

## **TUGAS**

### **MODUL 3 ORDER BY, FUNGSI AGREGASI, GROUP BY, HAVING**

Disusun Guna Memenuhi Tugas Mata Kuliah

Basis Data

Dosen pengampu:

**Adam Bachtiar, S.Kom, M.MT**



Oleh:

**NAMA : HALIMATUS SA'DIAH**

**NIM : (23241095)**

**PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI FAKULTAS SAINS, TEKNOLOGI**

**DAN TERAPAN UNIVERSITAS PENDIDIKAN MANDALIKA MATARAM**

**TAHUN 2024/2025**

## PRAKTEK 1:

```
9      -- ORDER BY
10     -- Mengurutkan data dari sebuah fieldy/kolom atau hasil oleh kolom
11     -- praktek 1
12     -- ambil nama produk dan qty dari tb penjualan urutan qty
13 •   SELECT nama_produk, qty FROM tr_penjualan ORDER BY qty; --urut dari kecil kebesar
```

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

nama_produk	qty
Flash disk DQLab 32 GB	1
Tas Travel Organizer DQLab	1
Kotak Pensil DQLab	1
Gift Voucher DQLab 100rb	2
Flashdisk DQLab 64 GB	2
Flashdisk DQLab 32 GB	2
Gantungan Kunci DQLab	2
Buku Planner Agenda DQSquad	3
Flashdisk DQLab 32 GB	3
Buku Planner Agenda DQSquad	3

tr\_penjualan 1 x

## PENJELASAN:

Mengambil kolom nama produk dan jumlah (qty), lalu diurutkan dari qty terkecil ke terbesar (ascending)

## PRAKTEK 2:

```
15     -- praktek 2
16     -- ambil nama produk, qty dari tb penjualan urutan qty dan nama produk
17 •   SELECT nama_produk, qty FROM tr_penjualan ORDER BY qty, nama_produk;
```

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

nama_produk	qty
Flash disk DQLab 32 GB	1
Kotak Pensil DQLab	1
Tas Travel Organizer DQLab	1
Flashdisk DQLab 32 GB	2
Flashdisk DQLab 64 GB	2
Gantungan Kunci DQLab	2
Gift Voucher DQLab 100rb	2
Buku Planner Agenda DQSquad	3
Buku Planner Agenda DQSquad	3
Flashdisk DQLab 32 GB	3

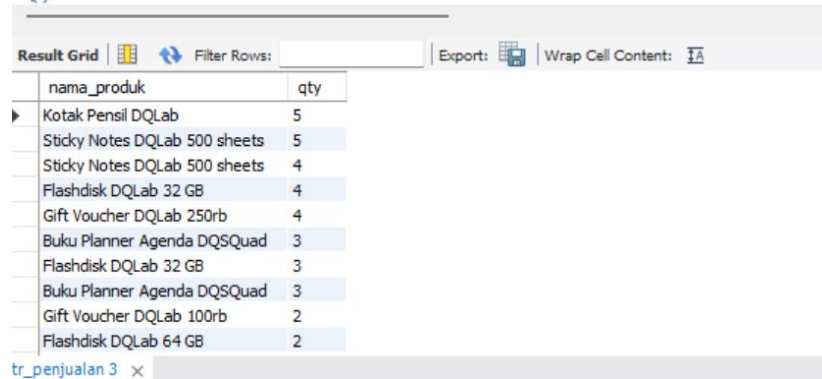
tr\_penjualan 2 x

## PENJELASAN:

Mengurutkan berdasarkan qty (kecil ke besar), lalu berdasarkan nama\_produk secara alfabet (default ASC) jika qty sama.

### PRAKTEK 3:

```
29 -- praktek 3
30 -- ambil nama produk, qty dari tb penjualanurut qty besar ke kecil
31 • SELECT nama_produk, qty from tr_penjualan ORDER BY qty DESC;_
32
```



The screenshot shows a database query result grid. The grid has two columns: 'nama\_produk' and 'qty'. The data is sorted by 'qty' in descending order. The results are as follows:

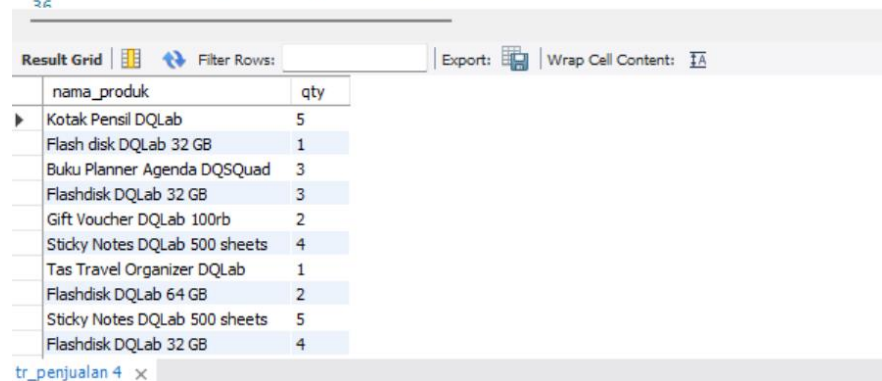
nama_produk	qty
Kotak Pensil DQLab	5
Sticky Notes DQLab 500 sheets	5
Sticky Notes DQLab 500 sheets	4
Flashdisk DQLab 32 GB	4
Gift Voucher DQLab 250rb	4
Buku Planner Agenda DQSquad	3
Flashdisk DQLab 32 GB	3
Buku Planner Agenda DQSquad	3
Gift Voucher DQLab 100rb	2
Flashdisk DQLab 64 GB	2

### PENJELASAN:

Mengurutkan berdasarkan qty dari besar ke kecil, lalu jika sama, nama\_produk secara abjad A-Z..

