

Nama : Anggi Juhnawan

Kelas : S4C

NPM : 201843500941

TUGAS 2 : PRAKTIKUM SISTEM BASIS DATA

```
MariaDB [tugas21] > DESC mahasiswa;
+-----+-----+-----+-----+-----+-----+
| Field | Type                | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| NPM   | int(15)             | NO   | PRI | NULL    |       |
| Nama  | varchar(150)        | NO   |     | NULL    |       |
| Alamat | varchar(150)        | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.004 sec)

MariaDB [tugas21] > DESC transaksi;
+-----+-----+-----+-----+-----+-----+
| Field | Type                | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Kode  | int(2) unsigned zerofill | NO   |     | NULL    |       |
| Jenis | varchar(150)        | NO   |     | NULL    |       |
| Harga | int(11)             | NO   |     | NULL    |       |
| Jumlah | int(11)             | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.004 sec)
```

PENJELASAN GAMBAR DIATAS:

- Pada tabel mahasiswa terdapat NPM int(15) primary key, Nama varchar(150), dan Alamat varchar(150). NPM diberi primary key karena tidak ingin memiliki nilai yang sama, jadi setiap orang-orang mempunyai NPM yang berbeda jadi ketika melakukan pencarian menggunakan NPM lebih mudah untuk menemukannya.
- Pada tabel transaksi terdapat Kode int(2) unsigned zerofill, Jenis varchar(150), Harga int(11), dan Jumlah int(11). Kode diberi tipe data int(2) karena ingin membuat angka "0" didepan dengan jumlah satu nol saja contoh : 01, 02, 03. Primary key digunakan agar data tidak ada yang sama. UNSIGNED dan ZEROFILL ini merupakan atribut default pada mysql. Pada Unsigned digunakan untuk tipe data numerik, namun berbeda sifatnya untuk tipe data INT, DECIMAL dan FLOAT. Untuk tipe data INT, atribut UNSIGNED berfungsi mengorbankan nilai negatif, untuk mendapatkan jangkauan nilai positif yang lebih tinggi. Namun untuk tipe data DECIMAL dan FLOAT, atribut UNSIGNED hanya akan menghilangkan nilai negatif, tanpa menambah jangkauan data.
ZEROFILL digunakan untuk tipe data numerik, dimana berfungsi untuk tampilan format data yang akan mengisi nilai 0 di sebelah kanan dari data. Di ZEROFILL ini lah yang bisa membuat angka nya menjadi 01,02,03. Jika menggunakan int(3) maka hasilnya akan menjadi seperti 001, 002, 003.

ALGORITMA :

- MULAI
- Fungsi menambah table mahasiswa dan transaksi di database tugas2
- Memasukan table (field1, field2, field3, ...)
- Sistem memeriksa data table yang dimasukan ke dalam sistem

- e. Bila data valid, sistem akan menyimpan data table ke dalam database di table yg digunakan
- f. Bila data tidak valid, sistem akan menampilkan informasi kesalahan data yang tidak valid.
- g. SELESAI

```
MariaDB [tugas21]> select * from mahasiswa;
```

NPM	Nama	Alamat
2009500579	Budi	Depok
2009500667	Pratiwi	Jakarta
2009500699	Lilik	Jakarta
2010500225	Quraish	Bogor
2010500239	Praningrum	Bogor
2011500193	Nurhayati	Jakarta
2011500331	Pipit	Bekasi

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

```
MariaDB [tugas21]> select * from transaksi;
```

Kode	Jenis	Harga	Jumlah
01		35000	5
03		25000	2
02		20000	2
03		27500	3

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

PENJELASAN GAMBAR DIATAS :

Syntax dibawah ini untuk menampilkan seluruh data pada table mahasiswa dan data table transaksi. Sebelum menampilkan harus menginputkan dengan cara yang berbeda disetiap tabelnya.

Untuk mahasiswa INSERT INTO nama_table VALUES (nil 1, nil 2, ...); ini digunakan agar field yang ada di table dimasukan semuanya.

