

Screenshot of a web browser showing a SQL study course on Progate. The page title is "Ayo Belajar SQL Secara Praktis". A sub-section title "Mengolah Hasil Pencarian" and section title "1. DISTINCT" are visible. A sidebar message says "Klik pada panah (atau tekan tombol →) untuk ke slide berikutnya." The Windows taskbar at the bottom shows various open applications.

Screenshot of a web browser showing a continuation of the SQL study course. The page title is "Mengolah Hasil". It features a conversation between "Guru Domba" and a student. Guru Domba asks if the student wants to learn how to "mengolah" data. The student asks what "mengolah" means. Guru Domba explains it involves calculations like summing up daily expenses. To the right, there is a table titled "Mengolah Hasil" showing data from October 10 to October 12, 2018. The Windows taskbar at the bottom shows various open applications.

Total	Tanggal
3	2018-10-10
4	2018-10-11
18	2018-10-12

Mengelola Hasil Pencarian > 1. DISTINCT

Mengecualikan Data Duplikat

Jadi, saya bisa mengetahui apa saja pengeluaran saya? Kalau begitu, saya harus membuat daftar semua barang yang sudah saya beli? Berarti saya bisa menggunakan kolom name!

Tetapi... Sepertinya sulit untuk menemukan **baris duplikat** seperti "puding" digambar kanan! Apakah ada cara pengolahan data yang bisa mengatasinya masalah ini?

Ya, bisa. Ada cara untuk **mengecualikan baris duplikat** dari sebuah kolom! Ayo kita coba bersama-sama.

21:34 05/09/2020

DISTINCT

DENGAN MENGGUNAKAN **DISTINCT**, ANDA DAPAT MENGECAULIKAN BARIS YANG MEMILIKI DATA YANG SAMA DARI HASIL PENCARIAN. ANDA DAPAT MENENTUKAN NAMA KOLOM UNTUK MENDAPATKAN BARIS UNIK DAN MENGECAULIKAN DUPLIKATNYA. UNTUK MELAKUKAN HAL INI, GUNAKAN SYNTAX BERIKUT:

DISTINCT(column_name)

sample.sql

```
DISTINCT(column_name)
```

menghilangkan baris duplikat untuk kolom yang dispesifikasi

21:34 05/09/2020

A screenshot of a Windows desktop showing a browser window for Progate. The page title is "SQL Study II | Progate - Belajar". The main content is about the **DISTINCT** keyword, specifically how it can be used to remove duplicate entries from a query. It includes a code editor window showing the following SQL code:

```
sample.sql
SELECT DISTINCT(name)
FROM purchases;
```

A screenshot of a Windows desktop showing a Progate exercise session. The title bar says "Pedoman Peserta - Ready Set Co...". The main area shows an exercise titled "Latihan" with the following instructions:

Untuk mempelajari cara kerja **DISTINCT**, kita dapat memulai dengan menggunakananya untuk mendapatkan baris data dengan cara yang sama seperti sebelumnya.

Beberapa code SQL sudah disediakan untuk Anda dalam file **exercise1.sql**. Tekan tombol ▶ untuk menjalankan code tersebut. Code ini akan mengambil baris dari kolom **character_name** tanpa menghapus baris duplikat.

The code editor shows the following SQL code:

```
-- Jalankan code untuk menampilkan kolom character_name dengan duplikasi
SELECT character_name
FROM purchases;
```

The results table shows the following data:

character_name
Ninja Ken
Guru Domba
Birdie
Ninja Ken

At the bottom, there are buttons for "Reset", "Lihat Jawaban", "Kirim", and "Hubungi kami!".

The image shows a Windows desktop with three separate windows open, each displaying a SQL exercise from the website progate.com.

Exercise 2: Using DISTINCT

Instruksi:

Selanjutnya, dengan menggunakan **DISTINCT**, kita akan mengambil data unik dari kolom `character_name` dan mengecualikan data duplikat.

Problema:

Dapatkan baris data dari kolom `character_name` menggunakan **DISTINCT** untuk menghapus baris duplikat.

Kode:

```
1 -- dapatkan baris dari kolom character_name dengan duplikat dihilangkan
2 SELECT DISTINCT(character_name)
3 FROM purchases;
```

Jawaban Anda (exercise2.sql):

character_name
Ninja Ken
Guru Domba
Birdie
Baby Ben

Jawaban Benar (exercise2.sql):

character_name
Ninja Ken
Guru Domba
Birdie
Baby Ben

Exercise 3: Using DISTINCT

Instruksi:

Anda untuk menjalankan code Anda dan pastikan didalam hasilnya tidak terdapat data duplikat.

Problema:

Selanjutnya, ayo kita buat daftar berisi hal-hal yang dibeli dengan uang tersebut.

Kode:

```
1 -- dapatkan baris dari kolom name tanpa duplikat
2 SELECT DISTINCT(name)
3 FROM purchases;
```

Jawaban Anda (exercise3.sql):

name
casing smartphone
rantangan
baju
buku pelajaran

Jawaban Benar (exercise3.sql):

name
casing smartphone
rantangan
baju
buku pelajaran

Both exercises show a completion status of "0/11 Sudah diselesaikan" and a "Hubungi Kami!" button.

Screenshot of a Progate SQL study session titled "Menggunakan Operator Aritmetika".

The page content includes:

- A user comment: "Saya berhasil membuat daftar barang yang saya beli! Terima kasih, Guru Domba!"
- A response: "Setelah dipikirkan, sepertinya akan lebih baik jika kita menambahkan pajak penjualan pada harga yang tersimpan dalam database. Bagaimana caranya?"
- An explanatory note: "Di SQL, **operator aritmetika** digunakan untuk melakukan penghitungan seperti perkalian. Kita bisa menggunakaninya untuk menghitung harga setelah pajak."
- A table titled "purchases" showing the following data:

name	price
puding	1
pena	1
buku tulis	2
puding	2
teh	2
rantangan	5
casing smartphone	15

A callout box on the right says: "ganti price dengan harga setelah pajak".

Screenshot of a Progate SQL study session titled "Operator Aritmetika".

The page content includes:

- A list of arithmetic operators:
 - + ... penjumlahan
 - ... pengurangan
 - * ... perkalian
 - / ... pembagian
- An explanation: "Penghitungan dapat dilakukan di SQL dengan **operator aritmetika**. Sebagaimana ditampilkan di bawah, dengan menggunakan operator ini, Anda dapat melakukan penghitungan di baris pada kolom yang Anda tentukan."
- A code editor window titled "sample.sql" showing:

```
column_name * 1.09  
kalikan baris di kolom yang dispesifikasi dengan 1.09!
```

```
column_name - 10  
kurangi 10 dari baris di kolom yang dispesifikasi!
```

Operator Aritmetika

Menggunakan Operator Aritmetika

sample.sql

```
SELECT name, price * 1.09  
      gunakan operator aritmetika  
   FROM purchases;
```

purchases	name	price
1	puding	1
2	pena	1
3	teh	2
4	puding	2
5	buku tulis	2
6	rantangan	5
7	casing smartphone	15

purchases (Hasil)	name	price * 1.09
1	puding	1.09
2	pena	1.09
3	teh	2.18
4	puding	2.18
5	buku tulis	2.18
6	rantangan	5.45
7	casing smartphone	16.35

dapatkan nilai kolom price yang telah dikalikan dengan 1.09!

Instruksi

Dalam latihan ini kita akan menggunakan operasi aritmetika untuk mendapatkan harga setelah pajak penjualan.

Karena Kueri SQL untuk mendapatkan harga di luar pajak penjualan sudah disediakan, Anda tinggal melanjutkannya saja.

progate/exercise1.sql

Dengan menambahkan `SELECT name, price`, kaliakan nilai dalam kolom price dengan tarif pajak sebesar 1.09.

Lihat Slide

Latihan

Database

```
/*  
ambil data harga setelah pajak penjualan dari kolom price  
dengan melanjutkan statement dibawah  
*/  
  
SELECT name, price, price * 1.09  
   FROM purchases;
```

Jawaban Anda (exercise1.sql)	Jawaban Benar (exercise1.sql)	
name	price	price * 1.09
casing smartphone	15	16.35
rantangan	5	5.45
baju	30	32.7
buku pelajaran	20	21.8

Reset Lihat Jawaban Kirim

1/11 Sudah diselesaikan

Lihat Slide Hubungi Kami!

Progate.com

Menggunakan Function Agregat

3. SUM

Agustina Nur Wulandari (Lv.9)

Menggunakan Function

Setelah menyertakan pajak penjualan, sepetinya saya sudah mengeluarkan uang yang sangat banyak. Saya harus menghemat! Apakah ada jenis analisis lainnya yang bisa saya lakukan?

Saya ingin mengetahui hal-hal seperti **total pengeluaran dan rata-rata pembelian barang.**

Tentu saja! Untuk itu kita dapat menggunakan **function agregat**. Dengan function tersebut, kita bisa mengumpulkan data seperti **total dan rata-rata data angka** yang disimpan didalam database.

Baiklah! Ayo mulai!

Data yang ditemukan dengan WHERE → Function Agregat → Hasil agregat sum, average, dll

Type here to search

22:17 05/09/2020

Progate.com

Mengumpulkan Jumlah Total

3. SUM

Agustina Nur Wulandari (Lv.9)

Pertama-tama, saya ingin mengetahui jumlah keseluruhan uang yang saya gunakan!

Ok, kalau begitu, sebaiknya kita lihat cara menghitung **nilai total data angka**.

