

LAPORAN SISTEM PENGAJIAN SMP IT DAARUL MUTTAQIN

Laporan ini disusun untuk memenuhi Ujian Akhir Semester Mata
Kuliah Basis Data (G)



Disusun Oleh:

Kelompok 6

Waode Fairuzh Ramadhani Somandeno	(4523210111)
Dian Ayu Azizah	(4523210140)
Nenden Nuraini	(4523210144)

Dosen Pengampu:

Adi Wahyu Pribadi, S.Si., M.Kom.

**PROGRAM STUDI S1 TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PANCASILA
2024/2025**

1. Rancang Database

Laporan Pertanggungjawaban Insentif Guru									
No	Nama	Honor			Transport			Jabatan	
		Nilai	Jumlah Jam	Jumlah	Nilai	Kehadiran	Jumlah	Nama	Jumlah
1	Eudi Pathurohman, S.Pd., Gr.	Rp 17.500	16	Rp 280.000	Rp 30.000	8	Rp 240.000	Wali Kelas IX C	Rp 100.000
2	Yanti Susanti, S.Pd.	Rp 17.500	32	Rp 560.000	Rp 30.000	20	Rp 600.000	Wali Kelas VIII D	Rp 100.000
3	Muhamad Salim	Rp 17.500	32	Rp 560.000	Rp 30.000	20	Rp 600.000	Waka Kurikulum	Rp 400.000
4	Dede Hamdani, S.Pd.	Rp 17.500	16	Rp 280.000	Rp 30.000	8	Rp 240.000	Wali Kelas IX A	Rp 100.000
5	Usep Supriatna, S.Pd.	Rp 17.500	16	Rp 280.000	Rp 30.000	12	Rp 360.000		
6	Iyul Yulinar, S.Pd.	Rp 17.500	32	Rp 560.000	Rp 30.000	20	Rp 600.000	Wali Kelas VIII A	Rp 100.000
7	Hadi Pamungkas, S.Pd.	Rp 17.500	40	Rp 700.000	Rp 30.000	20	Rp 600.000	Staf Keselamatan	Rp 250.000
8	Rani Dwi Rahayu, S. Pd.	Rp 17.500	24	Rp 420.000	Rp 30.000	20	Rp 600.000	Wali Kelas IX B	Rp 100.000
9	Ajat Murajat, S.Pd.	Rp 17.500	32	Rp 560.000	Rp 30.000	15	Rp 450.000	Pembina OSIS	Rp 150.000
10	Widy Mauliddina, S.Sn	Rp 17.500	24	Rp 420.000	Rp 30.000	20	Rp 600.000	Waka Keselamatan	Rp 400.000
11	Ria Elviana, M.Pd.	Rp 17.500	16	Rp 280.000	Rp 30.000	8	Rp 240.000		
12	Mohammad Tohirin, S.Pd.	Rp 17.500	8	Rp 140.000	Rp 30.000	8	Rp 240.000		
13	Diri Aprilia Ghaisani H, S.Pd.	Rp 17.500	16	Rp 280.000	Rp 30.000	8	Rp 240.000	Wali Kelas VII D	Rp 100.000
14	Wahyu Rustan Dityatman, S.Pd.	Rp 17.500	32	Rp 560.000	Rp 30.000	20	Rp 600.000	Wali Kelas VII B	Rp 100.000
15	Desky Jurdilah Nur El Haq, S.Pd., M.Pd.	Rp 17.500	24	Rp 420.000	Rp 30.000	16	Rp 480.000		
16	Mulyana Chandra, S.Pd.	Rp 17.500	24	Rp 420.000	Rp 30.000	7	Rp 210.000	Wali Kelas VIII C	Rp 100.000
17	Susan Susana Widiana, S.Pd.	Rp 17.500	24	Rp 420.000	Rp 30.000	11	Rp 330.000		
18	Fadhra Yuniarti Maulida, S.Sn	Rp 17.500	24	Rp 420.000	Rp 30.000	16	Rp 480.000	Wali Kelas VIII B	Rp 100.000
19	Embi Lila, S.Si., Gr.	Rp 17.500	16	Rp 280.000	Rp 30.000	7	Rp 210.000		
20	Muhamad Sulten Yasar, S.Pd.	Rp 17.500	16	Rp 280.000	Rp 30.000	10	Rp 300.000	Wali Kelas IX D	Rp 100.000
Total									Rp 18.540.000

Gambar 1. Data Asli Penggajian SMP IT Daarul Muttaqin

A. Identifikasi entitas utama, atribut, dan relasi antar entitas

Setelah melakukan wawancara dan mendapatkan data aslinya setelah dipelajari kami menyimpulkan.

1) Entitas Utama

1. Karyawan

Atribut:

- ID_Karyawan int: nomor unik yang mengidentifikasi setiap karyawan.
- Nama varchar(100): nama lengkap karyawan.
- NUPTK char 16: nomor induk karyawan.

2. Jabatan

Atribut:

- ID_Jabatan int: nomor unik yang mengidentifikasi setiap jabatan.
- Nama_Jabatan varchar(100): nama jabatan.
- Jumlah_Honor decimal(10,2): jumlah honor jabatan.

3. Transport

Atribut:

- ID_Transport: nomor unik yang mengidentifikasi transport.
- Nilai_Per_Hari: nilai atau biaya transport per hari.

4. Honor

Atribut:

- ID_Honor: nomor unik yang mengidentifikasi honor.
- Nilai_Per_Jam: nilai honor per jam.

2) Entitas Relasi

1. Penggajian

Atribut:

- ID_Penggajian int: nomor unik yang mengidentifikasi setiap proses penggajian.
- Tanggal_Penggajian date: tanggal penggajian dilakukan.
- Total_Gaji decimal(10,2): total gaji yang diterima.

3) Relasi Antar Entitas

1. Karyawan – Penggajian:

- Relasi: setiap karyawan dapat menerima satu atau lebih penggajian
- Tipe Relasi: *One-to-Many* (satu karyawan bisa memiliki banyak data penggajian)

2. Jabatan – Penggajian:

- Relasi: setiap jabatan menentukan nominal penggajian untuk karyawan di jabatan tersebut
- Tipe - Relasi: *One-to-Many* (satu jabatan dapat memengaruhi banyak penggajian).

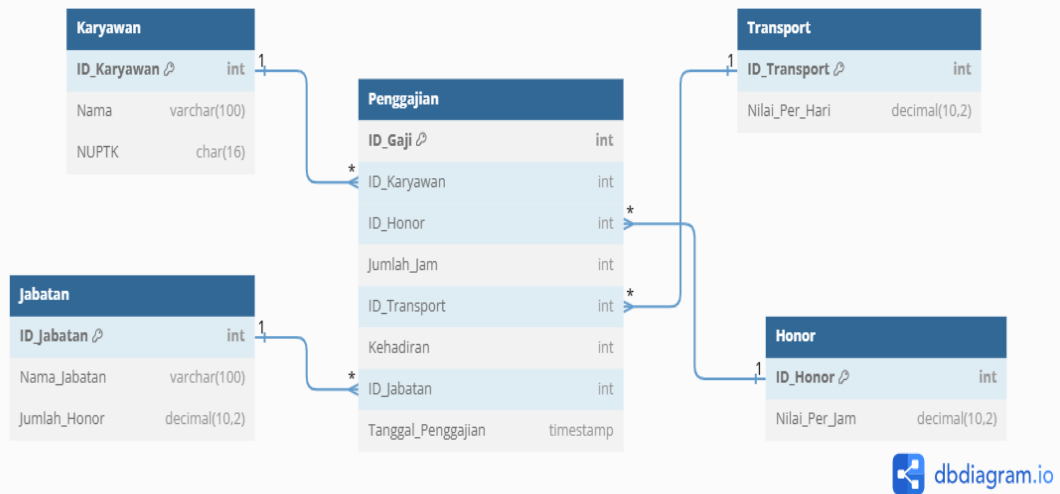
3. Transport -Penggajian

- Relasi: setiap jabatan menentukan nominal penggajian untuk karyawan di jabatan tersebut
- Tipe - Relasi: *One-to-Many* (satu Transport dapat memengaruhi banyak penggajian).

4. Honor -Penggajian

- Relasi: setiap jabatan menentukan nominal penggajian untuk karyawan di jabatan tersebut
- Tipe - Relasi: *One-to-Many* (satu Honor dapat memengaruhi banyak penggajian).

B. ERD untuk menggambarkan struktur *Database* Penggajian



Untuk gambaran dari penyusunan ERD dari analisis kami di atas jadi seperti ini, untuk tabel perhitungan gaji total dan laporan itu tidak kamu masukan kedalam rancangan *Database* karena bisa menggunakan prinsip join. Jadi disini fokus konsistensi sidatanya. Untuk laporan detail Penggajian per Karyawan dan Laporan Total Penggajian per bulan kami buat di bagian 2 implementasi *Database* dengan join.

2. Implementasi *Database*

A. Proses membuat *Database*, tabel, relasi, hingga input data *Update*, dan *Delete* menggunakan MySQL, InnoDB, dan UTF8.

1. Membuat *Database*

Run SQL query/queries on server "localhost":

```

1 CREATE DATABASE Penggajian_DMC
2 CHARACTER SET utf8mb4
3 COLLATE utf8mb4_general_ci;
4
        
```

☐ Bind parameters

Delimiter:
☐ Show this query here again
 ☐ Retain query box
 ☐ Rollback when finished

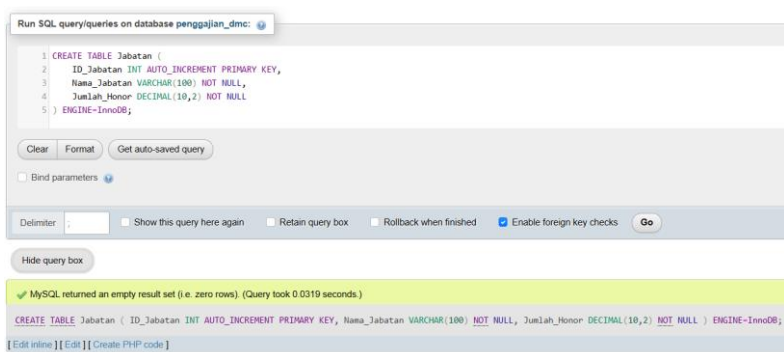
✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0125 seconds.)

[CREATE DATABASE Penggajian_DMC](#)
[CHARACTER SET utf8mb4](#)
[COLLATE utf8mb4_general_ci](#);

[\[Edit inline \]](#)
[\[Edit \]](#)
[\[Create PHP code \]](#)

	<pre>mysql> show databases; +-----+ Database +-----+ desty5_db information_schema mysql penggajian_dmc performance_schema sys +-----+ 6 rows in set (0.00 sec)</pre> <p>Query:</p> <pre>CREATE DATABASE Penggajian_DMC CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;</pre> <p>Penjelasan:</p> <p>Sesuai perintah, kami menggunakan MySQL untuk pembuatan <i>Database</i>, dan penyimpanan dengan InnoDB dan UTF8.</p>
<p>2. Membuat Tabel Karyawan</p>	 <p>The screenshot shows a MySQL query editor interface. The query entered is: <code>CREATE TABLE karyawan (ID_Karyawan INT AUTO_INCREMENT PRIMARY KEY, Nama VARCHAR(100) NOT NULL, NUPTK CHAR(16) UNIQUE NOT NULL) ENGINE=InnoDB;</code>. The query has been executed successfully, and the result shows: "MySQL returned an empty result set (i.e. zero rows). (Query took 0.0417 seconds)".</p> <p>Query:</p> <pre>CREATE TABLE karyawan (ID_Karyawan INT AUTO_INCREMENT PRIMARY KEY, Nama VARCHAR(100) NOT NULL, NUPTK CHAR(16) UNIQUE NOT NULL) ENGINE=InnoDB;</pre> <p>Penjelasan:</p> <p>Pembuatan tabel ‘Karyawan’ ini <i>primary key</i> ID_Karyawan menggunakan auto_increment agar dapat otomatis tidak perlu meninput ID, nama menggunakan tipe data varchar dengan ketentuan not null/harus diisi, serta NUPTK unique agar jika ada nama yang sama NUPTK tetap harus beda.</p>

3. Membuat Tabel Jabatan



```
1 CREATE TABLE Jabatan (
2   ID_Jabatan INT AUTO_INCREMENT PRIMARY KEY,
3   Nama_Jabatan VARCHAR(100) NOT NULL,
4   Jumlah_Honor DECIMAL(10,2) NOT NULL
5 ) ENGINE=InnoDB;
```

