

\*\*

## UJIAN PRAKTIKUM BASIS DATA

NAMA : Siti Nur Hasiza.A

NO. URUT : 29

Perhatikan tabel berikut:

### tabel\_guru

| id_guru | nama_depan  | nama_belakang | mapel                          | jabatan                | usia | tanggal_lahir |
|---------|-------------|---------------|--------------------------------|------------------------|------|---------------|
| 1       | Adrianty    |               | Pemrograman Web                | Ketua Jurusan          | 34   | 1982-06-29    |
| 2       | Ibrahim     | Mallombasang  | Basis Data                     | Kepala Sekolah         | 21   | 2000-09-21    |
| 3       | Muhammad    | Yusuf         | Pemodelan Perangkat Lunak      |                        | 28   | 1992-12-24    |
| 4       | Rusdyansyar |               | Pemrograman Berorientasi Objek | Asisten Kepala Sekolah | 25   | 1996-01-21    |

### Petunjuk Pengerjaan

1. Isi nama dan no. urut kalian pada lembar ini!
2. Buatlah database dengan nama “sekolah\_nama\_kalian” contohnya “sekolah\_adrian”!
3. Silahkan jawab setiap pertanyaan dengan men-screenshoot setiap query beserta hasil dari query tersebut dan letakkan pada setiap nomor di file ini!
4. Simpan file ini dalam ekstensi .pdf dengan format “Nomor Urut - Nama Lengkap” !  
Sebagai contoh “1 - Adrian.pdf”!

## Soal

### Query

```
create table tabel_guru (  
  -> id_guru int(10) primary key not null,  
  -> nama_depan varchar(25) not null,  
  -> nama_belakang varchar(25) null,  
  -> mapel varchar(50) not null,  
  -> jabatan varchar(25) null,
```

```
-> usia int(![[1.jpg]]10) not null,  
-> tanggal_lahir varchar(25) not null);
```

1. Buatlah tabel berikut dengan nama “tabel\_guru” dan simpan di dalam database bernama “sekolah\_nama\_kalian” contohnya “sekolah\_adrian”. Silahkan tentukan tipe data, maksimum karakter, constraint yang sesuai dengan kondisi tabel tersebut. Kemudian tampilkan struktur tabel “tabel\_guru” yang telah kalian buat!

```
MariaDB [sekolah_hasiza]> create table tabel_guru (  
-> id_guru int(10) primary key not null,  
-> nama_depan varchar(25) not null,  
-> nama_belakang varchar(25) null,  
-> mapel varchar(50) not null,  
-> jabatan varchar(25) null,  
-> usia int(10) not null,  
-> tanggal_lahir varchar(25) not null);  
Query OK, 0 rows affected (0.026 sec)
```

## Analisis

- **CREATE TABLE:** Ini adalah perintah SQL yang digunakan untuk membuat sebuah tabel baru di dalam database.
- **tabel\_guru:** Ini adalah nama tabel yang akan dibuat. Tabel ini akan digunakan untuk menyimpan informasi mengenai guru-guru.
- **( ... ):** Di antara tanda kurung ini, kita mendefinisikan struktur tabel, yaitu kolom-kolomnya.
- **id\_guru:** Ini adalah kolom pertama dalam tabel, yang digunakan untuk menyimpan ID unik setiap guru. Tipe datanya adalah INT(10), yang berarti itu adalah bilangan bulat dengan panjang maksimum 10 digit. Ditandai dengan PRIMARY KEY NOT NULL, yang berarti kolom ini harus unik dan tidak boleh kosong.
- **nama\_depan:** Ini adalah kolom kedua, yang digunakan untuk menyimpan nama depan guru. Tipe datanya adalah VARCHAR(25), yang berarti itu adalah string teks dengan panjang maksimum 25 karakter. Ditandai dengan NOT NULL, yang berarti kolom ini harus diisi setiap kali data baru dimasukkan.
- **nama\_belakang:** Ini adalah kolom ketiga, yang digunakan untuk menyimpan nama belakang guru. Tipe datanya adalah VARCHAR(25), yang berarti itu adalah string teks dengan panjang maksimum 25 karakter. Ditandai dengan NULL, yang berarti kolom ini bisa kosong (jika seorang guru tidak memiliki nama belakang).
- **mapel:** Ini adalah kolom keempat, yang digunakan untuk menyimpan mata pelajaran yang diajarkan oleh guru. Tipe datanya adalah VARCHAR(50), yang berarti itu adalah string teks dengan panjang maksimum 50 karakter. Ditandai dengan NOT NULL, yang berarti kolom ini harus diisi setiap kali data baru dimasukkan.