### PRAKTEK 4

```
33 -- praktek 4
34 -- ambil nama produk, qty dari tb penjualanurut qty desc, namaproduk asc
35 • SELECT nama_produk,qty from tr_penjualan;_
36
```



The screenshot shows a database query result grid. The grid has two columns: 'nama\_produk' and 'qty'. The data is sorted by 'qty' in descending order, and then by 'nama\_produk' in ascending order. The results are as follows:

nama_produk	qty
Kotak Pensil DQLab	5
Flash disk DQLab 32 GB	1
Buku Planner Agenda DQSquad	3
Flashdisk DQLab 32 GB	3
Gift Voucher DQLab 100rb	2
Sticky Notes DQLab 500 sheets	4
Tas Travel Organizer DQLab	1
Flashdisk DQLab 64 GB	2
Sticky Notes DQLab 500 sheets	5
Flashdisk DQLab 32 GB	4

### PENJELASAN:

Mengurutkan qty dari besar ke kecil, lalu jika sama, nama\_produk secara alfabet (default ASC)

## PRAKTEK 5:

```
47 -- praktek 5
48 -- menggunakan ORDER BY dari hasil perhitungan kolom
49 -- ambil nama produk, qty, harga dan perkalian qty*harga, urut qty*
50 • select nama_produk, qty, harga, qty*harga as total from tr_penjualan order by total desc;
51
```

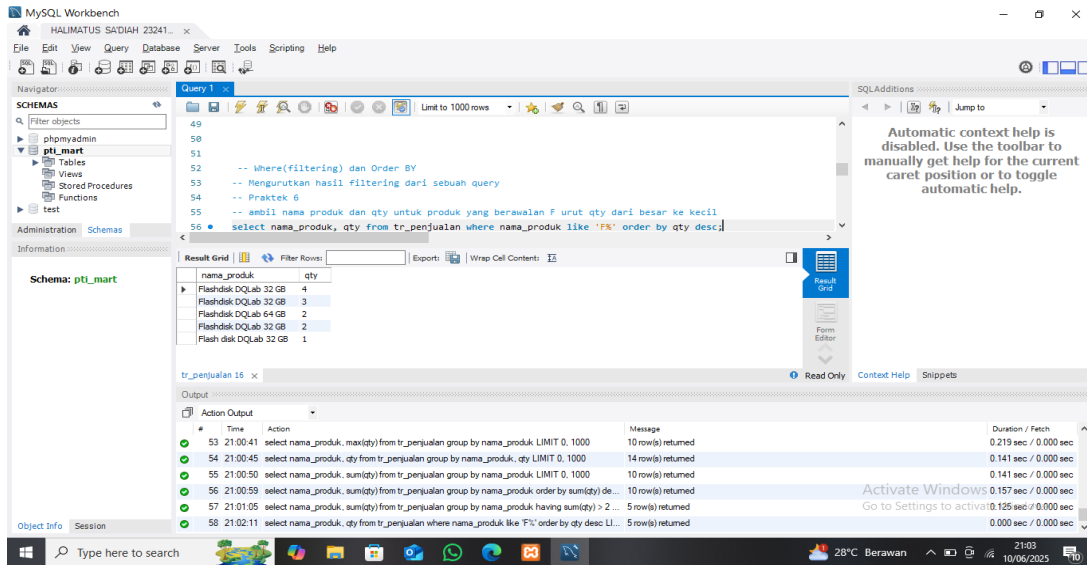
Result Grid   Filter Rows:   Export:   Wrap Cell Content:				
	nama_produk	qty	harga	total
▶	Gift Voucher DQLab 250rb	4	250000	1000000
	Kotak Pensil DQLab	5	62500	312500
	Buku Planner Agenda DQSquad	3	92000	276000
	Buku Planner Agenda DQSquad	3	92000	276000
	Sticky Notes DQLab 500 sheets	5	55000	275000
	Sticky Notes DQLab 500 sheets	4	55000	220000
	Gift Voucher DQLab 100rb	2	100000	200000
	Flashdisk DQLab 32 GB	4	40000	160000
	Flashdisk DQLab 32 GB	3	40000	120000
	Flashdisk DQLab 64 GB	2	55000	110000

Result 5 x

## PENJELASAN

Menghitung total penjualan (qty \* harga) dan mengurutkannya dari total terbesar ke terkecil.

## PRAKTEK 6

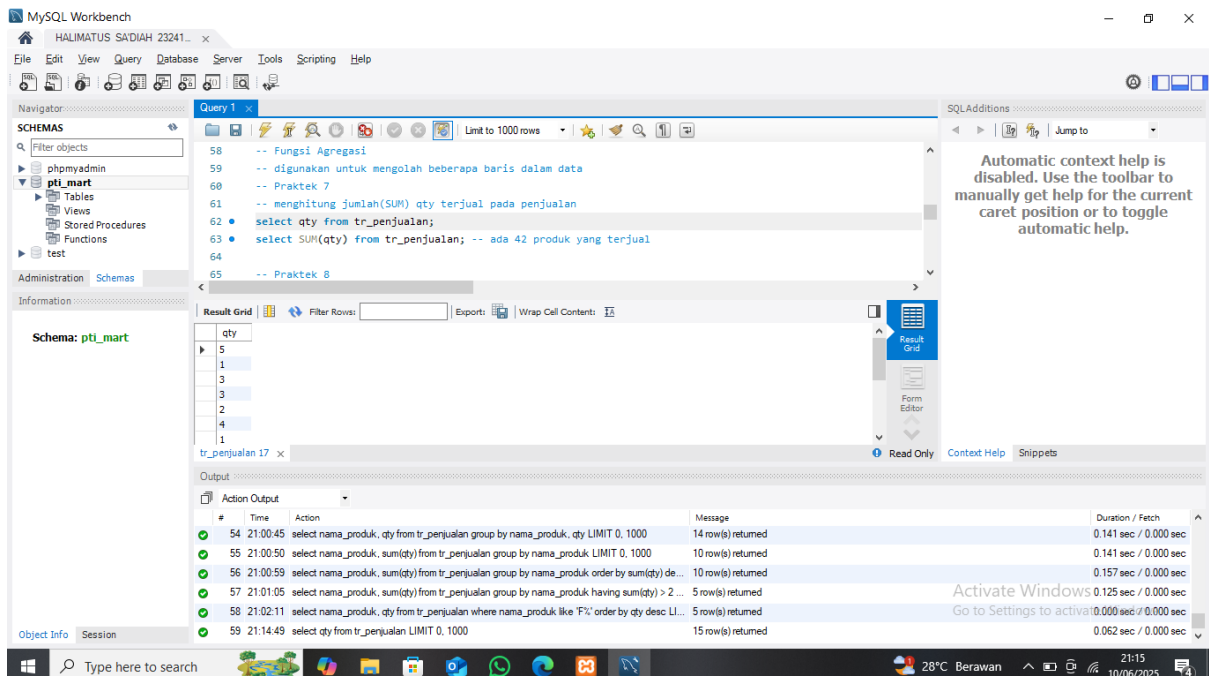


## PENJELASAN

Baris ini digunakan untuk memilih kolom nama\_produk dan qty dari tabel yang dituju (yaitu tr\_penjualan).

