

TUGAS MODUL 4
LAPORAN PRAKTIKUM BASIS DATA
“JOIN TABLE”



Disusun oleh:

Nama : Antonius Munthe

NIM : 121140032

Kelas : Basis Data RA

PROGRAM STUDI TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI SUMATERA
LAMPUNG SELATAN
2023

BAB I

TEORI DASAR

JOIN TABLE

Dalam SQL terdapat sebuah fungsi yang disebut dengan join. Join adalah sebuah operasi yang menggabungkan beberapa tabel menjadi satu set data baru. Ini merupakan salah satu fungsi dasar dalam database relasional yang sangat penting dan berperan dalam mendukung perencanaan database relasional yang baik. Melalui penjelasan dan gambaran mengenai kesatuan ini, kita dapat memahami lebih lanjut mengenai cara kerja fungsi join dalam SQL.

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

a) Cross Join

Cross join merupakan jenis tabel yang menghasilkan semua kombinasi baris dari dua tabel yang di-join. Dalam cross join, setiap baris dari tabel kedua dipasangkan dengan setiap baris dari tabel pertama, sehingga jumlah baris yang dihasilkan bisa sangat besar tergantung pada ukuran kedua tabel. Penggunaan cross join seringkali digunakan untuk menghasilkan tabel referensi silang atau dalam pengujian data pada saat pengembangan aplikasi. Namun, karena cross join dapat menghasilkan jumlah baris yang sangat besar, maka diperlukan manajemen data yang baik untuk meminimalkan penggunaan sumber daya sistem. Bentuk dari perintah CROSS JOIN adalah :

```
SELECT * FROM tabel1 CROSS JOIN tabel2;
```

b) Inner Join

Inner join adalah jenis join table yang menggabungkan tabel berdasarkan kondisi join yang diberikan. Dalam inner join, hanya baris data yang memiliki nilai yang sama pada kolom di-join yang akan diambil. Hasil dari inner join adalah tabel baru yang hanya berisi baris data yang cocok dari kedua tabel yang di-join. Inner join umumnya digunakan untuk menggabungkan data yang saling berkaitan dalam database, sehingga informasi dapat ditampilkan dalam satu tabel. Inner join sering digunakan pada tabel-tabel yang memiliki relasi one-to-many atau many-to-many. Bentuk dari perintah INNER JOIN adalah sebagai berikut:

```
SELECT * FROM table1 INNER JOIN table2 ON table1.field = table2.field;
```

c) Natural Join

Natural join adalah jenis inner join yang otomatis melakukan join pada kolom-kolom pada kedua tabel yang memiliki nama yang sama tanpa perlu menentukan kolom sebagai kondisi join. Natural join menghasilkan tabel baru yang hanya berisi baris data yang memiliki nilai yang sama pada kolom yang di-join. Natural join sangat berguna ketika menggabungkan tabel-tabel dengan struktur yang sama dan kolom-kolom yang memiliki nama yang sama pada tabel. Namun, perlu dilakukan dengan hati-hati karena dapat menyebabkan ambiguitas jika ada kolom-kolom lain pada tabel yang memiliki nama yang sama tetapi tidak seharusnya di-join. Bentuk dari perintah NATURAL JOIN adalah sebagai berikut:

```
SELECT * FROM table1 NATURAL JOIN table2;
```

d) Outer Join

Outer join merupakan jenis join tabel yang mempertahankan semua baris dari setidaknya satu tabel dan mencocokkan baris-baris tersebut dengan baris-baris yang cocok dari tabel lainnya. Dalam outer join, baris data yang tidak memiliki pasangan nilai yang sama pada tabel yang di-join akan tetap ditampilkan dalam tabel hasil join, tetapi nilainya akan diisi dengan NULL. Outer join biasanya digunakan ketika ingin mempertahankan semua data dari satu tabel, bahkan jika tidak memiliki pasangan nilai yang cocok pada tabel lainnya:

- LEFT OUTER JOIN

Outer join jenis left join mempertahankan semua baris dari tabel kiri dan mencocokkan baris-baris tersebut dengan baris-baris yang cocok dari tabel kanan. Jika tidak ada baris yang cocok di tabel kanan, maka kolom di sisi kanan akan diisi dengan nilai null. Dalam left join, semua baris dari tabel kiri akan ditampilkan dalam tabel hasil join, bahkan jika tidak memiliki pasangan nilai yang cocok pada tabel kanan. Bentuk dari perintahnya adalah :

```
SELECT kolom FROM table1 LEFT OUTER JOIN table2 ON table1.field =  
table2.field;
```