- **jabatan**: Ini adalah kolom kelima, yang digunakan untuk menyimpan jabatan atau posisi guru (misalnya, Kepala Sekolah, Guru Kelas, dll). Tipe datanya adalah VARCHAR(25), yang berarti itu adalah string teks dengan panjang maksimum 25 karakter. Ditandai dengan NULL, yang berarti kolom ini bisa kosong (jika informasi jabatan tidak tersedia).
- **usia**: Ini adalah kolom keenam, yang digunakan untuk menyimpan usia guru. Tipe datanya adalah INT(10), yang berarti itu adalah bilangan bulat dengan panjang maksimum 10 digit. Ditandai dengan NOT NULL, yang berarti kolom ini harus diisi setiap kali data baru dimasukkan.
- **tanggal\_lahir**: Ini adalah kolom terakhir, yang digunakan untuk menyimpan tanggal lahir guru. Tipe datanya adalah VARCHAR(25), yang berarti itu adalah string teks dengan panjang maksimum 25 karakter. Ditandai dengan NOT NULL, yang berarti kolom ini harus diisi setiap kali data baru dimasukkan.

## Kesimpulan

query tersebut membuat tabel "tabel\_guru" dengan struktur yang jelas untuk menyimpan informasi tentang guru. Setiap kolom memiliki ketentuan yang tepat mengenai keterisiannya, seperti NULL atau NOT NULL, dan tipe data yang sesuai dipilih untuk masing-masing informasi.

## Query

```
insert into tabel_guru
  -> values
  -> (1, "Adrianty", null, "Pemrograman web", "Ketua Jurusan", 34, "1982-06-29"),
  -> (2, "Ibrahim", "Mallombasang", "Basis Data", "Kepala Sekolah", 21, "2000-09-21"),
  -> (3, "Muhammad", "Yusuf", "Pemodelan Perangkat Lunak", null, 28, "1992-12-24"),
  -> (4, "Rusdyansyar", null, "Pemrograman Berorientasi Objek", "Asisten Kepala Sekolah", 25, "1996-01-21");
```

2. Masukkan data ke "tabel\_guru" yang telah kalian buat di soal nomor 1 seperti data yang ada pada "tabel\_guru"!

```
(2, Ibrahim, Mallombasang, Basis Data, Kepala Sekolah, 21, "2000-09-21") at line 3
MariaDB [sekolah_hasiza]> insert into tabel_guru
  -> values
  -> (1, "Adrianty", null, "Pemrograman web", "Ketua Jurusan", 34, "1982-06-29"),
  -> (2, "Ibrahim", "Mallombasang", "Basis Data", "Kepala Sekolah", 21, "2000-09-21"),
  -> (3, "Muhammad", "Yusuf", "Pemodelan Perangkat Lunak", null, 28, "1992-12-24"),
  -> (4, "Rusdyansyar", null, "Pemrograman Berorientasi Objek", "Asisten Kepala Sekolah", 25, "1996-01-21");
Query OK, 4 rows affected (0.118 sec)
Records: 4 Duplicates: 0 Warnings: 0
```

## Analisis

- **INSERT INTO:** Ini adalah perintah SQL yang digunakan untuk menyisipkan data ke dalam tabel yang sudah ada di dalam database.
- **tabel\_guru:** Ini adalah nama tabel yang data akan dimasukkan ke dalamnya.
- **VALUES:** Ini adalah bagian dari perintah yang menandakan nilai-nilai yang akan dimasukkan ke dalam kolom-kolom tabel.
- **Baris Pertama:** Data guru pertama yang akan dimasukkan.
  - ID Guru: 1
  - Nama Depan: Adrianty
  - Nama Belakang: NULL
  - Mata Pelajaran: Pemrograman Web
  - Jabatan: Ketua Jurusan
  - Usia: 34 tahun
  - Tanggal Lahir: 29 Juni 1982
- **D Guru (id\_guru):**
- **Tujuan:** Menyimpan ID unik untuk setiap guru dalam tabel.
- **Gunanya:**
  - Menjadi kunci utama (PRIMARY KEY) untuk tabel, memastikan setiap baris memiliki identifikasi unik.
  - Memudahkan pengambilan data spesifik tentang seorang guru dengan menggunakan ID tersebut.
- **Baris Kedua:** Data guru kedua yang akan dimasukkan.
  - ID Guru: 2
  - Nama Depan: Ibrahim
  - Nama Belakang: Mallombasang
  - Mata Pelajaran: Basis Data
  - Jabatan: Kepala Sekolah
  - Usia: 21 tahun
  - Tanggal Lahir: 21 September 2000