Artinya, hanya data nama produk dan jumlah (qty) yang akan ditampilkan.

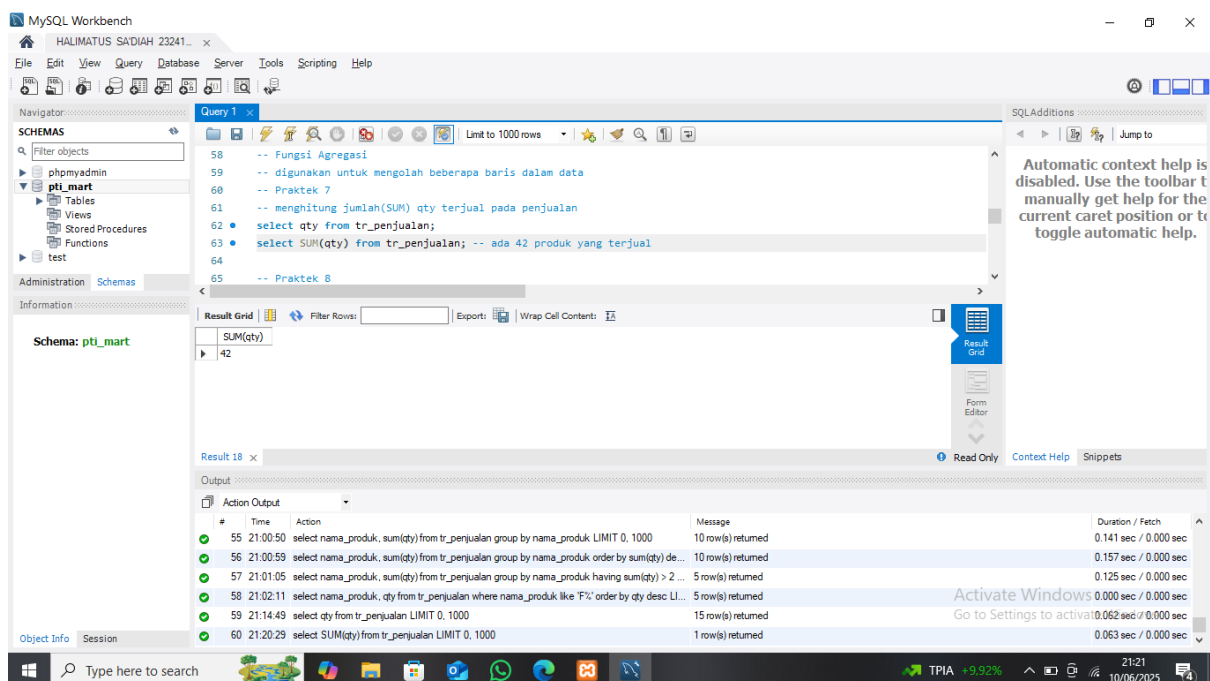
## PRAKTEK 7



## PENJELASAN

Kode ini digunakan untuk menampilkan data dari kolom qty (jumlah) yang terdapat di dalam tabel tr\_penjualan.

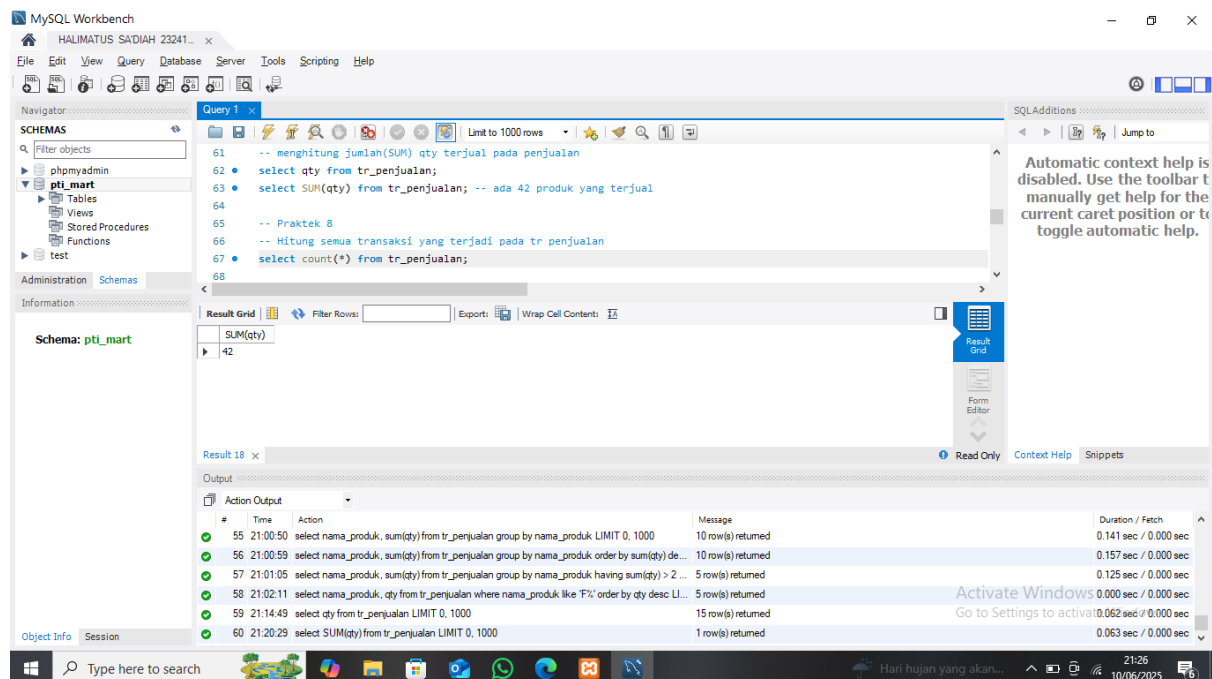
## PRAKTEK 7



## PENJELASAN

Baris kode ini digunakan untuk menghitung total jumlah (qty) dari semua data yang ada pada kolom qty di tabel tr\_penjualan.