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0319 seconds.)

```
CREATE TABLE Jabatan ( ID_Jabatan INT AUTO_INCREMENT PRIMARY KEY, Nama_Jabatan VARCHAR(100) NOT NULL, Jumlah_Honor DECIMAL(10,2) NOT NULL ) ENGINE=InnoDB;
```

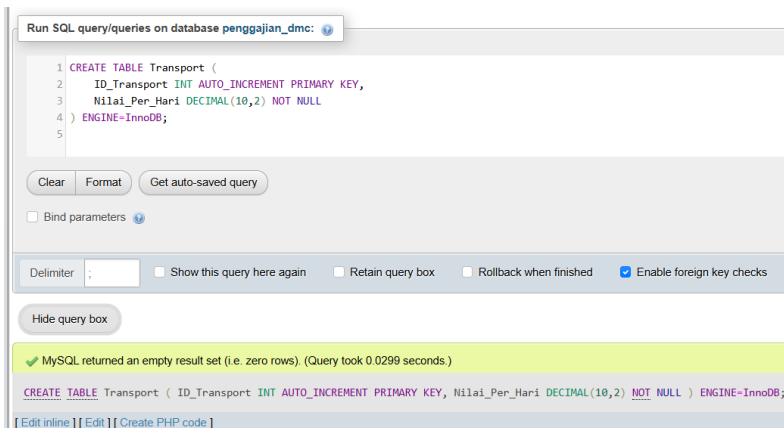
Query:

```
CREATE TABLE Jabatan ( ID_Jabatan INT AUTO_INCREMENT PRIMARY KEY, Nama_Jabatan VARCHAR(100) NOT NULL, Jumlah_Honor DECIMAL(10,2) NOT NULL ) ENGINE=InnoDB;
```

Penjelasan:

Pada tabel ‘Jabatan’ ID_Jabatan merupakan *primary key* yang *auto increment* sehingga tidak perlu menginput ID secara manual, nama menggunakan tipe data varchar dengan ketentuan harus terisi/*not null*, serta Jumlah_Honor dengan tipe data decimal karena menyangkut keuangan.

4. Membuat Tabel Transport



```
1 CREATE TABLE Transport (
2   ID_Transport INT AUTO_INCREMENT PRIMARY KEY,
3   Nilai_Per_Hari DECIMAL(10,2) NOT NULL
4 ) ENGINE=InnoDB;
```

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0299 seconds.)

```
CREATE TABLE Transport ( ID_Transport INT AUTO_INCREMENT PRIMARY KEY, Nilai_Per_Hari DECIMAL(10,2) NOT NULL ) ENGINE=InnoDB;
```

Query:

```
CREATE TABLE Transport ( ID_Transport INT AUTO_INCREMENT PRIMARY KEY, Nilai_Per_Hari DECIMAL(10,2) NOT NULL ) ENGINE=InnoDB;
```

Penjelasan:

Tabel ‘Transport’ juga memiliki ID_Transport dengan *primary key* yang *auto increment* sehingga tanpa perlu meninput manual, dan Nilai_Per_Hari menggunakan tipe data decimal karena menyimpan data keuangan agar tidak ada kesalahan kalau dilakukan pembulatan.

5. Membuat Tabel Honor

```
Run SQL query/queries on database penggajian_dmc:

1 CREATE TABLE Honor (
2   ID_Honor INT AUTO_INCREMENT PRIMARY KEY,
3   Nilai_Per_Jam DECIMAL(10,2) NOT NULL
4 ) ENGINE=InnoDB;
5

Clear Format Get auto-saved query

☐ Bind parameters

Delimiter ; ☐ Show this query here again ☐ Retain query box ☐ Rollback when finished ☒ Enable foreign key checks

Hide query box

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0320 seconds.)

CREATE TABLE Honor ( ID_Honor INT AUTO_INCREMENT PRIMARY KEY, Nilai_Per_Jam DECIMAL(10,2) NOT NULL ) ENGINE=InnoDB;

[ Edit inline ] [ Edit ] [ Create PHP code ]
```

Query:

```
CREATE TABLE Honor ( ID_Honor INT AUTO_INCREMENT PRIMARY KEY, Nilai_Per_Jam DECIMAL(10,2) NOT NULL ) ENGINE=InnoDB;
```

Penjelasan:

Pada tabel ‘Honor’ juga *primary key*-nya menggunakan *auto_increment* agar tidak perlu mengisi ID secara manual, dan atribut Nilai_Per_Jam dengan tipe data varchar yang harus diisi/*not null*.

6. Membuat Tabel Penggajian

```
Run SQL query/queries on database penggajian_dmc:

1 CREATE TABLE Penggajian (
2   ID_Gaji INT AUTO_INCREMENT PRIMARY KEY,
3   ID_Karyawan INT NOT NULL,
4   ID_Honor INT NOT NULL,
5   Jumlah_Jam INT NOT NULL,
6   ID_Transport INT NOT NULL,
7   Kehadiran INT NOT NULL,
8   ID_Jabatan INT,
9   Tanggal_Penggajian TIMESTAMP NOT NULL,
10  FOREIGN KEY (ID_Karyawan) REFERENCES Karyawan(ID_Karyawan) ON DELETE CASCADE ON UPDATE CASCADE,
11  FOREIGN KEY (ID_Honor) REFERENCES Honor(ID_Honor) ON DELETE CASCADE ON UPDATE CASCADE,
12  FOREIGN KEY (ID_Transport) REFERENCES Transport(ID_Transport) ON DELETE CASCADE ON UPDATE CASCADE,
13  FOREIGN KEY (ID_Jabatan) REFERENCES Jabatan(ID_Jabatan) ON DELETE CASCADE ON UPDATE CASCADE
14 ) ENGINE=InnoDB;
15

Clear Format Get auto-saved query

☐ Bind parameters

Delimiter ; ☐ Show this query here again ☐ Retain query box ☐ Rollback when finished ☒ Enable foreign key checks

Hide query box

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.1048 seconds.)

CREATE TABLE Penggajian ( ID_Gaji INT AUTO_INCREMENT PRIMARY KEY, ID_Karyawan INT NOT NULL, ID_Honor INT NOT NULL, Jumlah_Jam INT NOT NULL, ID_Transport INT NOT NULL, Kehadiran INT NOT NULL, ID_Jabatan INT, Tanggal_Penggajian TIMESTAMP NOT NULL, FOREIGN KEY (ID_Karyawan) REFERENCES Karyawan(ID_Karyawan) ON DELETE CASCADE ON UPDATE CASCADE, FOREIGN KEY (ID_Honor) REFERENCES Honor(ID_Honor) ON DELETE CASCADE ON UPDATE CASCADE, FOREIGN KEY (ID_Transport) REFERENCES Transport(ID_Transport) ON DELETE CASCADE ON UPDATE CASCADE, FOREIGN KEY (ID_Jabatan) REFERENCES Jabatan(ID_Jabatan) ON DELETE CASCADE ON UPDATE CASCADE ) ENGINE=InnoDB;

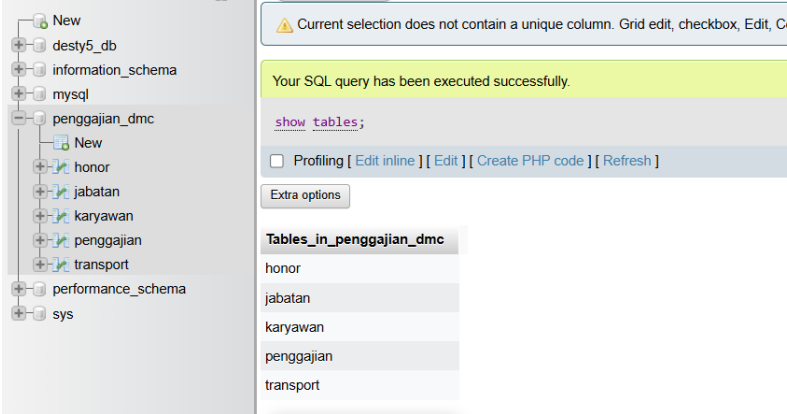
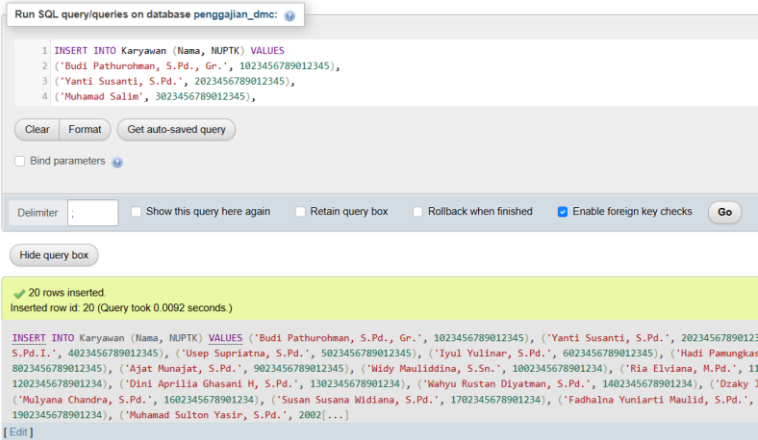
[ Edit inline ] [ Edit ] [ Create PHP code ]
```

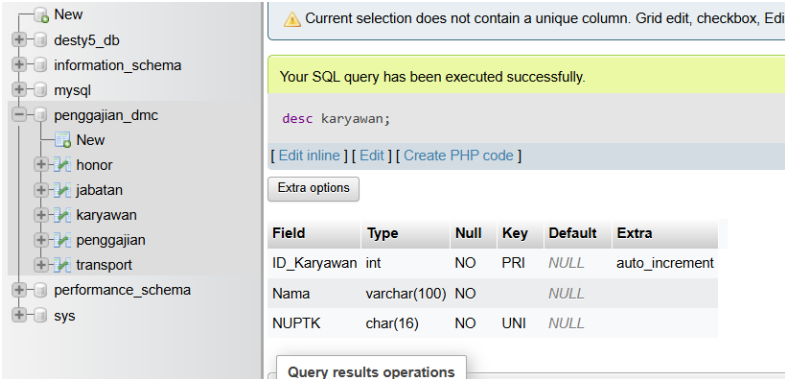
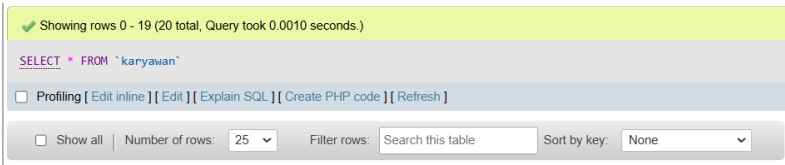
Query:

```
CREATE TABLE Penggajian ( ID_Gaji INT AUTO_INCREMENT PRIMARY KEY, ID_Karyawan INT NOT NULL, ID_Honor INT NOT NULL, Jumlah_Jam INT NOT NULL, ID_Transport INT NOT NULL, Kehadiran INT NOT NULL, ID_Jabatan INT, Tanggal_Penggajian TIMESTAMP NOT NULL, FOREIGN KEY (ID_Karyawan) REFERENCES Karyawan(ID_Karyawan) ON DELETE CASCADE ON UPDATE CASCADE, FOREIGN KEY (ID_Honor) REFERENCES Honor(ID_Honor) ON DELETE CASCADE ON UPDATE CASCADE, FOREIGN KEY (ID_Transport) REFERENCES Transport(ID_Transport) ON DELETE CASCADE ON UPDATE CASCADE, FOREIGN KEY (ID_Jabatan) REFERENCES Jabatan(ID_Jabatan) ON DELETE CASCADE ON UPDATE CASCADE ) ENGINE=InnoDB;
```

Penjelasan:

Tabel ‘Penggajian’ juga memiliki ID_Penggajian yang merupakan *primary key* yang *auto_increment*. Tabel ini