#### **Nama Depan (nama\_depan):**

- **Tujuan:** Menyimpan nama depan dari setiap guru.
- **Gunanya:**
  - Mengidentifikasi guru secara individu.
  - Menampilkan nama depan guru dalam hasil pencarian atau laporan.
- **Baris Ketiga:** Data guru ketiga yang akan dimasukkan.
  - ID Guru: 3
  - Nama Depan: Muhammad
  - Nama Belakang: Yusuf
  - Mata Pelajaran: Pemodelan Perangkat Lunak
  - Jabatan: NULL

- Usia: 28 tahun
- Tanggal Lahir: 24 Desember 1992

### **Nama Belakang (nama\_belakang):**

- **Tujuan:** Menyimpan nama belakang dari setiap guru (opsional).
- **Gunanya:**
  - Melengkapi informasi identitas guru jika tersedia.
  - Menampilkan nama lengkap guru dalam laporan atau tampilan lainnya.
- **Baris Keempat:** Data guru keempat yang akan dimasukkan.
  - ID Guru: 4
  - Nama Depan: Rusdyansyar
  - Nama Belakang: NULL
  - Mata Pelajaran: Pemrograman Berorientasi Objek
  - Jabatan: Asisten Kepala Sekolah
  - Usia: 25 tahun
  - Tanggal Lahir: 21 Januari 1996

### **Mata Pelajaran (mapel):**

- **Tujuan:** Menyimpan informasi tentang mata pelajaran yang diajarkan oleh guru.
  - **Gunanya:**
    - Memungkinkan pengelompokan guru berdasarkan mata pelajaran yang diajarkan.
    - Memfasilitasi pencarian guru berdasarkan mata pelajaran tertentu.

## **Kesimpulan**

- **Pengisian Data:** Query ini digunakan untuk mengisi data ke dalam tabel "tabel\_guru". Setiap baris data mencakup informasi lengkap tentang seorang guru, seperti nama, mata pelajaran yang diajarkan, jabatan (jika ada), usia, dan tanggal lahir.
- **Penambahan Data:** Dengan menggunakan perintah `INSERT INTO`, kita menambahkan empat baris data baru ke dalam tabel. Ini memperluas jumlah catatan tentang guru di dalam database.
- **Konsistensi Kolom:** Data yang dimasukkan sesuai dengan struktur kolom yang telah ditentukan sebelumnya saat pembuatan tabel. Setiap kolom diisi dengan nilai yang relevan, dan kolom-kolom opsional yang dibiarkan kosong ditandai dengan nilai NULL.
- **Penggunaan ID Guru:** Penggunaan ID Guru sebagai kunci utama memastikan bahwa setiap baris memiliki identifikasi unik, mempermudah pengambilan data tentang guru tertentu dan mempertahankan konsistensi dalam basis data.
- **Kelengkapan Informasi:** Meskipun beberapa kolom, seperti "nama\_belakang" dan "jabatan", tidak selalu diisi untuk setiap baris, data yang dimasukkan cukup lengkap untuk memberikan gambaran yang baik tentang setiap guru.

# Query

```
insert into tabel_guru
-> values
-> (5, "Nur", "Hasiza", "Pendidikan Agama Islam", null, 17, "2006-12-28");
```

3. Tambahkan lagi satu baris data berisi nama kalian dan isi data lainnya sesuai keinginan kalian!

```
ERROR 1062 (23000): Duplicate entry '5' for key 'PRIMARY'
MariaDB [sekolah_hasiza]> insert into tabel_guru
-> values
-> (5, "Nur", "Hasiza", "Pendidikan Agama Islam", null, 17, "2006-12-28");
Query OK, 1 row affected (0.015 sec)
```

## Analisis

- **INSERT INTO:** Ini adalah perintah SQL yang digunakan untuk menyisipkan data ke dalam tabel yang sudah ada di dalam database.
- **tabel\_guru:** Ini adalah nama tabel yang data akan dimasukkan ke dalamnya.
- **VALUES:** Ini adalah bagian dari perintah yang menandakan nilai-nilai yang akan dimasukkan ke dalam kolom-kolom tabel.
- **Baris Data Baru:**
  - **ID Guru:** 5
  - **Nama Depan:** Nur
  - **Nama Belakang:** Hasiza
  - **Mata Pelajaran:** Pendidikan Agama Islam
  - **Jabatan:** NULL
  - **Usia:** 17 tahun
  - **Tanggal Lahir:** 28 Desember 2006
- **ID Guru:** Angka 5 digunakan sebagai ID guru baru. Harap pastikan bahwa nilai ID guru ini unik dalam tabel agar tidak ada konflik dengan data yang ada sebelumnya.
- **Nama Depan dan Nama Belakang:** Nur Hasiza dimasukkan sebagai nama lengkap guru. Dalam kasus ini, kedua kolom nama belakang dan jabatan diisi dengan nilai yang relevan.
- **Mata Pelajaran:** Guru ini mengajar mata pelajaran "Pendidikan Agama Islam".
- **Jabatan:** Dalam query ini, jabatan guru tidak diisi (NULL). Mungkin karena guru ini tidak memiliki jabatan khusus dalam institusi tersebut.
- **Usia dan Tanggal Lahir:** Usia guru ini adalah 17 tahun dengan tanggal lahir pada 28 Desember 2006.

