

LATIHAN 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 table “Nasabah” pada Database Bank Mino dan isi nilai pada setiap kolom seperti berikut.

Untuk membuat tabel baru kita menggunakan perintah ‘create’ kemudian dilanjutkan dengan menginputkan nama-nama kolom. ‘id_Nasabah’ sebagai kolom untuk id nasabah dengan tipe data ‘int(3)’, ‘no_rekening’ sebagai kolom untuk nomor rekening dengan tipe data ‘int(6)’, ‘Nama_nasabah’ sebagai kolom untuk nama nasabah dengan tipe data ‘char(20)’, ‘Alamat’ sebagai kolom untuk alamat nasabah dengan tipe data ‘char(30)’, ‘No_telepon’ sebagai kolom untuk nomor telepon nasabah dengan tipe data ‘int(15)’ dengan id nasabah sebagai primary key.

Untuk menambahkan data pada tabel nasabah kita menggunakan perintah “Insert into”. Data yang ditambahkan adalah nasabah dengan id nasabah, no_rekening, nama nasabah, alamat, dan no_telepon 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 [bank_mino]> select * from nasabah;
```

id_nasabah	no_rekening	nama_nasabah	alamat	no_telepon
1	145678	Puspa Sari	Sei Batang Kuis	2147483647
2	256132	Sinta Jojo	Iskandar Muda	2147483647
3	247182	Bunga Putri	Jalan Martapura	2147483647
4	350067	Grace Sebayang	Gang Keluarga	2147483647
5	114115	Christine Talia	Sei Kambah	2147483647
6	100111	Rimayka Natalia	Perumahan Puri K	2147483647

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

2. Masukkan data dalam tabel “ rekening”

Untuk menambahkan data pada tabel rekening kita menggunakan perintah “Insert into”. Data yang ditambahkan adalah 6 rekening dengan no rekening, kode, cabang pin dan saldo 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 [bank_mino]> select * from rekening;
```

no_rekening	kode_cabang	pin	saldo
100111	KKM69	213671	998500
114115	SGS01	203010	500000
120023	JGJ01	837242	500000
140932	JOJ02	024422	1200000
145678	SGS02	908723	1000000
220983	SMG12	450817	5422000
247182	JKT48	091298	4000000
256132	KKM70	340987	230000
332102	JKT48	228883	400500
350067	RGS02	230987	5500000

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

3. Masukkan data ke dalam tabel “transaksi”

Untuk menambahkan data pada tabel transaksi kita menggunakan perintah “Insert into”. Data yang ditambahkan adalah 7 transaksi dengan no transaksi, no rekening, jenis transaksi, tanggal transaksi dan jumlah 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 [bank_mino]> select * from transaksi;
```

no_transaksi	no_rekening	jenis_transaksi	tgl_transaksi	jumlah
1	220983	DEBET	2014-06-12 12:00:45	500000
2	220983	DEBET	2014-06-13 08:20:33	2150000
3	140932	ATM	2014-06-14 10:56:52	100000
4	140932	ATM	2014-06-14 15:16:23	100000
6	220983	ATM	2014-06-17 20:34:22	150000
7	220983	ATM	2014-06-15 15:40:12	100000
8	220983	DEBET	2014-06-16 08:23:23	2500000
11	100111	ATM	2017-06-09 08:00:45	100000
12	114115	ATM	2017-06-09 11:10:12	200000
13	114115	DEBET	2017-06-11 13:00:30	50000
14	256132	DEBET	2017-06-15 11:20:00	150000
15	145678	ATM	2017- 06-15 23:11:22	250000
16	145678	DEBET	2017- 06-18 15:30:08	300000
17	350067	ATM	2017- 06-18 10:02:45	100000

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

4. Dengan menggunakan cross join ,tampilkan data dari nasabah dan rekening yang sesuai (sama).

Untuk menampilkan data nasabah dan rekening yang sama kita menggunakan perintah ‘Select kolom from <nama_tabel> cross join <nama_tabel> where <nama_tabel.nama_kolom> = <nama_tabel.nama_kolom>’. Kita perlu menentukan kolom yang digunakan sebagai kunci join antara kedua tabel. Pada kasus ini kita menggunakan no rekening sebagai kunci join dari kedua tabel tersebut.