- RIGHT OUTER JOIN

Outer join jenis right join mempertahankan semua baris dari tabel kanan dan mencocokkan baris-baris tersebut dengan baris-baris yang cocok dari tabel kiri. Jika tidak ada baris yang cocok di tabel kiri, maka kolom di sisi kiri akan diisi dengan nilai null. Dalam right join, semua baris dari tabel kanan akan ditampilkan dalam tabel hasil join, bahkan jika tidak memiliki pasangan nilai yang cocok pada tabel kiri:

```
SELECT kolom FROM table1 RIGHT OUTER JOIN table2 ON table1.field =  
table2.field;
```

- FULL OUTER JOIN

Outer join jenis full join mempertahankan semua baris dari kedua tabel dan mencocokkan baris-baris yang cocok. Jika tidak ada pasangan nilai yang cocok pada kedua tabel, maka kolom pada sisi yang kosong akan diisi dengan nilai null. Dalam full join, semua baris dari kedua tabel akan ditampilkan dalam tabel hasil join, bahkan jika tidak memiliki pasangan nilai yang cocok pada tabel lainnya:

```
SELECT kolom FROM table1 FULL OUTER JOIN table2 ON table1.field =  
table2.field;
```

BAB II

ANALISIS DAN PEMBAHASAN

1. Tambahkan data pada tabel produk.

Dalam kasus ini untuk menambahkan data pada tabel produk dapat menggunakan “insert into produk (Produk_Id, Produk_nama, Jumlah_stok, Suppliers_id) values (“P552”, “Aqua 1 L”, “300 pcs”, “S001”)” dan seterusnya.

```
MariaDB [northwind]> insert into produk values
-> ("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 (0.014 sec)
Records: 6 Duplicates: 0 Warnings: 0

MariaDB [northwind]> select * from produk;
```

produk_id	produk_nama	jumlah_stok	suppliers_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 5kg	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 (0.002 sec)
```

2. Tambahkan data pada tabel suppliers.

Dalam kasus ini untuk menambahkan data pada tabel suppliers dapat menggunakan “insert into suppliers (suppliers_id, company_nama, nama_kontak) values (“S001”, “Pelita Baru”, “Puspa”)” dan seterusnya.

```
MariaDB [northwind]> insert into suppliers values
-> ("S004","Pelita Baru","Puspa"),
-> ("S005","Surya Kun","Siti"),
-> ("S006","Ceria Kasih","Topan");
Query OK, 3 rows affected (0.004 sec)
Records: 3  Duplicates: 0  Warnings: 0

MariaDB [northwind]> select * from suppliers;
+-----+-----+-----+
| suppliers_id | company_nama | nama_kontak |
+-----+-----+-----+
| S001         | Semua Terang | Ali         |
| S002         | Suka Maju    | Rahmat      |
| S003         | Maju Terus   | Daryono     |
| 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.

Dalam kasus ini untuk menambahkan data pada tabel pegawai dapat menggunakan “insert into pegawai (id_pegawai, pegawai_nama, jabatan) values (“Pg_001”, “Siska”, “Casier”)” dan seterusnya.

```
MariaDB [northwind]> insert into pegawai values
-> ("pg_002","Siska","Casier"),
-> ("pg_003","Nuri","Casier"),
-> ("pg_004","Jamal","Casier");
Query OK, 3 rows affected (0.005 sec)
Records: 3  Duplicates: 0  Warnings: 0

MariaDB [northwind]> select * from pegawai;
+-----+-----+-----+
| id_pegawai | pegawai_nama | jabatan |
+-----+-----+-----+
| pg_001     | Santi       | Casier  |
| pg_002     | Siska       | Casier  |
| pg_003     | Nuri        | Casier  |
| pg_004     | Jamal       | Casier  |
| pg_201     | Santo       | Casier  |
| pg_300     | Yaya        | Manager |
+-----+-----+-----+
6 rows in set (0.001 sec)
```

4. Tabel pembeli tetap seperti minggu lalu.

Untuk menampilkan tabelnya dapat menggunakan sintaks “select * from nama_tabel;”. Sehingga dalam kasus ini untuk menampilkan tabel pembeli kita dapat menuliskan “select * from pembeli;”.

```
MariaDB [northwind]> select * from pembeli;
+-----+-----+-----+
| id_pembeli | pembeli_nama | pembeli_kontak |
+-----+-----+-----+
| C_890      | Prassetya    | 085211116464   |
| C_901      | Rudi         | 081231511      |
| C_991      | Andi         | 085212021111   |
+-----+-----+-----+
3 rows in set (0.001 sec)
```