Dalam kasus seperti ini, jika kita **menghitung total semua nilai dalam kolom price**, kita akan mengetahui berapa banyak uang yang sudah Anda keluarkan.

OK!

Kalkulasikan total-nya

name	price
puding	1
pena	1
buku tulis	2
puding	2
teh	2
rantangan	3
casing smartphone	15

Type here to search

22:18 05/09/2020

SUM

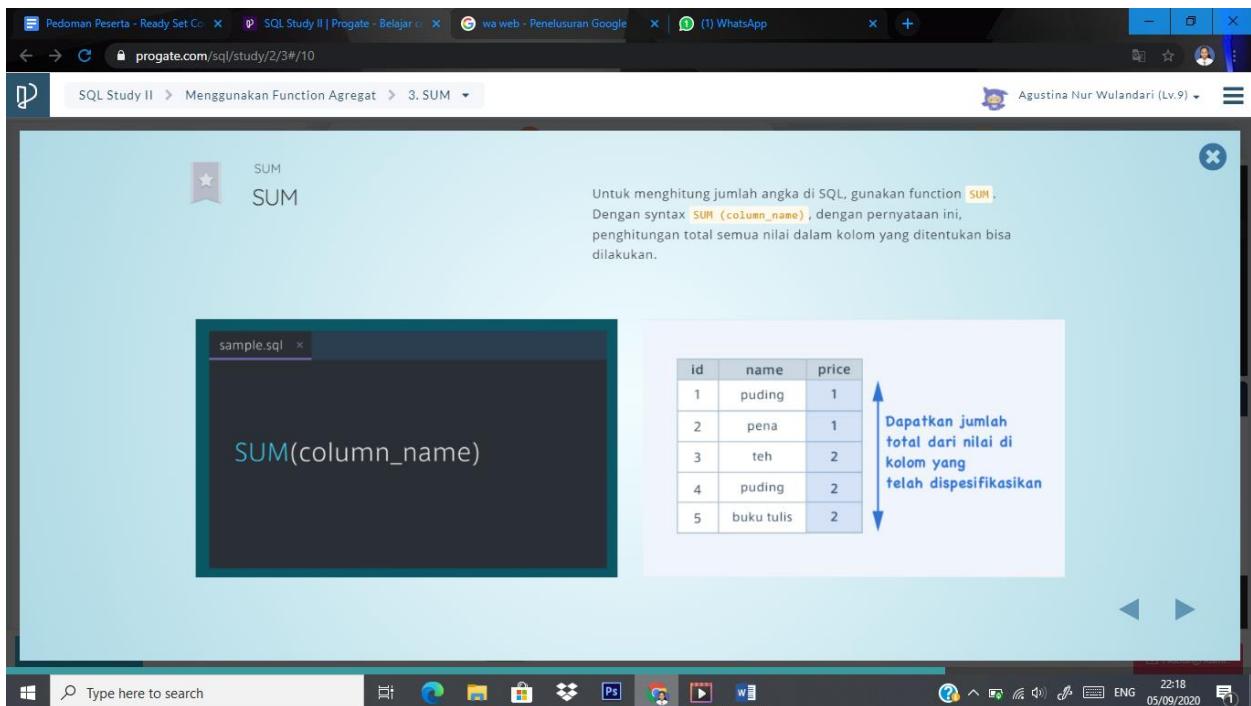
Untuk menghitung jumlah angka di SQL, gunakan function **SUM**. Dengan syntax **SUM (column_name)**, dengan pernyataan ini, penghitungan total semua nilai dalam kolom yang ditentukan bisa dilakukan.

sample.sql

```
SUM(column_name)
```

id	name	price
1	puding	1
2	pena	1
3	teh	2
4	puding	2
5	buku tulis	2

Dapatkan jumlah total dari nilai di kolom yang telah dispesifikasi



Menggunakan SUM

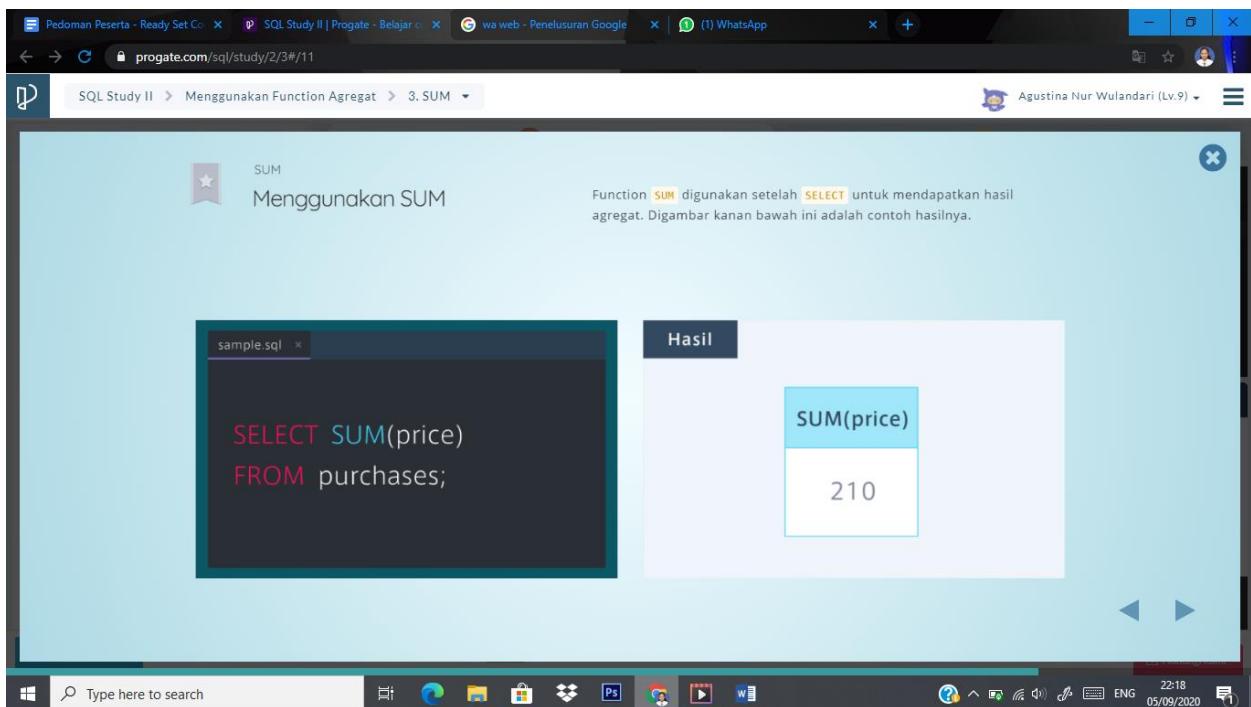
Function **SUM** digunakan setelah **SELECT** untuk mendapatkan hasil agregat. Digambar kanan bawah ini adalah contoh hasilnya.

sample.sql

```
SELECT SUM(price)  
FROM purchases;
```

Hasil

SUM(price)
210



The screenshot shows a Windows desktop with a browser window open to a SQL study session on Progate. The title of the lesson is "Menggabungkan WHERE & SUM". The main content area displays a code editor with the following SQL query:

```
sample.sql
SELECT SUM(price) -- ② dapatkan jumlah total dari kolom price dari hasil penyaringan baris dibawah
FROM purchases
WHERE character_name = "Ninja Ken";
```

Annotations explain the code: ① indicates the WHERE clause filters rows for "Ninja Ken", and ② indicates the SUM function calculates the total from the filtered rows.

The screenshot shows a Windows desktop with a browser window open to a SQL exercise session on Progate. The exercise title is "Latihan". The code editor contains the following SQL query:

```
exercise1sql
-- dapatkan total jumlah dari kolom price
SELECT SUM(price)
FROM purchases;
```

The "Jawaban Anda" tab shows the student's answer: `SUM(price)`. The "Jawaban Benar" tab shows the correct answer: `372`.

On the left, there is an "Instruksi" panel with text explaining the use of the SUM function to calculate the total value in the price column of the purchases table. It also mentions using WHERE to calculate the total amount spent by Ninja Ken.

At the bottom, there are buttons for "Reset", "Lihat Jawaban", and "Kirim". A progress bar indicates "2/11 Sudah diselesaikan".

Selanjutnya, gunakan `WHERE` untuk menghitung jumlah total yang dibelanjakan Ninja Ken.

Menggunakan `SUM`,apatkan jumlah total kolom `price`. Namun, kali ini, gunakan `WHERE` untuk mengambil baris berisi `Ninja Ken` saja.

```
/*  
 * dapatkan jumlah total dari kolom price dimana  
 * nilai character_name adalah "Ninja Ken"  
 */  
  
SELECT SUM(price)  
FROM purchases  
WHERE character_name = "Ninja Ken";
```

Jawaban Anda (exercise2.sql) Jawaban Benar (exercise2.sql)

Kirim

2 / 11 Sudah diselesaikan

Hubungi Kami!

Selanjutnya, gunakan `WHERE` untuk menghitung jumlah total yang dibelanjakan Ninja Ken.

Menggunakan `SUM`,apatkan jumlah total kolom `price`. Namun, kali ini, gunakan `WHERE` untuk mengambil baris berisi `Ninja Ken` saja.

```
/*  
 * dapatkan jumlah total dari kolom price dimana  
 * nilai character_name adalah "Ninja Ken"  
 */  
  
SELECT SUM(price)  
FROM purchases  
WHERE character_name = "Ninja Ken";
```

Level Up!
Lv.10

Anda berhasil naik ke level yang baru! Bagikan berita baik ini!