Dan untuk transaksi menggunakan INSERT INTO nama_table(field1, field2, ...) VALUES (nil1, nil2, ...); ini digunakan untuk memasukan sesuai field yang diinginkan.

ALGORITMA :

- a. MULAI
- b. Fungsi menambah data pada setiap field di table mahasiswa dan transaksi
- c. Memasukan data pada setiap field di table mahasiswa dan transaksi (nilai1, nilai2, nilai3, ...)
- d. Menampilkan tabel mahasiswa
- e. Menampilkan tabel transaksi
- f. Sistem memeriksa data yang sudah dimasukan tadi ke dalam sistem
- g. Bila data valid, sistem akan menyimpan data ke dalam table yg digunakan
- h. Bila data tidak valid, sistem akan menampilkan informasi kesalahan data yang tidak valid.
- i. Menampilkan seluruh data mahasiswa
- j. Menampilkan seluruh data transaksi
- k. SELESAI

1. Tampilkan seluruh data Mahasiswa yang namanya mengandung huruf "i"!
Syntax dibawah ini digunakan untuk menampilkan nama mahasiswa yang hanya mengandung huruf i saja. Untuk menampilkan yang mengandung huruf i saja harus menggunakan '%' dibagian depan dan belakang huruf i.

ALGORITMA :

- a. MULAI
- b. Fungsi menampilkan seluruh data mahasiswa
- b. Menampilkan seluruh data Mahasiswa yang namanya mengandung huruf "i"
- c. SELESAI

```
MariaDB [tugas21]> SELECT * FROM mahasiswa WHERE Nama LIKE '%i%';
```

NPM	Nama	Alamat
2009500579	Budi	Depok
2009500667	Pratiwi	Jakarta
2009500699	Lilik	Jakarta
2010500225	Quraish	Bogor
2010500239	Praningrum	Bogor
2011500193	Nurhayati	Jakarta
2011500331	Pipit	Bekasi

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

2. Tampilkan Nama, Alamat yang beralamat di Jakarta!
Syntax dibawah ini hanya menampilkan dari table mahasiswa yang isi fieldnya hanya Nama dan Alamat dimana Alamatnya hanya berada di Jakarta saja.

ALGORITMA :

- a. MULAI
- b. Fungsi menampilkan nama dan alamat pada data mahasiswa
- d. Menampilkan Nama dan Alamat yang beralamat di Jakarta saja
- e. SELESAI

```
MariaDB [tugas21]> SELECT Nama, Alamat FROM mahasiswa WHERE Alamat = 'Jakarta';
```

Nama	Alamat
Pratiwi	Jakarta
Lilik	Jakarta
Nurhayati	Jakarta

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

3. Tampilkan seluruh data yang alamatnya di Bogor dan Bekasi!
Syntax dibawah ini untuk menampilkan data mahasiswa hanya yang beralamatkan di Bogor dan Bekasi dengan Bantuan "OR" bisa menampilkan keduanya.

ALGORITMA :

- a. MULAI
- b. Fungsi menampilkan table mahasiswa
- c. Menampilkan seluruh data mahasiswa dimana hanya alamat Bogor dan Bekasi saja yang muncul.
- d. SELESAI

```
MariaDB [tugas21]> SELECT * FROM mahasiswa WHERE Alamat='Bogor' OR Alamat='Bekasi';
```

NPM	Nama	Alamat
2010500225	Quraish	Bogor
2010500239	Praningrum	Bogor
2011500331	Pipit	Bekasi

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

4. Tampilkan seluruh data yang namanya hanya terdiri dari 4 karakter!
 Syntax dibawah ini untuk menampilkan seluruh data mahasiswa yang hanya terdiri dari 4 karakter dengan bantuan "LIKE" dan "_" (underscore). Dimana setiap satu "_" terdapat satu karakter, jadi jika ingin 4 karakter membutuhkan 4x underscore.