## Kesimpulan

- **Penambahan Data:** Query ini bertujuan untuk menambahkan data seorang guru baru

ke dalam tabel "tabel\_guru" dalam basis data.

- **Kelengkapan Informasi:** Data yang dimasukkan mencakup informasi lengkap tentang guru, termasuk nama lengkap, mata pelajaran yang diajarkan, usia, dan tanggal lahir.
- **Penggunaan ID Guru:** ID Guru yang diberikan (5) harus unik dalam tabel untuk memastikan bahwa setiap guru memiliki identifikasi yang berbeda.
- **Konsistensi Struktur Tabel:** Data dimasukkan sesuai dengan struktur kolom yang telah ditetapkan sebelumnya saat pembuatan tabel. Kolom yang bersifat opsional, seperti "nama\_belakang" dan "jabatan", diisi atau dibiarkan kosong sesuai kebutuhan.

## Query

```
SELECT * FROM tabel_guru;
```

4. Tampilkan seluruh data yang telah kalian masukkan!

```
MariaDB [sekolah_hasiza]> SELECT * FROM tabel_guru;
```

| id_guru | nama_depan  | nama_belakang | mapel                          | jabatan                | usia | tanggal_lahir |
|---------|-------------|---------------|--------------------------------|------------------------|------|---------------|
| 1       | Adrianty    | NULL          | Pemrograman web                | Ketua Jurusan          | 34   | 1982-06-29    |
| 2       | Ibrahim     | Mallombasang  | Basis Data                     | Kepala Sekolah         | 21   | 2000-09-21    |
| 3       | Muhammad    | Yusuf         | Pemodelan Perangkat Lunak      | NULL                   | 28   | 1992-12-24    |
| 4       | Rusdyansyar | NULL          | Pemrograman Berorientasi Objek | Asisten Kepala Sekolah | 25   | 1996-01-21    |
| 5       | Nur         | Hasiza        | Pendidikan Agama Islam         | NULL                   | 17   | 2006-12-28    |

```
5 rows in set (0.001 sec)
```

## Analisis

- **Pengambilan Data Keseluruhan:** Perintah ini digunakan untuk mengambil semua data yang tersimpan dalam tabel "tabel\_guru". Ini memungkinkan pengguna untuk melihat semua informasi yang tersedia tentang para guru tanpa memfilter atau membatasi hasil.
- **Penelusuran dan Inspeksi:** Dengan menampilkan semua data dari tabel "tabel\_guru", pengguna dapat menelusuri dan memeriksa setiap catatan secara menyeluruh. Hal ini berguna untuk memeriksa data yang baru dimasukkan, memverifikasi integritas data, atau melakukan penelusuran informasi spesifik.
- **Analisis Data:** Dengan melihat keseluruhan data, pengguna dapat menganalisis pola, tren, atau statistik terkait dengan staf pengajar. Ini bisa termasuk analisis usia rata-rata, distribusi mata pelajaran, atau frekuensi jabatan yang berbeda di antara guru-guru.
- **Pemantauan dan Pelaporan:** Perintah ini dapat digunakan untuk membuat laporan atau pemantauan rutin tentang staf pengajar dalam sebuah institusi pendidikan. Informasi ini dapat digunakan oleh manajemen atau pihak yang berkepentingan untuk membuat keputusan atau evaluasi.
- **Integrasi dengan Aplikasi atau Sistem:** Data yang diambil dari tabel "tabel\_guru" dapat diintegrasikan dengan aplikasi atau sistem lain untuk tujuan seperti pembayaran gaji, penjadwalan, atau manajemen sumber daya manusia (SDM).

## Kesimpulan

erintah `SELECT * FROM tabel_guru;` digunakan untuk mengambil semua data yang tersimpan dalam tabel "tabel\_guru". Ini memudahkan untuk menelusuri, menganalisis, dan memantau informasi tentang staf pengajar dalam sebuah institusi pendidikan secara komprehensif.

