

# remedial basis data

## 1

### hasil & query

```
MariaDB [sekolah_jordan]> desc tabel_guru;
```

Field	Type	Null	Key	Default	Extra
id_guru	int(2)	NO	PRI	NULL	
nama_depan	varchar(100)	YES		NULL	
nama_belakang	varchar(100)	YES		NULL	
mapel	varchar(100)	YES		NULL	
jabatan	varchar(100)	YES		NULL	
usia	int(10)	YES		NULL	
tanggal_lahir	date	YES		NULL	

7 rows in set (0.225 sec)

### analisis query

- `CREATE TABLE tabel_guru` : Ini adalah perintah untuk membuat sebuah tabel baru dengan nama "tabel\_guru". Ini adalah langkah pertama dalam membuat struktur tabel baru dalam basis data.
- `( id_guru int(2) PRIMARY KEY,` : Ini mendefinisikan kolom pertama dalam tabel dengan nama "id\_guru". Tipe datanya adalah "int" (integer) dengan panjang maksimum 2 digit. Kata kunci "PRIMARY KEY" menetapkan kolom ini sebagai kunci utama (primary key) untuk tabel. Kunci utama adalah kolom atau kelompok kolom yang unik di setiap baris dan digunakan untuk mengidentifikasi secara unik setiap baris dalam tabel.
- `nama_depan varchar(100) ,` : Ini mendefinisikan kolom kedua dengan nama "nama\_depan". Tipe datanya adalah "varchar" (variabel karakter), yang memungkinkan penyimpanan string teks dengan panjang maksimum 100 karakter.
- `nama_belakang varchar(100),` : Ini mendefinisikan kolom ketiga dengan nama "nama\_belakang", dengan tipe data yang sama seperti kolom sebelumnya.
- `mapel varchar(100),` : Ini mendefinisikan kolom keempat dengan nama "mapel" (mata pelajaran). Tipe datanya adalah "varchar" dengan panjang maksimum 100 karakter.
- `jabatan varchar(100),` : Ini mendefinisikan kolom kelima dengan nama "jabatan". Tipe datanya adalah "varchar" dengan panjang maksimum 100 karakter.
- `usia int(10),` : Ini mendefinisikan kolom keenam dengan nama "usia". Tipe datanya adalah "int" (integer) dengan panjang maksimum 10 digit.

- `tanggal_lahir date);` : Ini mendefinisikan kolom terakhir dengan nama "tanggal\_lahir". Tipe datanya adalah "date" untuk menyimpan tanggal.

## kesimpulan

Program tersebut adalah perintah SQL yang digunakan untuk membuat sebuah tabel bernama "tabel\_guru" dalam sebuah basis data. Tabel ini dirancang untuk menyimpan informasi tentang guru-guru, dengan kolom-kolom sebagai berikut:

1. `id_guru` : Sebuah kolom bertipe data integer (int) dengan panjang maksimum 2 digit. Kolom ini ditetapkan sebagai kunci utama (PRIMARY KEY) untuk tabel, yang berarti nilainya harus unik dan digunakan untuk mengidentifikasi setiap baris secara unik.
2. `nama_depan` : Sebuah kolom bertipe data varchar yang digunakan untuk menyimpan nama depan guru. Panjang maksimum string yang bisa disimpan adalah 100 karakter.
3. `nama_belakang` : Sebuah kolom bertipe data varchar yang digunakan untuk menyimpan nama belakang guru. Panjang maksimum string yang bisa disimpan adalah 100 karakter.
4. `mapel` : Sebuah kolom bertipe data varchar yang digunakan untuk menyimpan mata pelajaran yang diajarkan oleh guru. Panjang maksimum string yang bisa disimpan adalah 100 karakter.
5. `jabatan` : Sebuah kolom bertipe data varchar yang digunakan untuk menyimpan jabatan guru. Panjang maksimum string yang bisa disimpan adalah 100 karakter.
6. `usia` : Sebuah kolom bertipe data integer (int) dengan panjang maksimum 10 digit. Kolom ini digunakan untuk menyimpan usia guru.
7. `tanggal_lahir` : Sebuah kolom bertipe data date yang digunakan untuk menyimpan tanggal lahir guru..

