



Free Kelas WAPPSANDROID

Materi Pertemuan #04

Sabtu, 24 Maret 2018

Pukul 20.00 – 23.00 WIB

Praktek PHP dan Mysql Tingkat Dasar

oleh :

Agus Muhajir

<http://github.com/hajirodeon>

Pada tutorial Lengkap belajar PHP MySQL untuk pemula ini mencakup dasar-dasar pemrograman MySQL dengan PHP. MySQL adalah sistem manajemen database yang populer. PHP tersedia gratis dan bisa bekerja dengan hampir semua perangkat lunak database, termasuk Oracle dan Sybase tetapi paling sering digunakan adalah database MySQL. Seperti PHP, perangkat lunak database MySQL tersedia secara gratis.

Karena PHP dan MySQL disediakan secara gratis, maka tidak heran jika banyak pengembang web menggunakan PHP dan MySQL. Selain gratis, PHP dan MySQL sangat stabil untuk diaplikasikan pada software skala kecil maupun besar. Banyak CMS yang telah memanfaatkan PHP dan MySQL, contohnya WordPress, Joomla, Drupal dan sebagainya.

Tutorial PHP: PHP secara singkat

- PHP adalah perangkat lunak open-source alias gratis
- PHP kependekan dari PHP: Hypertext Preprocessor (Personal Home Page).
- PHP adalah bahasa skrip server-side, lawan dari Javascript yang client-side.
- PHP hampir bisa mendukung kebanyakan database (MySQL, Informix, Oracle, Sybase, Solid, PostgreSQL, Generic ODBC, dll.)
- PHP didukung hampir semua Sistem Operasi (OS).

Tutorial MySQL: MySQL secara singkat

- MySQL adalah perangkat lunak server database open-source alias gratis
- MySQL bisa diaplikasikan dengan baik pada aplikasi skala kecil ataupun besar.
- MySQL didukung hampir semua Sistem Operasi (OS).

Belajar PHP MySQL: Persiapan awal

Sebelum melangkah lebih jauh, diharapkan anda sudah memahami poin dibawah ini:

- Anda sudah memahami dasar-dasar MySQL dan PHP.
- Download dan sudah menginstal versi terbaru dari MySQL.
- Download dan sudah menginstal Apache versi terbaru.
- Mengetahui username dan password server MySQL.

Untuk WebServer, kita telah menggunakan HP Android, sebagai tempat untuk dijadikan server, dengan menggunakan PALAPA WEB SERVER ataupun PENGUIN WEBSERVER.

Cara Membuka Koneksi Database MySQL PHP

PHP menyediakan `mysql_connect` yang berfungsi untuk membuka koneksi database. Fungsi ini membutuhkan lima parameter dan mengembalikan link identifier MySQL ketika berhasil, atau FALSE ketika gagal.

```
connection mysql_connect(server,user,passwd,new_link,client_flag);
```

Berikut penjelasan masing-masing parameter:

- **server** : nama host yang menjalankan server database. Jika tidak ditentukan maka nilai default adalah localhost: 3306.
- **user** : nama pengguna yang mengakses database. Jika tidak ditentukan maka default adalah nama user default server.
- **passwd** : password dari pengguna yang mengakses database. Jika tidak ditentukan maka akan diisi kosong.
- **new_link** : jika ada panggilan kedua yang dibuat untuk `mysql_connect ()` dengan argumen yang sama, maka tidak ada koneksi baru akan dibentuk; sebaliknya, identifier dari koneksi yang telah dibuka akan dikembalikan.

- `client_flags` : sebuah kombinasi dari konstanta berikut:
 - `MYSQL_CLIENT_SSL` – memungkinkan enkripsi SSL Gunakan
 - `MYSQL_CLIENT_COMPRESS` – memungkinkan menggunakan kompresi protokol
 - `MYSQL_CLIENT_IGNORE_SPACE` – memungkinkan spasi setelah nama fungsi
 - `MYSQL_CLIENT_INTERACTIVE` – memungkinkan menampilkan waktu timeout secara interaktif

Cara Menutup Koneksi Database MySQL PHP

Ini fungsi yang sederhana, PHP dengan fungsi `mysql_close` digunakan untuk menutup koneksi database. Fungsi ini membutuhkan koneksi yang telah dibuat oleh fungsi `mysql_connect` yang mengembalikan nilai `TRUE` jika berhasil atau `FALSE` jika gagal.

```
<?php
echo '<h1>Latihan 1 : Buka Tutup Koneksi Mysql</h1>
<hr>';

$dbhost = 'localhost';
$dbuser = 'root';
$dbpass = 'adminadmin';
$koneksi = mysql_connect($dbhost, $dbuser, $dbpass);
if(! $koneksi )
{
    die('Gagal Koneksi: ' . mysql_error());
}
```

```
}  
  
echo 'Koneksi Berhasil';  
  
mysql_close($koneksi);  
  
?>
```

