

Pemrograman Basis Data Berbasis Web

Pertemuan Ke10
(Basis Data MySQL)





Predikat diletakkan setelah klausa WHERE untuk pencarian *record database* agar mendapatkan *string*, karakter atau *range* tertentu.

1. LIKE dan NOT LIKE
2. BETWEEN
3. LIMIT
4. INNER JOIN
5. LEFT JOIN
6. RIGHT JOIN
7. UNION
8. UNION ALL
9. Sub-Query



1. LIKE dan NOT LIKE

- ❑ LIKE → untuk mendapatkan *record* yang memenuhi sebagian kriteria pencarian yaitu mencari data yang menyerupai atau seperti.
- ❑ Perintah LIKE sering dikombinasikan dengan tanda “persen” (%) dan “underscore” (_). “Persen” digunakan di awal atau akhir teks kriteria sedangkan “underscore” dimanapun diinginkan.
- ❑ Sintaks dasar dari SELECT yang melibatkan LIKE adalah
`select kolom1, kolom2, [...] FROM NAMA_TABEL
WHERE kolom_tabel LIKE %string%;`
- ❑ NOT LIKE → merupakan lawan dari LIKE. Semua data yang masuk kriteria LIKE secara otomatis tidak masuk kriteria NOT LIKE.

2. BETWEEN

- ❑ digunakan untuk menyeleksi nilai-nilai yang berada dalam kisaran (*range*), misalnya menampilkan “student” yang mempunyai IPK antara 3,00 sampai 3,50.
- ❑ Sintaks perintah SELECT dengan BETWEEN adalah
`select kolom1, kolom2, [...] FROM NAMA_TABEL
WHERE nama_kolom BETWEEN nilai1 and nilai2;`

3. LIMIT

- ❑ memungkinkan pembatasan jumlah *record* yang diambil dari database.
- ❑ Sintaks:
`select kolom1, kolom2, [...] FROM NAMA_TABEL LIMIT jumlah_limit;`



4. INNER JOIN

- JOIN digunakan bersama SELECT untuk mengkombinasikan kolom dari satu tabel dengan kolom pada tabel lain. Proses utama yang dilakukan oleh JOIN adalah mencocokkan nilai pada field kunci pada kedua tabel
- INNER JOIN mengembalikan semua baris sebagai hasil yang memenuhi suatu kondisi

5. LEFT JOIN

- Suatu LEFT JOIN mengembalikan semua baris sisi kiri kondisional bahkan jika tidak ada sisi kanan yang memenuhi sekalipun

6. RIGHT JOIN

- Suatu RIGHT JOIN akan menampilkan baris-baris sisi kanan kondisional yang memenuhi atau tidak memenuhi kondisi.



7. UNION

- ❑ untuk menggabung dua tabel, menempatkan dua *query* terpisah secara bersama membentuk satu tabel
- ❑ UNION akan memberikan hasil terbaik saat menggunakan dua tabel dengan kolom serupa karena setiap kolom harus mempunyai tipe data sama
- ❑ Sintaks dasar:

```
select kolom1, kolom2 FROM NAMA_TABEL  
UNION  
select kolom1, kolom2 FROM NAMA_TABEL;
```

8. UNION ALL

- ❑ memilih semua baris dari setiap tabel dan mengkombinasikannya ke dalam satu tabel
- ❑ Perbedaan antara UNION dan UNION ALL yaitu UNION ALL tidak akan menghapus (mengeliminir) baris-baris yang sama (*duplicate rows*) – hanya mengambil semua baris dari semua tabel sesuai *query* yang dikirim ke *server database*

9. Sub-Query

- ❑ adalah *query* SELECT yang ditempatkan di dalam suatu pernyataan SQL lain.
- ❑ dapat diletakkan sebagai bagian dari *query* utama : SELECT, INSERT, UPDATE, DELETE, SET, DO
- ❑ Sintaks:

```
select kolom1, kolom2, kolom3 [...] FROM NAMA_TABEL  
WHERE kolom_tabel = (select kolom1, kolom2, kolom3 [...] FROM  
NAME_TABEL_LAIN);
```



- PHP menyediakan fungsi untuk koneksi ke database dengan sejumlah fungsi untuk pengaturan baik menghubungkan maupun memutuskan koneksi dengan server database MySQL.
- Fungsi koneksi ke server database menggunakan pola yang sama yaitu server, port, user, password.
- Fungsi-fungsi tersebut adalah:



Fungsi ini digunakan untuk membuka koneksi dengan server MySQL. Parameter yang digunakan sebagai berikut:

- Host: nama server dengan server local dapat dengan menggunakan localhost.
- User: user yang terdaftar dalam MySQL yang digunakan untuk login ke server.
- Password: password dari user yang melakukan koneksi.
- \$conn : variabel untuk menampung hasil koneksi kepada server database

Contoh:

```
$host = "localhost";
```

```
$name = "root";
```

```
$pass = "";
```

```
$conn =mysql_connect($host,$name,$pass);
```



Fungsi ini digunakan untuk memilih database yang akan digunakan.