```
MariaDB [bank_mino]> select * from nasabah cross join rekening on nasabah.no_rekening = rekening.no_rekening;
```

id_nasabah	no_rekening	nama_nasabah	alamat	no_telepon	no_rekening	kode_cabang	pin	saldo
1	145678	Puspa Sari	Sei Batang Kuis	2147483647	145678	SGS02	908723	1000000
2	256132	Sinta Jojo	Iskandar Muda	2147483647	256132	KKM70	340987	230000
3	247182	Bunga Putri	Jalan Martapura	2147483647	247182	JKT48	091298	4000000
4	350067	Grace Sebayang	Gang Keluarga	2147483647	350067	RGS02	230987	5500000
5	114115	Christine Talia	Sei Kambah	2147483647	114115	SGS01	203010	500000
6	100111	Rimayka Natalia	Perumahan Puri K	2147483647	100111	KKM69	213671	998500

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

```
MariaDB [bank_mino]>
```

5. Dengan menggunakan Inner join ,tampilkan data dari rekening dan transaksi yang sesuai(sama).

Untuk menampilkan data rekening dan transaksi yang sama menggunakan inner join kita menggunakan perintah ‘Select kolom from <nama_tabel> inner join <nama_tabel> where <nama_tabel.nama_kolom> = <nama_tabel.nama_kolom>’. Kunci join dalam kasus ini adalah no rekening.

```
MariaDB [bank_mino]> select * from rekening inner join transaksi on rekening.no_rekening = transaksi.no_rekening;
```

no_rekening	kode_cabang	pin	saldo	no_transaksi	no_rekening	jenis_transaksi	tgl_transaksi	jumlah
220983	SMG12	450817	5422000	1	220983	DEBET	2014-06-12 12:00:45	500000
220983	SMG12	450817	5422000	2	220983	DEBET	2014-06-13 08:20:33	2150000
140932	JOJ02	024422	1200000	3	140932	ATM	2014-06-14 10:56:52	100000
140932	JOJ02	024422	1200000	4	140932	ATM	2014-06-14 15:16:23	100000
220983	SMG12	450817	5422000	6	220983	ATM	2014-06-17 20:34:22	150000
220983	SMG12	450817	5422000	7	220983	ATM	2014-06-15 15:40:12	100000
220983	SMG12	450817	5422000	8	220983	DEBET	2014-06-16 08:23:23	2500000
100111	KKM69	213671	998500	11	100111	ATM	2017-06-09 08:00:45	100000
114115	SGS01	203010	500000	12	114115	ATM	2017-06-09 11:10:12	200000
114115	SGS01	203010	500000	13	114115	DEBET	2017-06-11 13:00:30	50000
256132	KKM70	340987	230000	14	256132	DEBET	2017-06-15 11:20:00	150000
145678	SGS02	908723	1000000	15	145678	ATM	2017- 06-15 23:11:22	250000
145678	SGS02	908723	1000000	16	145678	DEBET	2017- 06-18 15:30:08	300000
350067	RGS02	230987	5500000	17	350067	ATM	2017- 06-18 10:02:45	100000

```
14 rows in set (0.000 sec)

MariaDB [bank_mino]> |
```

6. Dengan menggunakan natural Join, tampilkan nasabah dan rekening.

Untuk menampilkan data nasabah dan rekening menggunakan natural join kita menggunakan perintah ‘select kolom From <nama_tabel> Natural join <nama_tabel>’. Natural join akan secara otomatis melakukan join antara kedua tabel berdasarkan kolom-kolom yang memiliki nama yang sama dikedua tabel tersebut. Dalam hal ini, asumsi bahwa terdapat kolom dengan nama yang sama dikedua tabel yaitu “no_rekening”. Jika ada lebih dari satu kolom dengan nama yang sama, maka Natural join melakukan join pada semua kolom tersebut.

```
MariaDB [bank_mino]> select * from nasabah natural join rekening;
```

no_rekening	id_nasabah	nama_nasabah	alamat	no_telepon	kode_cabang	pin	saldo
145678	1	Puspa Sari	Sei Batang Kuis	2147483647	SGS02	908723	1000000
256132	2	Sinta Jojo	Iskandar Muda	2147483647	KKM70	340987	230000
247182	3	Bunga Putri	Jalan Martapura	2147483647	JKT48	091298	4000000
350067	4	Grace Sebayang	Gang Keluarga	2147483647	RGS02	230987	5500000
114115	5	Christine Talia	Sei Kambah	2147483647	SGS01	203010	500000
100111	6	Rimayka Natalia	Perumahan Puri K	2147483647	KKM69	213671	998500

