

10/1/2024

BASIS DATA

PERTEMUAN V

NAMA: RIFANDY ARNAS

NIM/NPM: 232310001

KELAS: TI – 23 – PA (LAB 2)

Tugas Pertemuan Ke-5
FUNCTION

Github: https://github.com/Rifandy232310001TeknologiInformasi/Basis_Data-Lab



















1. Duplikasi saat menggunakan fungsi **SUM()** dapat dihilangkan dengan menggunakan klausa **DISTINCT** di dalam fungsi **SUM()**. Klausa **DISTINCT** memastikan bahwa hanya nilai unik yang dijumlahkan.
2. **SUM()** digunakan untuk menjumlahkan suatu nilai-nilai numerik yang berasal dari suatu kolom ; Sedangkan **COUNT()** digunakan untuk menghitung jumlah record atau baris yang memenuhi kriteria tertentu.
3. **HAVING** adalah klausa dalam *SQL* yang digunakan untuk memfilter hasil setelah pengelompokan data dengan **GROUP BY**. Berbeda dengan **WHERE** yang digunakan sebelum pengelompokan, **HAVING** akan diterapkan setelah pengelompokan. Contohnya sebagai berikut!










```
SELECT column_name, SUM(some_column)
FROM table_name
GROUP BY column_name
HAVING SUM(some_column) > 100;
```

4. Membuat database **Raport** dengan record yang sesuai!

```
4
5 CREATE TABLE Siswa (
6     id INT AUTO_INCREMENT PRIMARY KEY,
7     nama VARCHAR(100)
8 );
9
10 CREATE TABLE Nilai (
11     id INT AUTO_INCREMENT PRIMARY KEY,
12     siswa_id INT,
13     mata_pelajaran VARCHAR(100),
14     nilai INT,
15     FOREIGN KEY (siswa_id) REFERENCES Siswa(id)
16 );
17
```

```
1 INSERT INTO Siswa (nama) VALUES ('Andi'), ('Budi'), ('Cici');
2
3 INSERT INTO Nilai (siswa_id, mata_pelajaran, nilai)
4 VALUES
5 (1, 'Matematika', 85),
6 (1, 'Biologi', 75),
7 (2, 'Matematika', 90),
8 (2, 'Biologi', 80),
9 (3, 'Matematika', 78),
10 (3, 'Biologi', 88);
11
```

<div><div><div>↩</div><div>T</div><div>→</div></div><div></div></div>				id	siswa_id	mata_pelajaran	nilai
<input type="checkbox"/>	 Edit	 Copy	 Delete	1	1	Matematika	85
<input type="checkbox"/>	 Edit	 Copy	 Delete	2	1	Biologi	75
<input type="checkbox"/>	 Edit	 Copy	 Delete	3	2	Matematika	90
<input type="checkbox"/>	 Edit	 Copy	 Delete	4	2	Biologi	80
<input type="checkbox"/>	 Edit	 Copy	 Delete	5	3	Matematika	78
<input type="checkbox"/>	 Edit	 Copy	 Delete	6	3	Biologi	88

<div>← T →</div>						id	nama	
<input type="checkbox"/>		Edit		Copy		Delete	1	Andi
<input type="checkbox"/>		Edit		Copy		Delete	2	Budi
<input type="checkbox"/>		Edit		Copy		Delete	3	Cici

5. Berikut ini adalah query untuk mendapatkan total dan rata-rata nilai siswa dengan nilai di atas 80!

```

1 SELECT siswa_id, SUM(nilai) AS total_nilai, AVG(nilai) AS rata_rata_nilai
2 FROM Nilai
3 WHERE nilai > 80
4 GROUP BY siswa_id;
5 |

```

<div><div><div></div><div></div><div></div></div></div>				siswa_id	total_nilai	rata_rata_nilai
<div><div><div></div></div></div>	<div><div><div></div></div></div> Edit	<div><div><div></div></div></div> Copy	<div><div><div></div></div></div> Delete	1	85	85.0000
<div><div><div></div></div></div>	<div><div><div></div></div></div> Edit	<div><div><div></div></div></div> Copy	<div><div><div></div></div></div> Delete	2	90	90.0000
<div><div><div></div></div></div>	<div><div><div></div></div></div> Edit	<div><div><div></div></div></div> Copy	<div><div><div></div></div></div> Delete	3	88	88.0000

6. Berikut ini adalah query untuk menampilkan nama siswa dengan nilai tertinggi!

```

1 SELECT S.nama, MAX(N.nilai) AS nilai_tertinggi
2 FROM Nilai N
3 JOIN Siswa S ON N.siswa_id = S.id
4 GROUP BY S.nama
5 ORDER BY nilai_tertinggi DESC
6 LIMIT 1;
7 |

```







nama	nilai_tertinggi
Budi	90

7. Berikut ini adalah query untuk menampilkan rata-rata nilai dari semua matapelajaran dengan rata-rata lebih dari 75!

```

1 SELECT mata_pelajaran, AVG(nilai) AS rata_rata_nilai
2 FROM Nilai
3 GROUP BY mata_pelajaran
4 HAVING AVG(nilai) > 75;
5 |

```

<div><div>↩️</div><div>T</div><div>➡️</div></div>					mata_pelajaran	rata_rata_nilai		
<input type="checkbox"/>		Edit		Copy		Delete	Matematika	84.3333
<input type="checkbox"/>		Edit		Copy		Delete	Biologi	81.0000

8. Berikut ini adalah query untuk menghitung total nilai siswa dalam setiap matapelajaran!

```

1 SELECT S.nama, N.mata_pelajaran, SUM(N.nilai) AS total_nilai
2 FROM Nilai N
3 JOIN Siswa S ON N.siswa_id = S.id
4 GROUP BY S.nama, N.mata_pelajaran;
5

```

nama	mata_pelajaran	total_nilai
Andi	Matematika	85
Andi	Biologi	75
Budi	Matematika	90
Budi	Biologi	80
Cici	Matematika	78
Cici	Biologi	88

9. Berikut ini adalah query untuk menampilkan jumlah matapelajaran!

```

1 SELECT COUNT(DISTINCT mata_pelajaran) AS jumlah_mata_pelajaran
2 FROM Nilai;
3

```

jumlah_mata_pelajaran

2

10. Berikut ini adalah query untuk menampilkan nama siswa dengan nilai terendah pada matapelajaran Biologi!

```

1 SELECT S.nama, N.nilai
2 FROM Nilai N
3 JOIN Siswa S ON N.siswa_id = S.id
4 WHERE N.mata_pelajaran = 'Biologi'
5 ORDER BY N.nilai ASC
6 LIMIT 1;
7

```

nama	nilai
Andi	75