## 2

## hasil & query

```
MariaDB [sekolah_jordan]> select * from 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	Malombassang	Basis Data	Kepala sekolah	21	2000-09-21
3	Muhammad	Yusuf	Permodelan Perangkat Lunak		28	1992-12-24
4	Rusdyansyar		Pemrograman Berorientasi Objek	Asisten kepala sekolah	25	1996-01-21

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

## analisis query

- `id_guru : 1`

- `nama_depan` : Adrianty
- `nama_belakang` : (kosong)
- `mapel` : Pemrograman Web
- `jabatan` : Ketua Jurusan
- `usia` : 34
- `tanggal_lahir` : 1982-06-29

Data tersebut dimasukkan dalam format yang benar, dengan tanggal lahir diberikan dalam tanda kutip.

3. Baris berikutnya memiliki data guru kedua yang dimasukkan dengan format yang benar juga.
4. Data guru ketiga dimasukkan dengan format yang benar, dengan tanggal lahir diberikan dalam tanda kutip.
5. Data guru keempat juga dimasukkan dengan format yang benar, dengan tanggal lahir diberikan dalam tanda kutip.

## kesimpulan

Program tersebut merupakan perintah SQL untuk memasukkan data ke dalam tabel "tabel\_guru". Data yang dimasukkan meliputi informasi tentang beberapa guru, seperti nama depan, nama belakang (opsional), mata pelajaran yang diajarkan, jabatan, usia, dan tanggal lahir. Setiap baris data diwakili oleh nilai-nilai yang sesuai untuk setiap kolom dalam tabel, dipisahkan oleh koma dan diapit oleh tanda kurung. Tanggal lahir dimasukkan dalam format yang benar, yaitu diapit oleh tanda kutip tunggal. Program ini berhasil memasukkan data guru ke dalam tabel "tabel\_guru".

## 3

## hasil & query

```
MariaDB [sekolah_jordan]> INSERT INTO `tabel_guru` (`id_guru`, `nama_depan`, `nama_belakang`, `mapel`, `jabatan`, `usia`, `tanggal_lahir`) VALUES
-> (5, "Jordan", "", "ApaSaja", "Siswa", 17, 2006-10-14);
Query OK, 1 row affected, 1 warning (0.102 sec)
```

## analisis query

- `id_guru` : Ini adalah atribut yang digunakan untuk mengidentifikasi secara unik setiap guru dalam tabel. Dalam contoh ini, atribut ini memiliki tipe data integer dan memiliki nilai 5

untuk guru yang baru dimasukkan. Ini adalah kunci utama (PRIMARY KEY) dari tabel, yang berarti nilainya harus unik dan tidak boleh null.

- `nama_depan` : Atribut ini digunakan untuk menyimpan nama depan dari setiap guru. Dalam contoh ini, nilai yang dimasukkan adalah "Jordan". Atribut ini memiliki tipe data varchar dan tidak memiliki batasan panjang tertentu, tetapi dalam praktiknya biasanya akan diberikan batasan panjang.
- `nama_belakang` : Atribut ini menyimpan nama belakang dari setiap guru. Dalam contoh ini, atribut ini memiliki nilai kosong (""). Ini adalah contoh penggunaan opsional atribut, di mana tidak semua guru mungkin memiliki nama belakang. Sama seperti `nama_depan`, atribut ini memiliki tipe data varchar.
- `mapel` : Atribut ini digunakan untuk menyimpan mata pelajaran yang diajarkan oleh setiap guru. Dalam contoh ini, nilai yang dimasukkan adalah "ApaSaja". Ini adalah contoh penggunaan atribut untuk menyimpan informasi tentang mata pelajaran yang diajarkan oleh guru. Atribut ini juga memiliki tipe data varchar.
- `usia` : Atribut ini digunakan untuk menyimpan usia dari setiap guru. Dalam contoh ini, nilai yang dimasukkan adalah 17. Ini adalah contoh penggunaan atribut untuk menyimpan informasi numerik tentang usia guru. Atribut ini memiliki tipe data integer.
- `tanggal_lahir` : Atribut ini digunakan untuk menyimpan tanggal lahir dari setiap guru. Dalam contoh ini, nilai yang dimasukkan adalah '2006-10-14'. Ini adalah contoh penggunaan atribut untuk menyimpan informasi tanggal. Atribut ini memiliki tipe data date.

## kesimpulan

- `id_guru` : Atribut ini berfungsi sebagai identifikasi unik untuk setiap guru dalam tabel. Nilainya adalah bilangan bulat yang unik, dan dalam program ini, kita dapat melihat bahwa guru yang baru dimasukkan memiliki ID 5. Ini memungkinkan untuk mengakses atau merujuk ke guru tertentu dalam operasi database.
- `nama_depan` dan `nama_belakang` : Atribut ini menyimpan nama lengkap guru. Dalam contoh ini, hanya nama depan yang diisi untuk setiap guru yang dimasukkan. `nama_belakang` bersifat opsional, yang berarti tidak semua guru mungkin memiliki nama belakang dalam data. Ini memberikan fleksibilitas dalam pengisian data.
- `mapel` : Atribut ini menyimpan informasi tentang mata pelajaran yang diajarkan oleh guru. Dalam contoh ini, kita melihat bahwa guru baru mengajar mata pelajaran "ApaSaja". Ini memungkinkan untuk mengetahui spesialisasi atau bidang keahlian dari masing-masing guru.
- `usia` : Atribut ini menyimpan informasi tentang usia guru. Dalam program ini, kita dapat melihat bahwa guru yang baru dimasukkan memiliki usia 17 tahun. Ini memungkinkan kita untuk melacak dan menganalisis distribusi usia guru dalam data.

## 4

### hasil & query