	<p>memiliki <i>foreign key</i> dengan tabel ‘Karyawan’, ‘Honor’, ‘Transport’, dan ‘Jabatan’. Lalu sesuai dengan yang kami pelajari di kelas, kami juga akan menambahkan <i>cascade</i> agar jika induknya dihapus maka data anaknya juga ikut terhapus.</p>
<p>7. Menunjuk kan tabel pada Database Penggajian_DMC</p>	 <p>Query: <code>show tables;</code></p> <p>Penjelasan: Perintah di atas untuk menunjukkan apa saja tabel yang ada pada Database Penggajian_DMC.</p>
<p>8. Insert values ke tabel Karyawan</p>	 <p>Query:</p> <pre>INSERT INTO Karyawan (Nama, NUPTK) VALUES ('Budi Pathurohman, S.Pd., Gr.', 1023456789012345), ('Yanti Susanti, S.Pd.', 2023456789012345), ('Muhamad Salim', 3023456789012345), ('Dede Hamdani, S.Pd.I.', 4023456789012345), ('Usep Supriatna, S.Pd.', 5023456789012345), ('Iyul Yulinar, S.Pd.', 6023456789012345), ('Hadi Pamungkas, S.Pd.', 7023456789012345), ('Rani Dwi Rahayu, S.Pd.', 8023456789012345), ('Ajat Munajat, S.Pd.', 9023456789012345), ('Widy Mauliddina, S.Sn.', 1002345678901234), ('Ria Elviana, M.Pd.', 1102345678901234), ('Mohammad Tohirin, S.Pd.', 1202345678901234), ('Dini Aprilia Ghasani H, S.Pd.', 1302345678901234), ('Wahyu Rustan Diyatman, S.Pd.', 1402345678901234), ('Dzaky Mulyana Chandra, S.Pd.', 1602345678901234), ('Susan Susana Widiana, S.Pd.', 1702345678901234), ('Fadhaina Yuniarti Maulid, S.Pd.', 1902345678901234), ('Muhamad Sulton Yasir, S.Pd.', 2002345678901234);</pre>

	<pre>('Wahyu Rustan Diyatman, S.Pd.', 1402345678901234), ('Dzaky Jundullah Nur El Haq, S.Pd., M.Pd.', 1502345678901234), ('Mulyana Chandra, S.Pd.', 1602345678901234), ('Susan Susana Widianana, S.Pd.', 1702345678901234), ('Fadhalna Yuniarti Maulid, S.Pd.', 1802345678901234), ('Embi Lilis, S.Si., Gr.', 1902345678901234), ('Muhamad Sulton Yasir, S.Pd.', 2002345678901234);</pre> <p>Penjelasan: Perintah <i>INSERT</i> dilakukan untuk penambahan data pada setiap <i>fields</i> yang ada pada tabel Karyawan.</p>
9. Melihat field tabel Karyawan	 <p>Query: <code>desc karyawan;</code></p> <p>Penjelasan: Perintah '<i>desc</i>' di sini kami gunakan untuk melihat <i>fields</i> apa saja yang ada pada tabel Karyawan.</p>
10. <i>Select</i> untuk melihat <i>Insert</i> yang sudah ditambahkan	

Server: localhost > Database: pengajaran_dmc > Table: karyawan

Browse Structure SQL Search Insert Export Import Privileges Operations

ID_Karyawan	Nama	NUPTK
<input type="checkbox"/> Edit Copy Delete	1 Budi Pathurohman, S.Pd., Gr.	1023456789012345
<input type="checkbox"/> Edit Copy Delete	2 Yanti Susanti, S.Pd.	2023456789012345
<input type="checkbox"/> Edit Copy Delete	3 Muhamad Salim	3023456789012345
<input type="checkbox"/> Edit Copy Delete	4 Dede Hamdani, S.Pd.I.	4023456789012345
<input type="checkbox"/> Edit Copy Delete	5 Usep Supriatna, S.Pd.	5023456789012345
<input type="checkbox"/> Edit Copy Delete	6 Iyul Yulinar, S.Pd.	6023456789012345
<input type="checkbox"/> Edit Copy Delete	7 Hadi Pamungkas, S.Pd.	7023456789012345
<input type="checkbox"/> Edit Copy Delete	8 Rani Dwi Rahayu, S.Pd.	8023456789012345
<input type="checkbox"/> Edit Copy Delete	9 Ajat Munajat, S.Pd.	9023456789012345
<input type="checkbox"/> Edit Copy Delete	10 Widy Mauliddina, S.Sn.	1002345678901234
<input type="checkbox"/> Edit Copy Delete	11 Ria Elviana, M.Pd.	1102345678901234
<input type="checkbox"/> Edit Copy Delete	12 Mohammad Tohirin, S.Pd.	1202345678901234
<input type="checkbox"/> Edit Copy Delete	13 Dini Aprilia Ghasani H, S.Pd.	1302345678901234
<input type="checkbox"/> Edit Copy Delete	14 Wahyu Rustan Diyatman, S.Pd.	1402345678901234
<input type="checkbox"/> Edit Copy Delete	15 Dzaky Jundullah Nur El Haq, S.Pd., M.Pd.	1502345678901234
<input type="checkbox"/> Edit Copy Delete	16 Mulyana Chandra, S.Pd.	1602345678901234
<input type="checkbox"/> Edit Copy Delete	17 Susan Susana Widiana, S.Pd.	1702345678901234
<input type="checkbox"/> Edit Copy Delete	18 Fadhalna Yuniarti Maulid, S.Pd.	1802345678901234
<input type="checkbox"/> Edit Copy Delete	19 Embi Lilis, S.Si., Gr.	1902345678901234
<input type="checkbox"/> Edit Copy Delete	20 Muhamad Sultan Yasir, S.Pd.	2002345678901234

Query: `SELECT * FROM `karyawan``

Penjelasan:

Perintah ‘*SELECT*’ ini kami gunakan untuk melihat seluruh *Insert* yang sudah ditambahkan sebelumnya.

11. *Insert* values ke tabel *Jabatan*

Server: localhost > Database: pengajaran_dmc > Table: karyawan

Browse Structure SQL Search Insert Export Import Privileges Operations Triggers

```

1 INSERT INTO jabatan (Nama_Jabatan, Jumlah_Honor) VALUES
2 ('Wali Kelas IX C', 100000),
3 ('Wali Kelas VIII D', 100000),
4 ('Waka Kurikulum', 400000),
5 ('Wali Kelas IX A', 100000),
6 ('Wali Kelas VIII A', 100000),
7 ('Staf Kesiswaan', 250000),
8 ('Wali Kelas IX B', 100000),
9 ('Pembina OSIS', 150000),
10 ('Waka Kesiswaan', 400000),
11 ('Wali Kelas VII D', 100000),
12 ('Wali Kelas VII B', 100000),
13 ('Wali Kelas VIII C', 100000),
14 ('Wali Kelas VIII B', 100000)

```

SELECT * SELECT INSERT UPDATE DELETE Clear Format Get auto-saved query

☐ Bind parameters

Delimiter: Show this query here again Retain query box Rollback when finished ☒ Enable foreign key checks Go

Hide query box

✓ 20 rows inserted
Inserted row id: 20 (Query took 0.0054 seconds)

```

INSERT INTO jabatan (Nama_Jabatan, Jumlah_Honor) VALUES ('Wali Kelas IX C', 100000), ('Wali Kelas VIII D', 100000), ('Waka Kurikulum', 400000), ('Staf Kesiswaan', 250000), ('Pembina OSIS', 150000), ('Waka Kesiswaan', 400000), ('Wali Kelas VIII C', 100000), ('Wali Kelas VIII B', 100000), ('Wali Kelas IX D', 100000), ('Wali Kelas VII C', 100000), ('Kepala Sekolah', 500000), ('Guru Mata Pelajaran Bahasa Indonesia', 300000), ('Guru Mata Pelajaran IPA', 300000);

```

[Edit inline] [Edit] [Create PHP code]

Query:

```

INSERT INTO jabatan (Nama_Jabatan, Jumlah_Honor) VALUES ('Wali Kelas IX C', 100000), ('Wali Kelas VIII D', 100000), ('Waka Kurikulum', 400000), ('Wali Kelas IX A', 100000), ('Wali Kelas VIII A', 100000), ('Staf Kesiswaan', 250000), ('Wali Kelas IX B', 100000), ('Pembina OSIS', 150000), ('Waka Kesiswaan', 400000), ('Wali Kelas VII D', 100000), ('Wali Kelas VII B', 100000), ('Wali Kelas VIII C', 100000), ('Wali Kelas VIII B', 100000), ('Wali Kelas IX D', 100000), ('Wali Kelas VII C', 100000), ('Kepala Sekolah', 500000), ('Wali Kelas VII C', 100000), ('Kepala Sekolah', 500000), ('Wali Kelas VII

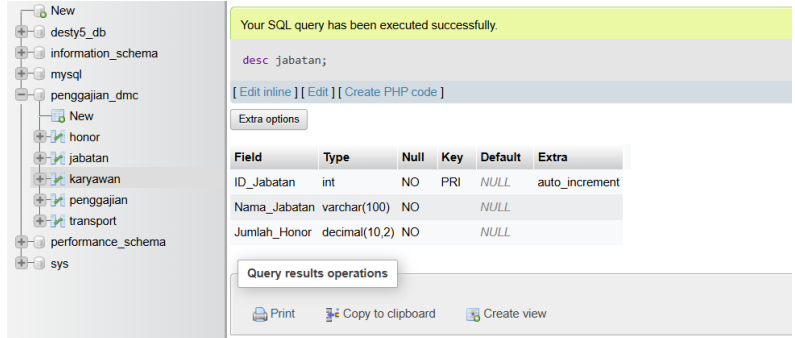
```

```
A', 100000), ('Guru Mata Pelajaran Matematika', 300000), ('Guru Mata Pelajaran Bahasa Indonesia', 300000), ('Guru Mata Pelajaran IPA', 300000);
```

Penjelasan:

Di sini kami melakukan perintah '*INSERT*' untuk menginput data ke tabel Jabatan. Kami menginput sebanyak 20 data untuk masing-masing *fields* (Nama_Jabatan dan Jumlah_Honor).

12. Melihat *fields* tabel Jabatan



Your SQL query has been executed successfully.

```
desc jabatan;
```

[Edit inline] [Edit] [Create PHP code]

Extra options

Field	Type	Null	Key	Default	Extra
ID_Jabatan	int	NO	PRI	NULL	auto_increment
Nama_Jabatan	varchar(100)	NO		NULL	
Jumlah_Honor	decimal(10,2)	NO		NULL	

Query results operations

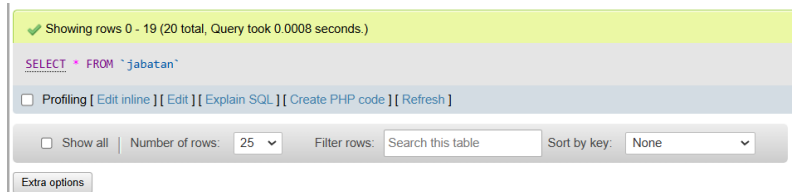
Print Copy to clipboard Create view

Query: `desc jabatan;`

Penjelasan:

Perintah untuk melihat apa saja *field* yang ada pada tabel Jabatan.

13. *Select* untuk melihat *Insert* yang sudah ditambahkan



Showing rows 0 - 19 (20 total, Query took 0.0008 seconds.)

```
SELECT * FROM `jabatan`
```

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

Server: localhost > Database: penggajian_dmc > Table: jabatan

	ID_Jabatan	Nama_Jabatan	Jumlah_Honor
<input type="checkbox"/> Edit Copy Delete	1	Wali Kelas IX C	100000.00
<input type="checkbox"/> Edit Copy Delete	2	Wali Kelas VIII D	100000.00
<input type="checkbox"/> Edit Copy Delete	3	Waka Kurikulum	400000.00
<input type="checkbox"/> Edit Copy Delete	4	Wali Kelas IX A	100000.00
<input type="checkbox"/> Edit Copy Delete	5	Wali Kelas VIII A	100000.00
<input type="checkbox"/> Edit Copy Delete	6	Staf Kesiswaan	250000.00
<input type="checkbox"/> Edit Copy Delete	7	Wali Kelas IX B	100000.00
<input type="checkbox"/> Edit Copy Delete	8	Pembina OSIS	150000.00
<input type="checkbox"/> Edit Copy Delete	9	Waka Kesiswaan	400000.00
<input type="checkbox"/> Edit Copy Delete	10	Wali Kelas VII D	100000.00
<input type="checkbox"/> Edit Copy Delete	11	Wali Kelas VII B	100000.00
<input type="checkbox"/> Edit Copy Delete	12	Wali Kelas VIII C	100000.00
<input type="checkbox"/> Edit Copy Delete	13	Wali Kelas VIII B	100000.00
<input type="checkbox"/> Edit Copy Delete	14	Wali Kelas IX D	100000.00
<input type="checkbox"/> Edit Copy Delete	15	Wali Kelas VII C	100000.00
<input type="checkbox"/> Edit Copy Delete	16	Kepala Sekolah	500000.00
<input type="checkbox"/> Edit Copy Delete	17	Wali Kelas VII A	100000.00
<input type="checkbox"/> Edit Copy Delete	18	Guru Mata Pelajaran Matematika	300000.00
<input type="checkbox"/> Edit Copy Delete	19	Guru Mata Pelajaran Bahasa Indonesia	300000.00
<input type="checkbox"/> Edit Copy Delete	20	Guru Mata Pelajaran IPA	300000.00

Check all With selected: Edit Copy Delete Export

Query: `SELECT * FROM `jabatan``

Penjelasan:

Perintah yang digunakan untuk melihat data apa saja yang sudah berhasil diinput setelah melakukan perintah ‘*INSERT*’ sebelumnya ke dalam tabel Jabatan.