## Query

```
SELECT * FROM tabel_guru
-> WHERE nama_depan = "Rusdyansyar";
```

5. Tampilkan seluruh data dari "Rusdyansyar" tanpa mengikutkan data guru-guru yang lain!

```
MariaDB [sekolah_hasiza]> SELECT * FROM tabel_guru
-> WHERE nama_depan = "Rusdyansyar";
```

| id_guru | nama_depan  | nama_belakang | mapel                          | jabatan                | usia | tanggal_lahir |
|---------|-------------|---------------|--------------------------------|------------------------|------|---------------|
| 4       | Rusdyansyar | NULL          | Pemrograman Berorientasi Objek | Asisten Kepala Sekolah | 25   | 1996-01-21    |

1 row in set (0.002 sec)

## Analisis

Perintah `SELECT * FROM tabel_guru WHERE nama_depan = "Rusdyansyar";` digunakan untuk mengambil semua data dari tabel "tabel\_guru" dimana nilai kolom "nama\_depan" sama dengan "Rusdyansyar". Ini akan menghasilkan semua informasi yang terkait dengan guru yang memiliki nama depan "Rusdyansyar"

- **Kondisi WHERE:** Dengan menggunakan klausa WHERE, perintah ini membatasi hasil query hanya untuk baris-baris di mana nilai kolom "nama\_depan" sama dengan "Rusdyansyar".
- **Kesesuaian Data:** Query ini akan memfilter data sehingga hanya baris-baris yang memiliki nilai "Rusdyansyar" di kolom "nama\_depan" yang akan ditampilkan.
- **Kolom yang Ditampilkan:** Dengan menggunakan "\*", query ini akan mengambil semua kolom dari tabel "tabel\_guru" untuk baris-baris yang memenuhi kondisi. Ini berarti seluruh informasi tentang guru yang bernama "Rusdyansyar" akan ditampilkan.
- **Konteks Pencarian:** Perintah ini membantu menemukan informasi spesifik tentang seorang guru dengan nama "Rusdyansyar" dari tabel "tabel\_guru".

## Kesimpulan

Perintah `SELECT * FROM tabel_guru WHERE nama_depan = "Rusdyansyar";` mengambil semua informasi dari tabel "tabel\_guru" hanya untuk guru yang bernama depan "Rusdyansyar", memungkinkan pengguna untuk fokus pada data spesifik yang dibutuhkan.

## Query



```
UPDATE tabel_guru set nama_belakang="Ganteng" WHERE id_guru="2";
```

6. Ganti "nama\_belakang" untuk "id\_guru" yang bernilai "2" menjadi "Ganteng", lalu tampilkan seluruh data kalian! (7 poin)

```
MariaDB [sekolah_hasiza]> UPDATE tabel_guru set nama_belakang="Ganteng" WHERE id_guru="2";
Query OK, 1 row affected (0.016 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [sekolah_hasiza]> SELECT * FROM tabel_guru
-> ;
+-----+-----+-----+-----+-----+-----+-----+
| id_guru | nama_depan | nama_belakang | mapel | jabatan | usia | tanggal_lahir |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | Adrianty | NULL | Pemrograman web | Ketua Jurusan | 34 | 1982-06-29 |
| 2 | Ibrahim | Ganteng | Basis Data | Kepala Sekolah | 21 | 2000-09-21 |
| 3 | Muhammad | Yusuf | Pemodelan Perangkat Lunak | NULL | 28 | 1992-12-24 |
| 4 | Rusdyansyar | NULL | Pemrograman Berorientasi Objek | Asisten Kepala Sekolah | 25 | 1996-01-21 |
| 5 | Nur | Hasiza | Pendidikan Agama Islam | NULL | 17 | 2006-12-28 |
+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.001 sec)
```

## Analisis

- **UPDATE:** Ini adalah perintah SQL yang digunakan untuk mengubah atau memperbarui data yang ada dalam tabel database.
- **tabel\_guru:** Ini adalah nama tabel yang akan diperbarui.
- **SET nama\_belakang="Ganteng":** Ini adalah bagian dari perintah yang menentukan kolom mana yang akan diperbarui dan nilai baru yang akan diberikan. Dalam hal ini, kita mengatur nilai kolom "nama\_belakang" menjadi "Ganteng".
- **WHERE id\_guru="2":** Ini adalah klausa yang menentukan kondisi untuk data yang akan diperbarui. Dalam hal ini, kita membatasi perubahan hanya untuk baris di mana nilai kolom "id\_guru" sama dengan "2".
- **id\_guru="2":** Ini adalah kondisi yang spesifik yang diterapkan pada kolom "id\_guru". Kondisi ini memastikan bahwa perubahan hanya akan terjadi pada baris dengan nilai "2" di kolom "id\_guru".
- **Tujuan:** Tujuan dari perintah ini adalah untuk mengubah nilai kolom "nama\_belakang" menjadi "Ganteng" hanya untuk guru dengan ID "2" dalam tabel "tabel\_guru".

