

1. *Apa yang Anda ketahui mengenai MySQL?*

MySQL menurut saya adalah sebuah software yang mengelola database / DBMS Data Base Management System

MySQL nantinya bisa menyimpan banyak data base didalamnya

dan kata SQL nya ini adalah bahasa / syntax cara untuk mengakses / berintraksi dengan data basenya,

dimana setiap DBMS yang memiliki kata SQL nantinya cara penulisan syntaxnya yaitu sama

2. *Berikan contoh perintah untuk beberapa perintah berikut pada MySQL?*

# SELECT = Menampilkan

*/\* Cara Menampilkan Data Biasa pada Select \*/*

*// untuk menampilkan semua data pada table users*

*SELECT \* FROM tb\_users*

*// untuk menampilkan semua username pada table users*

*SELECT username FROM tb\_users*

*// untuk menampilkan semua id, username, email, phone pada table users*

*SELECT Id, username, email, phone FROM tb\_users*

*/\* Cara Menampilkan Data Lebih Spesifik Select \*/*

*// untuk menampilkan semua data table users yang memiliki id = 1*

*SELECT \* FROM tb\_users WHERE id = 1*

# INSERT = Memasukan

*// cara melakukan insert value pada table users, dimana values berisi (nilai\_column1, nilai\_column2, ....dst)*

*INSERT INTO tb\_users VALUES ("Cookie", "cookie@gmail.com", 085889);*

# UPDATE = Merubah

*// cara melakukan perubahan data pada table users, dengan SET nama column = diisi value data baru, berdasarkan id = 1*

*UPDATE tb\_users SET city = "Bandung", phone = 08212 WHERE id = 1*

# DELETE = Menghapus

// untuk menghapus seluruh data yang berdasarkan username Cookie

```
DELETE FROM tb_users WHERE username="Cookie";
```

# DISTINCT

// cara untuk menampilkan semua data hasil record, dari address yang berbeda, dan sedangkan yang negeranya sama tidak akan dihitung

```
SELECT DISTINCT address FROM tb_users;
```

// cara untuk mencantumkan banyaknya jumlah alamat (address) pada tabel users yang memiliki nilai berbeda, cara ini hanya mengembalikan value hasil hitungannya saja tidak menampilkan berikut isi datanya

```
SELECT COUNT(DISTINCT address) FROM tb_users;
```

# LIKE = untuk mencari data tertentu pada column data table users

// cara menemukan nilai apapun dari table users yang dimulai dari karakter "a"

```
SELECT * FROM tb_users WHERE CustomerName LIKE "a%"
```

// cara menemukan nilai apapun dari table users yang diakhiri dari karakter "a"

```
SELECT * FROM tb_users WHERE CustomerName LIKE "%a"
```

// cara menemukan nilai apapun dari table users yang memiliki karakter "atau" di posisi apa pun

```
SELECT * FROM tb_users WHERE CustomerName LIKE "%or%"
```

// cara menemukan nilai apapun dari table users yang memiliki karakter "r" di posisi kedua

```
SELECT * FROM tb_users WHERE CustomerName LIKE "_r%"
```

// cara menemukan nilai apapun dari table users yang dimulai dengan karakter "a" dan panjangnya minimal 2 karakter

```
SELECT * FROM tb_users WHERE CustomerName LIKE "a_%"
```

// cara menemukan nilai apapun dari table users yang dimulai dengan karakter "a" dan panjangnya minimal 3 karakter

```
SELECT * FROM tb_users WHERE CustomerName LIKE "a_%"
```

// cara menemukan nilai apapun dari table users yang dimulai dengan karakter "a" dan diakhiri dengan "o"

```
SELECT * FROM tb_users WHERE ContactName LIKE "a%o"
```

### 3. Jelaskan jenis-jenis JOIN, dan berikan contoh-nya satu persatu?

Pada MySQL, perintah join ada beberapa macam yaitu :

- a.Inner Join
- b.Outer Join
- c.Cross Join
- d.Union Join

a.Inner join merupakan jenis join yang paling umum yang dapat digunakan pada semua database. Jenis ini dapat digunakan bila ingin merelasikan dua set data yang ada di tabel, letak relasinya setelah pada perintah ON pada join.