Membuat Database MySQL di PHP

PHP menggunakan fungsi `mysql_query` untuk membuat database MySQL, fungsi ini membutuhkan dua parameter dan mengembalikan nilai TRUE jika berhasil atau FALSE jika gagal.

```
mysql_query( sql, connection );
```

Deskripsi parameter pada skrip diatas:

- `sql` – parameter ini diperlukan untuk menempatkan query SQL untuk membuat database
- `connection` – parameter ini bersifat opsional, jika tidak ditentukan, maka akan menggunakan koneksi dengan fungsi `mysql_connect` yang paling terakhir

```
<?php  
echo '<h1>Latihan 2 : Membuat Database</h1>  
<hr>';
```

```
//membuat database
$dbhost = 'localhost';
$dbuser = 'root';
$dbpass = 'adminadmin';
$koneksi = mysql_connect($dbhost, $dbuser, $dbpass);
if(! $koneksi )
{
    die('Gagal Koneksi: ' . mysql_error());
}
echo 'Koneksi Berhasil';
$sql = 'CREATE Database test_db';
$buatdb = mysql_query( $sql, $koneksi );
if(! $buatdb )
{
    die('Pembuatan database, gagal: ' . mysql_error());
}
echo "Database test_db berhasil dibuat\n";
mysql_close($koneksi);

?>
```

Memilih Database MySQL di PHP

Setelah Anda membuat sambungan dengan database server maka langkah selanjutnya yaitu memilih database tertentu, hal ini diperlukan karena mungkin ada beberapa database yang berada pada server induk dan kita dapat melakukan pekerjaan dengan database yang telah dipilih ini pada suatu waktu.

PHP menyediakan fungsi `mysql_select_db` untuk memilih database. Fungsi ini mengembalikan nilai `TRUE` jika berhasil atau `FALSE` jika gagal.

```
mysql_select_db( db_name, connection )
```

Deskripsi parameter pada skrip diatas:

- `db_name` – parameter ini diperlukan untuk nama database yang akan dipilih
- `connection` – parameter ini bersifat opsional, jika tidak ditentukan, maka akan menggunakan koneksi dengan fungsi `mysql_connect` yang paling terakhir

```
<?php
$dbhost = 'localhost';
$dbuser = 'root';
$dbpass = 'password';
$koneksi = mysql_connect($dbhost, $dbuser, $dbpass);
if(! $koneksi )
{
    die('Gagal Koneksi: ' . mysql_error());
}
echo 'Koneksi Berhasil';
mysql_select_db( 'test_db' );
mysql_close($koneksi );
?>
```

Membuat Tabel di Database MySQL dengan PHP

Untuk membuat tabel dalam database, yang perlu kita lakukan sama seperti kita membuat database. Pertama kita membuat query SQL untuk membuat tabel dan kemudian jalankan query tersebut menggunakan fungsi `mysql_query()`.

```
<?php
echo '<h1>Latihan 2i : Membuat Table </h1>
<hr>';

//membuat table database
$dbhost = 'localhost';
$dbuser = 'root';
$dbpass = 'adminadmin';
$koneksi = mysql_connect($dbhost, $dbuser, $dbpass);
if(! $koneksi )
{
    die('Gagal Koneksi: ' . mysql_error());
}

echo 'Koneksi Berhasil';
$sql = 'CREATE TABLE karyawan( '
    'id_karyawan INT NOT NULL AUTO_INCREMENT, '
    'nama_karyawan VARCHAR(20) NOT NULL, '
    'alm_karyawan VARCHAR(20) NOT NULL, '
    'gaji_karyawan INT NOT NULL, '
    'tgl_gabung timestamp(14) NOT NULL, '
    'primary key ( id_karyawan ))';

mysql_select_db('test_db');
```



```
$buattabel = mysql_query( $sql, $koneksi );  
if(! $buattabel )  
{  
    die('Gagal Membuat Tabel: ' . mysql_error());  
}  
echo "Tabel Karyawan sukses dibuat\n";  
mysql_close($koneksi);  
?>
```

Cara Menghapus Database MySQL di PHP

Jika database tidak lagi diperlukan maka dapat dihapus selamanya. Kita dapat menggunakan perintah SQL dalam `mysql_query` untuk menghapus database.

Perhatikan contoh skrip menghapus database MySQL di PHP berikut ini:

```
<?php
$dbhost = 'localhost';
$dbuser = 'root';
$dbpass = 'password';
$koneksi = mysql_connect($dbhost, $dbuser, $dbpass);
if(! $koneksi )
{
    die('Gagal Koneksi: ' . mysql_error());
}
$sql = 'DROP DATABASE test_db';
$hapusdb = mysql_query( $sql, $koneksi );
if(! $hapusdb )
{
    die('Gagal hapus database test_db: ' . mysql_error());
}
echo "Database berhasil dihapus\n";
mysql_close($koneksi);
?>
```

Cara Menghapus Tabel MySQL di PHP

Sekali lagi kita akan menggunakan perintah SQL melalui fungsi `mysql_query`, kali ini untuk menghapus tabel dalam database. Namun saya ingatkan untuk berhati-hati saat menggunakan perintah ini karena dengan demikian, kita dapat menghapus beberapa informasi penting yang ada dalam tabel.

