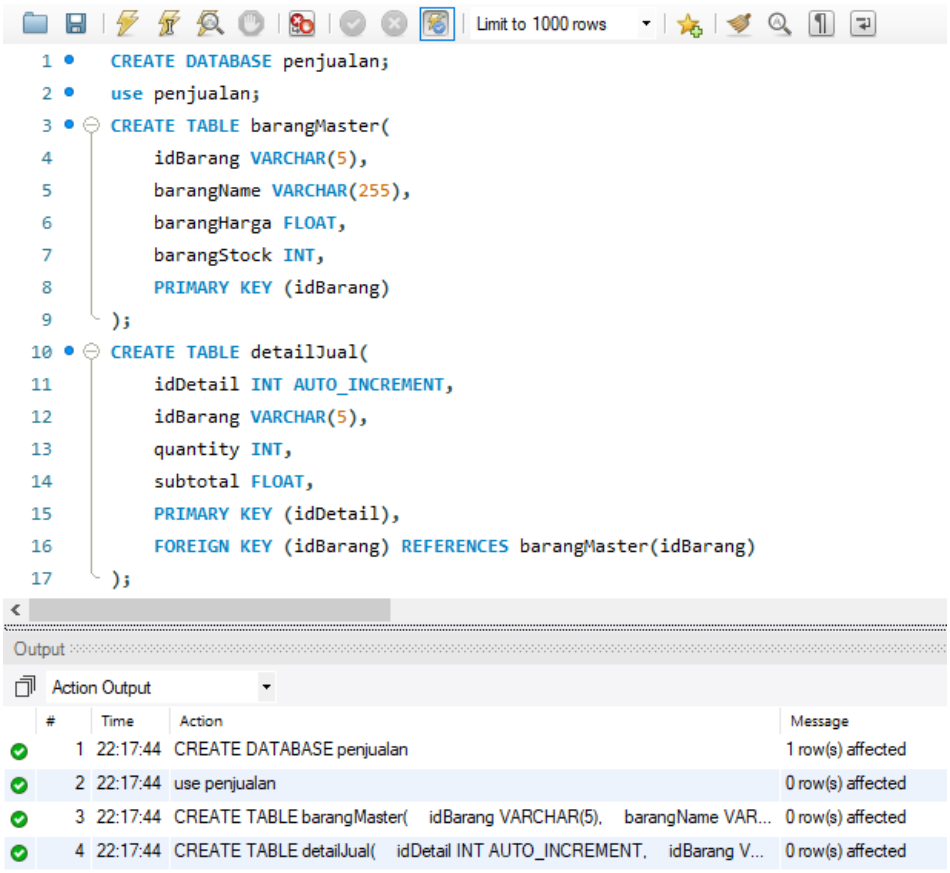
Nama : Ade Hikmat Pauji Ridwan

NPM : 22552011130

1. DDL
   a. Create DB use DB and Create table



2. DML
   a. Insert Data

```
Limit to 1000 rows
3   VALUES ('B0001', 'Baju', 100000, 10),
4       ('B0002', 'Celana Panjang', 200000, 20),
5       ('B0003', 'Celana Pendek', 150000, 30),
6       ('B0004', 'Kemeja', 125000, 25),
7       ('B0005', 'Jaket', 400000, 15),
8       ('B0006', 'Sepatu', 500000, 5);
9   INSERT INTO detailJual (idBarang, quantity, subtotal)
10  VALUES ('B0001', 2, 200000),
11      ('B0002', 3, 600000),
12      ('B0003', 4, 600000),
13      ('B0001', 1, 100000),
14      ('B0002', 2, 400000),
15      ('B0004', 3, 375000),
16      ('B0005', 1, 400000),
17      ('B0006', 2, 1000000),
18      ('B0002', 3, 600000),
19      ('B0003', 3, 450000);
```

```
Output
Action Output
#  Time      Action                                                    Message
1  22:24:52  use penjualan                                             0 row(s) affected
2  22:24:52  INSERT INTO barangMaster (idBarang, barangName, barangHarga, barangSto...  6 row(s) affected Records: 6  Duplicates: 0
3  22:24:52  INSERT INTO detailJual (idBarang, quantity, subtotal) VALUES ('B0001', 2, 200...  10 row(s) affected Records: 10  Duplicates:
```

1. Buat Query untuk menampilkan seluruh data master barang

```
Limit to 10
1  use penjualan;
2  SELECT *
3      FROM barangMaster;
```

Result Grid | Filter Rows: | Edit:

idBarang	barangName	barangHarga	barangStock
B0001	Baju	100000	10
B0002	Celana Panjang	200000	20
B0003	Celana Pendek	150000	30
B0004	Kemeja	125000	25
B0005	Jaket	400000	15
B0006	Sepatu	500000	5

arangMaster 1 ×

2. Buat Query untuk menampilkan kolom barangName dan barangHarga

```
1  use penjualan;
2  SELECT barangName,
3      barangHarga
4  from barangMaster;
```