ALGORITMA :

- MULAI
- Fungsi menampilkan seluruh data mahasiswa
- Menampilkan seluruh data mahasiswa hanya terdiri dari 4 karakter
- SELESAI

```
MariaDB [tugas21]> SELECT * FROM mahasiswa WHERE Nama LIKE '____';
```

NPM	Nama	Alamat
2009500579	Budi	Depok

```
1 row in set (0.001 sec)
```

5. Tampilkan NPM dan Nama yang namanya berawalan dari huruf "P"!
 Syntax dibawah ini untuk menampilkan NPM dan Nama pada table mahasiswa dimana Nama yang muncul hanya yang berawalan huruf "P". Dengan memakai LIKE mempermudah proses pencarian dan ditambahkan dengan 'p%'. '%' dibelakang ini menandakan bahwa yang dicari hanya huruf depannya saja.

ALGORITMA :

- MULAI
- Fungsi menampilkan NPM dan Nama pada data mahasiswa
- Menampilkan NPM dan Nama yang namanya berawalan huruf P
- SELESAI

```
MariaDB [tugas21]> SELECT NPM, Nama FROM mahasiswa WHERE Nama Like 'p%';
```

NPM	Nama
2009500667	Pratiwi
2010500239	Praningrum
2011500331	Pipit

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

6. Tampilkan seluruh data mahasiswa yang namanya diurut dari A ke Z !
 Syntax dibawah ini untuk menampilkan seluruh data mahasiswa diurutkan dengan ORDER BY dan menggunakan ASCENDING yang berfungsi mengurutkan data dari nomor kecil ke besar atau dari abjad A ke Z.

ALGORTIMA :

- MULAI

- b. Fungsi menampilkan seluruh data mahasiswa
- c. Menampilkan seluruh data mahasiswa dengan mengurutkan namanya dari A sampai Z
- d. SELESAI

```
MariaDB [tugas21] > SELECT * FROM mahasiswa ORDER BY Nama ASC;
```

NPM	Nama	Alamat
2009500579	Budi	Depok
2009500699	Lilik	Jakarta
2011500193	Murhayati	Jakarta
2011500331	Pipit	Bekasi
2010500239	Praningrum	Bogor
2009500667	Pratiwi	Jakarta
2010500225	Quraish	Bogor

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

7. Tampilkan seluruh data yang namanya tidak mengandung huruf "U"!
 Syntax dibawah ini untuk menampilkan seluruh mahasiswa dengan tidak ada huruf U yang muncul. Ini menggunakan NOT LIKE '%u%' karena NOT itu adalah Bukan, Tidak, Terkecuali. Jika tidak diberi NOT maka yang muncul huruf yang mengandung "U".

ALGORITMA :

- a. MULAI
- b. Fungsi menampilkan seluruh data mahasiswa
- c. Menampilkan seluruh data mahasiswa yang Namanya tidak mengandung huruf "U"
- d. SELESAI

```
MariaDB [tugas21] > SELECT * FROM mahasiswa WHERE Nama NOT LIKE '%u%';
```

NPM	Nama	Alamat
2009500667	Pratiwi	Jakarta
2009500699	Lilik	Jakarta
2011500331	Pipit	Bekasi

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

8. Input nilai untuk field Jenis ="Elektronik" untuk kode 01!
 Syntax dibawah ini untuk menambahkan field Jenis isinya Elektronik dimana berkode 01. Disini tidak menggunakan INSERT INTO karena itu akan menambahkan data baru, oleh karena itu menambahkannya dengan cara diupdate Kode 01 tersebut.