```
MariaDB [sekolah_jordan]> select * from 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	Malombassang	Basis Data	Kepala sekolah	21	2000-09-21
3	Muhammad	Yusuf	Permodelan Perangkat Lunak		28	1992-12-24
4	Rusdyansyar		Pemrograman Berorientasi Objek	Asisten kepala sekolah	25	1996-01-21
5	Jordan		ApaSaja	Siswa	17	2006-10-14

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

### analisis query

- "SELECT \* FROM" digunakan untuk mengambil atau menampilkan semua data dari sebuah tabel dalam basis data
- Tabel\_guru adalah nama tabel

### kesimpulan

Perintah SQL "SELECT \* FROM tabel\_guru;" mengambil semua data yang tersimpan di dalam tabel "tabel\_guru".

## 5

### hasil & query

```
MariaDB [sekolah_jordan]> SELECT * FROM tabel_guru WHERE id_guru = 4;
```

id_guru	nama_depan	nama_belakang	mapel	jabatan	usia	tanggal_lahir
4	Rusdyansyar		Pemrograman Berorientasi Objek	Asisten kepala sekolah	25	1996-01-21

```
1 row in set (0.009 sec)
```

### analisis query

- "SELECT \* FROM" digunakan untuk mengambil atau menampilkan semua data dari sebuah tabel dalam basis data
- Tabel\_guru adalah nama tabel
- WHERE id\_guru = 4 adalah data yang akan di tampilkan

## kesimpulan

- Perintah "SELECT \* FROM tabel\_guru WHERE id\_guru = 4" memiliki tujuan untuk mengambil data dari tabel "tabel\_guru".
- Data yang diambil hanya untuk baris yang memenuhi kriteria tertentu, yaitu baris dengan nilai "id\_guru" yang sama dengan 4.
- Hasilnya adalah tampilan data yang spesifik tentang guru dengan ID 4 dalam tabel "tabel\_guru".
- Kesimpulan tersebut menyoroti bahwa perintah tersebut memberikan informasi yang terperinci dan terfokus tentang satu entitas data (dalam hal ini, guru dengan ID 4), memungkinkan pengguna untuk memahami dan menganalisis data tersebut dengan lebih baik.

## 6

## hasil & query

```
MariaDB [sekolah_jordan]> UPDATE tabel_guru SET nama_belakang="Ganteng" WHERE id_guru = 2;
Query OK, 1 row affected (0.114 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [sekolah_jordan]> select * from 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	Ganteng	Basis Data	Kepala sekolah	21	2000-09-21
3	Muhammad	Yusuf	Permodelan Perangkat Lunak		28	1992-12-24
4	Rusdyansyar		Pemrograman Berorientasi Objek	Asisten kepala sekolah	25	1996-01-21
5	Jordan		ApaSaja	Siswa	17	2006-10-14

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

## analisis query

- `UPDATE table_guru` Perintah ini menunjukkan bahwa kita akan melakukan operasi pembaruan data pada tabel "table\_guru".
- `SET nama_belakang="Ganteng"` Bagian ini menetapkan nilai baru untuk kolom "nama\_belakang". Dalam kasus ini, nilai yang ditetapkan adalah "Ganteng".
- `WHERE id_guru = 2` WHERE digunakan untuk menentukan baris mana yang akan diperbarui. Dalam kasus ini, hanya baris yang memiliki nilai "id\_guru" sama dengan 2 yang akan diperbarui.

## kesimpulan

- UPDATE tabel\_guru`Perintah tersebut memiliki tujuan untuk mengubah data dalam tabel "tabel\_guru".
- set nama belakang="ganteng" Kolom "nama\_belakang" dalam tabel "tabel\_guru" akan diperbarui dengan nilai "ganteng" untuk baris yang memiliki nilai "id\_guru" sama dengan 2.
- WHERE id\_guru = 2 Ini berarti bahwa hanya data untuk guru dengan ID 2 yang akan mengalami perubahan. Kolom "nama\_belakang" untuk guru ini akan diubah menjadi "ganteng".

## 7

### hasil & query