Result Grid | Filter Rows:

barangName	barangHarga
Baju	100000
Celana Panjang	200000
Celana Pendek	150000
Kemeja	125000
Jaket	400000
Sepatu	500000

3. Buat Query untuk menampilkan data barang yang stocknya dibawah 20

```
1  use penjualan;
2  SELECT *
3  FROM barangMaster
4  WHERE barangStock < 20;
```

Result Grid | Filter Rows: | Edit:

idBarang	barangName	barangHarga	barangStock
B0001	Baju	100000	10
B0005	Jaket	400000	15
B0006	Sepatu	500000	5
NULL	NULL	NULL	NULL

4. Buat Query untuk menampilkan semua jenis celana

```
1  use penjualan;
2  SELECT *
3  FROM barangMaster
4  WHERE barangName LIKE "Celana%";
```

Result Grid | Filter Rows: | Edit:

idBarang	barangName	barangHarga	barangStock
B0002	Celana Panjang	200000	20
B0003	Celana Pendek	150000	30
NULL	NULL	NULL	NULL

5. Buat Query untuk menambahkan barang baru pada barangMaster

```
use penjualan;
INSERT INTO barangMaster (idBarang, barangName, barangHarga, barangStock)
VALUES ('B0007', 'Kaus Kaki', 30000, 10);
SELECT * FROM barangMaster WHERE idBarang = "B0007";
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Con

idBarang	barangName	barangHarga	barangStock
B0007	Kaus Kaki	30000	10
NULL	NULL	NULL	NULL

6. Buat Query untuk menampilkan barang yang stocknya antara 20 sampai 30

```
use penjualan;
SELECT *
FROM barangMaster
WHERE barangStock BETWEEN 20 AND 30;
```

Result Grid | Filter Rows: | Edit:

idBarang	barangName	barangHarga	barangStock
B0002	Celana Panjang	200000	20
B0003	Celana Pendek	150000	30
B0004	Kemeja	125000	25
NULL	NULL	NULL	NULL

7. Buat Query untuk mengurutkan harga barang dari yang paling mahal sampai paling murah

```
use penjualan;
SELECT *
FROM barangMaster
ORDER BY barangHarga DESC;
```

Result Grid | Filter Rows: | Edit:

idBarang	barangName	barangHarga	barangStock
B0006	Sepatu	500000	5
B0005	Jaket	400000	15
B0002	Celana Panjang	200000	20
B0003	Celana Pendek	150000	30
B0004	Kemeja	125000	25
B0001	Baju	100000	10
B0007	Kaus Kaki	30000	10
NULL	NULL	NULL	NULL

8. Buat Query untuk menentukan barang yang harganya antara 100000 sampai 200000

```
1  use penjualan;
2  SELECT *
3  FROM barangMaster
4  WHERE barangHarga BETWEEN 100000 AND 200000;
```

Result Grid | Filter Rows: | Edit:

idBarang	barangName	barangHarga	barangStock
B0001	Baju	100000	10
B0002	Celana Panjang	200000	20
B0003	Celana Pendek	150000	30
B0004	Kemeja	125000	25
NULL	NULL	NULL	NULL

9. Buat Query untuk menghitung jumlah data barang master

```
1  use penjualan;
2  SELECT COUNT(idBarang)
3  FROM barangMaster;
```

Result Grid | Filter Rows:

COUNT(idBarang)
7

10. Buat Query untuk menampilkan harga barang 400000 dan 500000

```
1  use penjualan;
2  SELECT *
3  FROM barangMaster
4  WHERE barangHarga IN(400000, 500000);
```

Result Grid | Filter Rows: | Edit:

idBarang	barangName	barangHarga	barangStock
B0005	Jaket	400000	15
B0006	Sepatu	500000	5
NULL	NULL	NULL	NULL

11. Buat Query untuk menentukan rata2 harga barang

```
use penjualan;
SELECT AVG(barangHarga) as average
from barangMaster;
```

Result Grid | Filter Rows: | Exp

average
215000

12. Buat Query untuk menampilkan detiljual kolom idbarang dan subtotal

```
use penjualan;
SELECT idBarang,
    subtotal
from detailJual;
```

Result Grid | Filter Rows:

idBarang	subtotal
B0001	200000
B0002	600000
B0003	600000

13. Buat Query untuk menambahkan detiljual yang baru

```
use penjualan;
INSERT INTO detailJual (idBarang, quantity, subtotal)
VALUES ('B0001', 3, 300000);
select * from detailJual order by idDetail DESC limit 1;

```

Limit to 1000 rows

Result Grid | Filter Rows: | Edit: | Export/Impo

idDetail	idBarang	quantity	subtotal
13	B0001	3	300000

14. Buat Query untuk menampilkan detiljual dengan quantity dibawah 3