Kembali ke Latihan Berikutnya

Lihat Jawaban Berikutnya

3 / 11 Sudah diselesaikan

Hubungi Kami!

Menghitung Rata-Rata

Saya sangat terkejut mengetahui banyaknya uang yang sudah saya keluarkan! Bisakah saya mencari tahu jumlah rata-rata yang saya keluarkan untuk setiap pembelian!

Tentu saja. Untuk hal ini, kita memerlukan **function agregat untuk menghitung rata-rata.**

Setelah kita menghitung rata-rata nilai yang disimpan dalam kolom price, kita akan mengetahui jumlah rata-rata uang yang Anda keluarkan per pembelian!

Dapatkan nilai rata-rata dari kolom price

id	name	price
1	puding	1
2	pena	1
3	teh	2
4	puding	2
5	buku tulis	2

Untuk menghitung rata-rata angka di SQL, Anda dapat menggunakan **AVG**. Menggunakan syntax **AVG (column_name)**, Anda dapat menghitung nilai rata-rata pada kolom yang dinginkan.

sample.sql

```
AVG(column_name)
```

Dapatkan nilai rata-rata dari kolom yang telah dispesifikasi

id	name	price
1	puding	1
2	pena	1
3	teh	2
4	puding	2
5	buku tulis	2

A screenshot of a Windows desktop showing a browser window for Progate. The page title is "Menggunakan Function Agregat > 4. AVG". The content shows a section titled "AVG" with a sub-section "Menggunakan AVG". It explains that the AVG function calculates the average of a column in a SELECT statement. Below this is a code editor window titled "sample.sql" containing the following SQL query:

```
SELECT AVG(price)
FROM purchases;
```

To the right of the code editor is a "Hasil" (Result) window showing the output:

AVG(price)
4

The status bar at the bottom of the screen shows the date and time as 05/09/2020 22:28.

A screenshot of a Windows desktop showing a browser window for Progate. The page title is "Menggunakan Function Agregat > 4. AVG". The content shows a section titled "AVG" with a sub-section "Menggabungkan WHERE & AVG". It explains that the AVG function can be combined with the WHERE clause to calculate the average price for a specific character. Below this is a code editor window titled "sample.sql" containing the following SQL query with annotations:

```
SELECT AVG(price) ② Kalkulasikan rata-rata kolom price dari hasil penyaringan baris dibawah
FROM purchases
WHERE ① dapatkan baris dengan nilai "Ninja Ken" !
character_name = "Ninja Ken";
```

The status bar at the bottom of the screen shows the date and time as 05/09/2020 22:28.

The screenshot shows a browser window with multiple tabs. The active tab is 'progate.com/sql/study/2/4#17'. The page title is 'SQL Study II | Progate - Belajar'. The main content area has a header 'Latihan' and a sidebar 'Instruksi'. The 'exercise1.sql' tab is selected, displaying the following SQL code:

```
1 -- dapatkan nilai rata-rata dari kolom price
2 SELECT AVG(price)
3 FROM purchases;
```

The results section shows the output: 'Jawaban Anda (exercise1.sql)' and 'Jawaban Benar (exercise1.sql)'. Both show 'AVG(price)' and the value '12.4'. Below the code editor is a toolbar with 'Reset', 'Lihat Jawaban', and a large green 'Kirim' button. The status bar at the bottom right shows '3/11 Sudah diselesaikan'.

This screenshot shows the same study interface after moving to exercise 2. The 'exercise2.sql' tab is selected, displaying the following SQL code:

```
1 /*
2  * dapatkan nilai rata-rata dari kolom price dimana
3  * nilai character_name adalah "Ninja Ken"
4  */
5
6 SELECT AVG(price)
7 FROM purchases
8 WHERE character_name = "Ninja Ken";
```

The results section shows the output: 'Jawaban Anda (exercise2.sql)' and 'Jawaban Benar (exercise2.sql)'. Both show 'AVG(price)' and the value '10.727272727272727'. Below the code editor is a toolbar with 'Reset', 'Lihat Jawaban', and a large green 'Kirim' button. The status bar at the bottom right shows '3/11 Sudah diselesaikan'.

Menghitung Jumlah Baris

Jumlah totalnya besar, tetapi harga rata-rata tiap pembeliannya kecil. Artinya, saya melakukan pembelian yang sangat banyak! Saya mau melihat berapa banyak pembelian yang sudah saya lakukan!

Pengamatan yang jitu Ken! Untuk kasus itu, kita dapat menggunakan **function agregat untuk menghitung jumlah baris.**

Cukup dengan menghitung jumlah baris yang disimpan dalam tabel pembelian, Anda dapat menentukan berapa kali sebuah item dibeli.

Dapatkan jumlah total baris!

id	name	price
1	puding	1
2	pena	1
3	teh	2
4	puding	2
5	buku tulis	2

COUNT

Function **COUNT** digunakan untuk menghitung jumlah total baris pada kolom yang ditargetkan. Ini dapat dilakukan dengan menggunakan syntax `COUNT(column_name)` sebagaimana ditampilkan pada contoh di bawah.

`sample.sql`

`COUNT(column_name)`

dapatkan jumlah total baris didalam kolom!

id	name	price
1	puding	1
2	pena	1
3	teh	2
4	puding	2
5	buku tulis	2

Screenshot of a Progate study session titled "Menggunakan Function Agregat" (Using Aggregate Functions) under "5. COUNT".

The main content area shows a sample SQL query:

```
sample.sql
COUNT(price)
```

A note explains: "Saat menggunakan COUNT, jumlah baris yang terhitung tidak mencakup baris dengan nilai NULL. Oleh karena itu, untuk kasus seperti gambar di sebelah kanan, hasil akhir hitungan COUNT adalah 4."

To the right, a table illustrates this:

id	name	price
1	puding	1
2	pena	1
3	teh	
4	puding	2
5	buku tulis	2

A callout box points to the empty cell in row 3: "Baris dengan nilai NULL tidak dihitung".

Screenshot of a Progate study session titled "Menggunakan COUNT" under "5. COUNT".

The main content area shows a sample SQL query:

```
sample.sql
SELECT COUNT(*)
FROM purchases;
```

A note explains: "Jika Anda ingin menghitung semua baris, termasuk baris dengan nilai NULL, Anda harus menggunakan * (tanda bintang) dengan function COUNT. Ketika digunakan, * akan menghitung jumlah total baris secara keseluruhan, termasuk yang nilainya NULL."

To the right, a table labeled "purchases" is shown:

id	name	price
1	puding	1
2	pena	1
3	teh	2
4	puding	
5	teh	2
6		3
7	casing smartphone	15

An arrow points from the table to a box labeled "Hasil" containing the result:

Menghitung jumlah total baris

Hasil

COUNT(*)
7

COUNT
WHERE & COUNT

Penggunaan function `COUNT` juga dapat dikombinasikan dengan `WHERE`. Pada contoh di bawah, kita menggunakan keduanya untuk melihat total pembelian yang dilakukan oleh Ninja Ken.

```
sample.sql
SELECT COUNT(*) ② Menghitung jumlah baris
FROM purchases
WHERE character_name = "Ninja Ken";
① Cari baris dengan nilai "Ninja Ken" !
```

Instruksi

Beberapa nilai dalam kolom `name` pada tabel `purchases` adalah `null`.

Dengan menggunakan `COUNT`, coba kita lihat berapa banyak nilai yang disimpan dalam kolom `name`.

progate/exercise1.sql

Gunakan `COUNT` untuk mendapatkan berapa banyak nilai yang disimpan dalam kolom `name`.

name

Lihat Slide

Latihan

Database

exercise1sql exercise2sql exercise3sql

```
1 -- dapatkan jumlah baris dikolom name dari table purchases
2 SELECT COUNT(name)
3 FROM purchases;
```

Jawaban Anda (exercise1sql) Jawaban Benar (exercise1sql)

COUNT(name)

20

Reset Lihat Jawaban Kirim

4/11 Sudah diselesaikan

Hubungi Kami!

Sekarang coba kita cari tahu berapa kali pembelian dilakukan (melalui jumlah baris dalam tabel).

Gunakan `COUNT` dan `*` untuk mendapatkan jumlah total baris dalam tabel `purchases`.

```
progate/exercise2.sql
1 -- dapatkan jumlah baris di tabel purchases
2 SELECT COUNT(*)
3 FROM purchases;
```

Jawaban Anda (exercise2.sql) Jawaban Benar (exercise2.sql)

COUNT(*)
30

Reset Lihat Jawaban Kirim Hubungi Kami!

Selanjutnya, gunakan juga `WHERE` untuk melihat berapa kali Ninja Ken melakukan pembelian.

Gunakan `COUNT` dan `*` untuk menghitung jumlah total baris dalam tabel.

Namun, kali ini, gunakan `WHERE` untuk mengambil baris `character_name` yang berisi `Ninja Ken` saja.

```
progate/exercise3.sql
1 -- dapatkan jumlah total baris dimana nilai character_name adalah "Ninja Ken"
2 SELECT COUNT(*)
3 FROM purchases
4 WHERE character_name = "Ninja Ken";
```

Jawaban Anda (exercise3.sql) Jawaban Benar (exercise3.sql)

COUNT(*)
11

Reset Lihat Jawaban Kirim Hubungi Kami!

Selanjutnya, gunakan juga `WHERE` untuk melihat berapa kali Ninja Ken melakukan pembelian.