Contoh:

```
$conn=mysql_connect("localhost","root","")
$sukses=mysql_select_db("akademik",$conn);
```


mysql_query(perintah_query,nama_koneksi)



- Perintah ini digunakan untuk mengirimkan query ke server database melalui link nama koneksi. Fungsi ini mengembalikan nilai FALSE baik CREATE, UPDATE, DELETE, DROP, dll.

- Contoh:

```
$conn=mysql_connect("localhost","root","");
```

```
$sql="select nama_anggota,password from anggota where  
    nomor_anggota='$username'";
```

```
$rs=mysql_query($sql,$conn);
```

Fungsi-fungsi MySQL



- `mysql_connect()`
 - Untuk melakukan koneksi ke server basis data (*database*) MySQL
 - Koneksi ke database akan secara otomatis terputus pada saat script program selesai dieksekusi seluruhnya, kecuali diberikan perintah fungsi `mysql_close()`
 - Akan menghasilkan nilai `true` jika koneksi berhasil, dan nilai `false` jika gagal
 - Sintaks: **`mysql_connect(nama_host,nama_user,password);`**
- `mysql_pconnect()`
 - Fungsinya sama dengan `mysql_connect`, akan tetapi koneksi tidak akan terputus meskipun script program telah selesai dieksekusi seluruhnya
- `mysql_create_db()`
 - Untuk membuat sebuah database
 - Sintaks: **`mysql_create_db("nama_database");`**



- `mysql_select_db()`
 - Untuk memilih atau mengaktifkan database
- `mysql_query()`
 - Untuk menjalankan query yang terdapat di MySQL
- `mysql_db_query()`
 - Untuk menjalankan suatu permintaan terhadap database
- `mysql_fetch_row()`
 - Menghasilkan suatu array yang berisi seluruh kolom dari sebuah baris pada suatu himpunan hasil
 - Mengambil data secara baris per baris
 - Data yang diambil dalam bentuk *array* (elemen dari array adalah *field-field* dari tabel data)
- `mysql_fetch_array()`
 - Fungsinya sama seperti `mysql_fetch_row()`, akan tetapi hasil yang diperoleh dalam bentuk array asosiatif

Mengelola MySQL dengan PHPMyAdmin



- PHPMyAdmin dibuat khusus untuk mengelola database MySQL
- Aplikasi ini *free* dan dapat di-*download* di mysql.com, sourceforge.net dan situs web lain yang berhubungan dengan PHP dan MySQL. Bahkan, dengan menginstal PHPTriad, maka secara otomatis Apache, MySQL, PHP dan PHPTriad juga telah terinstal
- Untuk membuka aplikasi PHPMyAdmin, harus menjalankan terlebih dahulu *server web* Apache, dan *server database* MySQL, sebelum membuka aplikasi PHPMyAdmin
- PHPMyAdmin dapat diakses melalui alamat <http://localhost/PHPMyAdmin> (jika PHPMyAdmin terinstal di komputer yang sedang digunakan) atau alamat http://nama_komputer_atau_ip_address/PHPMyAdmin (jika terinstal pada komputer lain)
- XAMPP merupakan sebuah tool yang menyediakan beberapa paket perangkat lunak kedalam satu buah paket.
- Dengan Menginstal XAMPP, kita tidak perlu lagi melakukan instalasi dan konfigurasi web server apache dan MySQL secara manual. Karena XAMPP akan menginstalasi dan mengkonfigurasinya secara otomatis

Tampilan awal PHPMyAdmin



localhost / localhost | phpMyAdmin 2.7.0-pl1 - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Search Favorites

Address http://localhost/phpmyadmin/

Welcome to phpMyAdmin 2.7.0-pl1

MySQL 3.23.54-nt running on localhost as root@localhost

phpMyAdmin

Database (Databases) ...

Please select a database

MySQL

- Create new database
- Show MySQL runtime information
- Show MySQL system variables
- Show processes
- Storage Engines
- Reload MySQL
- Privileges
- Databases
- Export
- Import

phpMy Admin

- Language: English (en-iso-8859-1)
- Theme / Style: XAMPP
- phpMy Admin documentation
- Official phpMyAdmin Homepage
- [ChangeLog] [CVS] [Lists]

Your configuration file contains settings (root with no password) that correspond to the default MySQL privileged account. Your MySQL server is running with this default, is open to intrusion, and you really should fix this security hole.