ALGORITMA :

- a. MULAI
- b. Fungsi mengubah data transaksi pada Kode=01
- c. Menambahkan data Elektronik pada field Jenis
- d. Mengubah data transaksi pada field Jenis di Kode=01
- e. SELESAI

```
MariaDB [tugas21]> UPDATE transaksi SET Jenis='Elektronik' WHERE Kode='01';
Query OK, 1 row affected (0.002 sec)
Rows matched: 1 Changed: 1 Warnings: 0

MariaDB [tugas21]> SELECT * FROM transaksi;
+-----+-----+-----+-----+
| Kode | Jenis      | Harga | Jumlah |
+-----+-----+-----+-----+
| 01   | Elektronik | 35000 | 5       |
| 03   |            | 25000 | 2       |
| 02   |            | 20000 | 2       |
| 03   |            | 27500 | 3       |
+-----+-----+-----+-----+
4 rows in set (0.001 sec)
```

9. Input nilai untuk field Jenis ="Komputer" untuk kode 02!

Syntax dibawah ini untuk menambahkan field Jenis isinya Komputer dimana berkode 02. Disini tidak menggunakan INSERT INTO karena itu akan menambahkan data baru, oleh karena itu menambahkannya dengan cara diupdate Kode 02 tersebut.

ALGORITMA :

- MULAI
- Fungsi mengubah data transaksi pada Kode=02
- Menambahkan data Komputer pada field Jenis
- Mengubah data transaksi pada field Jenis di Kode=02
- SELESAI

```
MariaDB [tugas21]> UPDATE transaksi SET Jenis = 'Komputer' WHERE Kode='02';
Query OK, 1 row affected (0.003 sec)
Rows matched: 1 Changed: 1 Warnings: 0

MariaDB [tugas21]> SELECT * FROM transaksi;
+-----+-----+-----+-----+
| Kode | Jenis      | Harga | Jumlah |
+-----+-----+-----+-----+
| 01   | Elektronik | 35000 | 5       |
| 03   |            | 25000 | 2       |
| 02   | Komputer   | 20000 | 2       |
| 03   |            | 27500 | 3       |
+-----+-----+-----+-----+
4 rows in set (0.001 sec)
```

10. Input nilai untuk field Jenis ="Game Station" untuk kode 03 !

Syntax dibawah ini untuk menambahkan field Jenis isinya Game Station dimana berkode 03. Disini tidak menggunakan INSERT INTO karena itu akan menambahkan data baru, oleh karena itu menambahkannya dengan cara diupdate Kode 03 tersebut.

ALGORITMA

- MULAI
- Fungsi mengubah data transaksi pada Kode=03
- Menambahkan data Game Station pada field Jenis
- Mengubah data transaksi pada field Jenis di Kode=03
- SELESAI

```
MariaDB [tugas21] > UPDATE transaksi SET Jenis='Game Station' WHERE Kode='03';
Query OK, 2 rows affected (0.003 sec)
Rows matched: 2 Changed: 2 Warnings: 0

MariaDB [tugas21] > SELECT * FROM transaksi;
+----+-----+-----+-----+
| Kode | Jenis      | Harga | Jumlah |
+----+-----+-----+-----+
| 01   | Elektronik | 35000 | 5       |
| 03   | Game Station | 25000 | 2       |
| 02   | Komputer  | 20000 | 2       |
| 03   | Game Station | 27500 | 3       |
+----+-----+-----+-----+
4 rows in set (0.001 sec)
```

11. Tampilkan seluruh data pada tabel transaksi yang harganya antara Rp25000 s.d. Rp35000 !
 Syntax dibawah ini menampilkan seluruh data transaksi dengan harganya antara Rp. 25000 s.d 35000. Caranya menampilkan ini adalah dengan menggunakan operator logika AND (yaitu Dan atau antara) dan menggunakan operator pembandingan >= (Lebih Besar Sama Dengan) dan juga <= (Lebih Kecil Sama Dengan). Untuk memilih dari harga berapa sampai harga berapa yang ingin dimunculkan.

