

# **LAPORAN AKHIR PRAKTIKUM**

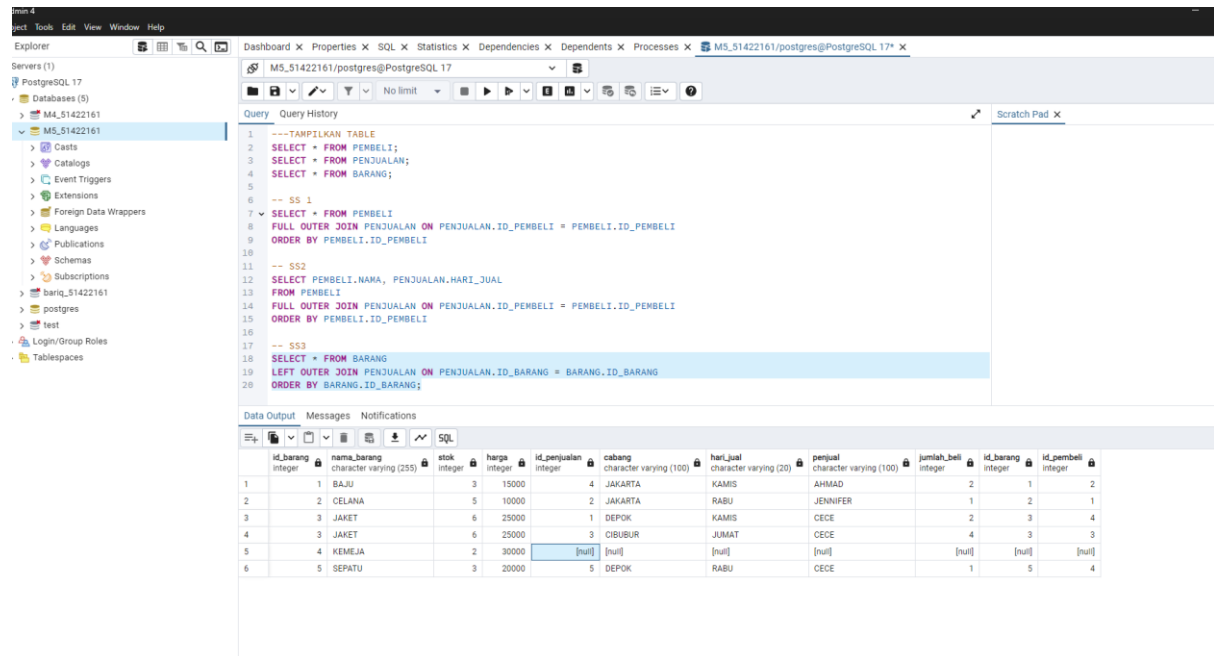
Mata Praktikum : SISTEM BASIS DATA 1  
Kelas : 3IA11  
Praktikum ke- : 6  
Tanggal : 6/12/2024  
Materi : OUTER JOIN  
NPM : 51422161  
Nama : MUHAMMAD TARMIDZI BARIQ  
Ketua Asisten : SEBASTIAN DWI  
Jumlah Lembar : 3



**LABORATORIUM TEKNIK INFORMATIKA**  
**UNIVERSITAS GUNADARMA**  
**2024**

Jelaskan kembali perbedaan left outer join dan right outer join pada kasus yang terdapat di activity

## LEFT JOIN



The screenshot shows a PostgreSQL query editor with a query window and a data output window. The query window contains the following SQL code:

```
1 ---TAMPILKAN TABLE
2 SELECT * FROM PENBELI;
3 SELECT * FROM PENJUALAN;
4 SELECT * FROM BARANG;
5
6 -- SS 1
7 SELECT * FROM PENBELI
8 FULL OUTER JOIN PENJUALAN ON PENJUALAN.ID_PEMBELI = PENBELI.ID_PEMBELI
9 ORDER BY PENBELI.ID_PEMBELI
10
11 -- SS2
12 SELECT PENBELI.NAMA, PENJUALAN.HARI_JUAL
13 FROM PENBELI
14 FULL OUTER JOIN PENJUALAN ON PENJUALAN.ID_PEMBELI = PENBELI.ID_PEMBELI
15 ORDER BY PENBELI.ID_PEMBELI
16
17 -- SS3
18 SELECT * FROM BARANG
19 LEFT OUTER JOIN PENJUALAN ON PENJUALAN.ID_BARANG = BARANG.ID_BARANG
20 ORDER BY BARANG.ID_BARANG;
```