Bentuk baku perintah inner join :

```
SELECT <field1>,<field2>,<fieldn> FROM <tabel1> INNER JOIN  
<tabel2>
```

```
ON <key.tabel1> = <key.tabel2>
```

b.Outer join merupakan jenis join yang sedikit berbeda dengan inner join. Pada MySQL, bentuk perintah untuk menerapkan outer join ada 2 yaitu :

- ```
SELECT <field1>,<field2>,<fieldn> FROM <tabel1> LEFT JOIN  
<tabel2> ON <key.tabel1> = <key.tabel2>
```
- ```
SELECT <field1>,<field2>,<fieldn> FROM <tabel1> RIGHT JOIN  
<tabel2> ON <key.tabel1> = <key.tabel2>
```

Left join digunakan dalam situasi ketika ingin mengembalikan semua elemen data set A, terlepas dari nilai kunci ada dalam data set B.

Right join digunakan dalam situasi ketika ingin mengembalikan semua elemen data set B, terlepas dari nilai kunci ada dalam data set A.

Bila ingin menampilkan semua data yang ada di tabel sebelah kiri baik yang mempunyai pasangan ataupun yang tidak mempunyai pasangan dengan

data pada tabel di sebelah kanan, maka gunakan left join. Sedangkan bila yang terjadi adalah kebalikannya, yaitu ingin menampilkan semua

data yang ada di sebelah kanan baik yang mempunyai pasangan ataupun yang tidak mempunyai pasangan dengan data pada tabel di sebelah kiri,

maka gunakan right join.

c. Cross join kadang kala disebut juga sebagai Cartesian Product.

Bila menggunakan cross join, maka hasil dari cross join akan menciptakan hasil yang didasarkan pada semua kemungkinan kombinasi baris dalam kedua set data. Dengan demikian, jumlah baris yang dikembalikan adalah  $N \times M$ , dimana N adalah jumlah baris dalam kumpulan data A dan M jumlah baris dalam kumpulan data B. Jelas, jumlah baris dalam cross join dapat menjadi sampah.

Bentuk perintah dari cross join adalah :

```
SELECT <field1>,<field2>,<fieldn> FROM <tabel1> CROSS JOIN  
<tabel2> atau SELECT <field1>,<field2>,<fieldn> FROM <tabel1>,  
<tabel2>
```

d. Union join didukung oleh MySQL mulai dari versi 4.0. Pemakaian union dapat menyederhanakan perintah persyaratan OR yang bertingkat. Bila dalam sebuah query menghasilkan pemakaian perintah OR yang lebih dari satu sehingga dapat membuat bingung, sebagai gantinya digunakan perintah UNION. Union dapat dikatakan sebagai perintah untuk menggabungkan hasil query sql yang fungsinya sama dengan perintah OR

#### 4. Berikan cara-cara untuk Backup dan Restore database pada MySQL?

- Backup Database MySQL Melalui Command Line / Command Prompt (Menggunakan XAMPP)

Untuk melakukan backup database MySQL melalui command line berikut perintahnya :

```
mysql -u [userdb] -p [pass] [dbname] > [backupfile.sql]
```

Keterangan.

[userdb] Nama user database Anda

[pass] Password database Anda (Pada contoh ini, database tidak diberikan password. Nantinya, perintah -p [pass] tidak dieksekusi)

