

MYSQL MENGGUNAKAN COMAND LINE

1. INSTALL XAMPP
2. FILE MYSQL ADA DI C:/xampp/mysql/bin.
3. Untuk membuka pada cmd kita arahkan ke folder mysqlnya
4. Dan untuk menjalankanya mysql -u root -p
-u (untuk username defaultnya root) -p(until passwordnya default kosong)
5. Show databases; (untuk melihat database yang ada)

MEMBUAT DATABASE

1. Create database namadatabase; (untuk membuat databases baru)

MENGHAPUS DATABASE

2. Drop database namadatabase; (untuk menghapus database)

MEMILIH DATABASE YANG AKAN DIJALANKAN

3. Use namadatabase; (untuk menggunakan/memilih databasenya)

MEMBUAT TABLE DIDATABSE

4. Create table namatable (
id int primary key auto_increment, (id nama baris , primary key ini untuk memilih key, auto increment untuk menambahkan angka 1 dari setiap kolom yang dibuat secara otomatis)
nama varchar(100), (nama yaitu nama baris , varchar(100) untuk memilih karakter yang digunakan dan 100 itu berapa digit character yang digunakan ada 100 digit.)
nrp int(9),
email varchar(100),
gambar varchar(100)
); (ini untuk membuat table)
5. Show tables; (untuk melihat table yang ada didalam database)
6. Describe namatable; (untuk melihat isi dalam table)

MEMASUKAN DATA KE DALAM TABLE TERSEBUT

7. insert into mahasiswa values ("','Rizat Sakmir',17190163,'rizatsakmir@gmail.com','Teknologi Informasi','rz2.png'); (digunakan untuk memasukan data didalam table)
8. select * from mahasiswa; (untuk memilih semua yang ada di from mahasiswa)
9. select nama from mahasiswa; (untuk memilih baris nama saja yang ada di from mahasiswa)
10. select nama, nim from mahasiswa; (untuk memilih baris nama dan nim saja yang ada di from mahasiswa)
11. select * from mahasiswa where nim = '17190163'; (untuk mencari kolom nim yang memiliki nim tersebut)

MENGUPDATE DATA YANG ADA DI TABLE

1. update mahasiswa set jurusan = ‘Teknik Informatika’ where nim = ‘17190131’; (untuk mengupdate table mahasiswa menjadi jurusan Teknik informatika yang memiliki nim 17190163)

MENGHAPUS DATA YANG ADA DI TABLE

1. delete from mahasiswa where nim=’17190137’; (untuk menghapus from mahasiswa yang memiliki nim tersebut jadi akan terhapus kolom yang memiliki nim tersebut)

PENGGUNAAN SQL UNTUK PHP

1. koneksi

```
$link = mysqli_connect('localhost','root','','namadatabase');

If(!$link){

    Die('ada error'. mysqli_connect_error());

}

Mysqli_close($link);
```

2. menampilkan data

```
$query = "SELECT * FROM murid";

$hasil = mysqli_query($link, $query);

If(mysqli_num_rows($hasil)>0){

    While($data = mysqli_fetch_assoc($hasil)){

        Echo $data['nama']."' ". $data['alamat']."'<br>";

    }

}
```

3. memfilter data yang ditampilkan

```
$query = "SELECT * FROM murid WHERE alamat=condet LIMIT 3 ORDER BY id ASC";

$hasil = mysqli_query($link, $query);

If(mysqli_num_rows($hasil)>0){
```

```
While($data = mysqli_fetch_assoc($hasil)){
    Echo $data['nama']." ". $data['alamat']."<br>";
}
}
```

4. memasukan data

```
$query = "INSERT INTO murid (id,nama,umur,alamat) VALUES ('emmy',24,'utara')";
If(mysqli_query($link, $query)){
    Echo 'berhasil';
}
```

5. Delete

```
$query = "DELETE FROM murid WHERE id=1"
```

6. Update

```
$query = "UPDATE murid SET nama='sangga',umur='22' WHERE id=1"
```

```
CREATE DATABASE TOKO
```

```
CREATE TABLE
```

```
parent
```

```
Satuan - satuanId
```

```
    - satuanNama
```

```
Supplier- supplierId
```

```
    - supplierNama
```

```
    - supplierAlamat
```

```
Cabang – cabangId
```

```
    - cabangName
```

```
    - cabangAlamat
```

```
child
```

Barang – barangId

- barangNama
- harga_beli
- harga_jual
- supplier_id -> index
- satuan_id -> index

Foreign key (supplier_id) references supplier(supplierId)

ON DELETE RESTRICT

RESTRICT digunakan jika di child ada data merujuk ke parent maka apabila parent di hapus datanya tidak bisa

ON UPDATE CASCADE = cascade jika parentnya dirubah maka table childnya akan ikut berubah