```
use penjualan;
SELECT *
FROM detailJual
WHERE quantity < 3;
```

Result Grid | Filter Rows:

idDetail	idBarang	quantity	subtotal
1	B0001	2	200000
4	B0001	1	100000
5	B0002	2	400000
7	B0005	1	400000
8	B0006	2	1000000

15. Buat Query untuk menampilkan detiljual dengan quantity antara 2 dan 4

```
use penjualan;
SELECT *
FROM detailJual
WHERE quantity BETWEEN 2 AND 4;
```

Result Grid | Filter Rows:

idDetail	idBarang	quantity	subtotal
1	B0001	2	200000
2	B0002	3	600000
3	B0003	4	600000
5	B0002	2	400000
6	B0004	3	375000
8	B0006	2	1000000

16. Buat Query untuk nemapilkan segala jenis subtotal dengan subtotal 600000 atau 400000

```
use penjualan;
SELECT *
FROM detailJual
WHERE subtotal IN(600000,400000)
```

Result Grid | Filter Rows:

idDetail	idBarang	quantity	subtotal
2	B0002	3	600000
3	B0003	4	600000
5	B0002	2	400000
7	B0005	1	400000
9	B0002	3	600000
NULL	NULL	NULL	NULL

400000

17. Buat Query untuk menghitung jumlah pembelian untuk jenis barang B0002

```
use penjualan;
SELECT COUNT(idBarang)
FROM detailJual
WHERE idBarang = "B0002";
```

Result Grid | Filter Rows:

COUNT(idBarang)
3

18. Buat Query untuk menampilkan detiljual urutkan idbarang, quantity dan berikutnya subtotal dari yang paling kecil

```
SELECT idBarang,
    quantity,
    subtotal
FROM detailJual
ORDER BY subtotal ASC;
```

Result Grid | Filter Rows:

idBarang	quantity	subtotal
B0001	1	100000
B0001	2	200000
B0001	3	300000
B0004	3	375000
B0002	2	400000

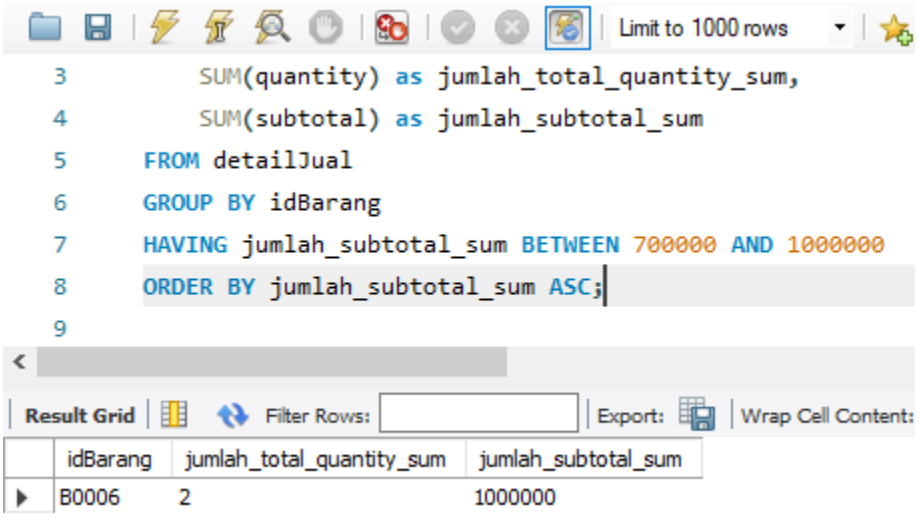
19. Buat Query untuk menampilan group by barangID , jumlah total quantity dan jumlah total sum

```
SELECT idBarang,
    SUM(quantity) as jumlah_total_quantity_sum,
    SUM(subtotal) as jumlah_subtotal_sum
FROM detailJual
GROUP BY idBarang;
```

Result Grid | Filter Rows: | Export: | Wrap C

idBarang	jumlah_total_quantity	jumlah_total
B0001	6	600000
B0002	8	1600000
B0003	7	1050000
B0004	3	375000
B0005	1	400000

20. Buat Query Groupby dan Having menggunakan column detiljual dan jelaskan



```
3        SUM(quantity) as jumlah_total_quantity_sum,
4        SUM(subtotal) as jumlah_subtotal_sum
5    FROM detailJual
6    GROUP BY idBarang
7    HAVING jumlah_subtotal_sum BETWEEN 700000 AND 1000000
8    ORDER BY jumlah_subtotal_sum ASC;
9
```

Limit to 1000 rows

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

idBarang	jumlah_total_quantity_sum	jumlah_subtotal_sum
B0006	2	1000000

Query HAVING ini sifatnya sama seperti kondisi WHARE dan hanya bisa digunakan bersamaan dengan query GROUP BY.