlah_beli
C_901	Rudi	2022-09-12	P552	3 pcs
C_901	Rudi	2022-09-15	P109	10 pcs
C_810	Ardi	2022-09-15	P114	2 pcs
C_991	Andi	2022-09-18	P333	3 pcs
C_800	Egi	2022-09-18	P552	1 pcs
C_810	Ardi	2022-09-20	P552	10 pcs

6 rows in set (0.001 sec)

14. Dengan melakukan salah satu dari jenis Join tabel, tampilkan Produk\_id, supplier\_id, Produk\_nama, Nama kontak ketika Produk\_nama “G”.

**Command Line : SELECT Produk\_id, Supplier\_id, Produk\_nama, Nama\_kontak FROM produk NATURAL JOIN Supplier WHERE Produk\_nama LIKE "G%";**

```
MariaDB [galeri_itera]> SELECT Produk_id, Supplier_id, Produk_nama, Nama_kontak FROM produk NATURAL JOIN Supplier WHERE Produk_nama LIKE "G%";
```

Produk_id	Supplier_id	Produk_nama	Nama_kontak
P123	S005	Gulaku 1 Kg	Siti
P311	S003	Grand 320 ml	Dayono
P453	S006	Garam 30 gram	Topan

3 rows in set (0.001 sec)

15. Dengan melakukan salah satu dari jenis Join tabel, tampilkan Produk\_id, Produk\_nama, id\_transaksi, jumlah stok, tanggal transaksi untuk Produk\_id “P1”.

**Command Line : SELECT Produk\_id, Produk\_nama, id\_transaksi, Jumlah\_stok, Tgl\_transaksi FROM produk NATURAL JOIN transaksi WHERE Produk\_id LIKE "P1%";**

```
MariaDB [galeri_itera]> SELECT Produk_id, Supplier_id, Produk_nama, Nama_kontak FROM produk NATURAL JOIN Supplier WHERE Produk_nama LIKE "G%";
```

Produk_id	Supplier_id	Produk_nama	Nama_kontak
P123	S005	Gulaku 1 Kg	Siti
P311	S003	Grand 320 ml	Dayono
P453	S006	Garam 30 gram	Topan

3 rows in set (0.001 sec)

16. Tampilkan Produk\_id, Produk\_nama, tanggal transaksi, id\_transaksi ketika Jumlah stok > 60pcs dan < 150pcs.

**Command Line : SELECT Produk\_id, Produk\_nama, Tgl\_transaksi, id\_transaksi FROM produk NATURAL JOIN transaksi WHERE Jumlah\_stok > 60 OR Jumlah\_stok < 150;**

```
XAMPP for Windows - mysql -u root -p
MariaDB [galeri_itera]> SELECT Produk_id, Produk_nama, Tgl_transaksi, id_transaksi FROM produk NATURAL JOIN transaksi WHERE Jumlah_stok > 60 OR Jumlah_stok < 150;
+-----+-----+-----+-----+
| Produk_id | Produk_nama | Tgl_transaksi | id_transaksi |
+-----+-----+-----+-----+
| P552      | Aqua 1 L    | 2022-09-12    | 1           |
| P109      | The Kotak 300 ml | 2022-09-15    | 2           |
| P114      | Milo 100 ml | 2022-09-15    | 3           |
| P333      | Sari Roti 100 gram | 2022-09-18    | 4           |
| P552      | Aqua 1 L    | 2022-09-18    | 5           |
| P552      | Aqua 1 L    | 2022-09-20    | 6           |
+-----+-----+-----+-----+
6 rows in set, 8 warnings (0.001 sec)
```

17. Tampilkan id\_pegawai, jabatan, id\_transaksi untuk transaksi dengan jumlah yang dibeli < 6pcs tanggal transaksi antara “2022-09-15 sampai dengan 2022-09-18”.

**Command Line : SELECT id\_pegawai, Jabatan, id\_transaksi FROM pegawai NATURAL JOIN transaksi WHERE jumlah\_beli < 6 AND Tgl\_transaksi BETWEEN "2022-09-15" AND "2022-09-18";**

```
MariaDB [galeri_itera]> SELECT id_pegawai, Jabatan, id_transaksi FROM pegawai NATURAL JOIN transaksi WHERE jumlah_beli < 6 AND Tgl_transaksi BETWEEN "2022-09-15" AND "2022-09-18";
+-----+-----+-----+
| id_pegawai | Jabatan | id_transaksi |
+-----+-----+-----+
| pg_001     | Cashier | 3           |
| pg_004     | Casier  | 4           |
| pg_001     | Cashier | 5           |
+-----+-----+-----+
3 rows in set, 6 warnings (0.013 sec)
```

## **BAB III**

### **KESIMPULAN**

---

Kesimpulannya, *Relationship* adalah suatu hubungan antara beberapa entitas. *Join* merupakan salah satu konstruksi dasar dari SQL dan basis data seperti Cross Join, Natural Join, Inner Join dan Outer Join.

## **DAFTAR PUSTAKA**

Modul 4 Praktikum Basis Data