SELECT * FROM namatable;

pilih semua dari table namatable

SELECT kolomid,kolomnama FROM namatable;

pilih dan tampilkan hanya kolomid dan kolomnama dari table namatable

SELECT kolomid,kolomnama FROM namatable LIMIT 10;

pilih dan batasi 10 kolom saja

SELECT kolomid AS id ,kolomnama AS nama FROM namatable LIMIT 10;

AS digunakan untuk menggantikan nama kolom

SELECT kolom id AS id ,kolomnama AS nama FROM namatable LIMIT 10 ,11 = 10,10;

untuk memilih 11 data dari urutan ke 10

SELECT CONCAT (firstname, ' ',lastname) as namalengkap;

concat berfungsi untuk menggabungkan karakter

WHERE

SELECT kolomid,kolomnama FROM namatable WHERE namakolom = 'namadata';

WHERE memfilter data dengan nama kolom motorcycles

```
SELECT kolomid,kolomnama FROM namatable WHERE namakolom != 'namadata';
```

= (!=) yang bukan termasuk

```
SELECT kolomid,kolomnama FROM namatable WHERE namakolom < '10';
```

pilih data yang kurang dari 10

```
SELECT kolomid,kolomnama FROM namatable WHERE namakolom < 10 AND namakolom2 > 15;
```

kurang dari 10 dan lebih besar 15 syaratnya harus terpenuhi keduanya

```
SELECT kolomid,kolomnama FROM namatable WHERE namakolom < 10 AND namakolom2 > 15;
```

kurang dari 10 dan lebih besar 15 salah satu syarat terpenuhi akan tampil

```
SELECT kolomid,kolomnama FROM namatable WHERE namakolom BETWEEN 50 AND 60;
```

ambil data diantara 50 sampai 60

```
SELECT * FROM namatable WHERE namakolom LIKE "%KUDA" ;
```

filter kolom dengan data seperti yang mirip dengan kuda dan diawali dengan huruf atau kalimat apapun

% = apapun karakter

_ = digunakan di tengah kalimat dengan karakter apapun

NOT LIKE = yang bukan seperti

IS NULL = memfilter data yang kosong saja

ORDER BY namakolom = mengurutkan sesuai dengan kolom namakolom defaultnya ASC

ORDER BY namakolom DESC = membalikan urutan

Contoh penggunaanya :

```
SELECT * FROM mahasiswa ORDER BY nik;
```

```
SELECT * FROM mahasiswa ORDER BY nama ASC;
```

```
SELECT * FROM mahasiswa ORDER BY nama DESC;
```

```
SELECT *
FROM mahasiswa
ORDER BY umur, nama;
```

```
SELECT * FROM mahasiswa ORDER BY umur ASC, nama DESC;
```

JOIN / INNER JOIN = digunakan untuk menggabungkan 2 table atau lebih ,mengambil data yang hanya berisian di tengah

LEFT JOIN = mengambil semua data pada parentnya walaupun tidak beririsan,

RIGHT JOIN = sama seperti left perbedaan posisi penempatan dan irisanya.

SELF JOIN = mengambil data dari table yang sama contoh kasus di table karyawan ada nama staf yang akan melapor ke atasan di table yang sama tidak menggunakan join!

Cara 1

```
Select kolomtable1,kolomtable2 From childtable1 INNER JOIN parenttable2 ON  
childtable1.idkolomtable1 = parenttable2.idkolomtab2 ;
```

ini digunakan untuk mengambil 2 data di table yang berbeda sesuai dengan idkolom

Cara 2

```
Select kolomtable1,kolomtable2 From childtable1 INNER JOIN parenttable2 USING (idkolomtable1);
```

Using dapat digunakan ketika nama kolom parent dan childnya sama

Contoh inner join

```
MariaDB [CLASSICMODELS]> SELECT a.firstName,a.jobTitle,b.firstName,b.jobTitle  
-> FROM employees AS a INNER JOIN employees AS b  
-> ON a.employeeNumber = b.reportsTo;
```

AGREGASI di isi pada SELECT

COUNT() = menghitung banyaknya data berdasarkan kolom

SUM() = menjumlah data sesuai dengan nama kolomnya

AVG() = rata rata sesuai nama kolom

MIN() = mencari data dengan nilai terendah

MAX () = mencari data dengan data tertinggi

GROUP BY digunakan bersamaan dengan fungsi agregasi karena agregasi bersifat keseluruhan data dalam kolom dan GROUP BY mengelompokkan lagi berdasarkan data

GROUP BY di tempatkan di akhir

GROUP BY namakolom

HAVING berfungsi hampir sama dengan WHERE berfungsi untuk memfilter tetapi di gunakan setelah di GROUP BY

```
MariaDB [classicmodels]> SELECT MONTH(paymentDate) AS bulan, SUM(amount) AS jumlah
    -> FROM payments
    -> WHERE YEAR(paymentDate)=2003
    -> GROUP BY MONTH(paymentDate)
    -> HAVING jumlah<150000;
+-----+-----+
| bulan | jumlah |
+-----+-----+
| 1 | 26267.620000000003 |
| 2 | 144384.36 |
| 4 | 136313.91999999998 |
+-----+
3 rows in set (0.00 sec)
```