Gunakan `COUNT` dan `*` untuk menghitung jumlah total baris dalam tabel.

Namun, kali ini, gunakan `WHERE` untuk mengambil baris `character_name` yang berisi `Ninja Ken` saja.

Level Up!
Lv.11
Anda berhasil naik ke level yang baru!
Bagikan berita baik ini!

Kembali ke Latihan Berikutnya

Pelajaran berikutnya adalah "MAX·MIN".

Menemukan Nilai Maksimum dan Minimum

Sekarang saya mulai memahami tentang jumlah uang yang saya keluarkan! Untuk mengurangi pengeluaran, sepertinya saya harus menemukan pembelian yang paling mahal. Hal itu bisa dilakukan dengan mengurutkan baris menurut harga, betul?

Benar sekali! Tapi ada teknik yang lebih mudah! Menggunakan function agregat, Anda dapat mencari nilai maksimum atau minimum!

Berarti, dengan cara ini, saya dapat melihat mana pembelian yang paling mahal. Dengan begitu, saya bisa mengurangi pengeluaran saya!

Dengan function **MAX** di SQL, Anda bisa mendapatkan nilai **maksimum** dari baris milik kolom tertentu. Sebaliknya, dengan menggunakan function **MIN**, Anda bisa mendapatkan nilai **minimum-nya**.

sample.sql

- ① **MAX(column_name)**
- ② **MIN(column_name)**

id	name	price	purchased_at
1	casing smartphone	<u>①</u> 15 Max	2018-10-10
2	rantangan	5	2018-10-10
3	kalkulator	7	2018-10-10
4	pena	<u>②</u> 1 Min	2018-10-11

Sama dengan function agregat lainnya, **MAX** dan **MIN** dapat digunakan setelah **SELECT**. Seperti contoh dibawah, dengan menetapkan kolom **price** di function **MAX**, Anda bisa mendapatkan item dengan harga **tertinggi** untuk semua baris dikolom **price**.

sample.sql

```
SELECT MAX(price)
FROM purchases;
```

Hasil

MAX(price)
15

The screenshot shows a Windows desktop with a browser window open to a Progate SQL study page. The title of the page is "Menggabungkan WHERE dengan MAX · MIN". The main content area contains a note about combining the WHERE clause with MAX and MIN functions to find the highest purchase price for a character named "Ninja Ken". Below the note is a code editor window titled "sample.sql" containing the following SQL query:

```
SELECT MAX(price) • Dapatkan nilai maksimum dari hasil yang telah ditemukan
FROM purchases
WHERE character_name = "Ninja Ken"; • Cari baris dengan nilai "Ninja Ken" !
```

The screenshot shows a Windows desktop with a browser window open to a Progate SQL exercise page. The title of the page is "Latihan" (Exercise). The exercise is titled "Dengan MAX, kita dapat melihat nilai tertinggi dalam tabel purchases." It includes a code editor window titled "exercise1sql" with the following SQL query:

```
-- dapatkan harga tertinggi di kolom price
SELECT MAX(price)
FROM purchases;
```

Below the code editor is a comparison table for "Jawaban Anda" and "Jawaban Benar". The "Jawaban Benar" row contains the value "80". At the bottom of the exercise page are buttons for "Reset", "Lihat Jawaban", and "Kirim". The status bar at the bottom indicates "5/11 Sudah diselesaikan".

Dengan `MIN`, kita juga dapat melihat nilai terendah dalam tabel.

Menggunakan statement SQL yang sudah ditulis dalam editor, gunakan `MIN` untuk mendapatkan nilai terendah dalam kolom `price`.

Selanjutnya kita dapat menggunakan `WHERE` untuk mengetahui secara spesifik pembelian termahal Ninja Ken.

```
-- dapatkan harga terendah di kolom price
SELECT MIN(price)
FROM purchases;
```

Jawaban Anda (exercise2.sql) Jawaban Benar (exercise2.sql)

MIN(price)

1

Reset Lihat Jawaban Kirim Hubungi Kami!

Selanjutnya kita dapat menggunakan `WHERE` untuk mengetahui secara spesifik pembelian termahal Ninja Ken.

Menggunakan statement SQL yang sudah ditulis dalam editor, gunakan `MAX` untuk mendapatkan nilai tertinggi dalam kolom `price`.

Namun kali ini, gunakan `WHERE` untuk mengambil baris dengan kolom `character_name` berisi `Ninja Ken` saja.

```
/*
untuk baris dimana nilai character_name adalah "Ninja Ken"
dapatkan harga tertinggi dari semua nilai di kolom price
*/
SELECT MAX(price)
FROM purchases
WHERE character_name = "Ninja Ken";
```

Jawaban Anda (exercise3.sql) Jawaban Benar (exercise3.sql)

MAX(price)

54

Reset Lihat Jawaban Kirim Hubungi Kami!

Mengelompokkan Baris

Function agregat seperti sangat berguna, tetapi, apakah klaus WHERE perlu digunakan untuk mendapatkan jumlah pembelanjaan ditanggal tertentu? Misalnya, saya ingin mendapatkan total pengeluaran per tanggal seperti digambar!

Tentu saja, Ayo pelajari teknik pengelompokan. Teknik ini dapat menerapkan function agregat dibaris dengan tanggal yang sama.

Mantap! Dengan ini, saya bisa mencari tahu berapa banyak pengeluaran bulanan saya!

Total jumlah dari setiap tanggal

SUM(price)	purchased_at
4	2018-10-10
3	2018-10-11
18	2018-10-12

GROUP BY

Dengan GROUP BY, Anda dapat mengelompokkan baris. Misalnya, menggunakan syntax GROUP BY column_name, sebagaimana ditampilkan di bawah ini, baris dengan nilai yang sama akan dikelompokkan untuk kolom yang Anda tentukan.

sample.sql

```
GROUP BY column_name
```

purchases

id	name	price	purchased_at
1	puding	1	2018-10-10
2	pena	1	2018-10-10
3	buku tulis	2	2018-10-10
4	puding	2	2018-10-11
5	teh	2	2018-10-11
6	rantangan	5	2018-10-12
7	casing smartphone	15	2018-10-12

grup "2018-10-10"
grup "2018-10-11"
grup "2018-10-12"

GROUP BY

Mengelompokkan Data & Mengagregat

Untuk menggunakan pengelompokan dengan data agregat, Anda dapat menambahkan `GROUP BY column_name` pada akhir statement SQL seperti gambar di sisi kiri. Dalam kasus seperti gambar dibawah, code ini akan menerapkan function agregat untuk menampilkan jumlah harga bagi setiap data tanggal unik dikolom "purchased_at".

sample.sql

```
SELECT SUM(price)
FROM purchases
GROUP BY purchased_at;
```

① Grup

name	price	purchased_at
pinggir	1	2018-10-10
pesa	1	2018-10-10
deku hama	2	2018-10-10

② Agregat

SUM(price)	purchased_at
4	2018-10-10
4	2018-10-11
20	2018-10-12

Catatan Tentang GROUP BY

Saat menggunakan `GROUP BY`, kolom yang ditentukan untuk mengelompokkan item atau function agregat sering digunakan didalam `SELECT`. Seperti contoh di bawah ini, karena function agregat `SUM()` tidak diberikan nama kolom, tidak ada yang dapat dikelompokkan oleh `GROUP BY` dan pada akhirnya yang ditampilkan adalah instance untuk setiap nilai "purchased_at".

sample.sql

○ SELECT `SUM(price)`, `purchased_at`
FROM purchases
GROUP BY `purchased_at`;

sample.sql

✗ SELECT `price`, `purchased_at`
FROM purchases
GROUP BY `purchased_at`;

Dalam latihan ini, kita akan menggunakan **GROUP BY** untuk menentukan total pembelian berdasarkan tanggal.

Gunakan function **SUM** untuk mendapatkan total dari kolom **price** dan pilih juga kolom **purchased_at**.

Lakukan pengelompokan pada kolom **purchased_at** menggunakan **GROUP BY**.

Lihat Slide

Sekarang coba gunakan **GROUP BY** untuk

Lihat Slide

Jawaban Anda (exercise1.sql) Jawaban Benar (exercise1.sql)

SUM(price)	purchased_at
75	2018-10-10
27	2018-10-11
22	2018-10-13
12	2018-10-17

6/11 Sudah diselesaikan

Kirim

Hubungi Kami!

Sekarang coba gunakan **GROUP BY** untuk mengetahui berapa kali pembelian dilakukan di setiap tanggal.

Gunakan function **COUNT** untuk mendapatkan jumlah baris dan pilih juga kolom **purchased_at**.

(untuk function COUNT, gunakan **price** sebagai nama kolom)

Gunakan **GROUP BY** untuk mengelompokkan data menurut **purchased_at**.

Lihat Slide

Jawaban Anda (exercise2.sql) Jawaban Benar (exercise2.sql)

COUNT(price)	purchased_at
5	2018-10-10
2	2018-10-11
3	2018-10-13
1	2018-10-17

6/11 Sudah diselesaikan

Kirim

Hubungi Kami!

Pengelompokan sangatlah berguna!

Memang, Anda bahkan dapat mengelompokkan banyak kolom. Misalnya, Anda dapat mengelompokkan hasil harga total yang dikeluaran oleh setiap karakter menurut tanggal pembelian.