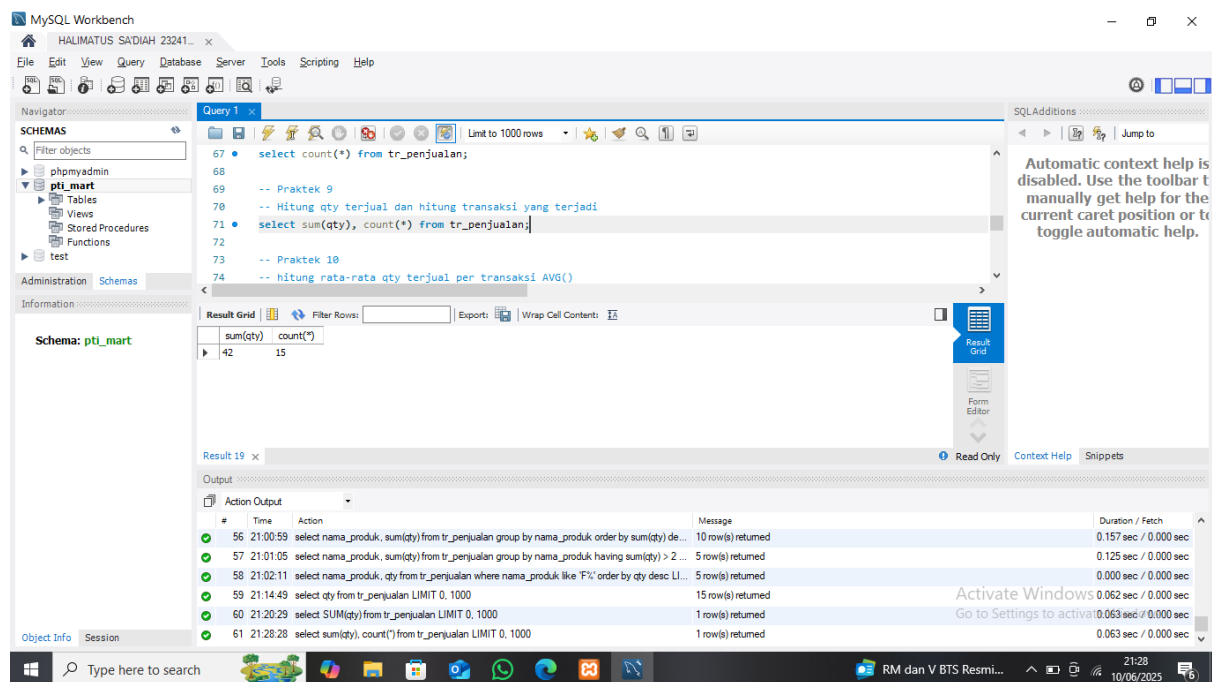
## PRAKTEK 8



## PENJELASAN

Baris kode ini digunakan untuk menghitung jumlah seluruh baris (data/record) yang terdapat dalam tabel tr\_penjualan.

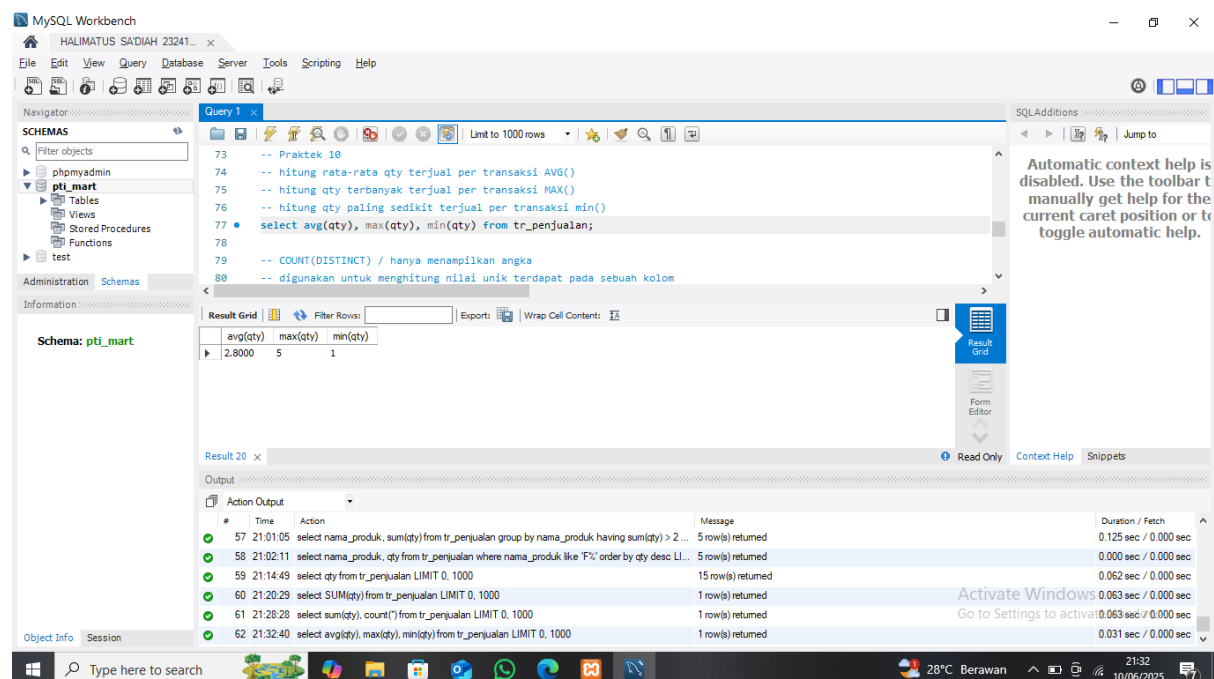
## PRAKTEK 9



## PENJELASAN

Baris kode ini digunakan untuk menampilkan dua informasi sekaligus dari tabel tr\_penjualan.