ALGORITMA :

- MULAI
- Fungsi menampilkan seluruh data transaksi
- Menampilkan seluruh data transaksi yang harganya antara Rp. 25000 s.d Rp. 35000
- SELESAI

```
MariaDB [tugas21] > SELECT * FROM transaksi WHERE Harga >= 25000 AND Harga <= 35000;
+----+-----+-----+-----+
| Kode | Jenis      | Harga | Jumlah |
+----+-----+-----+-----+
| 01   | Elektronik | 35000 | 5       |
| 03   | Game Station | 25000 | 2       |
| 03   | Game Station | 27500 | 3       |
+----+-----+-----+-----+
3 rows in set (0.001 sec)
```

12. Tampilkan seluruh data yang harganya bukan Rp20000!
 Syntax dibawah ini menampilkan seluruh data transaksi dimana harganya yang bukan 20000. Ini juga menggunakan banduan dari Operator pembandingan yaitu "<>" (Bukan atau Tidak Termasuk).

ALGORITMA :

- MULAI
- Fungsi menampilkan seluruh data transaksi
- Menampilkan seluruh data transaksi dimana harganya yang bukan 20000
- SELESAI

```
MariaDB [tugas21] > SELECT * FROM transaksi WHERE Harga <> 20000;
+----+-----+-----+-----+
| Kode | Jenis      | Harga | Jumlah |
+----+-----+-----+-----+
| 01   | Elektronik | 35000 | 5       |
| 03   | Game Station | 25000 | 2       |
| 03   | Game Station | 27500 | 3       |
+----+-----+-----+-----+
3 rows in set (0.001 sec)
```

13. Tampilkan Total Harga pada tabel transaksi!

Syntax dibawah ini menampilkan Total Harga pada tabel transaksi. Ini bisa menggunakan FUNGSI AGREGAT SUM untuk mencari total nilai keseluruhan yang terdapat pada suatu field yang bersifat numerik. `SELECT SUM(Harga)Total_Harga FROM transaksi;`

`SUM(Harga)` ini untuk mencari total nilai keseluruhan yang ada pada field Harga, `Total_Harga` ini untuk menjadi field sementara agar memunculkan hasil total dari nilai keseluruhan yang ada pada field Harga. Field sementara ini juga tidak bisa menggunakan spasi dan “-” (strip). Jadi bisa menggunakan `_` Underscore jika ingin menggabungkan

ALGORITMA :

- MULAI
- Fungsi SUM
- Menampilkan `Total_Harga` pada field Harga di tabel transaksi
- SELESAI

```
MariaDB [tugas21] > SELECT SUM(Harga)Total_Harga FROM transaksi;
+-----+
| Total_Harga |
+-----+
|      107500 |
+-----+
1 row in set (0.001 sec)
```

14. Tampilkan Total Jumlah pada tabel transaksi!

Syntax dibawah ini menampilkan Total Jumlah pada tabel transaksi. Ini juga bisa menggunakan FUNGSI AGREGAT COUNT untuk mencari berapa jumlah total baris yang terdapat pada suatu tabel.

ALGORITMA :

- MULAI
- Fungsi COUNT
- Menampilkan Total Jumlah yg ada pada tabel transaksi
- SELESAI

```
MariaDB [tugas21] > SELECT COUNT(*)Jumlah FROM transaksi;
+-----+
| Jumlah |
+-----+
|        4 |
+-----+
1 row in set (0.001 sec)
```

15. Tampilkan Rata-rata harga pada tabel transaksi!

Syntax dibawah ini untuk menampilkan Rata-rata harga pada tabel transaksi. Ini juga bisa menggunakan FUNGSI AGREGAT AVG untuk mencari nilai rata-rata pada suatu pada field bersifat numerik. `RataRata_Harga` disyntax ini merupakan field sementara, Field sementara ini juga tidak bisa menggunakan spasi dan “-” (strip). Jadi bisa menggunakan `_` Underscore jika ingin menggabungkan.

ALGORITMA :

- MULAI
- Fungsi AVG
- Menampilkan Rata-rata harga pada tabel transaksi
- SELESAI


```
ta_harga FROM transaksi at line 1
MariaDB [tugas21] > SELECT AVG(Harga)RataRata_Harga FROM transaksi;
+-----+
| RataRata_Harga |
+-----+
|      26875.0000 |
+-----+
1 row in set (0.001 sec)
```