```
6 rows in set (0.000 sec)

MariaDB [bank_mino]> |
```

7. Dengan menggunakan Left Join untuk tabel rekening dan transaksi tampilkan nomor rekening , saldo dan tanggal transaksi, id transaksi.

Untuk melakukan left outer join pada tabel rekening dan transaksi kita menggunakan perintah ‘Select kolom from <nama_tabel> left outer join <nama_tabel> ON <nama_tabel.nama_kolom> = <nama_tabel.nama_kolom>’. Left outer join digunakan untuk mengambil semua data dari tabel rekening, dan data yang sesuai dari tabel transaksi. Left outer

join mengambil semua data dari tabel kiri dan hanya data yang sesuai dari tabel kanan. Kunci join dari kasus ini adalah kolom no_rekening.

```
MariaDB [bank_mino]> select rekening.no_rekening, rekening.saldo, transaksi.tgl_transaksi, transaksi.no_transaksi
-> FROM rekening
-> left outer join transaksi on rekening.no_rekening = transaksi.no_rekening;
```

no_rekening	saldo	tgl_transaksi	no_transaksi
100111	998500	2017-06-09 08:00:45	11
114115	500000	2017-06-09 11:10:12	12
114115	500000	2017-06-11 13:00:30	13
120023	500000	NULL	NULL
140932	1200000	2014-06-14 10:56:52	3
140932	1200000	2014-06-14 15:16:23	4
145678	1000000	2017-06-15 23:11:22	15
145678	1000000	2017-06-18 15:30:08	16
220983	5422000	2014-06-12 12:00:45	1
220983	5422000	2014-06-13 08:20:33	2
220983	5422000	2014-06-17 20:34:22	6
220983	5422000	2014-06-15 15:40:12	7
220983	5422000	2014-06-16 08:23:23	8
247182	4000000	NULL	NULL
256132	230000	2017-06-15 11:20:00	14
332102	400500	NULL	NULL
350067	5500000	2017-06-18 10:02:45	17

17 rows in set (0.000 sec)

8. Dengan menggunakan Right Join untuk tabel nasabah dan rekening tampilkan

nomor rekening, nama nasabah , kode cabang dan saldo nasabah.

Untuk melakukan right outer join pada tabel nasabah dan rekening kita menggunakan perintah 'Select kolom from <nama_tabel> Right outer join <nama_tabel> ON <nama_tabel.nama_kolom> = <nama_tabel.nama_kolom>'. Right outer join digunakan untuk mengambil semua data dari tabel nasabah, dan data yang sesuai dari tabel rekening. Left outer join mengambil semua data dari tabel kanan dan hanya data yang sesuai dari tabel kiri. Kunci join dari kasus ini adalah kolom id no_rekening.

```
MariaDB [bank_mino]> select nasabah.no_rekening, nasabah.nama_nasabah, rekening.kode_cabang, rekening.saldo from nasabah right outer
join rekening on nasabah.no_rekening = rekening.no_rekening;
```

no_rekening	nama_nasabah	kode_cabang	saldo
100111	Rimayka Natalia	KKM69	998500
114115	Christine Talia	SGS01	500000
NULL	NULL	JGJ01	500000
NULL	NULL	JOJ02	1200000
145678	Puspa Sari	SGS02	1000000
NULL	NULL	SMG12	5422000
247182	Bunga Putri	JKT48	4000000
256132	Sinta Jojo	KKM70	230000
NULL	NULL	JKT48	400500
350067	Grace Sebayang	RGS02	5500000

10 rows in set (0.000 sec)

```
MariaDB [bank_mino]>
```

9. Dengan menggunakan salah satu dari Jenis Join di atas, tampilkan data rekening dan transaksi ketika jumlah transaksi > 100000.

Untuk menampilkan data rekening dan transaksi dengan ketentuan jumlah transaksi '> 100000' kita menggunakan inner join dengan perintah 'Select kolom from <nama_tabel> inner join <nama_tabel> ON <nama_tabel.nama_kolom> = <nama_tabel.nama_kolom> where <nama_tabel.nama_kolom> = kondisi kasus. Kunci join dalam kasus ini adalah kolom no rekening.