## PRAKTEK 10



The screenshot shows the MySQL Workbench interface. The 'Query' tab is active, displaying a SQL query. The 'Result Grid' shows the results of the query, which is a single row with three columns: avg(qty), max(qty), and min(qty). The values are 2.8000, 5, and 1 respectively. The 'Output' tab shows the execution log, including the query and its results.

Schema: ptj\_mart

Query 1

```
-- Praktek 10
73 -- hitung rata-rata qty terjual per transaksi AVG()
74 -- hitung qty terbanyak terjual per transaksi MAX()
75 -- hitung qty paling sedikit terjual per transaksi min()
76 select avg(qty), max(qty), min(qty) from tr_penjualan;
77
78
79 -- COUNT(DISTINCT) / hanya menampilkan angka
80 -- digunakan untuk menghitung nilai unik terdapat pada sebuah kolom
```

Result Grid

avg(qty)	max(qty)	min(qty)
2.8000	5	1

Output

#	Time	Action	Message	Duration / Fetch
57	21:01:05	select nama_produk, sum(qty) from tr_penjualan group by nama_produk having sum(qty) > 2 ...	5 row(s) returned	0.125 sec / 0.000 sec
58	21:02:11	select nama_produk, qty from tr_penjualan where nama_produk like 'F%'; order by qty desc LI...	5 row(s) returned	0.000 sec / 0.000 sec
59	21:14:49	select qty from tr_penjualan LIMIT 0, 1000	15 row(s) returned	0.062 sec / 0.000 sec
60	21:20:29	select SUM(qty) from tr_penjualan LIMIT 0, 1000	1 row(s) returned	0.063 sec / 0.000 sec
61	21:28:28	select sum(qty), count(*) from tr_penjualan LIMIT 0, 1000	1 row(s) returned	0.063 sec / 0.000 sec
62	21:32:40	select avg(qty), max(qty), min(qty) from tr_penjualan LIMIT 0, 1000	1 row(s) returned	0.031 sec / 0.000 sec

## PENJELASAN

Rata-rata (AVG), jumlah terbanyak (MAX), dan jumlah terkecil (MIN) per transaksi.



## PRAKTEK 11

The screenshot shows the MySQL Workbench interface. The 'Query 1' editor contains the following SQL code:

```
-- COUNT(DISTINCT) / hanya menampilkan angka
-- digunakan untuk menghitung nilai unik terdapat pada sebuah kolom
-- Praktek 11
select count(nama_produk) from tr_penjualan; -- ada 15
select count(distinct nama_produk) from tr_penjualan; -- ada 10 produk
-- Praktek 12
select count(nama_produk), count(distinct nama_produk) from tr_penjualan;
```

The 'Result Grid' shows the results of the first query:

count(nama_produk)
15

The 'Output' pane shows the execution log for the first query:

#	Time	Action	Message	Duration / Fetch
58	21:02:11	select nama_produk, qty from tr_penjualan where nama_produk like 'F%' order by qty desc LI...	5 row(s) returned	0.000 sec / 0.000 sec
59	21:14:49	select qty from tr_penjualan LIMIT 0, 1000	15 row(s) returned	0.062 sec / 0.000 sec
60	21:20:29	select SUM(qty) from tr_penjualan LIMIT 0, 1000	1 row(s) returned	0.063 sec / 0.000 sec
61	21:28:28	select sum(qty), count(*) from tr_penjualan LIMIT 0, 1000	1 row(s) returned	0.063 sec / 0.000 sec
62	21:32:40	select avg(qty), max(qty), min(qty) from tr_penjualan LIMIT 0, 1000	1 row(s) returned	0.031 sec / 0.000 sec
63	21:37:51	select count(nama_produk) from tr_penjualan LIMIT 0, 1000	1 row(s) returned	0.015 sec / 0.000 sec

## PENJELASAN

count(nama\_produk): menghitung total baris yang berisi nama\_produk (bisa ada yang sama).

The screenshot shows the MySQL Workbench interface. The 'Query 1' editor contains the following SQL code:

```
-- COUNT(DISTINCT) / hanya menampilkan angka
-- digunakan untuk menghitung nilai unik terdapat pada sebuah kolom
-- Praktek 11
select count(nama_produk) from tr_penjualan; -- ada 15
select count(distinct nama_produk) from tr_penjualan; -- ada 10 produk
-- Praktek 12
select count(nama_produk), count(distinct nama_produk) from tr_penjualan;
```

The 'Result Grid' shows the results of the second query:

count(distinct nama_produk)
10

The 'Output' pane shows the execution log for the second query:

#	Time	Action	Message	Duration / Fetch
59	21:14:49	select qty from tr_penjualan LIMIT 0, 1000	15 row(s) returned	0.062 sec / 0.000 sec
60	21:20:29	select SUM(qty) from tr_penjualan LIMIT 0, 1000	1 row(s) returned	0.063 sec / 0.000 sec
61	21:28:28	select sum(qty), count(*) from tr_penjualan LIMIT 0, 1000	1 row(s) returned	0.063 sec / 0.000 sec
62	21:32:40	select avg(qty), max(qty), min(qty) from tr_penjualan LIMIT 0, 1000	1 row(s) returned	0.031 sec / 0.000 sec
63	21:37:51	select count(nama_produk) from tr_penjualan LIMIT 0, 1000	1 row(s) returned	0.015 sec / 0.000 sec
64	21:38:47	select count(distinct nama_produk) from tr_penjualan LIMIT 0, 1000	1 row(s) returned	0.187 sec / 0.000 sec

## PENJELASAN

count(distinct nama\_produk): menghitung jumlah produk yang unik (tidak duplikat).