IN () adalah sub query digunakan

Contoh penerapannya

```
MariaDB [classicmodels]> SELECT firstName,lastName
    -> FROM employees
    -> WHERE reportsTo IN (
    ->     SELECT employeeNumber FROM employees
    ->     WHERE firstName='Diane' AND lastName='Murphy');
```

SUBSTRING

Cara Penggunaan Fungsi SUBSTRING, SUBSTR dan MID MySQL

SUBSTRING (nama_kolom, index_awal, jumlah_karakter)

SELECT SUBSTRING(NIP, 4) FROM daftar_dosen: ambil seluruh karakter pada kolom daftar_dosen mulai dari index (huruf) ke-4 hingga akhir string.

SELECT SUBSTRING(NIP, 4, 3) FROM daftar_dosen: ambil seluruh karakter pada kolom daftar_dosen mulai dari index (huruf) ke-4, dan 3 karakter berikutnya.

SELECT SUBSTRING(NIP, -4) FROM daftar_dosen: ambil 4 karakter terakhir dari kolom daftar dosen.

SELECT SUBSTRING(NIP, -4, 3) FROM daftar_dosen: ambil 4 karakter terakhir dari kolom daftar dosen, kecuali 1 huruf terakhir ($4-3 = 1$).

```
mysql> SELECT NIP FROM daftar_dosen;
+-----+
| NIP |
+-----+
| 0160436012 |
| 0260432002 |
| 0275430005 |
| 0480432066 |
| 0576431001 |
| 0770435006 |
| 0869437003 |
| 1080432007 |
+-----+
8 rows in set (0.00 sec)

mysql> SELECT LEFT(NIP, 5) FROM daftar_dosen;
+-----+
| LEFT(NIP, 5) |
+-----+
| 01604 |
| 02604 |
| 02754 |
| 04804 |
| 05764 |
| 07704 |
| 08694 |
| 10804 |
+-----+
8 rows in set (0.00 sec)

mysql> SELECT RIGHT(NIP, 5) FROM daftar_dosen;
+-----+
| RIGHT(NIP, 5) |
+-----+
| 36012 |
| 32002 |
| 30005 |
| 32066 |
| 31001 |
| 35006 |
| 37003 |
| 32007 |
+-----+
8 rows in set (0.00 sec)
```

LEFT AND RIGHT

Penggunaan from dengan table baru dengan hasil data yang ada, tanpa membuat table baru.

Menggunakan table baru yang dibuat menggunakan from menggunakan group by

```
MariaDB [terminal]> select kodebank.terminal,bank,prefix,kodebank.nomorkartu,sum(amount) as TotalAmount_BankC
    -> from
    -> (select terminal,nomorkartu,substr(nomorkartu from 1 for 4) as kodebank,amount from tb_transaksiatm) as kodebank
    -> join tb_issued on kodebank.kodebank = tb_issued.prefix
    -> group by bank;
+-----+-----+-----+-----+
| terminal | bank | prefix | nomorkartu | TotalAmount_BankC |
+-----+-----+-----+-----+
| SV0001   | A    | 5567  | 55679293945 |      3000000 |
| SV0003   | C    | 7790  | 77908473632 |     1400000 |
+-----+-----+-----+-----+
2 rows in set (0.026 sec)
```

Menggunakan WHERE

```
MariaDB [terminal]> select kodebank.terminal,bank,prefix,kodebank.nomorkartu,sum(amount) as TotalAmount_Bank
    -> from
    -> (select terminal,nomorkartu,substr(nomorkartu from 1 for 4) as kodebank,amount from tb_transaksiatm) as kodebank
    -> join tb_issued on kodebank.kodebank = tb_issued.prefix where bank = "c";
+-----+-----+-----+-----+
| terminal | bank | prefix | nomorkartu | TotalAmount_Bank |
+-----+-----+-----+-----+
| SV0003   | c    | 7790  | 77908473632 |     1400000 |
+-----+-----+-----+-----+
1 row in set (0.001 sec)
```

Membuat view dengan data yang menggunakan substr dan bisa juga membuat table baru juga.

```
MariaDB [terminal]> create view tb_kodebank as
    -> select substr(nomorkartu from 1 for 4) as kodebank,amount
    -> from tb_transaksiatm;
Query OK, 0 rows affected (0.054 sec)

MariaDB [terminal]> show tables;
+-----+
| Tables_in_terminal |
+-----+
| tb_issued           |
| tb_kodebank         |
| tb_terminalatm     |
| tb_transaksiatm    |
+-----+
4 rows in set (0.001 sec)
```