[dbname] Nama database Anda

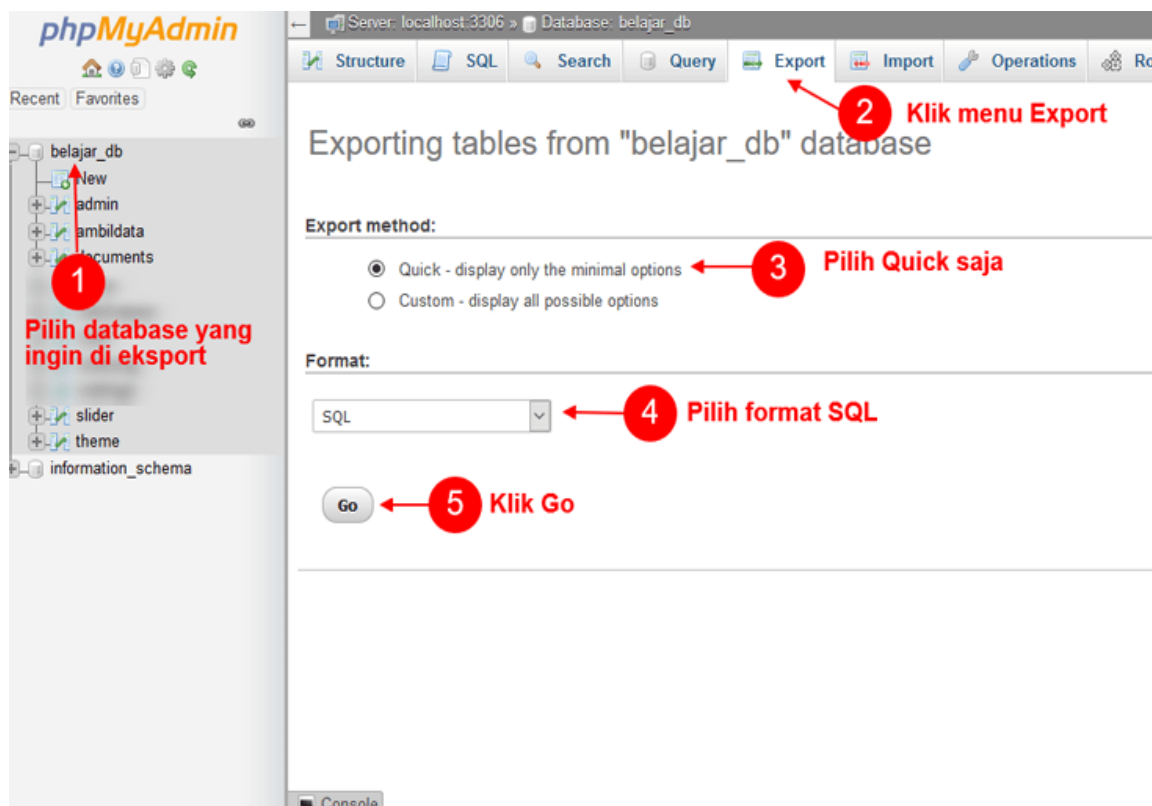
[backupfile.sql] Nama file untuk database yang Anda backup nantinya

Example:

```
mysqldump -u 1234 root myapp > myapp.sql
```

Perintah diatas akan menghasilkan sebuah file bernama myapp.sql. File tersebut berisi sintaks SQL dari database root yang di backup.

- Backup Database MySQL Menggunakan PHPMyAdmin



- *Restore Database MySQL Melalui Command Line / Command Prompt (Menggunakan XAMPP)*

berikut perintahnya :

```
mysql -u [uname] -p [pass] [db_to_restore] < [backupfile.sql]
```

keterangan.

[db\_to\_restore] Nama database yang sudah anda buat sebelum melakukan restore

Sebelum me-restore database, pastikan terlebih dahulu database yang akan di restore sudah dibuat. Nama database tidak harus sama dengan nama database backup, semua terserah pada Anda.

Berikut perintahnya :

```
mysql -u 1234 root < myapp.sql
```

## 5. *Looping Code Javascript*

```
function FizzBuzz() {  
  for (let i = 1; i <= 100; i++) {  
    if (i % 3 === 0 && i % 5 === 0) {  
      console.log(`FizzBuzz`);  
    } else if (i % 3 === 0) {  
      console.log(`Fizz`);  
    } else if (i % 5 === 0) {  
      console.log(`Buzz`);  
    } else {  
      console.log(i);  
    }  
  }  
}  
FizzBuzz();
```

```
function segitiga() {  
  let result = "";  
  for (let i = 9; i >= 1; i--) {  
    for (let j = 1; j <= i; j++) {  
      result += `${j} `;  
    }  
    result += "\n";  
  }  
  return result;  
}  
console.log(segitiga());
```