## PRAKTEK 12

The screenshot shows the MySQL Workbench interface. The 'Query 1' editor contains the following SQL code:

```
82 • select count(nama_produk) from tr_penjualan; -- ada 15
83 • select count(distinct nama_produk) from tr_penjualan; -- ada 10 produk
84
85 -- Praktek 12
86 • select count(nama_produk), count(distinct nama_produk) from tr_penjualan;
87
88 -- Praktek 13
89 -- Ambil nama produk dan hitung nilai max (terbanyak) dari qty
```

The 'Result Grid' shows the results of the query:

count(nama_produk)	count(distinct nama_produk)
15	10

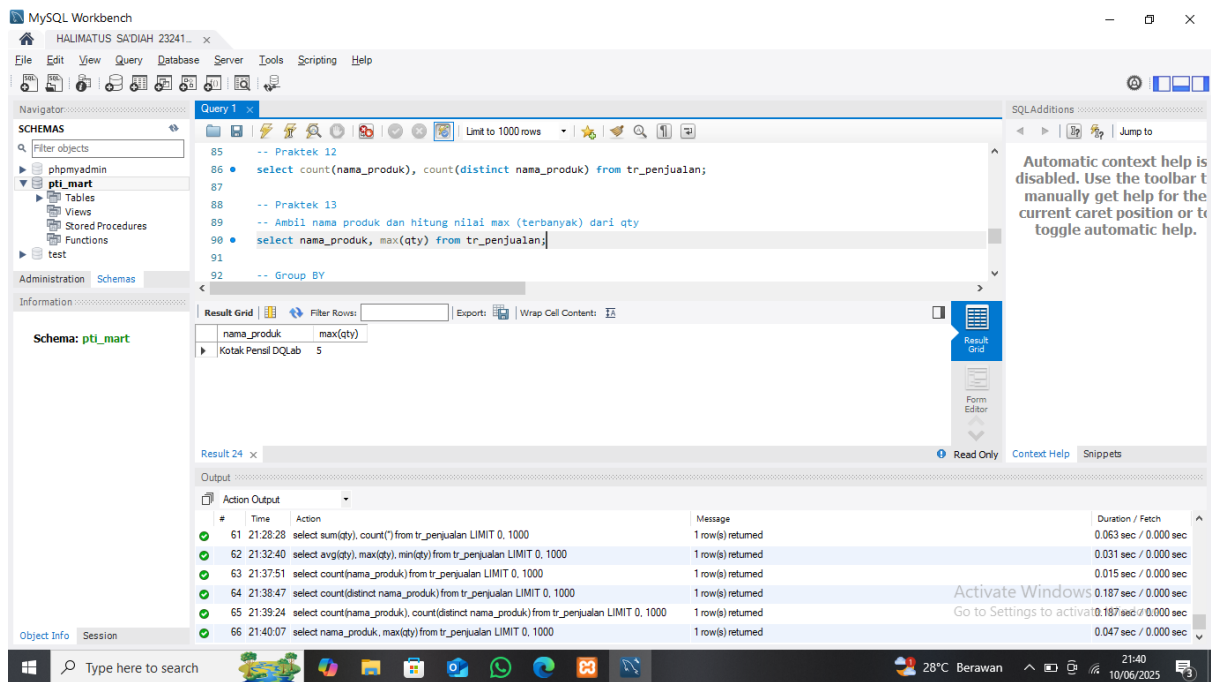
The 'Output' pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
60	21:20:29	select SUM(qty) from tr_penjualan LIMIT 0, 1000	1 row(s) returned	0.063 sec / 0.000 sec
61	21:28:28	select sum(qty), count(*) from tr_penjualan LIMIT 0, 1000	1 row(s) returned	0.063 sec / 0.000 sec
62	21:32:40	select avg(qty), max(qty), min(qty) from tr_penjualan LIMIT 0, 1000	1 row(s) returned	0.031 sec / 0.000 sec
63	21:37:51	select count(nama_produk) from tr_penjualan LIMIT 0, 1000	1 row(s) returned	0.015 sec / 0.000 sec
64	21:38:47	select count(distinct nama_produk) from tr_penjualan LIMIT 0, 1000	1 row(s) returned	0.187 sec / 0.000 sec
65	21:39:24	select count(nama_produk), count(distinct nama_produk) from tr_penjualan LIMIT 0, 1000	1 row(s) returned	0.187 sec / 0.000 sec

## PENJELASAN

Menampilkan kedua hasil dalam satu query.