14. Insert values ke tabel Transport

Run SQL query/queries on table penggajian_dmc.jabatan:

```

1 INSERT INTO transport (Nilai_Per_Hari) VALUES
2 (30000), (35000), (40000), (25000), (28000), (32000), (31000), (33000), (34000),
3 (36000), (37000), (29000), (38000), (39000), (41000), (42000), (43000), (44000),
4 (45000), (46000);

```

SELECT * SELECT INSERT UPDATE DELETE Clear Format Get auto-saved query

☐ Bind parameters

Delimiter: Show this query here again Retain query box Rollback when finished ☒ Enable foreign key checks Go

Hide query box

✓ 20 rows inserted.
Inserted row id: 20 (Query took 0.0085 seconds.)

```

INSERT INTO transport (Nilai_Per_Hari) VALUES (30000), (35000), (40000), (25000), (28000), (32000), (31000), (33000), (34000), (36000), (37000), (29000), (38000), (39000), (41000), (42000), (43000), (44000), (45000), (46000);

```

Query: `INSERT INTO transport (Nilai_Per_Hari) VALUES (30000), (35000), (40000), (25000), (28000), (32000), (31000), (33000), (34000), (36000), (37000), (29000), (38000), (39000), (41000), (42000), (43000), (44000), (45000), (46000);`

Penjelasan:

Query untuk menambahkan/menginput data ke dalam tabel Transport sesuai *fieldsnya*.

15. Melihat *fields* tabel Transport

Server: localhost » Database: penggajian_dmc » Table: jabatan

Structure

Show query box

Your SQL query has been executed successfully.

```
desc transport;
```

[Edit inline] [Edit] [Create PHP code]

Extra options

Field	Type	Null	Key	Default	Extra
ID_Transport	int	NO	PRI	NULL	auto_increment
Nilai_Per_Hari	decimal(10,2)	NO		NULL	

Query results operations

Print Copy to clipboard Create view

Query: `desc transport;`

Penjelasan:

Desc digunakan untuk melihat deskripsi tabel Transport atau melihat isi *fields* dari tabel tersebut.

16. Select untuk melihat *Insert* yang sudah ditambahkan

Showing rows 0 - 19 (20 total, Query took 0.0010 seconds.)

```
SELECT * FROM `transport`
```

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

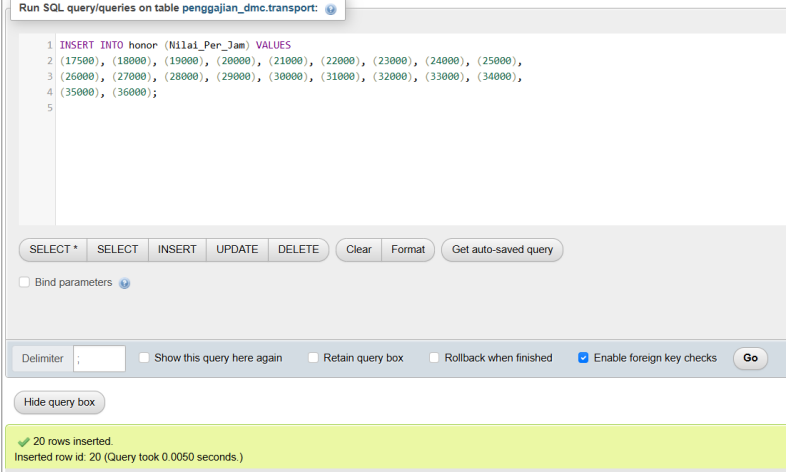
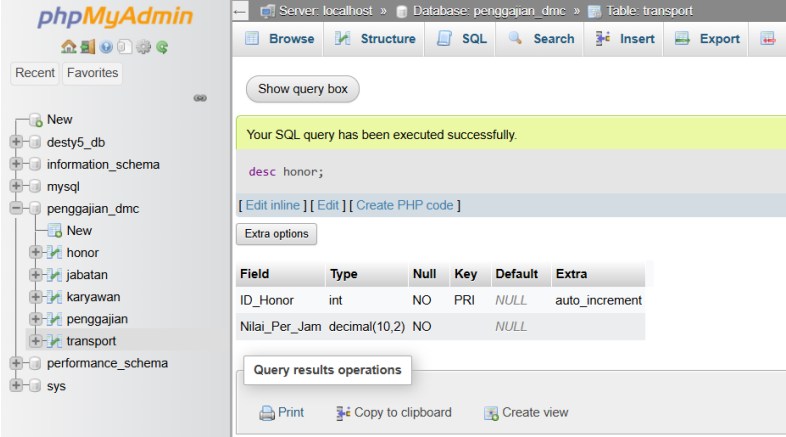
☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Server: localhost » Database: penggajian_dmc » Table: transport

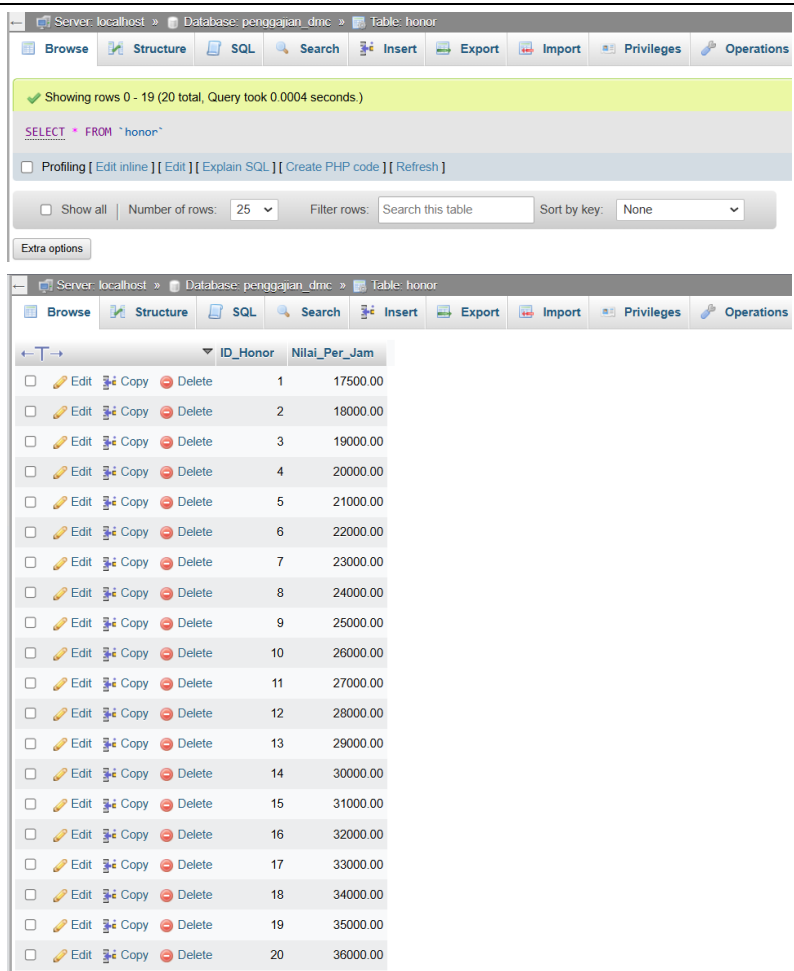
Structure

	ID_Transport	Nilai_Per_Hari
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	30000.00
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	35000.00
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	40000.00
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	4	25000.00
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	5	28000.00
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	6	32000.00
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	7	31000.00
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	8	33000.00
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	9	34000.00
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	10	36000.00
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	11	37000.00
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	12	29000.00
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	13	38000.00
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	14	39000.00
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	15	41000.00
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	16	42000.00
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	17	43000.00
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	18	44000.00
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	19	45000.00
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	20	46000.00

Query: `SELECT * FROM `transport``

	<p>Penjelasan:</p> <p>Untuk melihat data yang sudah diinput ke dalam tabel Transport.</p>
<p>17. Insert values ke tabel Honor</p>	 <p>Query:</p> <pre>INSERT INTO honor (Nilai_Per_Jam) VALUES (17500), (18000), (19000), (20000), (21000), (22000), (23000), (24000), (25000), (26000), (27000), (28000), (29000), (30000), (31000), (32000), (33000), (34000), (35000), (36000);</pre> <p>Penjelasan:</p> <p>Insert dilakukan untuk menambahkan <i>values</i> atau data ke dalam tabel Honor.</p>
<p>18. Melihat fields tabel Honor</p>	 <p>Query: <code>desc honor;</code></p> <p>Penjelasan:</p> <p>Gunanya untuk melihat ada <i>fields</i> apa saja di tabel Honor.</p>

19. *Select* untuk melihat *Insert* yang sudah ditambahkan



The screenshot shows the MySQL Workbench interface. At the top, the 'Table: honor' tab is active. Below the toolbar, a message states 'Showing rows 0 - 19 (20 total, Query took 0.0004 seconds)'. The SQL editor contains the query: `SELECT * FROM `honor``. Below the editor, there are options for 'Show all', 'Number of rows: 25', 'Filter rows: Search this table', and 'Sort by key: None'. An 'Extra options' button is also present. Below this, the 'Table: honor' is displayed with columns 'ID_Honor' and 'Nilai_Per_Jam'. The table contains 20 rows of data, with 'ID_Honor' ranging from 1 to 20 and 'Nilai_Per_Jam' ranging from 17500.00 to 36000.00 in increments of 1000.00.

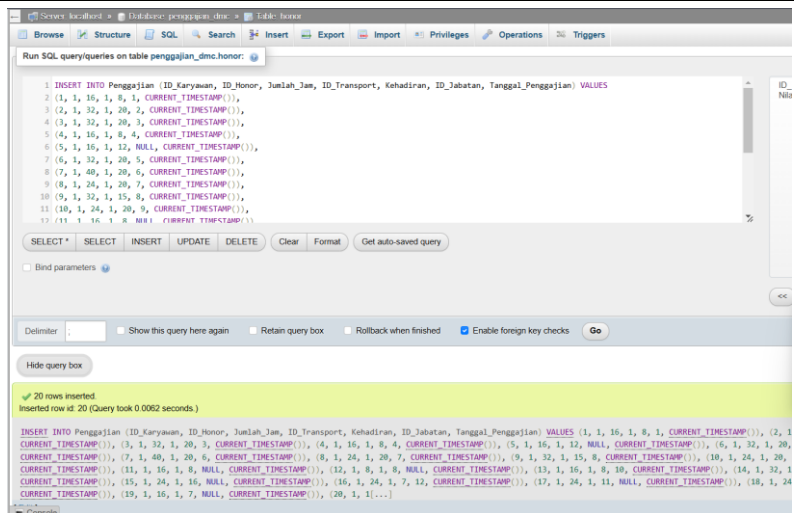
ID_Honor	Nilai_Per_Jam
1	17500.00
2	18000.00
3	19000.00
4	20000.00
5	21000.00
6	22000.00
7	23000.00
8	24000.00
9	25000.00
10	26000.00
11	27000.00
12	28000.00
13	29000.00
14	30000.00
15	31000.00
16	32000.00
17	33000.00
18	34000.00
19	35000.00
20	36000.00

Query: `SELECT * FROM `honor``

Penjelasan:

Query di atas untuk melihat data-data yang sudah diinput ke dalam tabel Honor.