## Kesimpulan

Perintah tersebut mengubah nilai kolom "nama\_belakang" menjadi "Ganteng" hanya untuk guru dengan ID "2" dalam tabel "tabel\_guru".

## Query

```
DELETE FROM tabel_guru WHERE id_guru="5";
```

7. Hapuslah data yang kalian masukkan pada soal nomor 3, lalu tampilkan seluruh data kalian! (7 poin)

```
MariaDB [sekolah_hasiza]> DELETE FROM tabel_guru WHERE id_guru="5";
Query OK, 1 row affected (0.015 sec)

MariaDB [sekolah_hasiza]> SELECT * FROM tabel_guru;
+-----+-----+-----+-----+-----+-----+-----+
| id_guru | nama_depan | nama_belakang | mapel | jabatan | usia | tanggal_lahir |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | Adrianty | NULL | Pemrograman web | Ketua Jurusan | 34 | 1982-06-29 |
| 2 | Ibrahim | Ganteng | Basis Data | Kepala Sekolah | 21 | 2000-09-21 |
| 3 | Muhammad | Yusuf | Pemodelan Perangkat Lunak | NULL | 28 | 1992-12-24 |
| 4 | Rusdyansyar | NULL | Pemrograman Berorientasi Objek | Asisten Kepala Sekolah | 25 | 1996-01-21 |
+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.001 sec)
```

## Analisis

- **DELETE FROM:** Ini adalah perintah SQL yang digunakan untuk menghapus baris atau data dari tabel database.
- **tabel\_guru:** Ini adalah nama tabel dari mana data akan dihapus.
- **WHERE id\_guru="5":** Klausa WHERE digunakan untuk menentukan kondisi untuk data yang akan dihapus. Dalam hal ini, kita membatasi penghapusan hanya untuk baris yang memiliki nilai "5" di kolom "id\_guru".
- **id\_guru="5":** Ini adalah kondisi yang spesifik yang diterapkan pada kolom "id\_guru". Kondisi ini memastikan bahwa hanya baris dengan nilai "5" di kolom "id\_guru" yang akan dihapus.
- **Tujuan:** Tujuan dari perintah ini adalah untuk menghapus semua informasi tentang guru dengan ID "5" dari tabel "tabel\_guru".

## Kesimpulan

Perintah `DELETE FROM tabel_guru WHERE id_guru="5";` digunakan untuk menghapus data guru yang memiliki ID "5" dari tabel "tabel\_guru".

## Query

```
SELECT * FROM tabel_guru
-> WHERE mapel like 'Pem%' and usia < 30
-> ORDER BY usia ASC;
```

8. Tampilkan seluruh data yang memiliki “usia” kurang dari 30 dan sekaligus memiliki data “mapel” dengan awalan “Pem”. Lalu kemudian urutkan data tersebut berdasarkan kolom “usia”-nya dari usia terkecil ke usia terbesar! Hasilnya akan seperti di bawah ini! (15 poin)

| id_guru | nama_depan  | nama_belakang | mapel                          | jabatan                | usia | tanggal_lahir |
|---------|-------------|---------------|--------------------------------|------------------------|------|---------------|
| 4       | Rusdyansyar |               | Pemrograman Berorientasi Objek | Asisten Kepala Sekolah | 25   | 1996-01-21    |
| 3       | Muhammad    | Yusuf         | Pemodelan Perangkat            |                        | 28   | 1992-12-24    |

|  |  |  |       |  |  |  |
|--|--|--|-------|--|--|--|
|  |  |  | Lunak |  |  |  |
|--|--|--|-------|--|--|--|

```
MariaDB [sekolah_hasiza]> SELECT * FROM tabel_guru
-> WHERE mapel like 'Pem%' and usia < 30
-> ORDER BY usia ASC;
```

| id_guru | nama_depan  | nama_belakang | mapel                          | jabatan                | usia | tanggal_lahir |
|---------|-------------|---------------|--------------------------------|------------------------|------|---------------|
| 4       | Rusdyansyar | NULL          | Pemrograman Berorientasi Objek | Asisten Kepala Sekolah | 25   | 1996-01-21    |
| 3       | Muhammad    | Yusuf         | Pemodelan Perangkat Lunak      | NULL                   | 28   | 1992-12-24    |

```
2 rows in set (0.001 sec)
```