5. Buat tabel transaksi dan isi data ke dalam tabel.

Untuk membuat tabel baru kita menggunakan perintah 'create' kemudian dilanjutkan dengan menginputkan nama-nama kolom. 'id_transaksi' sebagai kolom untuk id transaksi dengan tipe data int(5), 'id_pembeli' sebagai kolom untuk id transaksi dengan tipe data 'varchar(20)', 'id_pembeli' sebagai kolom untuk id pembeli dengan tipe data 'varchar(20)', 'produk_id' sebagai kolom untuk produk id dengan tipe data 'varchar(20)', 'tgl_transaksi' sebagai kolom untuk tanggal transaksi dengan tipe data 'date', jumlah beli sebagai kolom untuk jumlah pembelian dengan tipe data varchar(20), dengan id transaksi sebagai primary key.

Untuk menambahkan data pada tabel transaksi kita menggunakan perintah "Insert into". Data yang ditambahkan adalah transaksi dengan id transaksi, id pembeli, id pegawai, produk id, tanggal transaksi dan jumlah beli dengan data tertentu. Setiap data diwakili oleh sebuah pasangan nilai dalam kurung (), yang diikuti oleh kata kunci "VALUES". Pasangan nilai tersebut mewakili untuk nilai untuk setiap kolom pada tabel, diurutkan dalam urutan yang sama seperti kolom-kolom pada tabel

```
MariaDB [northwind]> Create table transaksi (
  -> id_transaksi varchar(10),
  -> id_pembeli varchar(10),
  -> id_pegawai varchar(10),
  -> produk_id varchar(10),
  -> tgl_transaksi date,
  -> jumlah_beli varchar(10),
  -> primary key (id_transaksi)
  -> ) engine = innodb;
Query OK, 0 rows affected (0.060 sec)

MariaDB [northwind]> desc transaksi;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id_transaksi | varchar(10)   | NO   | PRI | NULL    |       |
| id_pembeli   | varchar(10)   | YES  |     | NULL    |       |
| id_pegawai   | varchar(10)   | YES  |     | NULL    |       |
| produk_id    | varchar(10)   | YES  |     | NULL    |       |
| tgl_transaksi | date          | YES  |     | NULL    |       |
| jumlah_beli  | varchar(10)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.019 sec)
```

```

MariaDB [northwind]> update pembeli set id_pembeli = "C_800", pembeli_nama = "Egi", pembeli_kontak = "0812521221" WHERE id_pembeli = "C_890";
Query OK, 1 row affected (0.010 sec)
Rows matched: 1 Changed: 1 Warnings: 0

MariaDB [northwind]> select * from pembeli;
+-----+-----+-----+
| id_pembeli | pembeli_nama | pembeli_kontak |
+-----+-----+-----+
| C_800      | Egi          | 0812521221     |
| C_901      | Rudi         | 0812315111     |
| C_991      | Andi         | 085212021111   |
+-----+-----+-----+
3 rows in set (0.001 sec)

MariaDB [northwind]> insert into pembeli values
-> ("C_810","Ardi","0862145121");
Query OK, 1 row affected (0.005 sec)

MariaDB [northwind]> select * from pembeli;
+-----+-----+-----+
| id_pembeli | pembeli_nama | pembeli_kontak |
+-----+-----+-----+
| C_800      | Egi          | 0812521221     |
| C_810      | Ardi         | 0862145121     |
| C_901      | Rudi         | 0812315111     |
| C_991      | Andi         | 085212021111   |
+-----+-----+-----+
4 rows in set (0.001 sec)

```

```

MariaDB [northwind]> ALTER TABLE transaksi
-> ADD FOREIGN KEY (id_pembeli) REFERENCES pembeli (id_pembeli),
-> ADD FOREIGN KEY (id_pegawai) REFERENCES pegawai (id_pegawai),
-> ADD FOREIGN KEY (produk_id) REFERENCES produk (produk_id)
-> ON DELETE CASCADE
-> ON UPDATE CASCADE;
Query OK, 0 rows affected (0.133 sec)
Records: 0 Duplicates: 0 Warnings: 0