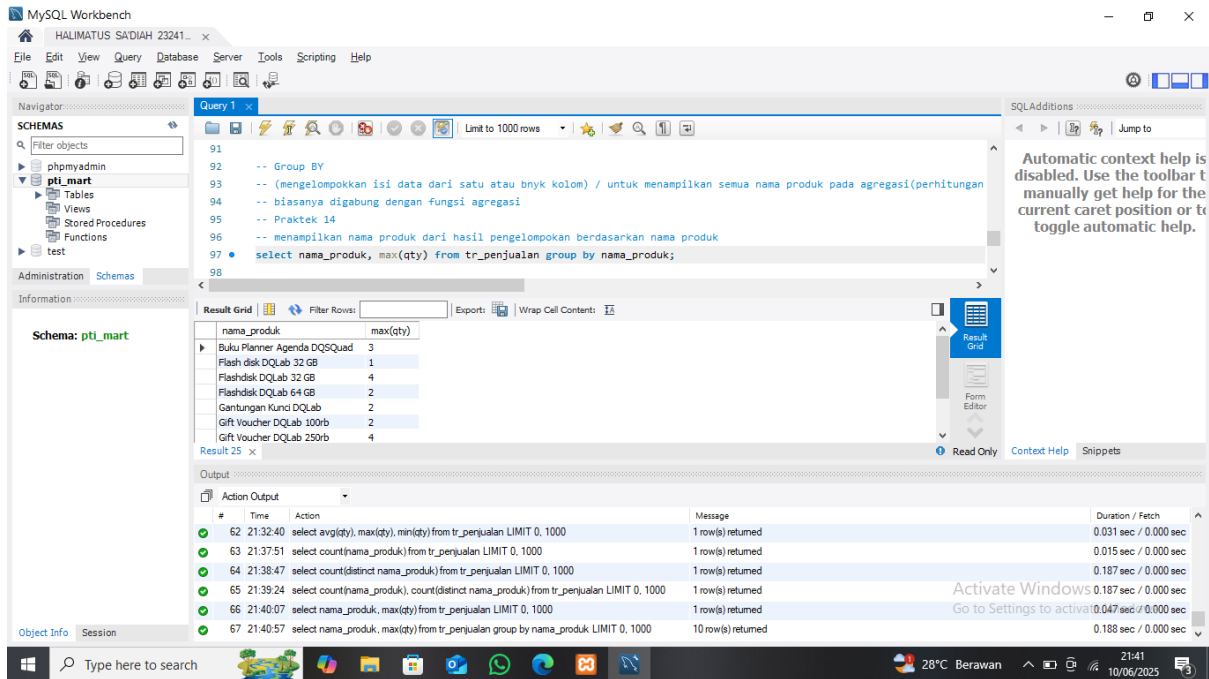
## PRAKTEK 13



## PENJELASAN

Tidak tepat secara logika karena nama\_produk tidak dikelompokkan (group by), sehingga hasilnya bisa membingungkan. Biasanya digunakan bersama GROUP BY seperti di Praktek 14.

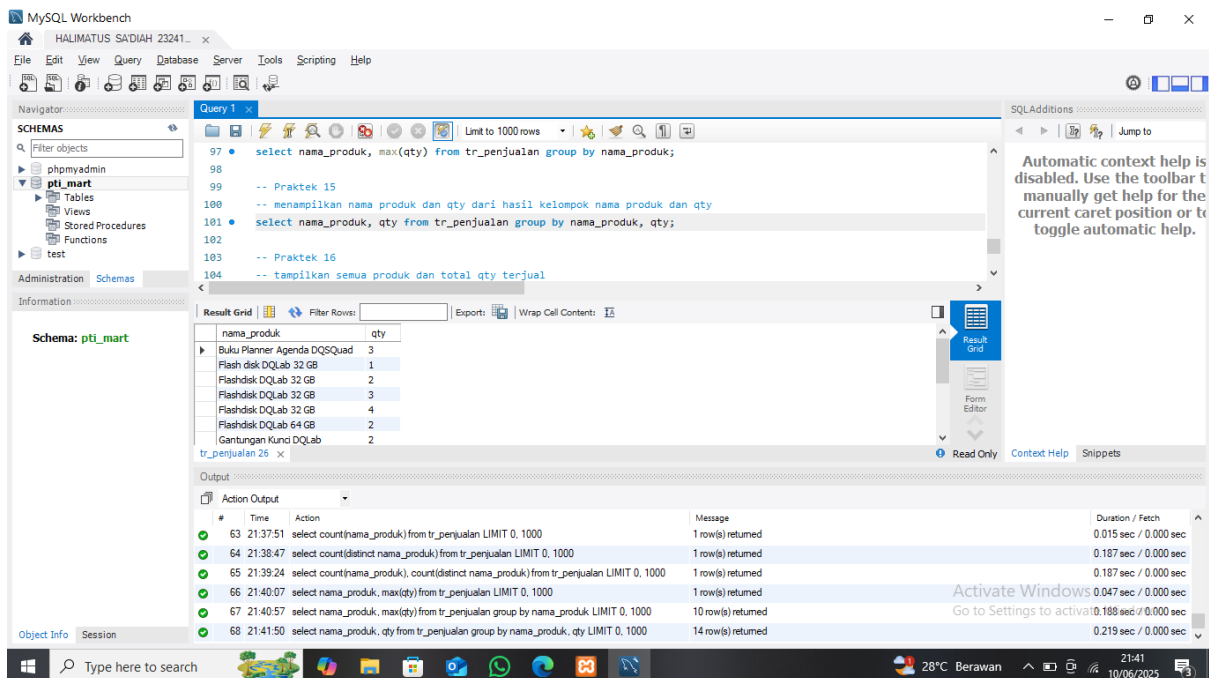
## PRAKTEK 14



## PENJELASAN

Kelompokkan berdasarkan nama\_produk, tampilkan nilai qty tertinggi untuk setiap produk.

## PRAKTEK 15



## PENJELASAN

Kelompokkan berdasarkan kombinasi nama\_produk dan qty, sering digunakan jika datanya sudah unik.

## PRAKTEK 16

The screenshot shows the MySQL Workbench interface. The 'Query' tab is active, displaying the following SQL code:

```
-- menampilkan nama produk dan qty dari hasil kelompok nama produk dan qty
100 select nama_produk, qty from tr_penjualan group by nama_produk, qty;
101
102
103 -- Praktek 16
104 -- tampilkan semua produk dan total qty terjual
105 select nama_produk, sum(qty) from tr_penjualan group by nama_produk;
106
107 -- Praktek 17
```

The 'Result Grid' shows the results of the query in the 'Result 27' tab:

nama_produk	sum(qty)
Buku Planner Agenda DQ5Squad	6
Flash disk DQLab 32 GB	1
Flashdisk DQLab 32 GB	9
Flashdisk DQLab 64 GB	2
Gantungan Kunci DQLab	2
Gift Voucher DQLab 100rb	2
Gift Voucher DQLab 250rb	4