SUM(price)	purchased_at	character_name
68	2018-10-10	Birdie
83	2018-10-10	Ninja Ken
279	2018-10-10	Guru Domba
8	2018-10-11	Baby Ben
35	2018-10-11	Ninja Ken
50	2018-10-12	Ninja Ken

GROUP BY (lebih dari satu)

Cara Menggunakan GROUP BY dengan banyak Kolom

Anda dapat menggunakan **GROUP BY** untuk banyak kolom dengan memasukkan nama kolom-kolom tersebut dan memisahkannya dengan koma (,). Kita akan lihat apa saja jenis pengelompokan dan agregat yang dapat dibuat dislide berikutnya.

```
sample.sql
GROUP BY kolom1, kolom2, . . .
gunakan koma
```

```
sample.sql
SELECT SUM(price), purchased_at, character_name
FROM purchases
GROUP BY purchased_at, character_name;
```

Screenshot of a Windows desktop showing a browser window on Progate.com. The page title is "SQL Study II | Mengelompokkan Data | 8. GROUP BY (lebih dari satu)". The content discusses using GROUP BY with multiple columns to combine data into groups. It shows four examples of grouped data:

- grup "2018-10-10" untuk "Ninja Ken"**

SUM(price)	purchased_at	character_name
25	2018-10-10	Ninja Ken
50	2018-10-10	Ninja Ken
- grup "2018-10-10" untuk "Guru Domba"**

SUM(price)	purchased_at	character_name
50	2018-10-10	Guru Domba
150	2018-10-10	Guru Domba
- grup "2018-10-10" untuk "Baby Ben"**

SUM(price)	purchased_at	character_name
3	2018-10-10	Baby Ben
- grup "2018-10-11" untuk "Ninja Ken"**

SUM(price)	purchased_at	character_name
50	2018-10-11	Ninja Ken
250	2018-10-11	Ninja Ken
- grup "2018-10-11" untuk "Guru Domba"**

SUM(price)	purchased_at	character_name
270	2018-10-11	Guru Domba
- grup "2018-10-12" untuk "Ninja Ken"**

SUM(price)	purchased_at	character_name
56	2018-10-12	Ninja Ken

Screenshot of a Windows desktop showing a browser window on Progate.com. The page title is "SQL Study II | Mengelompokkan Data | 8. GROUP BY (lebih dari satu)". The content discusses using aggregate functions on grouped results. It shows two examples:

- total per tanggal dan nama karakter**

SUM(price)	purchased_at	character_name
50	2018-10-10	Ninja Ken
150	2018-10-10	Guru Domba
2	2018-10-10	Baby Ben
10	2018-10-11	Ninja Ken
27	2018-10-11	Guru Domba
5	2018-10-12	Ninja Ken
- Jumlah pembelian per tanggal dan nama karakter**

COUNT(*)	purchased_at	character_name
2	2018-10-10	Ninja Ken
2	2018-10-10	Guru Domba
1	2018-10-10	Baby Ben
3	2018-10-11	Ninja Ken
1	2018-10-11	Guru Domba
1	2018-10-12	Ninja Ken

Pedoman Peserta - Ready Set Co SQL Study II | Progate - Belajar wa web - Penelusuran Google (1) WhatsApp

← → C progate.com/sql/study/2/8#34

SQL Study II > Mengelompokkan Data > 8. GROUP BY (lebih dari satu)

Agustina Nur Wulandari (Lv.11)

Instruksi

Gunakan `GROUP BY` untuk mengetahui jumlah total uang yang digunakan setiap karakter pada tanggal tertentu.

progate/exercise1.sql

Gunakan function `SUM` untuk mendapatkan total dari kolom `price` dan kelompokkan datanya berdasarkan kolom `purchased_at` dan `character_name`.

Gunakan `GROUP BY` untuk mengelompokkan kolom `purchased_at` dan `character_name`.

Latihan

exercise1.sql exercise2.sql

```
1 -- dapatkan total harga purchases berdasarkan purchased_at dan character_name
2 SELECT SUM(price), purchased_at, character_name
3 FROM purchases
4 GROUP BY purchased_at, character_name;
5
```

Jawaban Anda (exercise1.sql) Jawaban Benar (exercise1.sql)

SUM(price)	purchased_at	character_name
30	2018-10-10	Birdie
10	2018-10-10	Guru Domba
35	2018-10-10	Ninja Ken
25	2018-10-11	Baby Ben

Reset Lihat Jawaban Kirim

7 / 11 Sudah diselesaikan

Lihat Slide Hubungi Kami!

Type here to search

23:00 05/09/2020 ENG

Pedoman Peserta - Ready Set Co SQL Study II | Progate - Belajar wa web - Penelusuran Google (1) WhatsApp

← → C progate.com/sql/study/2/8#34

SQL Study II > Mengelompokkan Data > 8. GROUP BY (lebih dari satu)

Agustina Nur Wulandari (Lv.11)

Instruksi

Selanjutnya, gunakan `GROUP BY` untuk mengetahui berapa kali pembelian dilakukan oleh setiap karakter untuk setiap tanggal.

progate/exercise2.sql

Gunakan function `COUNT` untuk mendapatkan total jumlah baris yang dikelompokkan menurut kolom `purchased_at` dan `character_name`.

Gunakan `GROUP BY` untuk mengelompokkan baris dari `purchased_at` dan `character_name`.

Latihan

exercise1.sql exercise2.sql

```
1 /*
2  dapatkan total berapa kali purchases terjadi berdasarkan
3  purchased_at and character_name
4 */
5
6 SELECT COUNT(*), purchased_at, character_name
7 FROM purchases
8 GROUP BY purchased_at, character_name;
9
```

Jawaban Anda (exercise2.sql) Jawaban Benar (exercise2.sql)

COUNT(*)	purchased_at	character_name
1	2018-10-10	Birdie
2	2018-10-10	Guru Domba
2	2018-10-10	Ninja Ken
1	2018-10-11	Baby Ben

Reset Lihat Jawaban Kirim

7 / 11 Sudah diselesaikan

Lihat Slide Hubungi Kami!

Type here to search

23:05 05/09/2020 ENG

Selanjutnya, gunakan `GROUP BY` untuk mengetahui berapa kali pembelian dilakukan oleh setiap karakter untuk setiap tanggal.

Gunakan function `COUNT` untuk mendapatkan total jumlah baris yang dikelompokkan menurut kolumn `purchased_at` dan `character_name`.

Gunakan `GROUP BY` untuk mengelompokkan baris dari kolumn `purchased_at` dan `character_name`.

Pelajaran berikutnya adalah "GROUP BY (WHERE)".

Latihan

Level Up!

Lv.12

Anda berhasil naik ke level yang baru! Bagikan berita baik ini!

Kembali ke Latihan Berikutnya

8/11 Sudah diselesaikan

Agustina Nur Wulandari (Lv.11)

Mengelompokkan Menurut Kondisi Tepat

Saya sudah bisa mendapatkan jumlah total yang dikeluarkan setiap karakter untuk setiap tanggal, tetapi saya ingin mencari hasil yang lebih spesifik!

Seperti **berapa banyak uang yang dikeluarkan seseorang untuk makanan pada tanggal tertentu?**

Untuk data tersebut, kita harus **mengombinasikan GROUP BY dengan klausu WHERE**. Saya akan mengajarmu dipelajaran ini.

SUM(price)	purchased_at	character_name
25	2018-10-10	Birdie
32	2018-10-10	Ninja Ken
71	2018-10-10	Guru Domba
8	2018-10-11	Baby Ben
22	2018-10-11	Ninja Ken
36	2018-10-13	Ninja Ken

total pengeluaran untuk makanan per tanggal dan nama

Agustina Nur Wulandari (Lv.12)

GROUP BY (WHERE)

WHERE & GROUP BY

`sample.sql`

```
SELECT aggregate_function
FROM table_name
WHERE kondisi
GROUP BY kolom1, kolom2;
diletakkan setelah WHERE!
```

Pencarian	WHERE
Gruping	GROUP BY
Function	COUNT · SUM · AVG · MAX · MIN

GROUP BY (WHERE)

Urutan Penggunaan WHERE dan GROUP BY (1)

Untuk mendapatkan jumlah total uang makanan yang dibelanjakan setiap karakter pada hari tertentu, ikuti 3 langkah berikut:

- ① Cari baris dengan nilai category "makanan"
- ② Kelompokkan nilai menurut kolom character_name dan purchased_at. Untuk langkah ketiga akan dijelaskan pada slide berikutnya.

① Pencarian

price	purchased_at	character_name	category
8	2018-10-10	Baby Ben	makanan
19	2018-10-10	Birdie	makanan
9	2018-10-10	Ninja Ken	makanan
47	2018-10-10	Ninja Ken	makanan
35	2018-10-10	Guru Domba	makanan
122	2018-10-10	Guru Domba	makanan

② Gruping

grup "2018-10-10" untuk "Ninja Ken"

price	purchased_at	character_name
9	2018-10-10	Ninja Ken
47	2018-10-10	Ninja Ken

grup "2018-10-10" untuk "Guru Domba"

SUM(price)	purchased_at	character_name
35	2018-10-10	Guru Domba
122	2018-10-10	Guru Domba

grup "2018-10-10" untuk "Baby Ben"

SUM(price)	purchased_at	character_name
8	2018-10-10	Baby Ben

The screenshot shows a Progate study session titled "9. GROUP BY (WHERE)". The main content is a table titled "Aggregate" with the following data:

SUM(price)	purchased_at	character_name
300	2018-10-10	Baby Ben
1000	2018-10-10	Ninja Ken
350	2018-10-10	Guru Domba
700	2018-10-11	Ninja Ken
270	2018-10-11	Guru Domba
500	2018-10-12	Ninja Ken

The screenshot shows a Progate study session titled "Menulis SQL dengan WHERE dan GROUP BY". It includes a message from a character named "Guru Domba" and a SQL code editor window.