20. *Insert* values ke tabel Pengajian



The screenshot shows the MySQL Workbench interface with the 'Run SQL query/queries on table pengajian_dmc.honor:' window open. The SQL editor contains a large `INSERT INTO` statement for the 'Pengajian' table. The statement is as follows: `INSERT INTO Pengajian (ID_Karyawan, ID_Honor, Jumlah_Jam, ID_Transport, Kehadiran, ID_Jabatan, Tanggal_Pengajian) VALUES (1, 1, 16, 1, 8, 1, CURRENT_TIMESTAMP()), (2, 1, 32, 1, 20, 2, CURRENT_TIMESTAMP()), (3, 1, 32, 1, 20, 3, CURRENT_TIMESTAMP()), (4, 1, 16, 1, 8, 4, CURRENT_TIMESTAMP()), (5, 1, 16, 1, 12, NULL, CURRENT_TIMESTAMP()), (6, 1, 32, 1, 20, 5, CURRENT_TIMESTAMP()), (7, 1, 40, 1, 20, 6, CURRENT_TIMESTAMP()), (8, 1, 24, 1, 20, 7, CURRENT_TIMESTAMP()), (9, 1, 32, 1, 15, 8, CURRENT_TIMESTAMP()), (10, 1, 24, 1, 20, 9, CURRENT_TIMESTAMP()), (11, 1, 16, 1, 8, NULL, CURRENT_TIMESTAMP()), (12, 1, 8, 1, 8, NULL, CURRENT_TIMESTAMP()), (13, 1, 16, 1, 8, 10, CURRENT_TIMESTAMP()), (14, 1, 32, 1, 20, 11, CURRENT_TIMESTAMP()), (15, 1, 24, 1, 16, NULL, CURRENT_TIMESTAMP()), (16, 1, 24, 1, 7, 12, CURRENT_TIMESTAMP()), (17, 1, 24, 1, 11, NULL, CURRENT_TIMESTAMP()), (18, 1, 24, 1, 16, 1, 7, NULL, CURRENT_TIMESTAMP()), (19, 1, 16, 1, 7, NULL, CURRENT_TIMESTAMP()), (20, 1, 1, 1, 1, NULL, CURRENT_TIMESTAMP())`. Below the editor, there are buttons for 'SELECT *', 'SELECT', 'INSERT', 'UPDATE', 'DELETE', 'Clear', 'Format', and 'Get auto-saved query'. There is also a 'Bind parameters' checkbox. At the bottom, there is a 'Delimit' dropdown, checkboxes for 'Show this query here again', 'Retain query box', 'Rollback when finished', and 'Enable foreign key checks', and a 'Go' button. Below the editor, a message states '20 rows inserted. Inserted row id 20 (Query took 0.0062 seconds)'. The SQL editor also shows the full `INSERT INTO` statement again.

```
1 INSERT INTO Pengajian (ID_Karyawan, ID_Honor, Jumlah_Jam, ID_Transport, Kehadiran, ID_Jabatan, Tanggal_Pengajian) VALUES
2 (1, 1, 16, 1, 8, 1, CURRENT_TIMESTAMP()),
3 (2, 1, 32, 1, 20, 2, CURRENT_TIMESTAMP()),
4 (3, 1, 32, 1, 20, 3, CURRENT_TIMESTAMP()),
5 (4, 1, 16, 1, 8, 4, CURRENT_TIMESTAMP()),
6 (5, 1, 16, 1, 12, NULL, CURRENT_TIMESTAMP()),
7 (6, 1, 32, 1, 20, 5, CURRENT_TIMESTAMP()),
8 (7, 1, 40, 1, 20, 6, CURRENT_TIMESTAMP()),
9 (8, 1, 24, 1, 20, 7, CURRENT_TIMESTAMP()),
10 (9, 1, 32, 1, 15, 8, CURRENT_TIMESTAMP()),
11 (10, 1, 24, 1, 20, 9, CURRENT_TIMESTAMP()),
12 (11, 1, 16, 1, 8, NULL, CURRENT_TIMESTAMP()),
13 (12, 1, 8, 1, 8, NULL, CURRENT_TIMESTAMP()),
14 (13, 1, 16, 1, 8, 10, CURRENT_TIMESTAMP()),
15 (14, 1, 32, 1, 20, 11, CURRENT_TIMESTAMP()),
16 (15, 1, 24, 1, 16, NULL, CURRENT_TIMESTAMP()),
17 (16, 1, 24, 1, 7, 12, CURRENT_TIMESTAMP()),
18 (17, 1, 24, 1, 11, NULL, CURRENT_TIMESTAMP()),
19 (18, 1, 24, 1, 16, 1, 7, NULL, CURRENT_TIMESTAMP()),
20 (19, 1, 16, 1, 7, NULL, CURRENT_TIMESTAMP()),
21 (20, 1, 1, 1, 1, NULL, CURRENT_TIMESTAMP())
```

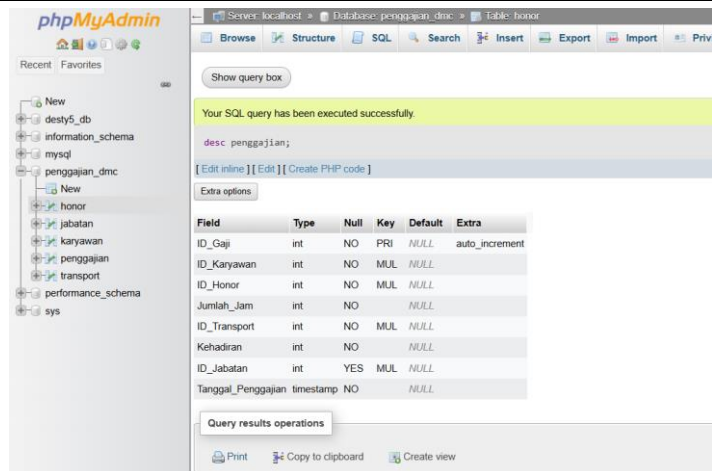
Query:

```
INSERT INTO Penggajian (ID_Karyawan, ID_Honor, Jumlah_Jam, ID_Transport, Kehadiran, ID_Jabatan, Tanggal_Penggajian) VALUES (1, 1, 16, 1, 8, 1, CURRENT_TIMESTAMP()), (2, 1, 32, 1, 20, 2, CURRENT_TIMESTAMP()), (3, 1, 32, 1, 20, 3, CURRENT_TIMESTAMP()), (4, 1, 16, 1, 8, 4, CURRENT_TIMESTAMP()), (5, 1, 16, 1, 12, NULL, CURRENT_TIMESTAMP()), (6, 1, 32, 1, 20, 5, CURRENT_TIMESTAMP()), (7, 1, 40, 1, 20, 6, CURRENT_TIMESTAMP()), (8, 1, 24, 1, 20, 7, CURRENT_TIMESTAMP()), (9, 1, 32, 1, 15, 8, CURRENT_TIMESTAMP()), (10, 1, 24, 1, 20, 9, CURRENT_TIMESTAMP()), (11, 1, 16, 1, 8, NULL, CURRENT_TIMESTAMP()), (12, 1, 8, 1, 8, NULL, CURRENT_TIMESTAMP()), (13, 1, 16, 1, 8, 10, CURRENT_TIMESTAMP()), (14, 1, 32, 1, 20, 11, CURRENT_TIMESTAMP()), (15, 1, 24, 1, 16, NULL, CURRENT_TIMESTAMP()), (16, 1, 24, 1, 7, 12, CURRENT_TIMESTAMP()), (17, 1, 24, 1, 11, NULL, CURRENT_TIMESTAMP()), (18, 1, 24, 1, 16, 13, CURRENT_TIMESTAMP()), (19, 1, 16, 1, 7, NULL, CURRENT_TIMESTAMP()), (20, 1, 16, 1, 10, 14, CURRENT_TIMESTAMP());
```

Penjelasan:

Untuk tabel Penggajian ini juga sama, kami hanya melakukan id saja karena *foreign key* ke tabel-tabel lain, lalu di bagian jabatan mengapa isinya null, karena pada tabel Jabatan itu ada guru yang memiliki jabatan, ada juga yang tidak. Kami juga menggunakan *current_timestamp* untuk memudahkan kami dalam melakukan join khususnya untuk laporan penggajian yang membutuhkan tanggal saat itu juga saat laporan dibuat.

21. Melihat *fields* tabel Penggajian

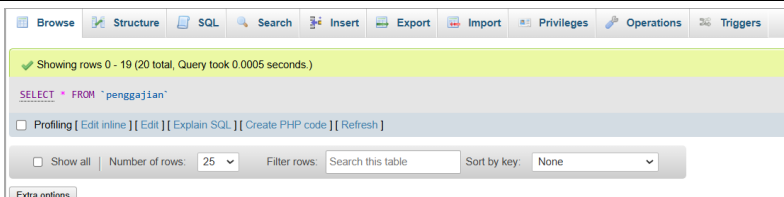


Query: `desc penggajian;`

Penjelasan:

Dilakukan untuk melihat ada *fields* apa saja di tabel Penggajian.

22. *Select* untuk melihat *Insert* yang sudah ditambahkan



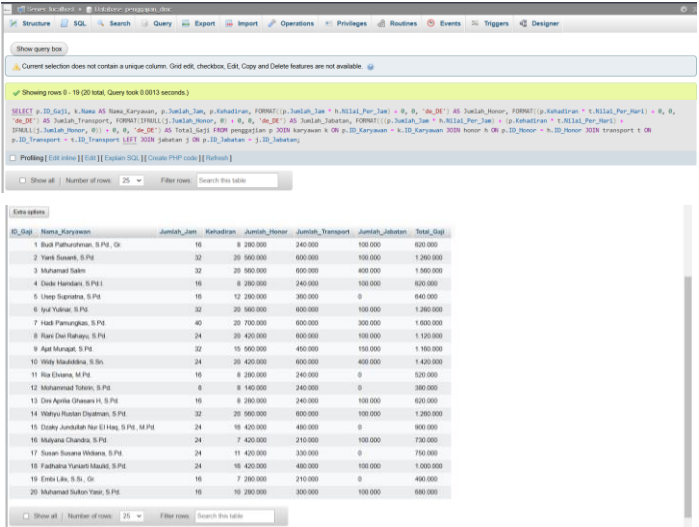
	ID_Gaji	ID_Karyawan	ID_Honor	Jumlah_Jam	ID_Transport	Kehadiran	ID_Jabatan	Tanggal_Penggajian
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	1	1	16	1	8	1	2024-12-29 12:39:32
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	2	1	32	1	20	2	2024-12-29 12:39:32
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	3	1	32	1	20	3	2024-12-29 12:39:32
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	4	4	1	16	1	8	4	2024-12-29 12:39:32
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	5	5	1	16	1	12	NULL	2024-12-29 12:39:32
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	6	6	1	32	1	20	5	2024-12-29 12:39:32
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	7	7	1	40	1	20	6	2024-12-29 12:39:32
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	8	8	1	24	1	20	7	2024-12-29 12:39:32
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	9	9	1	32	1	15	8	2024-12-29 12:39:32
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	10	10	1	24	1	20	9	2024-12-29 12:39:32
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	11	11	1	16	1	8	NULL	2024-12-29 12:39:32
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	12	12	1	8	1	8	NULL	2024-12-29 12:39:32
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	13	13	1	16	1	8	10	2024-12-29 12:39:32
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	14	14	1	32	1	20	11	2024-12-29 12:39:32
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	15	15	1	24	1	16	NULL	2024-12-29 12:39:32
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	16	16	1	24	1	7	12	2024-12-29 12:39:32
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	17	17	1	24	1	11	NULL	2024-12-29 12:39:32
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	18	18	1	24	1	16	13	2024-12-29 12:39:32
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	19	19	1	16	1	7	NULL	2024-12-29 12:39:32
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	20	20	1	16	1	10	14	2024-12-29 12:39:32

Query: `SELECT * FROM `penggajian``

Penjelasan:

Perintah untuk melihat hasil *insert* yang sebelumnya ditambahkan di tabel ‘Penggajian’.