```
MariaDB [bank_mino]> select * from rekening inner join transaksi
-> on rekening.no_rekening = transaksi.no_rekening
-> where transaksi.jumlah > 100000;
```

no_rekening	kode_cabang	pin	saldo	no_transaksi	no_rekening	jenis_transaksi	tgl_transaksi	jumlah
220983	SMG12	450817	5422000	1	220983	DEBET	2014-06-12 12:00:45	500000
220983	SMG12	450817	5422000	2	220983	DEBET	2014-06-13 08:20:33	2150000
220983	SMG12	450817	5422000	6	220983	ATM	2014-06-17 20:34:22	150000
220983	SMG12	450817	5422000	8	220983	DEBET	2014-06-16 08:23:23	2500000
114115	SGS01	203010	500000	12	114115	ATM	2017-06-09 11:10:12	200000
256132	KKM70	340987	230000	14	256132	DEBET	2017-06-15 11:20:00	150000
145678	SGS02	908723	1000000	15	145678	ATM	2017- 06-15 23:11:22	250000
145678	SGS02	908723	1000000	16	145678	DEBET	2017- 06-18 15:30:08	300000

8 rows in set (0.001 sec)

10. Dengan menggunakan salah satu dari Jenis Join di atas, tampilkan data nasabah dan rekening ketika saldo rekening >1500000.

Untuk menampilkan data nasabah dan rekening dengan ketentuan 'saldo rekening > 1500000' kita menggunakan inner join dengan perintah 'Select kolom from <nama_tabel> inner join <nama_tabel> ON <nama_tabel.nama_kolom> = <nama_tabel.nama_kolom> where <nama_tabel.nama_kolom> = kondisi kasus. Kunci join dalam kasus ini adalah kolom no rekening.

```
MariaDB [bank_mino]> select * from nasabah inner join rekening
-> on nasabah.no_rekening = rekening.no_rekening
-> where rekening.saldo > 1500000;
```

id_nasabah	no_rekening	nama_nasabah	alamat	no_telepon	no_rekening	kode_cabang	pin	saldo
3	247182	Bunga Putri	Jalan Martapura	2147483647	247182	JKT48	091298	4000000
4	350067	Grace Sebayang	Gang Keluarga	2147483647	350067	RGS02	230987	5500000

2 rows in set (0.000 sec)

11. Dengan menggunakan salah satu dari Jenis Join di atas, tampilkan data rekening dan nasabah ketika kode cabang “KKM” dan saldo <1000000.

Untuk menampilkan data rekening dan nasabah dengan ketentuan ‘saldo < 1000000 dan kode cabang “KKM”’ kita menggunakan inner join dengan perintah ‘Select kolom from <nama_tabel> inner join <nama_tabel> ON <nama_tabel.nama_kolom> = <nama_tabel.nama_kolom> where <nama_tabel.nama_kolom> <kondisi kasus> AND <nama_tabel.nama_kolom> LIKE <kondisi kasus%>’. Like digunakan untuk mencocokkan pola pada nilai kolom. Kunci join dalam kasus ini adalah kolom no_rekening.

```
MariaDB [bank_mino]> SELECT *
-> FROM rekening
-> INNER JOIN nasabah ON rekening.no_rekening = nasabah.no_rekening
-> WHERE rekening.kode_cabang LIKE 'KKM%' AND rekening.saldo < 1000000;
```

no_rekening	kode_cabang	pin	saldo	id_nasabah	no_rekening	nama_nasabah	alamat	no_telepon
256132	KKM70	340987	230000	2	256132	Sinta Jojo	Iskandar Muda	2147483647
100111	KKM69	213671	998500	6	100111	Rimayka Natalia	Perumahan Puri K	2147483647

2 rows in set (0.000 sec)

12. Dengan menggunakan salah satu dari Jenis Join di atas, tampilkan nama nasabah, alamat, nomor telepon, nomor rekening dan kode cabang ketika alamat nasabah “Se”.

Untuk menampilkan data nama nasabah, alamat, nomor telepon, nomor rekening dan kode cabang dengan ketentuan alamat nasabah ‘se’ kita menggunakan inner join dengan perintah ‘Select kolom from <nama_tabel> inner join <nama_tabel> ON <nama_tabel.nama_kolom> = <nama_tabel.nama_kolom> where <nama_tabel.nama_kolom> LIKE <kondisi kasus%>’. Like digunakan untuk mencocokkan pola pada nilai kolom. Kunci join dalam kasus ini adalah kolom no rekening.