Baiklah! Jika saya mencari baris dengan WHERE terlebih dahulu, lalu mengelompokkannya berdasarkan tanggal dan nama, saya dapat melihat bagaimana penggunaan uang saya sehari-hari!

Tepat sekali! Lihatlah contoh di kanan ini. Perhatikan urutannya ketika menggunakan WHERE dengan GROUP BY. Sekarang adalah waktunya untuk menguji pengetahuan Anda!

```
sample.sql x
SELECT SUM(price), purchased_at, character_name
FROM purchases
WHERE category = "makanan"
① Mencari baris yang memiliki "makanan" !
GROUP BY purchased_at, character_name;
② Grup hasil berdasarkan tanggal dan nama karakter!
```

Pedoman Peserta - Ready Set Co SQL Study II | Progate - Belajar wa web - Penelusuran Google (1) WhatsApp

← → 🔒 progate.com/sql/study/2/9#39

SQL Study II > Mengelompokkan Data > 9. GROUP BY (WHERE)

Instruksi

Gunakan WHERE untuk mengambil baris dengan kategori makanan dan kelompokkan hasilnya.

progate/exercise2.sql

Menggunakan WHERE, agregasikan baris dengan nilai makanan dalam kolom category.

Selain itu, kelompokkan berdasarkan kolom berikut:

- purchased_at
- character_name

Untuk setiap kelompok, dapatkan baris dari data berikut:

- Jumlah price

Jawaban Anda (exercise2.sql) Jawaban Benar (exercise2.sql)

SUM(price)	purchased_at	character_name
5	2018-10-10	Guru Domba
2	2018-10-11	Ninja Ken
15	2018-10-13	Ninja Ken
1	2018-10-18	Baby Ben

Reset Lihat Jawaban Kirim

8 / 11 Sudah diselesaikan

Hubungi Kami!

Lihat Slide

Type here to search

23:18 05/09/2020 ENG

Agustina Nur Wulandari (Lv.12)

Database

The screenshot shows a browser window with multiple tabs. The main content is a SQL study interface. On the left, there's an 'Instruksi' section with instructions and a code editor for 'exercise2.sql'. The code is as follows:

```
3 setelah itu, kelompokkan hasilnya berdasarkan kolom
4 purchased_at dan character_name
5 */
6
7 SELECT SUM(price),purchased_at,character_name
8 FROM purchases
9 WHERE category = "makanan"
10 GROUP BY purchased_at,character_name;
11
```

The right side shows the 'Latihan' section with two tabs: 'Jawaban Anda (exercise2.sql)' and 'Jawaban Benar (exercise2.sql)'. Both tabs show the same results table:

SUM(price)	purchased_at	character_name
5	2018-10-10	Guru Domba
2	2018-10-11	Ninja Ken
15	2018-10-13	Ninja Ken
1	2018-10-18	Baby Ben

At the bottom, there are buttons for 'Reset', 'Lihat Jawaban', and a large green 'Kirim' button. A progress bar indicates '8 / 11 Sudah diselesaikan' (8/11 completed). A red 'Hubungi Kami!' button is also visible.

Pedoman Peserta - Ready Set Co SQL Study II | Progate - Belajar wa web - Penelusuran Google (1) WhatsApp

← → 🔒 progate.com/sql/study/2/9#39

SQL Study II > Mengelompokkan Data > 9. GROUP BY (WHERE)

Instruksi

Ayo gunakan WHERE untuk mengelompokkan data pembelanjaan Ninja Ken!

progate/exercise1.sql

Pilih data berikut dan kelompokkan menurut purchased_at.

- total price
- purchased_at

Gunakan WHERE untuk mengagregasikan baris dengan kolom character_name.

berisi Ninja Ken .

Jawaban Anda (exercise1.sql) Jawaban Benar (exercise1.sql)

SUM(price)	purchased_at
35	2018-10-10
2	2018-10-11
15	2018-10-13
2	2018-10-18

Reset Lihat Jawaban Kirim

8 / 11 Sudah diselesaikan

Hubungi Kami!

Lihat Slide

Type here to search

23:15 05/09/2020 ENG

Agustina Nur Wulandari (Lv.12)

Database

This screenshot is similar to the one above but for exercise 1. The 'Instruksi' section has different text, and the code in the editor is:

```
1 /*
2 tambahkan aturan di klausa WHERE untuk mengelompokkan
3 baris, dimana character_name adalah "Ninja Ken"
4 */
5
6 SELECT SUM(price),purchased_at
7 FROM purchases
8 WHERE character_name = "Ninja Ken"
9 GROUP BY purchased_at;
```

The results table is identical to the one in the first screenshot:

SUM(price)	purchased_at
35	2018-10-10
2	2018-10-11
15	2018-10-13
2	2018-10-18

Mempersempit Data yang Dikelompokkan

Dengan pengelompokan, saya bisa mendapatkan jumlah total uang yang saya habiskan setiap hari! Bisakah saya merincikannya lagi? Misalnya, hanya melihat tanggal di mana saya menghabiskan **lebih dari \$10**? Atau tanggal ketika saya **menghabiskan uang banyak**?

Tentu saja! Anda bisa mendapatkan grup tertentu dari grup yang dibuat dengan GROUP BY di SQL dengan menggunakan klausula **HAVING**.

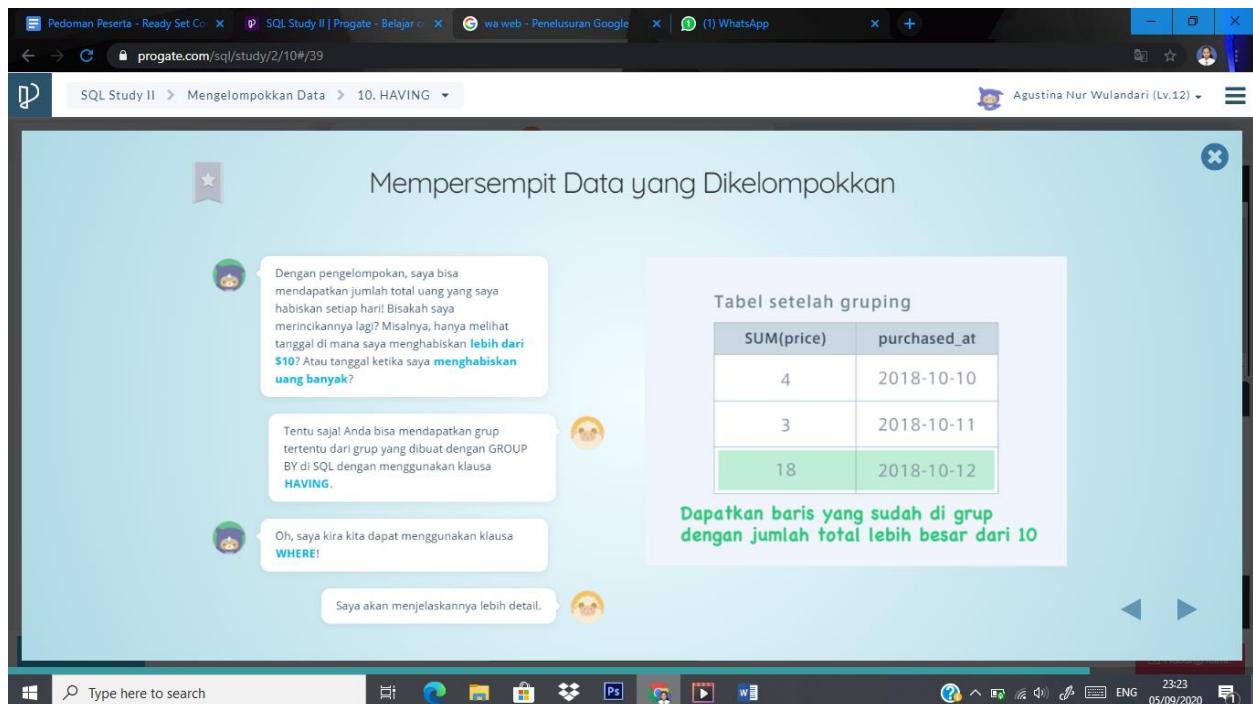
Oh, saya kira kita dapat menggunakan klausula **WHERE**!

Saya akan menjelaskannya lebih detail.

Tabel setelah gruping

SUM(price)	purchased_at
4	2018-10-10
3	2018-10-11
18	2018-10-12

Dapatkan baris yang sudah di grup dengan jumlah total lebih besar dari 10



HAVING

Jika Anda ingin mempersempit penyaringan hasil data yang dikelompokkan menggunakan **GROUP BY**, Anda dapat menggunakan **HAVING**. Sebagaimana ditampilkan pada contoh dibawah, kita bisa mendapatkan grup dengan kondisi tertentu dengan menggunakan syntax: **GROUP BY column_name HAVING kondisi**.