23. Join untuk mendapatkan Detail Penggajian per Karyawan



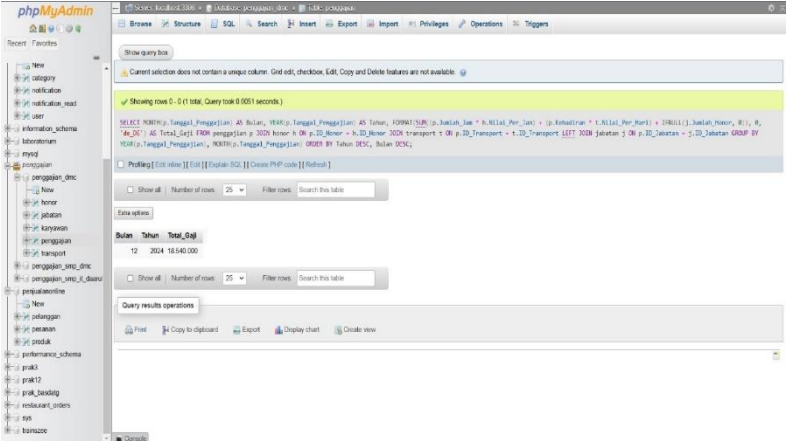
Query:

```
SELECT
    p.ID_Gaji,
    k>Nama AS Nama_Karyawan,
    p.Jumlah_Jam,
    p.Kehadiran,
    FORMAT((p.Jumlah_Jam * h.Nilai_Per_Jam) + 0, 0, 'de_DE') AS
    Jumlah_Honor,
    FORMAT((p.Kehadiran * t.Nilai_Per_Hari) + 0, 0, 'de_DE') AS
    Jumlah_Transport,
    FORMAT(IFNULL(j.Jumlah_Honor, 0) + 0, 0, 'de_DE') AS
    Jumlah_Jabatan,
    FORMAT(((p.Jumlah_Jam * h.Nilai_Per_Jam) + (p.Kehadiran *
    t.Nilai_Per_Hari) + IFNULL(j.Jumlah_Honor, 0)) + 0, 0, 'de_DE') AS
    Total_Gaji
FROM penggajian p
JOIN karyawan k ON p.ID_Karyawan = k.ID_Karyawan
JOIN honor h ON p.ID_Honor = h.ID_Honor
JOIN transport t ON p.ID_Transport = t.ID_Transport
LEFT JOIN jabatan j ON p.ID_Jabatan = j.ID_Jabatan;
```

Penjelasan:

Pada *Query* di atas, kami melakukan join dan alias karena tabel ini adalah tabel laporan yang akan disuguhkan secara visual. Maka outputnya juga bukan ID, tapi langsung nama orang yang bersangkutan, dan di sini juga terjadi perhitungan perkalian dari jumlah jam dan honor, jumlah kehadiran dan transport, juga terjadi penambahan yang menjadi total seluruh gaji yang diterima.

24. Join untuk mendapatkan Laporan Total Penggajian per Bulan



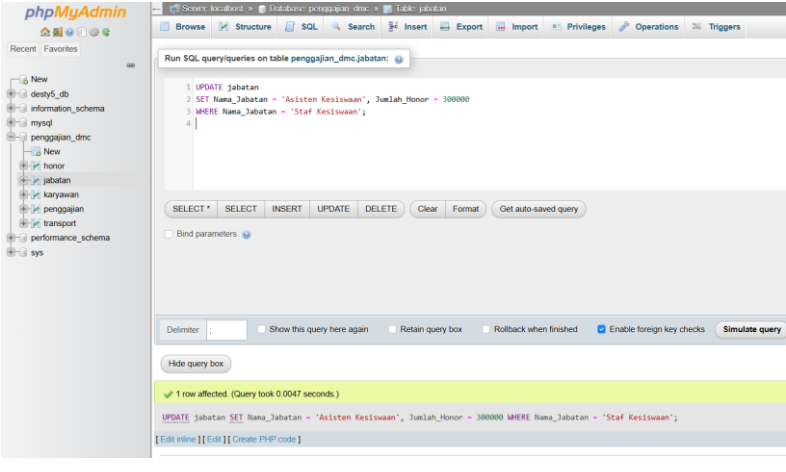
Query:

```
SELECT
    MONTH(p.Tanggal_Penggajian) AS Bulan,
    YEAR(p.Tanggal_Penggajian) AS Tahun,
    FORMAT(SUM((p.Jumlah_Jam * h.Nilai_Per_Jam) + (p.Kehadiran *
t.Nilai_Per_Hari) + IFNULL(j.Jumlah_Honor, 0)), 0, 'de_DE') AS
Total_Gaji
FROM penggajian p
JOIN honor h ON p.ID_Honor = h.ID_Honor
JOIN transport t ON p.ID_Transport = t.ID_Transport
LEFT JOIN jabatan j ON p.ID_Jabatan = j.ID_Jabatan
GROUP BY YEAR(p.Tanggal_Penggajian), MONTH(p.Tanggal_Penggajian)
ORDER BY Tahun DESC, Bulan DESC;
```

Penjelasan:

Untuk mendapatkan laporan penggajian setiap bulannya di sini kami juga melakukan perhitungan alias serta join.

25. Perintah Update pada Tabel Jabatan



Server: localhost > Database: pengajaran_dmc > Table: jabatan

Browse Structure SQL Search Insert Export Import Privileges Operations

DATA UPDATES

	ID_Jabatan	Nama_Jabatan	Jumlah_Honor
<input type="checkbox"/> Edit Copy Delete	1	Wali Kelas IX C	100000.00
<input type="checkbox"/> Edit Copy Delete	2	Wali Kelas VIII D	100000.00
<input type="checkbox"/> Edit Copy Delete	3	Waka Kurikulum	400000.00
<input type="checkbox"/> Edit Copy Delete	4	Wali Kelas IX A	100000.00
<input type="checkbox"/> Edit Copy Delete	5	Wali Kelas VIII A	100000.00
<input type="checkbox"/> Edit Copy Delete	6	Asisten Kesiswaan	300000.00
<input type="checkbox"/> Edit Copy Delete	7	Wali Kelas IX B	100000.00
<input type="checkbox"/> Edit Copy Delete	8	Pembina OSIS	150000.00
<input type="checkbox"/> Edit Copy Delete	9	Waka Kesiswaan	400000.00
<input type="checkbox"/> Edit Copy Delete	10	Wali Kelas VII D	100000.00
<input type="checkbox"/> Edit Copy Delete	11	Wali Kelas VII B	100000.00
<input type="checkbox"/> Edit Copy Delete	12	Wali Kelas VIII C	100000.00
<input type="checkbox"/> Edit Copy Delete	13	Wali Kelas VIII B	100000.00
<input type="checkbox"/> Edit Copy Delete	14	Wali Kelas IX D	100000.00
<input type="checkbox"/> Edit Copy Delete	15	Wali Kelas VII C	100000.00
<input type="checkbox"/> Edit Copy Delete	16	Kepala Sekolah	500000.00
<input type="checkbox"/> Edit Copy Delete	17	Wali Kelas VII A	100000.00
<input type="checkbox"/> Edit Copy Delete	18	Guru Mata Pelajaran Matematika	300000.00
<input type="checkbox"/> Edit Copy Delete	19	Guru Mata Pelajaran Bahasa Indonesia	300000.00
<input type="checkbox"/> Edit Copy Delete	20	Guru Mata Pelajaran IPA	300000.00

Query:

```
UPDATE jabatan SET Nama_Jabatan = 'Asisten Kesiswaan', Jumlah_Honor = 300000 WHERE Nama_Jabatan = 'Staf Kesiswaan';
```

Penjelasan:

Perintah di atas dilakukan untuk mengubah salah satu nama jabatan,, yang asalnya ‘Staf Kesiswaan’ menjadi ‘Asisten Kesiswaan’.

26. Perintah Delete pada Tabel Transport

phpMyAdmin

Server: localhost > Database: pengajaran_dmc > Table: transport

Browse Structure SQL Search Insert Export Import Privileges Operations Triggers

Run SQL query/queries on table pengajaran_dmc.transport:

```
1 DELETE FROM transport
2 WHERE Nilai_Per_Hari > 20000 AND Nilai_Per_Hari < 31000;
3
```

SELECT * SELECT INSERT UPDATE DELETE Clear Format Get auto-saved query

☐ Bind parameters

Delimiter: ; Show this query here again Retain query box Rollback when finished ☒ Enable foreign key checks Simulate query

Hide query box

4 rows affected (Query took 0.0055 seconds.)

```
DELETE FROM transport WHERE Nilai_Per_Hari > 20000 AND Nilai_Per_Hari < 31000;
```

[Edit inline] [Edit] [Create PHP code]

Query:

```
DELETE FROM transport WHERE Nilai_Per_Hari > 20000 AND Nilai_Per_Hari < 31000;
```

Penjelasan:

Merupakan perintah untuk menghapus data pada tabel Transport menggunakan dua kondisi, kondisi pertama untuk menghapus data yang dari Nilai_Per_Hari yang bernilai >

20000 dan kondisi kedua untuk data Nilai_Per_Hari yang bernilai < 31000.

Server: localhost > Database: pengajian_dmc > Table: transport

Browse Structure SQL Search Insert Export Import Privileges Operations

Showing rows 0 - 15 (16 total, Query took 0.0002 seconds.)

SELECT * FROM `transport`

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

	ID_Transport	Nilai_Per_Hari
<input type="checkbox"/> Edit Copy Delete	2	35000.00
<input type="checkbox"/> Edit Copy Delete	3	40000.00
<input type="checkbox"/> Edit Copy Delete	6	32000.00
<input type="checkbox"/> Edit Copy Delete	7	31000.00
<input type="checkbox"/> Edit Copy Delete	8	33000.00
<input type="checkbox"/> Edit Copy Delete	9	34000.00
<input type="checkbox"/> Edit Copy Delete	10	36000.00
<input type="checkbox"/> Edit Copy Delete	11	37000.00
<input type="checkbox"/> Edit Copy Delete	13	38000.00
<input type="checkbox"/> Edit Copy Delete	14	39000.00
<input type="checkbox"/> Edit Copy Delete	15	41000.00
<input type="checkbox"/> Edit Copy Delete	16	42000.00
<input type="checkbox"/> Edit Copy Delete	17	43000.00
<input type="checkbox"/> Edit Copy Delete	18	44000.00
<input type="checkbox"/> Edit Copy Delete	19	45000.00
<input type="checkbox"/> Edit Copy Delete	20	46000.00

Check all With selected: Edit Copy Delete Export

B. Implementasikan konsep *Transaction*

1) *Atomicity* manual transaksi sukses

```
START TRANSACTION;
```

```
INSERT INTO penggajian (ID_Karyawan, ID_Honor, Jumlah_Jam,
ID_Transport, Kehadiran, ID_Jabatan, Tanggal_Penggajian)
VALUES (21, 1, 20, 1, 20, NULL, CURRENT_TIMESTAMP());
INSERT INTO penggajian (ID_Karyawan, ID_Honor, Jumlah_Jam,
ID_Transport, Kehadiran, ID_Jabatan, Tanggal_Penggajian)
VALUES (22, 1, 18, 1, 22, NULL, CURRENT_TIMESTAMP());
```

```
COMMIT;
```

Penjelasan:

Jadi maksud dari *atomicity* ini adalah saat kita melakukan *start transaction* dan melakukan perintah maka si perintah itu bakalan ketahan sebelum kita *commit* kalau misalnya sudah *commit* baru akan terjadi perubahan di dalam tabel kita. Kalau misalnya belum *commit* seperti pada tabel kami yang sebelumnya ada 20 isinya kemudian kami menggunakan metode transaksi kemudian memasukan perintah buat menambahkan 2 *values* terus di *commit* berhasil jadi data di tabel penggajian ada 22. Berikut adalah bukti *atomicity* berhasil atau *commit* berhasil:

```
mysql> select *from penggajian;
```

ID_Gaji	ID_Karyawan	ID_Honor	Jumlah_Jam	ID_Transport	Kehadiran	ID_Jabatan	Tanggal_Penggajian
1	1	1	16	1	8	1	2024-12-25 11:43:40
2	2	1	32	1	20	2	2024-12-25 11:43:40
3	3	1	32	1	20	3	2024-12-25 11:43:40
4	4	1	16	1	8	4	2024-12-25 11:43:40
5	5	1	16	1	12	NULL	2024-12-25 11:43:40
6	6	1	32	1	20	5	2024-12-25 11:43:40
7	7	1	40	1	20	6	2024-12-25 11:43:40
8	8	1	24	1	20	7	2024-12-25 11:43:40
9	9	1	32	1	15	8	2024-12-25 11:43:40
10	10	1	24	1	20	9	2024-12-25 11:43:40
11	11	1	16	1	8	NULL	2024-12-25 11:43:40
12	12	1	8	1	8	NULL	2024-12-25 11:43:40
13	13	1	16	1	8	10	2024-12-25 11:43:40
14	14	1	32	1	20	11	2024-12-25 11:43:40
15	15	1	24	1	16	NULL	2024-12-25 11:43:40
16	16	1	24	1	7	12	2024-12-25 11:43:40
17	17	1	24	1	11	NULL	2024-12-25 11:43:40
18	18	1	24	1	16	13	2024-12-25 11:43:40
19	19	1	16	1	7	NULL	2024-12-25 11:43:40
20	20	1	16	1	10	14	2024-12-25 11:43:40

20 rows in set (0.00 sec)

Data Tabel Penggajian (data sebelum ditambah lagi values)

```
mysql> select *from penggajian;
```