The data output window shows the results of the query. The table has 10 columns: `id_barang`, `nama_barang`, `stok`, `harga`, `id_penjualan`, `cabang`, `hari_jual`, `penjual`, `jumlah_beli`, `id_barang`, and `id_pembeli`. The results are as follows:

| id_barang | nama_barang | stok | harga | id_penjualan | cabang  | hari_jual | penjual  | jumlah_beli | id_barang | id_pembeli |
|-----------|-------------|------|-------|--------------|---------|-----------|----------|-------------|-----------|------------|
| 1         | BAJU        | 3    | 15000 | 4            | JAKARTA | KAMIS     | AHMAD    | 2           | 1         | 2          |
| 2         | CELANA      | 5    | 10000 | 2            | JAKARTA | RABU      | JENNIFER | 1           | 2         | 1          |
| 3         | JAKET       | 6    | 25000 | 1            | DEPOK   | KAMIS     | CECE     | 2           | 3         | 4          |
| 4         | JAKET       | 6    | 25000 | 3            | CIBUBUR | JUMAT     | CECE     | 4           | 3         | 3          |
| 5         | KEMEJA      | 2    | 30000 | [null]       | [null]  | [null]    | [null]   | [null]      | [null]    | [null]     |
| 6         | SEPATU      | 3    | 20000 | 5            | DEPOK   | RABU      | CECE     | 1           | 5         | 4          |

## SELECT \* FROM BARANG

- Memilih semua kolom dari table BARANG

## LEFT OUTER JOIN PENJUALAN ON PENJUALAN.ID\_BARANG = BARANG.ID\_BARANG

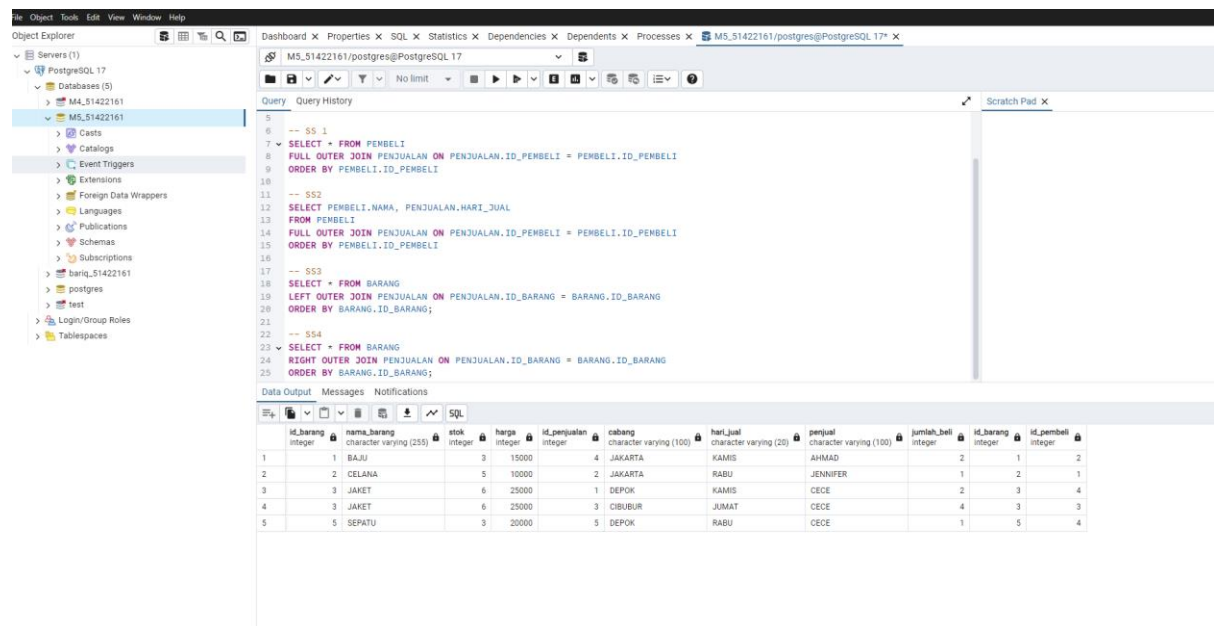
- Melakukan LEFT JOIN pada tabel PENJUALAN dengan kondisi join `PENJUALAN.ID_BARANG = BARANG.ID_BARANG`. Artinya, semua baris dari tabel BARANG akan ditampilkan, dan baris yang cocok dari tabel PENJUALAN akan disertakan. Jika tidak ada kecocokan, kolom dari tabel PENJUALAN akan berisi nilai NULL.