```

```

MariaDB [northwind]> INSERT INTO transaksi VALUES
-> (1,"C_901","pg_002","P552","2017-09-12","3 pcs"),
-> (2,"C_901","pg_003","P109","2017-09-15","10 pcs"),
-> (3,"C_810","pg_001","P114","2017-09-15","2 pcs"),
-> (4,"C_991","pg_004","P333","2017-09-18","3 pcs"),
-> (5,"C_800","pg_001","P552","2017-09-18","1 pcs"),
-> (6,"C_810","pg_002","P123","2017-09-20","10 pcs");
Query OK, 6 rows affected (0.009 sec)
Records: 6 Duplicates: 0 Warnings: 0

```

6. Tampilkan data Produk dan Supplier menggunakan Natural join.

Untuk menampilkan data produk dan supplier menggunakan natural join dapat menggunakan perintah “select kolom From <nama_tabel1> Natural join <nama_tabel2>;”. Natural join akan secara otomatis melakukan join antara kedua tabel berdasarkan kolom-kolom yang memiliki nama yang sama dikedua tabel tersebut.

```
MariaDB [northwind]> select * from produk natural join suppliers;
```

suppliers_id	produk_id	produk_nama	jumlah_stok	company_nama	nama_kontak
S001	P114	Milo 100 ml	800 pcs	Semua Terang	Ali
S001	P235	Aqua 250 ml	300 pcs	Semua Terang	Ali
S001	P552	Aqua 1 L	300 pcs	Semua Terang	Ali
S002	P109	The Kotak 300 ml	40 pcs	Suka Maju	Rahmat
S002	P441	Rojo Lele 5kg	60 pcs	Suka Maju	Rahmat
S003	P115	Milo 150 ml	50 pcs	Maju Terus	Daryono
S003	P311	Grand 320 ml	400 pcs	Maju Terus	Daryono
S004	P882	Indomilk 25 ml	200 pcs	Pelita Baru	Puspa
S005	P123	Gulaku 1 Kg	100 pcs	Surya Kun	Siti
S005	P333	Sari Roti 100 gram	30 pcs	Surya Kun	Siti
S006	P453	Garam 30 gram	20 pcs	Ceria Kasih	Topan

11 rows in set (0.010 sec)

7. Tampilkan data Produk dan Supplier yang sesuai(sama) menggunakan join.

Dalam kasus ini untuk menampilkan data produk dan Supplier yang sesuai(sama) kita dapat menuliskan “select * from produk inner join suppliers where produk.suppliers_id = suppliers.suppliers_id;;”.

```
MariaDB [northwind]> select * from produk inner join suppliers
-> on produk.suppliers_id = suppliers.suppliers_id;
```

produk_id	produk_nama	jumlah_stok	suppliers_id	suppliers_id	company_nama	nama_kontak
P114	Milo 100 ml	800 pcs	S001	S001	Semua Terang	Ali
P235	Aqua 250 ml	300 pcs	S001	S001	Semua Terang	Ali
P552	Aqua 1 L	300 pcs	S001	S001	Semua Terang	Ali
P109	The Kotak 300 ml	40 pcs	S002	S002	Suka Maju	Rahmat
P441	Rojo Lele 5kg	60 pcs	S002	S002	Suka Maju	Rahmat
P115	Milo 150 ml	50 pcs	S003	S003	Maju Terus	Daryono
P311	Grand 320 ml	400 pcs	S003	S003	Maju Terus	Daryono
P882	Indomilk 25 ml	200 pcs	S004	S004	Pelita Baru	Puspa
P123	Gulaku 1 Kg	100 pcs	S005	S005	Surya Kun	Siti
P333	Sari Roti 100 gram	30 pcs	S005	S005	Surya Kun	Siti
P453	Garam 30 gram	20 pcs	S006	S006	Ceria Kasih	Topan

11 rows in set (0.001 sec)

8. Lakukan Left Outer join pada tabel Pegawai dan Transaksi.

Dalam kasus ini kita dapat menuliskan “select pegawai.*, transaksi.* from pegawai left outer join transaksi on pegawai.id_Pegawai = transaksi.id_pegawai;”. Left outer join berfungsi untuk menampilkan semua data pada tabel 1 (kiri) meskipun data tersebut tidak memiliki pasangan pada tabel 2 (kanan). Sehingga didapatkan luara seperti gambar dibawah.

```
MariaDB [northwind]> select * from pegawai left outer join transaksi
-> on pegawai.id_pegawai = transaksi.id_pegawai;
```