ID_Gaji	ID_Karyawan	ID_Honor	Jumlah_Jam	ID_Transport	Kehadiran	ID_Jabatan	Tanggal_Penggajian
1	1	1	16	1	8	1	2024-12-25 11:43:40
2	2	1	32	1	20	2	2024-12-25 11:43:40
3	3	1	32	1	20	3	2024-12-25 11:43:40
4	4	1	16	1	8	4	2024-12-25 11:43:40
5	5	1	16	1	12	NULL	2024-12-25 11:43:40
6	6	1	32	1	20	5	2024-12-25 11:43:40
7	7	1	40	1	20	6	2024-12-25 11:43:40
8	8	1	24	1	20	7	2024-12-25 11:43:40
9	9	1	32	1	15	8	2024-12-25 11:43:40
10	10	1	24	1	20	9	2024-12-25 11:43:40
11	11	1	16	1	8	NULL	2024-12-25 11:43:40
12	12	1	8	1	8	NULL	2024-12-25 11:43:40
13	13	1	16	1	8	10	2024-12-25 11:43:40
14	14	1	32	1	20	11	2024-12-25 11:43:40
15	15	1	24	1	16	NULL	2024-12-25 11:43:40
16	16	1	24	1	7	12	2024-12-25 11:43:40
17	17	1	24	1	11	NULL	2024-12-25 11:43:40
18	18	1	24	1	16	13	2024-12-25 11:43:40
19	19	1	16	1	7	NULL	2024-12-25 11:43:40
20	20	1	16	1	10	14	2024-12-25 11:43:40

20 rows in set (0.00 sec)

Data Tabel Penggajian (data ketika berhasil dilakukan *commit*)

2) Atomicity manual transaksi batal / rollback pada Penggajian

Query:

```
START TRANSACTION;
DELETE FROM penggajian WHERE ID_Gaji = 1;
DELETE FROM penggajian WHERE ID_Gaji = 2;
ROLLBACK;
```

Penjelasan:

Jadi saat kita *start transaction*, lalu memasukkan perintah hapus 1 dan 2 seperti di gambar, saat dilakukan pengecekan id 1 dan 2 berhasil terhapus, seperti pada gambar pertama. Akan tetapi hal itu bersifat sementara karena kita tidak melakukan *commit*. Jadi kalau kita melakukan *rollback* datanya akan kembali lagi seperti pada gambar 2. Berikut bukti kegiatan *rollback* berhasil:

```
mysql> START TRANSACTION; DELETE FROM penggajian WHERE ID_Gaji = 1; DELETE FROM penggajian WHERE ID_Gaji = 2;
Query OK, 0 rows affected (0.00 sec)

Query OK, 1 row affected (0.00 sec)

Query OK, 1 row affected (0.00 sec)

mysql> select *from penggajian;
```

ID_Gaji	ID_Karyawan	ID_Honor	Jumlah_Jam	ID_Transport	Kehadiran	ID_Jabatan	Tanggal_Penggajian
3	3	1	32	1	20	3	2024-12-25 11:43:40
4	4	1	16	1	8	4	2024-12-25 11:43:40
5	5	1	16	1	12	NULL	2024-12-25 11:43:40
6	6	1	32	1	20	5	2024-12-25 11:43:40
7	7	1	40	1	20	6	2024-12-25 11:43:40
8	8	1	24	1	20	7	2024-12-25 11:43:40
9	9	1	32	1	15	8	2024-12-25 11:43:40
10	10	1	24	1	20	9	2024-12-25 11:43:40
11	11	1	16	1	8	NULL	2024-12-25 11:43:40
12	12	1	8	1	8	NULL	2024-12-25 11:43:40
13	13	1	16	1	8	10	2024-12-25 11:43:40
14	14	1	32	1	20	11	2024-12-25 11:43:40
15	15	1	24	1	16	NULL	2024-12-25 11:43:40
16	16	1	24	1	7	12	2024-12-25 11:43:40
17	17	1	24	1	11	NULL	2024-12-25 11:43:40
18	18	1	24	1	16	13	2024-12-25 11:43:40
19	19	1	16	1	7	NULL	2024-12-25 11:43:40
20	20	1	16	1	10	14	2024-12-25 11:43:40
21	21	1	20	1	20	NULL	2024-12-25 14:34:08
22	22	1	18	1	22	NULL	2024-12-25 14:34:08

20 rows in set (0.00 sec)

Data ID_Gaji = 1 dan ID_Gaji = 2 sudah dihapus, namun belum di *rollback*

```
mysql> ROLLBACK;
Query OK, 0 rows affected (0.00 sec)

mysql> select *from penggajian;
```

ID_Gaji	ID_Karyawan	ID_Honor	Jumlah_Jam	ID_Transport	Kehadiran	ID_Jabatan	Tanggal_Penggajian
1	1	1	16	1	8	1	2024-12-25 11:43:40
2	2	1	32	1	20	2	2024-12-25 11:43:40
3	3	1	32	1	20	3	2024-12-25 11:43:40
4	4	1	16	1	8	4	2024-12-25 11:43:40
5	5	1	16	1	12	NULL	2024-12-25 11:43:40
6	6	1	32	1	20	5	2024-12-25 11:43:40
7	7	1	40	1	20	6	2024-12-25 11:43:40
8	8	1	24	1	20	7	2024-12-25 11:43:40
9	9	1	32	1	15	8	2024-12-25 11:43:40
10	10	1	24	1	20	9	2024-12-25 11:43:40
11	11	1	16	1	8	NULL	2024-12-25 11:43:40
12	12	1	8	1	8	NULL	2024-12-25 11:43:40
13	13	1	16	1	8	10	2024-12-25 11:43:40
14	14	1	32	1	20	11	2024-12-25 11:43:40
15	15	1	24	1	16	NULL	2024-12-25 11:43:40
16	16	1	24	1	7	12	2024-12-25 11:43:40
17	17	1	24	1	11	NULL	2024-12-25 11:43:40
18	18	1	24	1	16	13	2024-12-25 11:43:40
19	19	1	16	1	7	NULL	2024-12-25 11:43:40
20	20	1	16	1	10	14	2024-12-25 11:43:40
21	21	1	20	1	20	NULL	2024-12-25 14:34:08
22	22	1	18	1	22	NULL	2024-12-25 14:34:08

22 rows in set (0.00 sec)

Tampilan data kembali ketika sudah dilakukan *rollback*

3) *Consistency Invalid Update* pada penggajian

Query:

```
START TRANSACTION;
UPDATE penggajian SET ID_Honor = NULL WHERE ID_Gaji = 1;
ID_Honor tidak boleh NULL
COMMIT;
```

Penjelasan:

Jadi meskipun kita sudah *commit* jika data yang kita masukan tidak sesuai dengan pengaturan yang sudah kita buat di awal maka tidak akan berhasil. Contohnya seperti tabel penggajian kami di bawah karena saat membuat tabel kita sudah settingbiar gak nul terus kita ubah menjadi null maka tidak bisa. Berikut bukti kegiatan *consistency*:

```
mysql> START TRANSACTION; UPDATE penggajian SET ID_Honor = NULL WHERE ID_Gaji = 1; ID_H
onor tidak boleh NULL COMMIT;
Query OK, 0 rows affected (0.00 sec)

ERROR 1048 (23000): Column 'ID_Honor' cannot be null
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresp
onds to your MySQL server version for the right syntax to use near 'ID_Honor tidak bole
h NULL COMMIT' at line 1
mysql> |
```

Meskipun sudah melakukan *commit*, tetapi jika data yang diinput tidak sesuai maka akan *invalid*

4) *Isolation* (isolasi antar transaksi pada Penggajian)

Query:

```
START TRANSACTION;
UPDATE penggajian
SET Jumlah_Jam = Jumlah_Jam + 5
WHERE ID_Karyawan = 1;

SELECT * FROM penggajian WHERE ID_Karyawan = 1 FOR UPDATE;
UPDATE penggajian
SET Kehadiran = Kehadiran - 1
WHERE ID_Karyawan = 1;

COMMIT;
```

Penjelasan:

Jadi maksud dari transaksi 1 harus sebelum transaksi 2 mulai, nanti tidak akan ada perminataan atau perintahnya yang bentrok. Seperti yang terjadi pada gambar di bawah ini, yang pertama dijalankan itu permintaan untuk menambah jam, di mana asalnya 16 ditambah 5, menjadi 21 jam. Tetapi bisa dilihat pada kehadiran belum dikurangi, karena dia menyelesaikan satu-satu dulu. Setelah itu baru permintaan kedua bisa dijalankan, dan akan terlihat berkurang, yang tadinya 8 menjadi 7. Berikut bukti kegiatan ilustrasi *isolation*:


```
mysql> START TRANSACTION; UPDATE penggajian SET Jumlah_Jam = Jumlah_Jam + 5 WHERE ID_Karyawan = 1; SELECT * FROM penggajian
WHERE ID_Karyawan = 1 FOR UPDATE; UPDATE penggajian SET Kehadiran = Kehadiran - 1 WHERE ID_Karyawan = 1; COMMIT;
Query OK, 0 rows affected (0.00 sec)

Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

+-----+-----+-----+-----+-----+-----+-----+-----+
| ID_Gaji | ID_Karyawan | ID_Honor | Jumlah_Jam | ID_Transport | Kehadiran | ID_Jabatan | Tanggal_Penggajian |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | 1 | 1 | 21 | 1 | 8 | 1 | 2024-12-25 11:43:40 |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

Query OK, 0 rows affected (0.01 sec)
```

Telah meninput 2 perintah, namun yang dieksekusi perintah pertama

```
mysql> SELECT * FROM penggajian WHERE ID_Karyawan = 1 FOR UPDATE; UPDATE penggajian SET Kehadiran = Kehadiran - 1 WHERE ID_Ka
rnyawan = 1; COMMIT;
+-----+-----+-----+-----+-----+-----+-----+-----+
| ID_Gaji | ID_Karyawan | ID_Honor | Jumlah_Jam | ID_Transport | Kehadiran | ID_Jabatan | Tanggal_Penggajian |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | 1 | 1 | 21 | 1 | 7 | 1 | 2024-12-25 11:43:40 |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for
the right syntax to use near '= 1' at line 1
Query OK, 0 rows affected (0.00 sec)
```

Setelah perintah di atas selesai, baru perintah kedua dijalankan, maka kehadiran baru akan berkurang.

C. Buatkan skenario apabila ada kesalahan maka seluruh transaksi batal

Query:

```
START TRANSACTION;
DECLARE EXIT HANDLER FOR SQLEXCEPTION
BEGIN
    ROLLBACK;
    SELECT 'ada kesalahan pengisian, ualngi kembali.' AS
Error_Message;
END;
UPDATE penggajian
SET Jumlah_Jam = Jumlah_Jam - 10
WHERE ID_Karyawan = 1;
UPDATE penggajian
SET Kehadiran = NULL
WHERE ID_Karyawan = 2;
COMMIT;
```

Penjelasan:

Dengan menggunakan *rollback*, kita bisa membuat ilustrasi jika ada kesalahan dalam *input* data ataupun saat dilakukan *commit*. Dia tidak akan mengubah apapun atau transaksi seluruhnya batal meski dilakukan *commit*. Kalau untuk tabel Penggajian, untuk perintah pertama berhasil dibuat, tetapi karena perintah kedua itu ada kesalahan jadi dia membatalkan seluruh transaksi dan mengembalikan data ke posisi awal dengan metode *rollback*. Berikut bukti skenario apabila ada kesalahan maka seluruh transaksi batal:

```
mysql> START TRANSACTION; DECLARE EXIT HANDLER FOR SQLEXCEPTION BEGIN ROLLBACK; SELECT 'ada kesalahan pengisian, ualngi
kembali.' AS Error_Message; END; UPDATE penggajian SET Jumlah_Jam = Jumlah_Jam - 10 WHERE ID_Karyawan = 1; UPDATE penggajian
SET Kehadiran = NULL WHERE ID_Karyawan = 2; COMMIT;
Query OK, 0 rows affected (0.00 sec)

ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for
the right syntax to use near 'DECLARE EXIT HANDLER FOR SQLEXCEPTION BEGIN ROLLBACK' at line 1

+-----+-----+
| Error_Message |
+-----+-----+
| ada kesalahan pengisian, ualngi kembali. |
+-----+-----+
1 row in set (0.00 sec)

ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for
the right syntax to use near 'END' at line 1
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

ERROR 1048 (23000): Column 'Kehadiran' cannot be null
Query OK, 0 rows affected (0.00 sec)
```

Ilustrasi jika terjadi pengisian data yang salah maka tidak akan berhasil dan data tidak akan berubah karena *rollback*.