```
MariaDB [sekolah_jordan]> DELETE FROM tabel_guru WHERE id_guru = 5;
Query OK, 1 row affected (0.102 sec)

MariaDB [sekolah_jordan]> select * from 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	Ganteng	Basis Data	Kepala sekolah	21	2000-09-21
3	Muhammad	Yusuf	Permodelan Perangkat Lunak		28	1992-12-24
4	Rusdyansyar		Pemrograman Berorientasi Objek	Asisten kepala sekolah	25	1996-01-21

```
4 rows in set (0.129 sec)
```

### analisis query

- DELETE FROM table\_guru : Ini adalah pernyataan SQL yang menandakan bahwa kita ingin menghapus data dari tabel table\_guru .
- WHERE id\_guru = 5; : Ini adalah klausa yang menentukan kriteria untuk penghapusan data. Dalam hal ini, hanya data yang memiliki nilai kolom id\_guru sama dengan 5 yang akan dihapus.

### kesimpulan

- DELETE FROM table\_guru : Menunjukkan bahwa operasi yang dilakukan adalah penghapusan data dari tabel table\_guru .
- WHERE id\_guru = 5; : Menentukan kriteria untuk data yang akan dihapus, yaitu data yang memiliki nilai kolom id\_guru sama dengan 5.

## 8

## hasil & query

```
MariaDB [sekolah_jordan]> SELECT * FROM tabel_guru WHERE usia < 30 AND mapel LIKE 'Pem%' ORDER BY usia ASC;
```

id_guru	nama_depan	nama_belakang	mapel	jabatan	usia	tanggal_lahir
4	Rusdyansyar		Pemrograman Berorientasi Objek	Asisten kepala sekolah	25	1996-01-21

1 row in set (0.001 sec)

## analisis query

- `SELECT * FROM table_guru` : Ini adalah pernyataan SQL yang digunakan untuk memilih atau mengambil data dari tabel `table_guru`.
- `WHERE usia < 30` : Ini adalah klausa yang menentukan kriteria untuk data yang akan dipilih. Dalam hal ini, hanya data dengan nilai kolom `usia` kurang dari 30 yang akan dipilih.
- `AND mapel LIKE 'Pem%'` : Ini adalah klausa tambahan yang menambahkan kriteria. Klausa ini menyatakan bahwa hanya data dengan nilai kolom `mapel` yang dimulai dengan string 'Pem' yang akan dipilih. Penggunaan operator `LIKE` dengan '%', berarti mencocokkan pola dimana string dimulai dengan 'Pem' dan diikuti oleh karakter apa pun.
- `ORDER BY usia ASC` : Ini adalah klausa yang digunakan untuk mengurutkan hasil berdasarkan kolom `usia` secara menaik (ASC berarti ascending, atau dari yang terkecil ke yang terbesar).

## kesimpulan

`SELECT * FROM table_guru WHERE usia < 30 AND mapel LIKE 'Pem%' ORDER BY usia ASC ;` adalah bahwa program ini akan mengambil semua data dari tabel `table_guru` dimana usia guru kurang dari 30 tahun dan dia mengajar mata pelajaran yang dimulai dengan 'Pem'. Hasilnya akan diurutkan berdasarkan usia guru dari yang paling muda.

## 9

## hasil & query

```
MariaDB [sekolah_jordan]> 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 (0.002 sec)



## analisis query

1. `SELECT id_guru, nama_depan` : Ini adalah pernyataan SQL yang digunakan untuk memilih atau mengambil data dari tabel `table_guru`, khususnya hanya kolom `id_guru` dan `nama_depan` yang akan ditampilkan dalam hasilnya.
2. `FROM table_guru` : Ini menunjukkan bahwa data akan diambil dari tabel `table_guru`.
3. `WHERE nama_depan LIKE '%i%'` : Ini adalah klausa yang menentukan kriteria untuk data yang akan dipilih. Klausa ini menyatakan bahwa hanya data dengan nilai kolom `nama_depan` yang mengandung huruf "i" (tidak peduli dengan letaknya di dalam string) yang akan dipilih. Penggunaan operator `LIKE` dengan `'%i%'` berarti mencocokkan pola dimana string mengandung "i" di mana pun di dalamnya.

## kesimpulan