id_pegawai	pegawai_nama	jabatan	id_transaksi	id_pembeli	id_pegawai	produk_id	tgl_transaksi	jumlah_beli
pg_001	Santi	Casier	3	C_810	pg_001	P114	2017-09-15	2 pcs
pg_001	Santi	Casier	5	C_800	pg_001	P552	2017-09-18	1 pcs
pg_002	Siska	Casier	1	C_901	pg_002	P552	2017-09-12	3 pcs
pg_002	Siska	Casier	6	C_810	pg_002	P123	2017-09-20	10 pcs
pg_003	Nuri	Casier	2	C_901	pg_003	P109	2017-09-15	10 pcs
pg_004	Jamal	Casier	4	C_991	pg_004	P333	2017-09-18	3 pcs
pg_201	Santo	Casier	NULL	NULL	NULL	NULL	NULL	NULL
pg_300	Yaya	Manager	NULL	NULL	NULL	NULL	NULL	NULL

8 rows in set (0.003 sec)

9. Lakukan Right Outer join pada tabel Pembeli dan Transaksi.

Dalam kasus ini kita dapat menuliskan “select pembeli.*, transaksi.* from pembeli right outer join transaksi on pembeli.id_pembeli = transaksi.id_pembeli;”. Right outer join berfungsi untuk menampilkan semua data pada tabel 2 (kanan) meskipun data tersebut tidak memiliki pasangan pada tabel 1 (kiri). Sehingga didapatkan luara seperti gambar dibawah.

```
MariaDB [northwind]> select * from pembeli right outer join transaksi
-> on pembeli.id_pembeli = transaksi.id_pembeli;
```

id_pembeli	pembeli_nama	pembeli_kontak	id_transaksi	id_pembeli	id_pegawai	produk_id	tgl_transaksi	jumlah_beli
C_901	Rudi	081231511	1	C_901	pg_002	P552	2017-09-12	3 pcs
C_901	Rudi	081231511	2	C_901	pg_003	P109	2017-09-15	10 pcs
C_810	Ardi	0862145121	3	C_810	pg_001	P114	2017-09-15	2 pcs
C_991	Andi	085212021111	4	C_991	pg_004	P333	2017-09-18	3 pcs
C_800	Egi	0812521221	5	C_800	pg_001	P552	2017-09-18	1 pcs
C_810	Ardi	0862145121	6	C_810	pg_002	P123	2017-09-20	10 pcs

6 rows in set (0.004 sec)

10. Tampilkan data transaksi dan produk yang sesuai(sama) menggunakan join.

Dalam kasus ini untuk menampilkan transaksi dan produk yang sesuai(sama) kita dapat menuliskan “select * from transaksi cross join produk where transaksi.produk_id = produk.produk_id;”. Sehingga didapatkan luara seperti gambar dibawah.

```
MariaDB [northwind]> select * from transaksi inner join produk
-> on transaksi.produk_id = produk.produk_id;
```

id_transaksi	id_pembeli	id_pegawai	produk_id	tgl_transaksi	jumlah_beli	produk_id	produk_nama	jumlah_stok	suppliers_id
1	C_901	pg_002	P552	2017-09-12	3 pcs	P552	Aqua 1 L	300 pcs	S001
2	C_901	pg_003	P109	2017-09-15	10 pcs	P109	The Kotak 300 ml	40 pcs	S002
3	C_810	pg_001	P114	2017-09-15	2 pcs	P114	Milo 100 ml	800 pcs	S001
4	C_991	pg_004	P333	2017-09-18	3 pcs	P333	Sari Roti 100 gram	30 pcs	S005
5	C_800	pg_001	P552	2017-09-18	1 pcs	P552	Aqua 1 L	300 pcs	S001
6	C_810	pg_002	P123	2017-09-20	10 pcs	P123	Gulaku 1 Kg	100 pcs	S005

6 rows in set (0.004 sec)

11. Dengan menggunakan Cross join tampilkan data pegawai dan transaksi yang sesuai(sama).

Dalam kasus ini untuk menampilkan data pegawai dan transaksi yang sesuai(sama) kita dapat menuliskan “select * from pegawai cross join transaksi where pegawai.id_pegawai = transaksi.id_pegawai;”. Sehingga didapatkan luara seperti gambar dibawah.

```
MariaDB [northwind]> select * from pegawai cross join transaksi
-> on pegawai.id_pegawai = transaksi.id_pegawai;
```

id_pegawai	pegawai_nama	jabatan	id_transaksi	id_pembeli	id_pegawai	produk_id	tgl_transaksi	jumlah_beli
pg_002	Siska	Casier	1	C_901	pg_002	P552	2017-09-12	3 pcs
pg_003	Nuri	Casier	2	C_901	pg_003	P109	2017-09-15	10 pcs
pg_001	Santi	Casier	3	C_810	pg_001	P114	2017-09-15	2 pcs
pg_004	Jamal	Casier	4	C_991	pg_004	P333	2017-09-18	3 pcs
pg_001	Santi	Casier	5	C_800	pg_001	P552	2017-09-18	1 pcs
pg_002	Siska	Casier	6	C_810	pg_002	P123	2017-09-20	10 pcs