## ORDER BY BARANG.ID\_BARANG;

- Mengurutkan hasil berdasarkan kolom ID\_BARANG dari tabel BARANG.

Hasil dari query ini akan mengembalikan semua baris dari tabel BARANG dan baris yang cocok dari tabel PENJUALAN. Jika tidak ada kecocokan pada kolom ID\_BARANG, hasilnya akan berisi nilai NULL dari tabel PENJUALAN.

## RIGHT JOIN



The screenshot shows a PostgreSQL IDE interface. The left pane displays the database structure, including tables like 'PENJUALAN' and 'BARANG'. The main pane shows a SQL query with three statements: a SELECT from 'PEMBELI' using a FULL OUTER JOIN, a SELECT from 'PEMBELI' using a FULL OUTER JOIN, and a SELECT from 'BARANG' using a RIGHT OUTER JOIN. The bottom pane displays the results of the query, showing columns from 'BARANG' and 'PENJUALAN'.

```
5 -- SS 1
6
7 SELECT * FROM PEMBELI
8 FULL OUTER JOIN PENJUALAN ON PENJUALAN.ID_PEMBELI = PEMBELI.ID_PEMBELI
9 ORDER BY PEMBELI.ID_PEMBELI
10
11 -- SS2
12 SELECT PEMBELI.NAMA, PENJUALAN.HARI_JUAL
13 FROM PEMBELI
14 FULL OUTER JOIN PENJUALAN ON PENJUALAN.ID_PEMBELI = PEMBELI.ID_PEMBELI
15 ORDER BY PEMBELI.ID_PEMBELI
16
17 -- SS3
18 SELECT * FROM BARANG
19 LEFT OUTER JOIN PENJUALAN ON PENJUALAN.ID_BARANG = BARANG.ID_BARANG
20 ORDER BY BARANG.ID_BARANG;
21
22 -- SS4
23 SELECT * FROM BARANG
24 RIGHT OUTER JOIN PENJUALAN ON PENJUALAN.ID_BARANG = BARANG.ID_BARANG
25 ORDER BY BARANG.ID_BARANG;
```

| id_barang | nama_barang | stok | harga | id_penjualan | cabang  | hari_jual | penjual  | jumlah_beli | id_barang | id_pembeli |
|-----------|-------------|------|-------|--------------|---------|-----------|----------|-------------|-----------|------------|
| 1         | BAJU        | 3    | 15000 | 4            | JAKARTA | KAMIS     | AHMAD    | 2           | 1         | 2          |
| 2         | CELANA      | 5    | 10000 | 2            | JAKARTA | RABU      | JENNIFER | 1           | 2         | 1          |
| 3         | JAKET       | 6    | 25000 | 1            | DEPOK   | KAMIS     | CECE     | 2           | 3         | 4          |
| 4         | JAKET       | 6    | 25000 | 3            | CIBUBUR | JUMAT     | CECE     | 4           | 3         | 3          |
| 5         | SEPATU      | 3    | 20000 | 5            | DEPOK   | RABU      | CECE     | 1           | 5         | 4          |

## SELECT \* FROM BARANG

- Memilih semua kolom dari table BARANG

## RIGHT OUTER JOIN PENJUALAN ON PENJUALAN.ID\_BARANG = BARANG.ID\_BARANG

- Melakukan RIGHT JOIN pada tabel PENJUALAN dengan kondisi join `PENJUALAN.ID_BARANG = BARANG.ID_BARANG`. Artinya, semua baris dari tabel PENJUALAN akan ditampilkan, dan baris yang cocok dari tabel BARANG akan disertakan. Jika tidak ada kecocokan, kolom dari tabel BARANG tidak akan ditampilkan

## ORDER BY BARANG.ID\_BARANG;

- Mengurutkan hasil berdasarkan kolom ID\_BARANG dari tabel BARANG.