## Analisis

1. **SELECT \* FROM tabel\_guru:** Ini adalah perintah untuk memilih semua kolom dari tabel "tabel\_guru".
2. **WHERE mapel LIKE 'Pem%':** Klausa WHERE digunakan untuk memfilter baris-baris yang sesuai dengan kriteria tertentu. Kriteria dalam hal ini adalah bahwa nilai kolom "mapel" harus dimulai dengan string "Pem".
3. **AND usia < 30:** Ini adalah bagian dari klausa WHERE yang menambahkan syarat tambahan. Kriteria kedua adalah bahwa nilai kolom "usia" harus kurang dari 30.
4. **ORDER BY usia ASC:** Ini digunakan untuk mengurutkan hasil berdasarkan nilai kolom "usia" secara ascending (ASC), atau dari yang terkecil hingga terbesar.

## Kesimpulan

Perintah tersebut mengambil semua informasi tentang guru-guru yang mengajar mata pelajaran yang dimulai dengan "Pem" dan berusia kurang dari 30 tahun dari tabel "tabel\_guru", lalu hasilnya diurutkan berdasarkan usia dari yang terkecil.

## Query

```
SELECT id_guru, nama_depan from tabel_guru
-> WHERE nama_depan LIKE '___i%';
```

9. Tampilkan data hanya pada kolom "nama\_depan" dengan kriteria yang "nama\_depan" - nya mengandung karakter "i" baik di depan atau di tengah atau di belakang! Hasilnya akan seperti di bawah ini! (10 poin)

| id_guru | nama_depan |
|---------|------------|
| 1       | Adrianty   |
| 2       | Ibrahim    |

```
MariaDB [sekolah_hasiza]> SELECT id_guru, nama_depan from tabel_guru
-> WHERE nama_depan LIKE '__%i%';
```

| id_guru | nama_depan |
|---------|------------|
| 1       | Adrianty   |
| 2       | Ibrahim    |

2 rows in set (0.001 sec)

## Analisis

- **SELECT id\_guru, nama\_depan FROM tabel\_guru:** Ini adalah perintah untuk memilih dua kolom, yaitu "id\_guru" dan "nama\_depan", dari tabel "tabel\_guru".
- **WHERE nama\_depan LIKE '\_\_%i%':** Klausa WHERE digunakan untuk memfilter baris-baris yang sesuai dengan kriteria tertentu. Kriteria dalam hal ini adalah bahwa nilai kolom "nama\_depan" harus memiliki minimal dua karakter sebelum karakter "i" dan karakter "i" harus hadir di suatu tempat di dalam nama.

## Kesimpulan

Perintah tersebut mengambil ID guru dan nama depan dari tabel "tabel\_guru" untuk semua guru yang memiliki setidaknya dua karakter sebelum karakter "i" dan karakter "i" terdapat di dalam nama depan mereka.

## Query

```
SELECT CONCAT_WS (" ", nama_depan, nama_belakang) AS nama_lengkap FROM
tabel_guru;
```

10. Gabungkan data pada kolom "nama\_depan" dengan kolom "nama\_belakang" dan berikan pemisah antara data dari kedua kolom dengan karakter spasi. Berikan alias atau nama pengganti sementara dari dua kolom yang digabungkan dengan nama kolom sementara yaitu "nama\_lengkap"! Hasilnya akan seperti di bawah ini! (8 poin)

| nama_lengkap    |
|-----------------|
| Adrianty        |
| Ibrahim Ganteng |
| Muhammad Yusuf  |
| Rusdyansyar     |

```
MariaDB [sekolah_hasiza]> SELECT CONCAT_WS (" ", nama_depan, nama_belakang) AS nama_lengkap FROM tabel_guru;
```

| nama_lengkap    |
|-----------------|
| Adrianty        |
| Ibrahim Ganteng |
| Muhammad Yusuf  |
| Rusdyansyar     |

```
4 rows in set (0.001 sec)
```

## Analisis

- **SELECT CONCAT\_WS(" ", nama\_depan, nama\_belakang) AS nama\_lengkap:** Ini adalah perintah untuk membuat kolom baru dengan nama "nama\_lengkap". Fungsi CONCAT\_WS() digunakan untuk menggabungkan nilai dari beberapa kolom menjadi satu nilai, dalam hal ini "nama\_depan" dan "nama\_belakang", dengan menggunakan spasi sebagai pemisah. Alias "nama\_lengkap" diberikan kepada kolom baru yang dihasilkan.
- **FROM tabel\_guru:** Ini menunjukkan tabel dari mana data akan diambil, yaitu "tabel\_guru".