6 rows in set (0.001 sec)

12. Dengan melakukan salah satu dari jenis join tabel, tampilkan jabatan pegawai, id pegawai, tanggal transaksi untuk pegawai dengan id “Pg_001”.

Dalam kasus ini untuk Untuk menampilkan jabatan pegawai, id pegawai, tanggal transaksi untuk pegawai dengan id “Pg_001” kita dapat menuliskan “select pegawai.jabatan, transaksi.id_pegawai, transaksi.Tgl_transaksi from transaksi inner join pegawai on pegawai.id_pegawai = transaksi.id_pegawai where transaksi.id_pegawai = "Pg_001";”. Sehingga didapatkan luaran seperti pada gambar dibawah.

```
MariaDB [northwind]> select pegawai.jabatan, pegawai.id_pegawai, transaksi.tgl_transaksi from pegawai inner join transaksi
-> on pegawai.id_pegawai = transaksi.id_pegawai
-> where pegawai.id_pegawai = "pg_001";
```

jabatan	id_pegawai	tgl_transaksi
Casier	pg_001	2017-09-15
Casier	pg_001	2017-09-18

2 rows in set (0.005 sec)

13. Dengan melakukan salah satu dari jenis join tabel, tampilkan company nama, produk id, supplier id ketika jumlah stok > 50 pcs.

Dalam kasus ini untuk Untuk menampilkan company nama, produk id, supplier id ketika jumlah stok > 50 pcs kita dapat menuliskan “select suppliers.company_nama, produk.produk_id, produk.suppliers_id from suppliers inner join produk on suppliers.suppliers_id = produk.suppliers_id where produk.jumlah_stok > 50;”. Sehingga didapatkan luaran seperti pada gambar dibawah.

```
MariaDB [northwind]> SELECT suppliers.company_nama, produk.produk_id, suppliers.suppliers_id
-> FROM suppliers
-> INNER JOIN produk ON suppliers.suppliers_id = produk.suppliers_id
-> WHERE REGEXP_REPLACE(produk.jumlah_stok, '^[^0-9]', '') > 50;
```

company_nama	produk_id	suppliers_id
Semua Terang	P114	S001
Semua Terang	P235	S001
Semua Terang	P552	S001
Suka Maju	P441	S002
Maju Terus	P311	S003
Pelita Baru	P882	S004
Surya Kun	P123	S005

7 rows in set (0.001 sec)

14. Dengan melakukan salah satu dari jenis join tabel, tampilkan produk id, id transaksi, jumlah dan tanggal transaksi ketika jumlah yang dibeli > 6 pcs.

Dalam kasus ini untuk Untuk menampilkan produk id, id transaksi, jumlah dan tanggal transaksi ketika jumlah yang dibeli > 6 pcs kita dapat menuliskan “select produk.produk_id, transaksi.id_transaksi, transaksi.jumlah_beli, transaksi.Tgl_transaksi from produk inner join transaksi on produk.produk_id = transaksi.produk_id where transaksi.jumlah_beli > 6;”.

```
MariaDB [northwind]> SELECT produk.produk_id, transaksi.id_transaksi, transaksi.jumlah_beli, transaksi.tgl_transaksi
-> FROM produk
-> INNER JOIN transaksi ON produk.produk_id = transaksi.produk_id
-> WHERE REGEXP_REPLACE(transaksi.jumlah_beli, '^[^0-9]', '') > 6;
```

produk_id	id_transaksi	jumlah_beli	tgl_transaksi
P109	2	10 pcs	2017-09-15
P123	6	10 pcs	2017-09-20

2 rows in set (0.001 sec)