- Langkah-langkah membuat database:
 1. Di halaman awal PHPMyAdmin, masukkan nama *database*, misalnya db_company pada isian **Create new database**
 2. Klik tombol **Create**

Jika berhasil, akan terdapat keterangan sebagai berikut:

Database db_company has been created.

No tables found in database.



- Langkah-langkah pembuatan tabel:

1. Pada menu HOME, pilih nama database yang akan ditambahkan tabel di dalamnya. Misalkan nama *database* adalah *db_company*.

- Create new table on database *db_company* :

Name :

Fields :

2. Pada bagian [Create new table](#) on database *db_company*, masukkan nama tabel, misalnya *EMPLOYEE* dan jumlah *field* (kolom)-nya adalah 3. Klik [Go](#).
3. Tentukan nama ketiga *field* berikut tipe datanya.

Database *db_company* - table *employee* running on *localhost*

Field	Type	Length/Values*	Attributes	Null	Default	Extra	Primary
ID	INT	4		not null			<input checked="" type="checkbox"/>
Name	VARCHAR	20		not null			<input type="checkbox"/>
Department	VARCHAR	10		not null			<input type="checkbox"/>

Table comments :

Table type :



4. Klik **Save**. Jika pembuatan tabel berhasil, maka akan muncul pesan seperti berikut ini:

Database db_company - table employee running on localhost

table employee has been created.

```
SQL-query : [Edit]
CREATE TABLE `employee` (
  `ID` INT(4) NOT NULL,
  `Name` VARCHAR(20) NOT NULL,
  `Department` VARCHAR(10) NOT NULL,
  PRIMARY KEY (`ID`)
)
COMMENT = 'Menyimpan data karyawan';
```

[Browse] [Select] [**Insert**] [Empty] [Drop]

Menyimpan data karyawan

	Field	Type	Attributes	Null	Default	Extra	Action					
<input type="checkbox"/>	ID	int(4)		No	0		Change	Drop	Primary	Index	Unique	Fulltext
<input type="checkbox"/>	Name	varchar(20)		No			Change	Drop	Primary	Index	Unique	Fulltext
<input type="checkbox"/>	Department	varchar(10)		No			Change	Drop	Primary	Index	Unique	Fulltext

↑ With selected: Or



- Langkah-langkah memasukkan data:
 - Pada sisi kiri (menu HOME), terdapat nama database db_company dan tabel-tabel di dalamnya. Klik tabel yang akan dimasukkan datanya, misalnya tabel EMPLOYEE.
 - Terdapat *link* Browse, Select, Insert, Empty, Drop. Untuk memasukkan data, klik *link* **Insert**. Pada halaman entri data, masukkan data yang diinginkan pada kolom kanan.
 - Di bawah form isian, terdapat dua buah radio button. Pilihan “Return” artinya kembali ke halaman tabel dan pengisian selesai). Sedangkan pilihan “Insert a new record” artinya memasukkan record baru berikutnya. Misalnya yang dipilih adalah **Return**.
 - Klik **Go**.

Database db_company - table employee running on localhost

Field	Type	Function	Null	Value
ID	int(4)			11
Name	varchar(20)			Arifin Akbar
Department	varchar(10)			EDP

Insert as new row -- **And** -- Return
Or
 Insert a new record



- Langkah-langkah untuk mengedit (meng-*update*) dan menghapus data:
 1. Klik link *Browse* atau *Select* pada tabel yang akan diubah atau dihapus datanya
 2. Data ditampilkan dalam bentuk tabel dan pada sisi kirinya terdapat link *Edit* dan *Delete*
 3. Untuk menghapus data, klik *Delete* dan jawab **OK** sebagai konfirmasi
 4. Untuk meng-*update* data, klik *Edit*. Pada form isian (seperti pada penambahan) betulkan data yang perlu diupdate. Akhiri dengan klik **Go**



- MySQL adalah merupakan RDBMS, sehingga ia dapat digunakan untuk memanipulasi data seperti menambah, menghapus, menampilkan, dan mengubah suatu data
- Tipe-tipe data yang terdapat di dalam MySQL adalah Tipe data Numerik, Tipe data *String*, Tipe data `char()` dan `varchar()` serta Tipe data Tanggal
- Operator SQL yang dapat digunakan di dalam MySQL: Operator matematika, Operator range, Operator kesamaan dan Operator logika
- PHP tidak menyediakan fungsi-fungsi khusus untuk operasi data, sehingga sintaks yang dipakai adalah sintaks perintah-perintah MySQL
- PHPMyAdmin dapat digunakan untuk mengelola basis data MySQL, sehingga dapat digunakan untuk membuat *database*, membuat tabel, memasukkan data ataupun mengedit (meng-*update*) dan menghapus data