## Kesimpulan

Perintah tersebut mengambil semua nilai dari kolom "nama\_depan" dan "nama\_belakang" dalam tabel "tabel\_guru", menggabungkannya dengan menggunakan spasi sebagai pemisah, dan menampilkannya dalam satu kolom baru yang disebut "nama\_lengkap".

## Query

```
ALTER table tabel_guru
-> ADD STATUS ENUM ("PNS", "PPPK", "Honorer");
```

11. Tambahkan sebuah kolom baru dengan nama "status". Lengkapi seluruh data pada kolom tersebut! Data yang bisa dimasukkan oleh user hanya dapat bernilai "PNS" atau "PPPK" atau "Honorer", sehingga berikan tipe data yang tepat untuk kolom tersebut! Tampilkan struktur tabel dan hasil dari inputan data kalian! (10 poin)

```
MariaDB [sekolah_hasiza]> ALTER table tabel_guru
-> ADD STATUS ENUM ("PNS", "PPPK", "Honorer");
Query OK, 0 rows affected (0.025 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [sekolah_hasiza]> SELECT * FROM tabel_guru;
```

| id_guru | nama_depan  | nama_belakang | mapel                          | jabatan                | usia | tanggal_lahir | STATUS |
|---------|-------------|---------------|--------------------------------|------------------------|------|---------------|--------|
| 1       | Adrianty    | NULL          | Pemrograman web                | Ketua Jurusan          | 34   | 1982-06-29    | NULL   |
| 2       | Ibrahim     | Ganteng       | Basis Data                     | Kepala Sekolah         | 21   | 2000-09-21    | NULL   |
| 3       | Muhammad    | Yusuf         | Pemodelan Perangkat Lunak      | NULL                   | 28   | 1992-12-24    | NULL   |
| 4       | Rusdyansyar | NULL          | Pemrograman Berorientasi Objek | Asisten Kepala Sekolah | 25   | 1996-01-21    | NULL   |

```
4 rows in set (0.001 sec)
```

## Analisis

- **ALTER TABLE:** Ini adalah perintah SQL yang digunakan untuk mengubah struktur tabel yang sudah ada.
- **tabel\_guru:** Ini adalah nama tabel yang akan diubah.
- **ADD STATUS ENUM("PNS", "PPPK", "Honorar"):** Ini adalah perintah untuk menambahkan kolom baru ke tabel "tabel\_guru" dengan nama "STATUS". Tipe data ENUM digunakan, yang berarti nilai yang diperbolehkan untuk kolom ini terbatas pada daftar nilai yang telah ditentukan. Nilai-nilai yang diperbolehkan dalam ENUM adalah "PNS", "PPPK", dan "Honorar".

## Kesimpulan

Perintah tersebut menambahkan kolom baru bernama "STATUS" ke tabel "tabel\_guru" dengan tipe data ENUM. Kolom ini memungkinkan nilai-nilai tertentu, yaitu "PNS", "PPPK", atau "Honorar".

## Query

```
SELECT nama_depan, MAX(usia) AS usia FROM tabel_guru;
```

12. Seleksi dan tampilkan hanya kolom "nama\_depan" dan "usia" yang memiliki nilai kolom "usia" paling tua menggunakan fungsi agregasi! Hasilnya akan seperti di bawah ini! (7 poin)

| nama_depan | usia |
|------------|------|
| Adrianty   | 34   |

```
ERROR 1630 (42000): FUNCTION sekolah_hasiza.MAX does not exist. Check the Funct
MariaDB [sekolah_hasiza]> SELECT nama_depan, MAX(usia) AS usia FROM tabel_guru;
+-----+-----+
| nama_depan | usia |
+-----+-----+
| Adrianty   | 34   |
+-----+-----+
1 row in set (0.001 sec)
```

## Analisis

- **SELECT nama\_depan, MAX(usia) AS usia:** Ini adalah perintah untuk memilih dua kolom dari tabel "tabel\_guru". Kolom pertama adalah "nama\_depan", dan kolom kedua adalah nilai maksimum dari kolom "usia", yang diberi alias "usia" menggunakan fungsi agregat MAX().
- **FROM tabel\_guru:** Ini adalah klausa yang menunjukkan bahwa data akan diambil dari tabel "tabel\_guru".

## Kesimpulan

Perintah tersebut mengambil nama depan dari guru yang memiliki usia tertinggi dari tabel "tabel\_guru". Usia maksimum dihitung menggunakan fungsi agregat MAX() dan ditampilkan dalam kolom yang diberi alias "usia".

\*\*