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

Dalam kasus ini untuk menampilkan data dari tabel Transaksi dan pegawai yang sesuai (sama) kita dapat menuliskan “select * from transaksi inner join pegawai on transaksi.id_pegawai = pegawai.id_pegawai;”. Sehingga menghasilkan luaran seperti pada gambar dibawah.

```
MariaDB [northwind]> select * from transaksi inner join pegawai
-> on 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	2017-09-12	3 pcs	pg_002	Siska	Casier
2	C_901	pg_003	P109	2017-09-15	10 pcs	pg_003	Nuri	Casier
3	C_810	pg_001	P114	2017-09-15	2 pcs	pg_001	Santi	Casier
4	C_991	pg_004	P333	2017-09-18	3 pcs	pg_004	Jamal	Casier
5	C_800	pg_001	P552	2017-09-18	1 pcs	pg_001	Santi	Casier
6	C_810	pg_002	P123	2017-09-20	10 pcs	pg_002	Siska	Casier

```
6 rows in set (0.001 sec)
```

16. Tampilkan id_pembeli, pembeli_nama, tanggal transaksi, produk_id dan jumlah_beli yang sama(sesuai) dengan melakukan join pada tabel pembeli dan transaksi.

Dalam kasus ini untuk menampilkan id_pembeli, pembeli_nama, tanggal transaksi, produk_id dan jumlah_beli yang sama(sesuai) dengan melakukan join pada tabel pembeli dan transaksi kita dapat menuliskan “select pembeli.id_pembeli, pembeli.pembeli_nama, transaksi.Tgl_transaksi, transaksi.produk_id, transaksi.jumlah_beli from pembeli inner join transaksi on pembeli.id_pembeli = transaksi.id_pembeli;”.

```
MariaDB [northwind]> select pembeli.id_pembeli, pembeli.pembeli_nama, transaksi.tgl_transaksi, transaksi.produk_id, transaksi.jumlah_beli
-> from pembeli inner join transaksi
-> on pembeli.id_pembeli = transaksi.id_pembeli;
```

id_pembeli	pembeli_nama	tgl_transaksi	produk_id	jumlah_beli
C_901	Rudi	2017-09-12	P552	3 pcs
C_901	Rudi	2017-09-15	P109	10 pcs
C_810	Ardi	2017-09-15	P114	2 pcs
C_991	Andi	2017-09-18	P333	3 pcs
C_800	Egi	2017-09-18	P552	1 pcs
C_810	Ardi	2017-09-20	P123	10 pcs

```
6 rows in set (0.001 sec)
```

17. Dengan melakukan salah satu dari jenis join tabel, tampilkan produk_id, supplier_id, produk_nama, nama kontak 12etika produk_nama “G”.

Dalam kasus ini untuk menampilkan produk_id, supplier_id, produk_nama, nama kontak ketika produk_nama “G” kita dapat menuliskan “select produk.produk_id, suppliers.suppliers_id, produk.produk_nama, suppliers.nama_kontak from produk inner join suppliers on produk.suppliers_id = suppliers.suppliers_id where produk.produk_nama like “G%”;”. Sehingga menghasilkan luaran seperti gambar dibawah.

```
MariaDB [northwind]> select produk.produk_id, suppliers.suppliers_id, suppliers.nama_kontak
-> from produk inner join suppliers
-> on produk.suppliers_id = suppliers.suppliers_id
-> where produk.produk_nama like "G%";
```

produk_id	suppliers_id	nama_kontak
P311	S003	Daryono
P123	S005	Siti
P453	S006	Topan

```
3 rows in set (0.001 sec)
```

18. Dengan melakukan salah satu dari jenis join tabel, tampilkan produk_id, produk_nama, id transaksi, jumlah stok, tanggal transaksi untuk Produk_id “P1”.

Dalam kasus ini untuk menampilkan produk_id, produk_nama, id transaksi, jumlah stok, tanggal transaksi untuk Produk_id “P1” kita dapat menuliskan “select produk.produk_id, produk.produk_nama, transaksi.id_transaksi, produk.jumlah_stok, transaksi.tgl_transaksi from produk inner join transaksi on produk.produk_id = transaksi.produk_id where produk.produk_id like "P1%";”. Sehingga didapatkan luaran seperti gambar dibawah.