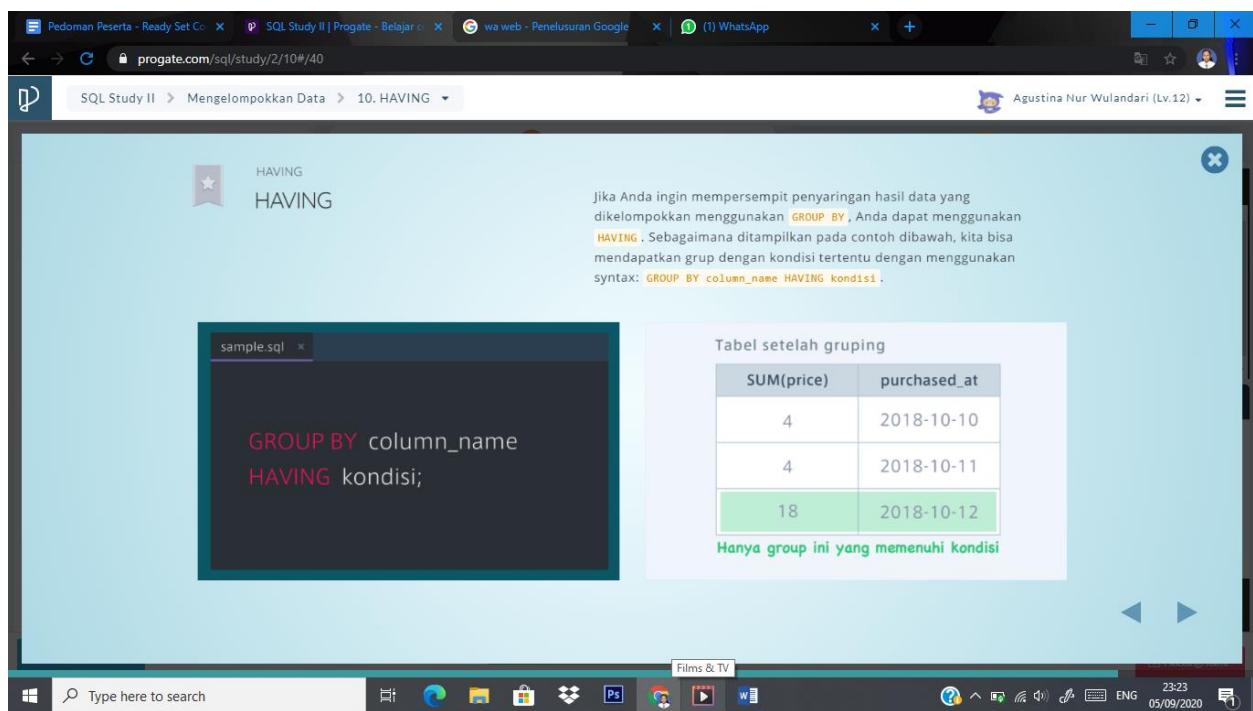
sample.sql

```
GROUP BY column_name  
HAVING kondisi;
```

Tabel setelah gruping

SUM(price)	purchased_at
4	2018-10-10
4	2018-10-11
18	2018-10-12

Hanya group ini yang memenuhi kondisi



Pedoman Peserta - Ready Set Co SQL Study II | Progate - Belajar wa web - Penelusuran Google WhatsApp

← → 🔍 progate.com/sql/study/2/10#41

SQL Study II > Mengelompokkan Data > 10. HAVING

Agustina Nur Wulandari (Lv.12)

HAVING

WHERE & HAVING

Untuk mempersempit data setelah pengelompokan, gunakanlah **HAVING**, bukan **WHERE**, karena SQL memiliki urutan specific dalam menjalankan kueri-nya. Sebagaimana ditampilkan pada gambar di bawah, **WHERE** dijalankan pertama, lalu **GROUP BY**, lalu **function**, dan **HAVING** dijalankan terakhir.

Pencarian	WHERE
Gruping	GROUP BY
Funtion	COUNT · SUM · AVG · MAX · MIN
HAVING	HAVING

← →

Pedoman Peserta - Ready Set Co SQL Study II | Progate - Belajar wa web - Penelusuran Google WhatsApp

← → 🔍 progate.com/sql/study/2/10#42

SQL Study II > Mengelompokkan Data > 10. HAVING

Agustina Nur Wulandari (Lv.12)

HAVING

Perbedaan Antara WHERE dan HAVING

Bergantung pada urutannya, **WHERE** dan **HAVING** mencari target yang berbeda. **WHERE** mencari seluruh tabel sebelum pengelompokan, sementara **HAVING** mencari grup yang dibuat dengan **GROUP BY**.

WHERE	HAVING
id name price	SUM(price) date
1 puding 1	4 2018-10-10
2 pena 1	4 2018-10-11
3 buku tulis 2	18 2018-10-12
4 puding 2	

Mencari keseluruhan tabel

Hanya mencari baris yang telah di grup

← →

Progate.com/sql/study/2/10#43

HAVING

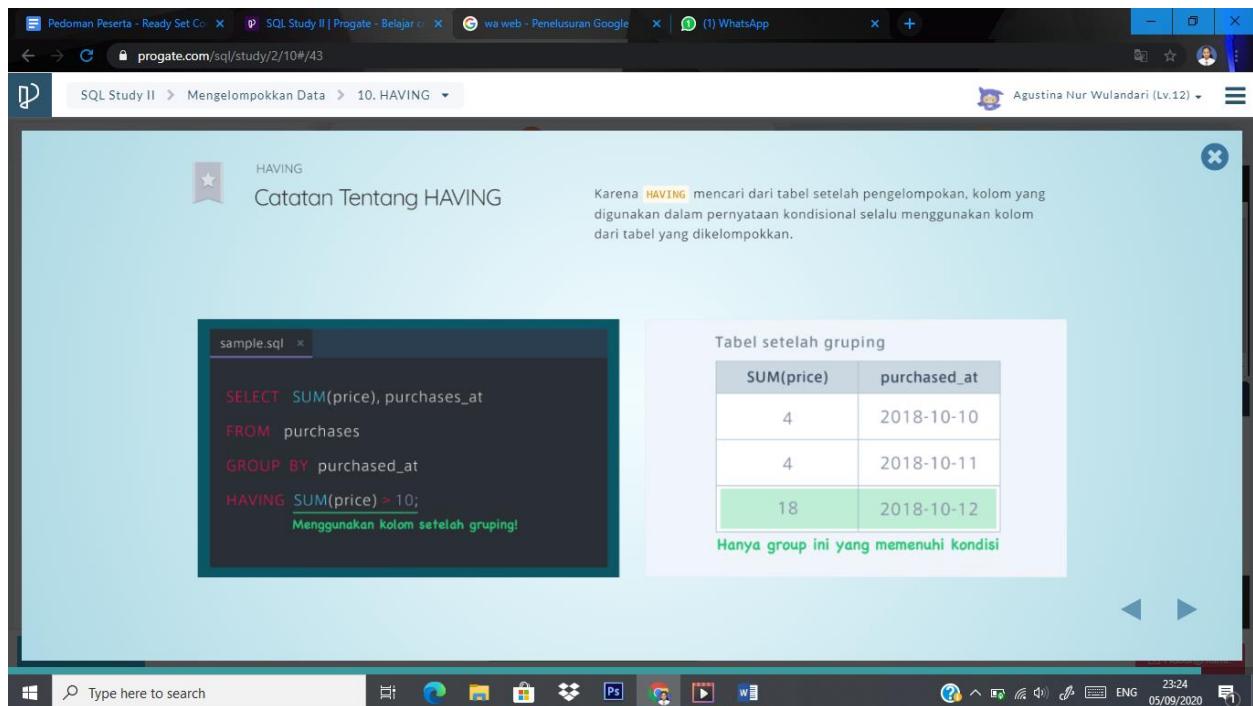
Catatan Tentang HAVING

Karena **HAVING** mencari dari tabel setelah pengelompokan, kolom yang digunakan dalam pernyataan kondisional selalu menggunakan kolom dari tabel yang dikelompokkan.

```
sample.sql x
SELECT SUM(price), purchases_at
FROM purchases
GROUP BY purchased_at
HAVING SUM(price) > 10;
Menggunakan kolom setelah gruping!
```

SUM(price)	purchased_at
4	2018-10-10
4	2018-10-11
18	2018-10-12

Hanya group ini yang memenuhi kondisi



Progate.com/sql/study/2/10#44

Instruksi

Dalam latihan ini, coba gunakan **GROUP BY** dan **HAVING** untuk mengelompokkan dan mempersempit hasil.

progate/exercise1.sql

Pertama, kelompokkan menurut kolom **purchased_at** dan dapatkan total kolom **price** serta kolom **purchased_at**. Setelah itu, gunakan **HAVING** untuk mencari grup yang total nilai pada kolom **price** lebih besar dari **20** saja (bersifat eksklusif).