3. Implementasikan *User Management* ke *Database*

A. Siapa aja yang boleh akses ke *Database*

Role yang bisa mengakses *Database* Penggajian_DMC adalah *user* root, admin, KepalaSekolah dan umum.

B. Siapa aja yang boleh *input* data

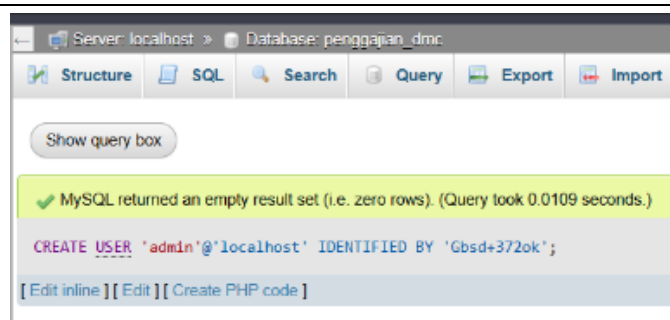
Sedangkan yang bisa menginput data hanyalah *user* root dan admin saja.

C. Siapa aja yang cuman boleh *Query Select*

Untuk *user* yang bisa melakukan *Query Select* atau *user* yang hanya bisa melakukan kegiatan “melihat” adalah *user* KepalaSekolah dan umum.

D. Proses membuat banyak *user* dengan *privileged* tertentu

1. Membuat beberapa account/user

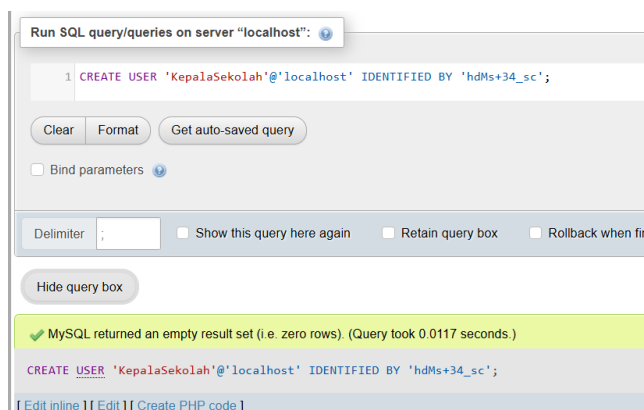


Query:

```
CREATE USER 'admin'@'localhost' IDENTIFIED BY 'Gbsd+372ok';
```

Penjelasan:

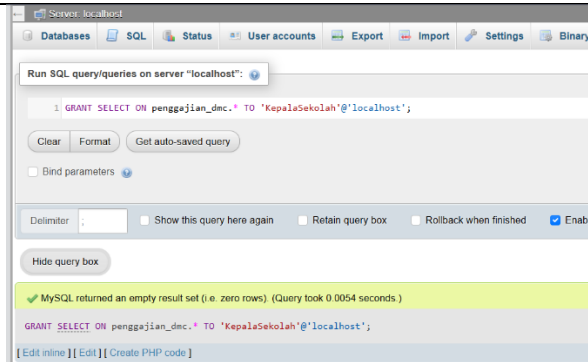
Perintah untuk membuat *user* baru bernama ‘admin’ yang hanya bisa mengakses *database* dari *localhost* serta memiliki kata sandi seperti pada *query* di atas.



Query:

```
CREATE USER 'KepalaSekolah'@'localhost' IDENTIFIED BY 'hdMs+34_sc';
```

	<p>Penjelasan:</p> <p>Ini juga sama seperti di atas, hanya saja yang membedakan ini <i>user</i> baru yang bernama ‘KepalaSekolah’.</p>  <p>Query:</p> <pre>CREATE USER 'KepalaSekolah'@'localhost' IDENTIFIED BY 'g3n4Gbsd+_y';</pre> <p>Penjelasan:</p> <p>Dan ini juga sama, untuk membuat user baru bernama ‘umum’.</p>
<p>2. Memberikan hak akses untuk masing-masing <i>user</i> yang baru dibuat</p>	 <p>Query:</p> <pre>GRANT SELECT, INSERT, UPDATE, DELETE, DROP ON penggajian_dmc.* TO 'admin'@'localhost';</pre> <p>Penjelasan:</p> <p><i>Query</i> di atas adalah perintah untuk memberikan hak akses kepada <i>user</i> ‘admin’ agar bisa melakukan operasi <i>SELECT</i>, <i>INSERT</i>, <i>UPDATE</i>, <i>DELETE</i>, dan <i>DROP</i> pada semua tabel di <i>database</i> Penggajian_DMC. Akan tetapi semua itu hanya bisa dilakukan di <i>localhost</i> saja.</p>

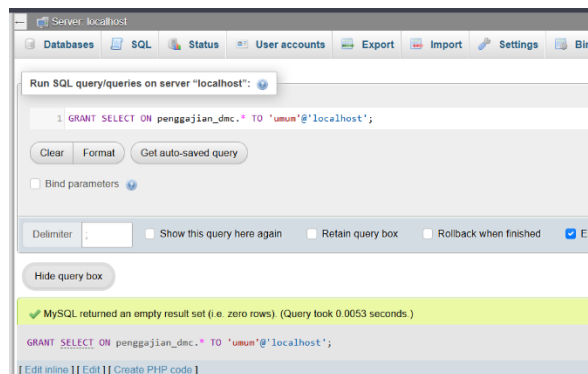


Query:

```
GRANT SELECT ON penggajian_dmc.* TO 'KepalaSekolah'@'localhost';
```

Penjelasan:

Sedangkan perintah ini juga sama dengan yang di atas, bedanya *query* ini memberikan hak akses pada *user* 'KepalaSekolah' hanya bisa melakukan operasi *SELECT* atau melihat saja pada database Penggajian_DMC.



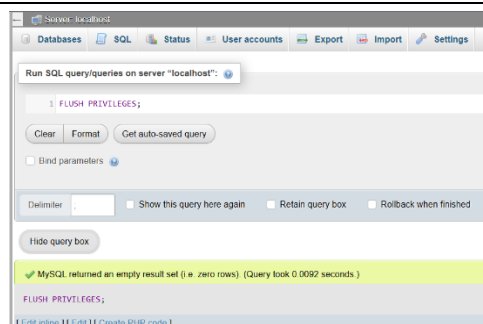
Query:

```
GRANT SELECT ON penggajian_dmc.* TO 'umum'@'localhost';
```

Penjelasan:

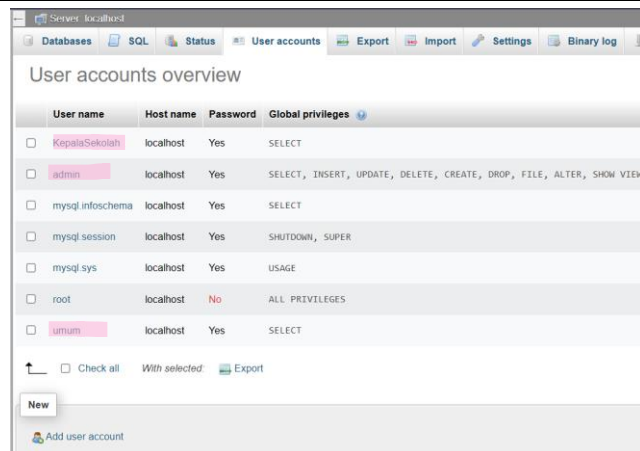
Untuk *query* ini sama persis dengan sebelumnya, yaitu memberikan hak akses hanya untuk melakukan *SELECT* saja untuk *user* 'umum' pada database Penggajian_DMC.

3. Melakukan muat ulang



	<p>Query: <code>FLUSH PRIVILEGES;</code></p> <p>Penjelasan:</p> <p>Perintah untuk melakukan <i>refresh</i> setelah dilakukan perubahan/penambahan untuk setiap hak akses atau pembuatan <i>user</i> baru.</p>
4. Menampilkan hak akses yang sudah ditambahkan	 <p>Query: <code>SHOW GRANTS FOR 'admin'@'localhost';</code></p> <p>Penjelasan:</p> <p><i>Query</i> untuk menampilkan hak akses <i>user</i> ‘admin’.</p>  <p>Query: <code>SHOW GRANTS FOR 'KepalaSekolah'@'localhost';</code></p> <p>Penjelasan:</p> <p>Perintah untuk menampilkan hak akses apa <i>user</i> ‘KepalaSekolah’.</p>  <p>Query: <code>SHOW GRANTS FOR 'umum'@'localhost';</code></p> <p>Penjelasan:</p> <p><i>Query</i> untuk menampilkan hak akses pada <i>user</i> ‘umum’.</p>

5. Daftar user pada localhost



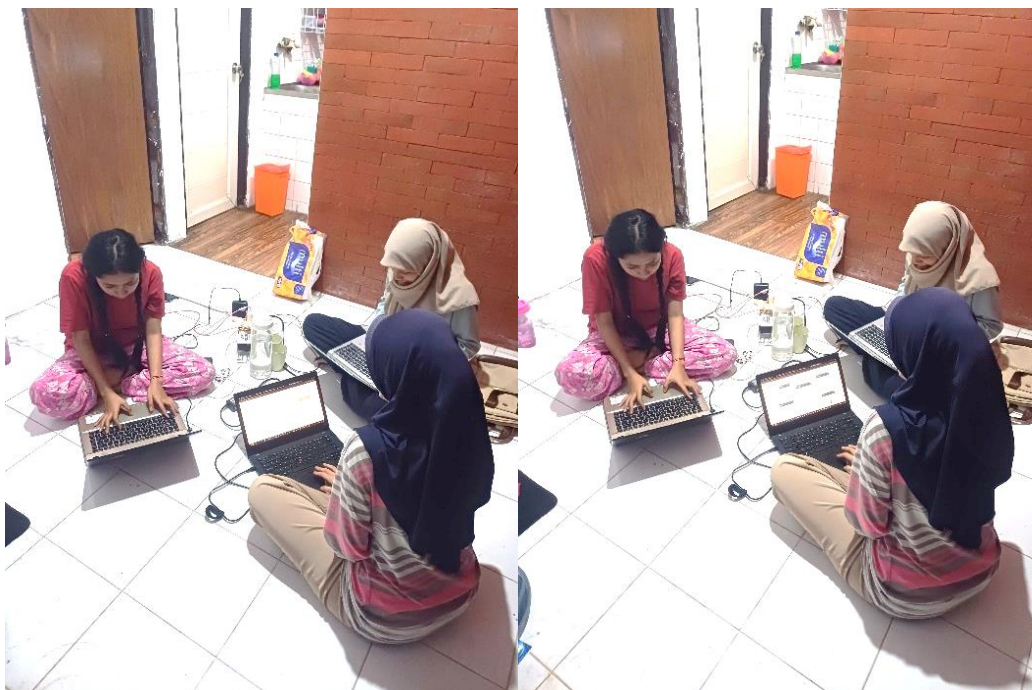
The screenshot shows the 'User accounts overview' page in a MySQL management tool. It displays a table of user accounts on the localhost. The table has columns for User name, Host name, Password, and Global privileges. The users listed are KepalaSekolah, admin, mysql.infoschema, mysql.session, mysql.sys, root, and umum. The root user has 'No' for password and 'ALL PRIVILEGES' for global privileges. The other users have 'Yes' for password and specific privileges like SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, FILE, ALTER, SHOW VIEW, SHUTDOWN, SUPER, and USAGE.

User name	Host name	Password	Global privileges
KepalaSekolah	localhost	Yes	SELECT
admin	localhost	Yes	SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, FILE, ALTER, SHOW VIEW
mysql.infoschema	localhost	Yes	SELECT
mysql.session	localhost	Yes	SHUTDOWN, SUPER
mysql.sys	localhost	Yes	USAGE
root	localhost	No	ALL PRIVILEGES
umum	localhost	Yes	SELECT

Penjelasan:

Mengapa demikian? Karena root sebagai administrator utama, dan saya membuat *Database Penggajian_DMC* di sana. Sedangkan *user* admin, saya atur agar dia bisa mengubah, menambah, namun dia tidak memiliki *All Privilege* seperti *user* root. Kemudian untuk *user* KepalaSekolah dan umum adalah *user* untuk pengguna yang tidak bisa mengedit, hanya bisa melihat saja atau hanya bisa melakukan perintah *SELECT* saja, sesuai dengan bagaimana saya mengatur *grant* mereka.

4. Dokumentasi Pengerjaan



5. Link Github

https://github.com/nura215/UASBASDAT_SitemPenggajian