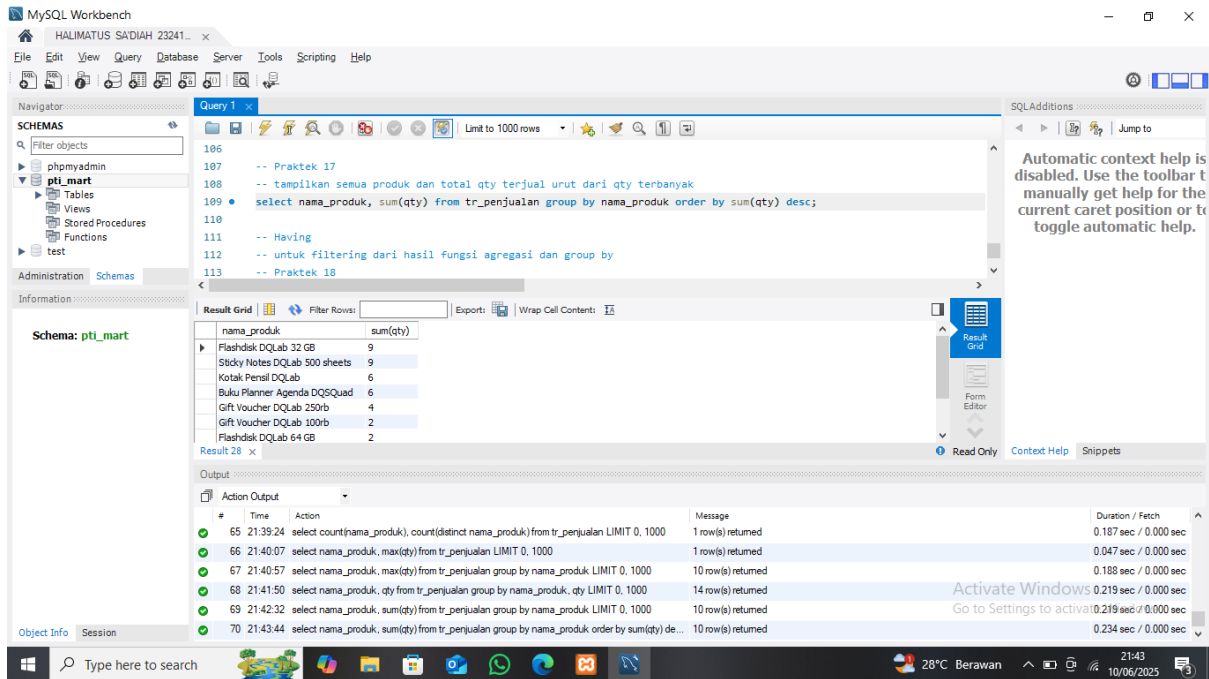
The 'Output' tab shows the execution log with the following messages:

#	Time	Action	Message	Duration / Fetch
64	21:38:47	select count(distinct nama_produk) from tr_penjualan LIMIT 0, 1000	1 row(s) returned	0.187 sec / 0.000 sec
65	21:39:24	select count(nama_produk), count(distinct nama_produk) from tr_penjualan LIMIT 0, 1000	1 row(s) returned	0.187 sec / 0.000 sec
66	21:40:07	select nama_produk, max(qty) from tr_penjualan LIMIT 0, 1000	1 row(s) returned	0.047 sec / 0.000 sec
67	21:40:57	select nama_produk, max(qty) from tr_penjualan group by nama_produk LIMIT 0, 1000	10 row(s) returned	0.188 sec / 0.000 sec
68	21:41:50	select nama_produk, qty from tr_penjualan group by nama_produk, qty LIMIT 0, 1000	14 row(s) returned	0.219 sec / 0.000 sec
69	21:42:32	select nama_produk, sum(qty) from tr_penjualan group by nama_produk LIMIT 0, 1000	10 row(s) returned	0.219 sec / 0.000 sec

## PENJELASAN

Total qty untuk masing-masing nama\_produk.

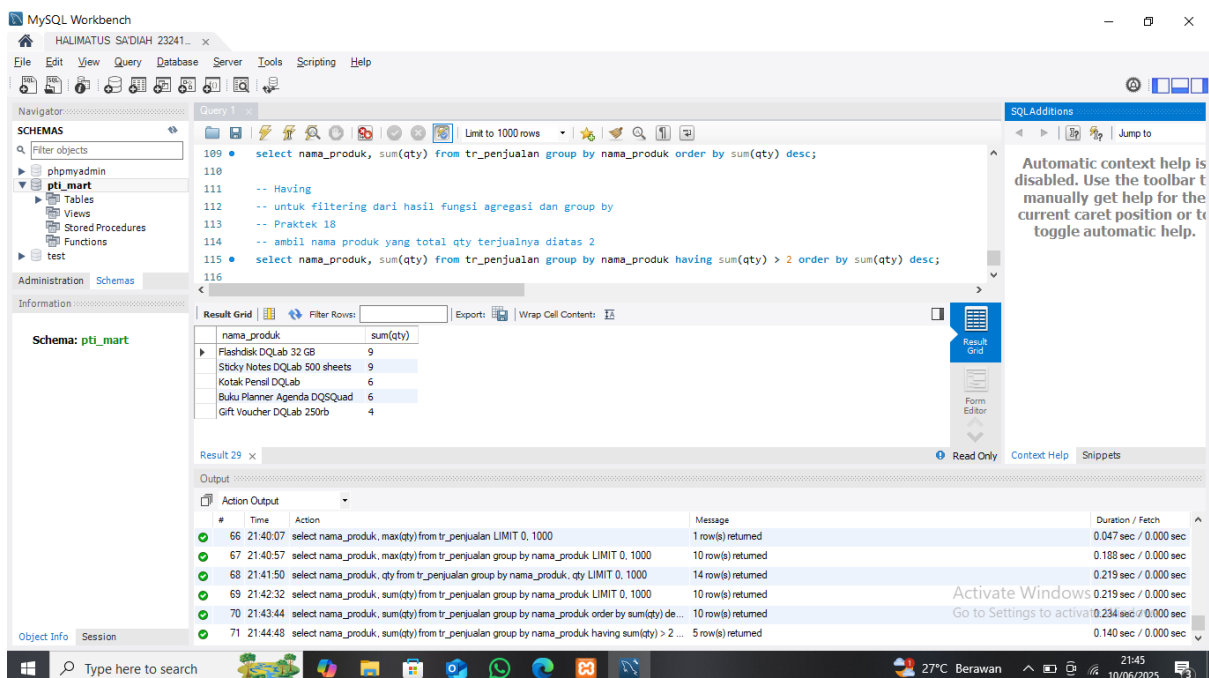
## PRAKTEK 17



## PENJELASAN

Sama seperti di atas, tapi hasilnya diurut berdasarkan total qty dari terbesar ke terkecil.

## PRAKTEK 18



## PENJELASAN

Ambil produk yang total qty-nya lebih dari 2. HAVING digunakan setelah GROUP BY karena WHERE tidak bisa digunakan untuk hasil agregasi.