Latihan

Database

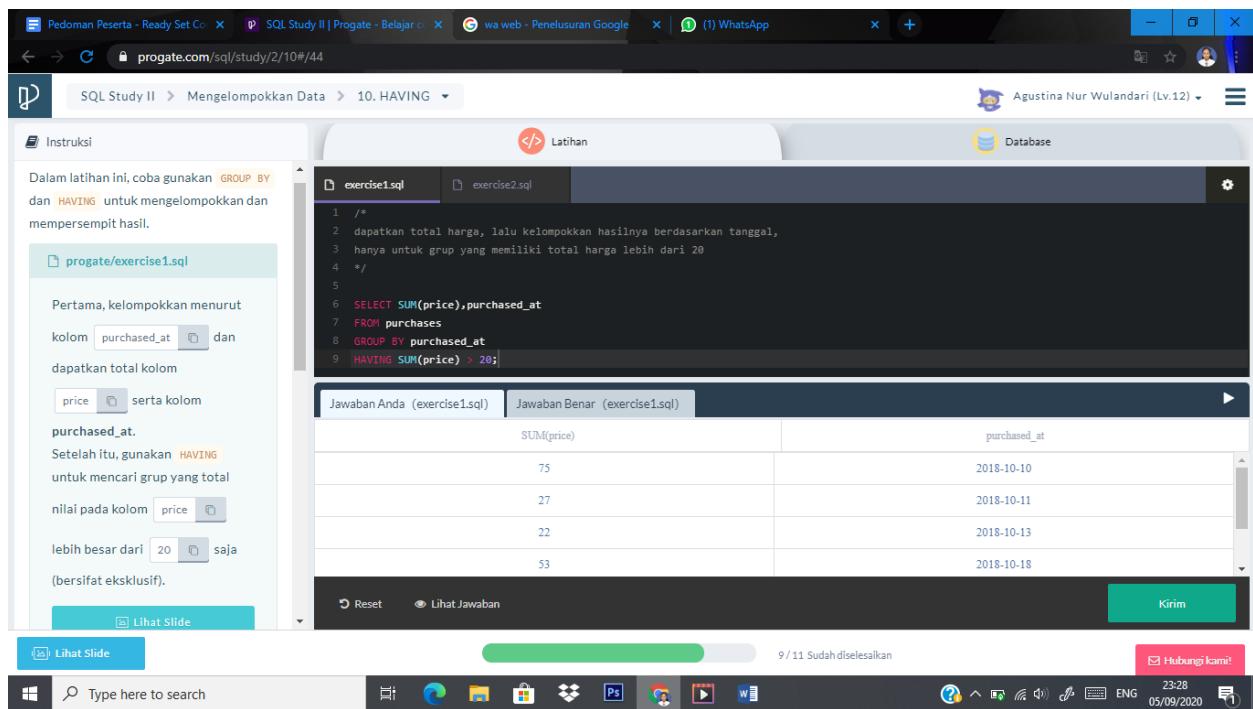
exercise1.sql

```
/*
1  dapatkan total harga, lalu kelompokkan hasilnya berdasarkan tanggal,
2  hanya untuk grup yang memiliki total harga lebih dari 20
3  */
4
5
6 SELECT SUM(price),purchased_at
7 FROM purchases
8 GROUP BY purchased_at
9 HAVING SUM(price) > 20;
```

Jawaban Anda (exercise1.sql) | Jawaban Benar (exercise1.sql)

SUM(price)	purchased_at
75	2018-10-10
27	2018-10-11
22	2018-10-13
53	2018-10-18

Reset | Lihat Jawaban | Kirim | Lihat Slide | Hubungi Kami! | 9/11 Sudah diselesaikan | 23:28 05/09/2020



Screenshot of a Progate SQL study session showing a completed exercise.

The exercise details:

```
-- dapatkan total berdasarkan tanggal dan karakter dimana totalnya lebih dari 30
SELECT SUM(price), purchased_at, character_name
FROM purchases
GROUP BY purchased_at, character_name
HAVING SUM(price) > 30;
```

The results table shows:

SUM(price)	purchased_at	character_name
35	2018-10-10	Ninja Ken
50	2018-10-18	Guru Domba
83	2018-10-20	Birdie
54	2018-11-07	Ninja Ken

Exercise status: 9 / 11 Sudah diselesaikan

Screenshot of a Progate SQL study session showing a completed exercise.

The exercise details:

```
-- d
2 SELECT
3   FROM
4   GROUP
5   HAVING
```

A modal window displays the "Level Up!" achievement for Level 13, featuring a cartoon owl character and social sharing buttons for Facebook and Twitter.

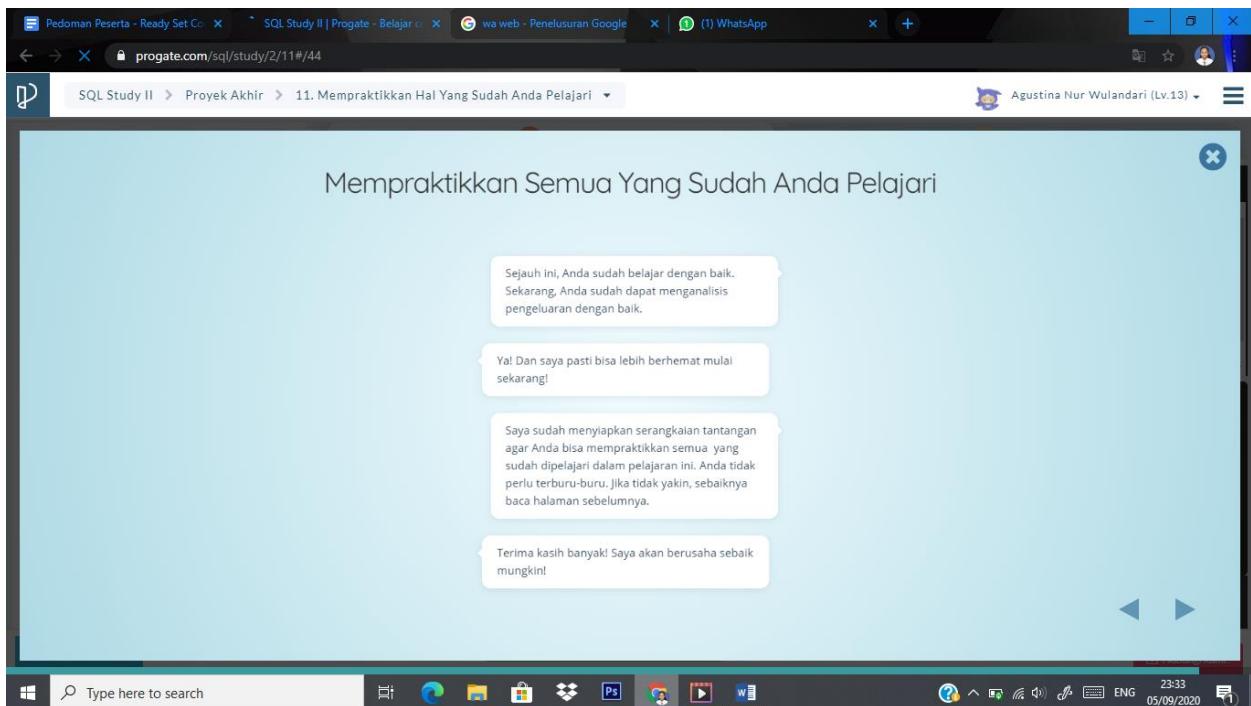
Level Up! Lv.13

Anda berhasil naik ke level yang baru!
Bagikan berita baik ini!

Kembali ke Latihan Berikutnya

Pelajaran berikutnya adalah "Mempraktikkan Hal Yang Sudah Anda Pelajari".

Exercise status: 10 / 11 Sudah diselesaikan



Instruksi

Anda berhasil sampai ke latihan akhir! Sejauh ini, Anda fokus pada pembelanjaan milik Ninja Ken. Namun, dalam latihan ini, kita akan menganalisis menurut kolom category.

progate/exercise1.sql

Buat grup menggunakan kolom category , lalu dapatkan total kolom price untuk setiap kategori beserta kolom category .

Lihat Slide

Latihan

Database

Jawaban Anda (exercise1.sql)	Jawaban Benar (exercise1.sql)	category
SUM(price)	298	lainnya
53	53	makanan
21	21	perjalanan

Reset Lihat Jawaban Kirim

Tekan tombol ► di jendela Jawaban Anda

Lihat Slide

Hubungi Kami!

Pedoman Peserta - Ready Set Co SQL Study II | Progate - Belajar wa web - Penelusuran Google (2) WhatsApp

← → 🔍 progate.com/sql/study/2/11#45

SQL Study II > Proyek Akhir > 11. Mempraktikkan Hal Yang Sudah Anda Pelajari

Agustina Nur Wulandari (Lv.13)

Instruksi

Latihan

Database

exercise1.sql exercise2.sql

```
1 -- buatlah sebuah query yang benar, korrekter
2 dan dapatkan total jumlah dari kolom price dan character_name
3 tetapi, aggregatkan hasil hanya untuk kolom category dengan nilai "lainnya"
4 */
5
6
7 SELECT SUM(price), character_name
8 FROM purchases
9 WHERE category = "lainnya"
10 GROUP BY character_name
11 ;
```

Jawaban Anda (exercise2.sql) Jawaban Benar (exercise2.sql)

SUM(price)	character_name
28	Baby Ben
124	Birdie
57	Guru Domba
89	Ninja Ken

Reset Lihat Jawaban Kirim

Tekan tombol ► di jendela Jawaban Anda

Lihat Slide Type here to search Hubungi Kami!

10 / 11 Sudah diselesaikan 23:46 05/09/2020

Pedoman Peserta - Ready Set Co SQL Study II | Progate - Belajar wa web - Penelusuran Google (2) WhatsApp

← → 🔍 progate.com/sql/study/2/11#45

SQL Study II > Proyek Akhir > 11. Mempraktikkan Hal Yang Sudah Anda Pelajari

Agustina Nur Wulandari (Lv.13)

Instruksi

Latihan

Database

Congratulations!

Selamat! Anda telah menyelesaikan SQL Study II!

Bagikan berita baik ini!

f t

Terima kasih telah menyelesaikan latihan ini! Berikutnya

Lanjutkan ➔

Tekan tombol ► di jendela Jawaban Anda

Lihat Slide Type here to search Hubungi Kami!

11 / 11 Sudah diselesaikan 23:46 05/09/2020