```
SELECT id_guru, nama_depan FROM table_guru WHERE nama_depan LIKE '%i%';
```

adalah bahwa program ini akan mengambil data dari tabel `table_guru` dimana nama depan guru mengandung huruf "i", dan hanya menampilkan kolom `id_guru` dan `nama_depan` dari data yang memenuhi kriteria tersebut.

## 10

## hasil & query

```
MariaDB [sekolah_jordan]> 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 (0.002 sec)

## analisis query

- `SELECT CONCAT_WS(" ", nama_depan, nama_belakang) AS nama_lengkap` : Ini adalah pernyataan SQL yang digunakan untuk mengambil data dari tabel `table_guru` dengan melakukan penggabungan (concatenation) kolom `nama_depan` dan `nama_belakang` menjadi satu kolom baru yang disebut `nama_lengkap`. Fungsi `CONCAT_WS` digunakan di sini untuk menggabungkan nilai dari kedua kolom dengan menggunakan spasi (" ") sebagai pemisah.

- `FROM table_guru` : Ini menunjukkan bahwa data akan diambil dari tabel `table_guru`.

## kesimpulan

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

program ini menghasilkan daftar nama lengkap dari setiap guru dalam tabel `table_guru`, yang diambil dari gabungan nilai kolom `nama_depan` dan `nama_belakang`.x

## 11

## hasil & query

```
MariaDB [sekolah_jordan]> ALTER TABLE tabel_guru ADD COLUMN status ENUM("PNS","PPPK","HONORER") NOT NULL;
Query OK, 0 rows affected (0.182 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [sekolah_jordan]> desc tabel_guru;
```

Field	Type	Null	Key	Default	Extra
id_guru	int(2)	NO	PRI	NULL	
nama_depan	varchar(100)	YES		NULL	
nama_belakang	varchar(100)	YES		NULL	
mapel	varchar(100)	YES		NULL	
jabatan	varchar(100)	YES		NULL	
usia	int(10)	YES		NULL	
tanggal_lahir	date	YES		NULL	
status	enum('PNS','PPPK','HONORER')	NO		NULL	

```
8 rows in set (0.054 sec)
```

## analisis query

- `ALTER TABLE table_guru` : Ini adalah pernyataan SQL yang digunakan untuk mengubah struktur tabel yang sudah ada, dalam hal ini `table_guru`.
- `ADD COLUMN status ENUM("PNS","PPPK","HONORER")` : Ini adalah bagian dari pernyataan yang menentukan perubahan yang ingin dilakukan pada tabel. Dalam hal ini, kita menambahkan kolom baru bernama `status` ke tabel `table_guru`. Tipe data `ENUM` digunakan untuk menentukan bahwa nilai yang diterima oleh kolom ini harus dipilih dari daftar nilai yang telah ditentukan, yaitu "PNS", "PPPK", atau "HONORER".
- `NOT NULL` : Ini menunjukkan bahwa kolom `status` tidak boleh memiliki nilai `NULL`, yang berarti setiap baris dalam tabel harus memiliki nilai untuk kolom `status`.

## kesimpulan

```
ALTER TABLE table_guru ADD COLUMN status ENUM("PNS","PPPK","HONORER") NOT NULL;
```

adalah bahwa program ini mengubah struktur tabel `table_guru` dengan menambahkan kolom `status` yang harus memiliki salah satu nilai dari tiga opsi yang telah ditentukan, dan kolom ini tidak boleh memiliki nilai `NULL`.

## 12

### hasil & query

```
MariaDB [sekolah_jordan]> SELECT nama_depan,MAX(usia) AS Usia FROM tabel_guru;
+-----+-----+
| nama_depan | Usia |
+-----+-----+
| Adrianty   | 34   |
+-----+-----+
1 row in set (0.016 sec)

MariaDB [sekolah_jordan]> _
```

### analisis query

- `SELECT nama_depan, MAX(usia) AS Usia`: Ini adalah pernyataan SQL yang digunakan untuk mengambil data dari tabel `table_guru`. Kolom `nama_depan` akan dipilih langsung tanpa modifikasi. Sedangkan untuk kolom `usia`, pernyataan `MAX(usia)` digunakan untuk mengambil nilai maksimum dari kolom `usia`. Penggunaan `AS Usia` memberikan alias "Usia" untuk nilai maksimum tersebut.
- `FROM table_guru`: Ini menunjukkan bahwa data akan diambil dari tabel `table_guru`.

### kesimpulan

```
SELECT nama_depan,MAX(usia) AS Usia FROM table_guru;
```

program ini memberikan nama depan dari guru bersama dengan usia maksimum yang tercatat dalam tabel `table_guru`.