```
MariaDB [northwind]> select produk.produk_id, produk.produk_nama, transaksi.id_transaksi, produk.jumlah_stok, transaksi.tgl_transaksi
-> from produk left outer join transaksi
-> on produk.produk_id = transaksi.produk_id
-> where produk.produk_id like "P1%";
```

produk_id	produk_nama	id_transaksi	jumlah_stok	tgl_transaksi
P109	The Kotak 300 ml	2	40 pcs	2017-09-15
P114	Milo 100 ml	3	800 pcs	2017-09-15
P115	Milo 150 ml	NULL	50 pcs	NULL
P123	Gulaku 1 Kg	6	100 pcs	2017-09-20

```
4 rows in set (0.000 sec)
```

19. Tampilkan Produk_id, Produk_nama, tanggal transaksi, id_transaksi, id_transaksi ketika jumlah stok > 60 pcs dan < 150 pcs.

Dalam kasus ini untuk menampilkan Produk_id, Produk_nama, tanggal transaksi, id_transaksi, id_transaksi ketika jumlah stok > 60 pcs dan < 150 pcs kita dapat menuliskan “select produk.produk_id, produk.produk_nama, transaksi.tgl_transaksi, transaksi.id_transaksi from produk inner join transaksi on produk.produk_id = transaksi.produk_id where produk.jumlah_stok > 60 pcs and produk.jumlah_stok < 150 pcs;”. Sehingga didapatkan luaran seperti pada gambar dibawah.

```
MariaDB [northwind]> select produk.produk_id, produk.produk_nama, transaksi.tgl_transaksi, transaksi.id_transaksi
-> from produk inner join transaksi
-> on produk.produk_id = transaksi.produk_id
-> where REGEXP_REPLACE(produk.jumlah_stok, '^0-9', '') > 60 AND REGEXP_REPLACE(produk.jumlah_stok, '^0-9', '') < 150;
```

produk_id	produk_nama	tgl_transaksi	id_transaksi
P123	Gulaku 1 Kg	2017-09-20	6

```
1 row in set (0.001 sec)
```

20. Tampilkan id_pegawai, jabatan, id_transaksi untuk transaksi dengan jumlah yang dibeli < 6 pcs tanggal transaksi antara “2017-09-15” sampai dengan “2017-09-18”.

Dalam kasus ini untuk menampilkan Tampilkan id_pegawai, jabatan, id_transaksi untuk transaksi dengan jumlah yang dibeli < 6 pcs tanggal transaksi antara “2017-09-15” sampai dengan “2017-09-18”. Kita dapat menuliskan “ select pegawai.id_pegawai, pegawai.jabatan, transaksi.id_transaksi from transaksi inner join pegawai on transaksi.id_pegawai = pegawai.id_pegawai where transaksi.tgl_transaksi between "2017-09-15" and "2017-09-18" and transaksi.jumlah_beli < 6; ”.

```
MariaDB [northwind]> select pegawai.id_pegawai, pegawai.jabatan, transaksi.id_transaksi
-> from pegawai inner join transaksi
-> on pegawai.id_pegawai = transaksi.id_pegawai
-> where REGEXP_REPLACE(transaksi.jumlah_beli, '^[^0-9]', '') < 6
-> AND transaksi.tgl_transaksi between "2017-09-15" AND "2017-09-18";
```

id_pegawai	jabatan	id_transaksi
pg_001	Casier	3
pg_004	Casier	4
pg_001	Casier	5

3 rows in set (0.005 sec)

BAB III

KESIMPULAN

1. Dalam Menyelesaikan soal ini digunakan beberapa fungsi join yaitu:

a. Cross Join

```
SELECT * FROM tabel1 CROSS JOIN tabel2;
```

b. Inner Join

```
SELECT * FROM table1 INNER JOIN table2 ON table1.field = table-2.field;
```

c. Natural Join

```
SELECT * FROM table1 NATURAL JOIN table2;
```

d. Outer Join:

- Left Join :

```
SELECT kolom FROM table1 LEFT OUTER JOIN table2  
ON table1.field = table2.field;
```
- Right Join :

```
SELECT kolom FROM table1 RIGHT OUTER JOIN table2  
ON table1.field = table2.field;
```
- Full Join :

```
SELECT kolom FROM table1 FULL OUTER JOIN table2  
ON table1.field = table2.field;
```

2. Dalam proses join, diperlukan penentuan kolom yang akan digunakan sebagai kunci untuk menghubungkan antara satu tabel dengan tabel lainnya. Dalam penggunaannya, jenis join yang tepat dapat membantu memudahkan pengambilan data dari beberapa tabel dalam satu query dan mempercepat kinerja database. Namun, penggunaan join secara berlebihan dapat mengakibatkan lambatnya kinerja query dan masalah pada database. Oleh karena itu, memahami jenis join yang tersedia dan menggunakan join yang sesuai dengan kebutuhan sangatlah penting.

DAFTAR PUSTAKA

- Modul Praktikum Basis Data Pertemuan 4