```
MariaDB [bank_mino]> select nasabah.nama_nasabah, nasabah.alamat, nasabah.no_telepon, rekening.kode_cabang
-> from nasabah inner join rekening on nasabah.no_rekening = rekening.no_rekening
-> where alamat like "Se%";
```

nama_nasabah	alamat	no_telepon	kode_cabang
Puspa Sari	Sei Batang Kuis	2147483647	SGS02
Christine Talia	Sei Kambah	2147483647	SGS01

2 rows in set (0.000 sec)

13. Dengan menggunakan salah satu dari Jenis Join di atas, tampilkan nomor rekening, saldo, jumlah transaksi , tanggal transaksi ketika nomor rekening sama dengan 145678.

Untuk menampilkan nomor rekening, saldo, jumlah transaksi dan tanggal transaksi dengan ketentuan ‘nomor rekening = 145678’ kita menggunakan inner join dengan perintah ‘Select kolom from <nama_tabel> inner join <nama_tabel> ON <nama_tabel.nama_kolom> = <nama_tabel.nama_kolom> where <nama_tabel.nama_kolom> = kondisi kasus. Kunci join dalam kasus ini adalah kolom no_rekening.

```
MariaDB [bank_mino]> select rekening.no_rekening, rekening.saldo, transaksi.jumlah, transaksi.tgl_transaksi
-> from rekening inner join transaksi on rekening.no_rekening = transaksi.no_rekening
-> where rekening.no_rekening = 145678;
```

no_rekening	saldo	jumlah	tgl_transaksi
145678	1000000	250000	2017- 06-15 23:11:22
145678	1000000	300000	2017- 06-18 15:30:08

2 rows in set (0.000 sec)

14. Dengan menggunakan salah satu dari Jenis Join di atas, tampilkan data rekening dan transaksi ketika nomor rekening “1” dan jenis transaksi menggunakan debet.

Untuk menampilkan data rekening dan transaksi dengan ketentuan ‘nomor rekening > 1 dan jenis transaksi = debet’ kita menggunakan inner join dengan perintah ‘Select kolom from <nama_tabel> inner join <nama_tabel> ON <nama_tabel.nama_kolom> = <nama_tabel.nama_kolom> where <nama_tabel.nama_kolom> <kondisi kasus> AND <nama_tabel.nama_kolom> <kondisi kasus>. Kunci join dalam kasus ini adalah kolom no rekening..

```
MariaDB [bank_mino]> select * from rekening inner join transaksi
-> on rekening.no_rekening = transaksi.no_rekening
-> where rekening.no_rekening like "1%" AND transaksi.jenis_transaksi = "DEBET";
```

no_rekening	kode_cabang	pin	saldo	no_transaksi	no_rekening	jenis_transaksi	tgl_transaksi	jumlah
114115	SGS01	203010	500000	13	114115	DEBET	2017-06-11 13:00:30	50000
145678	SGS02	908723	1000000	16	145678	DEBET	2017- 06-18 15:30:08	300000

2 rows in set (0.000 sec)

15. Dengan menggunakan salah satu dari jenis join di atas, tampilkan nama nasabah, alamat, nomor telepon, nomor rekening dan kode cabang ketika nomor rekening kode cabang “SGS” dan nomor telepon “0812”

Untuk menampilkan data nama nasabah, alamat, nomor telepon, nomor rekening dan kode cabang dengan ketentuan kode cabang ‘SGS’ dan nomor telepon ‘0812’ kita menggunakan inner join dengan perintah ‘Select kolom from <nama_tabel> inner join <nama_tabel> ON <nama_tabel.nama_kolom> = <nama_tabel.nama_kolom> AND <nama_tabel.nama_kolom> LIKE <kondisi kasus%> WHERE <nama_tabel.nama_kolom> LIKE <kondisi kasus%>’. Like digunakan untuk mencocokkan pola pada nilai kolom. Kunci join dalam kasus ini adalah kolom no rekening.

```
MariaDB [bank_mino]> select nasabah.nama_nasabah, nasabah.alamat, nasabah.no_telepon, rekening.no_rekening, rekening.kode_cabang
-> from nasabah cross join rekening
-> on nasabah.no_rekening = rekening.no_rekening
-> where rekening.kode_cabang like "SGS%" AND nasabah.no_telepon like "0812%";
Empty set (0.000 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