Perhatikan contoh skrip menghapus tabel MySQL di PHP berikut ini:

```
<?php
$dbhost = 'localhost';
$dbuser = 'root';
$dbpass = 'password';
$koneksi = mysql_connect($dbhost, $dbuser, $dbpass);
if(! $koneksi )
{
    die('Gagal Koneksi: ' . mysql_error());
}
$sql = 'DROP TABLE karyawan';
$hapustabel = mysql_query( $sql, $koneksi );
if(! $hapustabel )
{
    die('Gagal Hapus tabel karyawan: ' . mysql_error());
}
echo "Berhasil hapus tabel karywan\n";
mysql_close($koneksi );
?
```

Insert Data ke Tabel MySQL tanpa form HTML

Data dapat dimasukkan ke dalam tabel MySQL dengan mengeksekusi pernyataan INSERT SQL melalui fungsi PHP `mysql_query`. Di bawah contoh sederhana untuk menyisipkan catatan ke dalam tabel karyawan.

Perhatikan contoh skrip insert data ke tabel di database MySQL dengan PHP berikut ini:

```
<?php
$dbhost = 'localhost';
$dbuser = 'root';
$dbpass = 'password';
$koneksi = mysql_connect($dbhost, $dbuser, $dbpass);
if(! $koneksi )
{
    die('Gagal Koneksi: ' . mysql_error());
}
$sql = 'INSERT INTO karyawan '
        '(nama_karyawan,alm_karyawan, gaji_karyawan, tgl_gabung) '
        'VALUES ( "nyekrip", "alamat nyekrip", 34000, NOW() )';

mysql_select_db('test_db');
$tambahdata = mysql_query( $sql, $koneksi );
if(! $tambahdata )
{
    die('Gagal tambah data: ' . mysql_error());
}
echo "Berhasil tambah data\n";
mysql_close($koneksi);
?>
```

Insert Data ke Tabel MySQL dengan form HTML

Dalam aplikasi nyata, kebanyakan proses insert data melalui form dalam HTML, semua nilai akan diambil menggunakan formulir HTML dan kemudian nilai-nilai tersebut akan ditangkap dengan menggunakan skrip PHP (GET atau POST) dan akhirnya nilai-nilai tersebut dimasukkan ke dalam tabel MySQL.

Saat melakukan insert atau tambah data, praktek terbaik yaitu menambahkan fungsi `get_magic_quotes_gpc()` untuk memeriksa apakah konfigurasi saat ini untuk magic quote sudah diatur atau tidak. Fungsi ini sangat bermanfaat, kebanyakan fungsi ini dipakai untuk mencegah error ketika string query yang dijalankan (dieksekusi) mengandung karakter khusus.

```
<html>
<head>
<title>Tambah DATA ke DATABASE - Nyekrip</title>
</head>
<body>
<?php
if(isset($_POST['tambah']))
{
$dbhost = 'localhost';
$dbuser = 'root';
$dbpass = '';
$koneksi = mysql_connect($dbhost, $dbuser, $dbpass);
if(! $koneksi )
{
    die('Gagal Koneksi: ' . mysql_error());
}

if(! get_magic_quotes_gpc() )
{
    $nama_karyawan = addslashes ($_POST['nama_karyawan']);
    $alm_karyawan = addslashes ($_POST['alm_karyawan']);
}
else
{
    $nama_karyawan = $_POST['nama_karyawan'];
```

```

    $alm_karyawan = $_POST['alm_karyawan'];
}
$gaji_karyawan = $_POST['gaji_karyawan'];

$sql = "INSERT INTO karyawan ".
    "(nama_karyawan,alm_karyawan, gaji_karyawan, tgl_gabung) ".
    "VALUES('$nama_karyawan','$alm_karyawan',$gaji_karyawan, NOW())";
mysql_select_db('test_db');
$tambahdata = mysql_query( $sql, $koneksi );
if( ! $tambahdata )
{
    die('Gagal Tambah Data: ' . mysql_error());
}
echo "Berhasil tambah data\n";
mysql_close($koneksi);
}
else
{
    ?>
<form method="post" action="index.php">
<table width="500" border="0" cellspacing="1" cellpadding="2">
<tr>
<td width="100">Nama Karyawan</td>
<td><input name="nama_karyawan" type="text" id="nama_karyawan"></td>
</tr>
<tr>
<td width="110">Alamat Karyawan</td>
<td><input name="alm_karyawan" type="text" id="alm_karyawan"></td>
</tr>
<tr>

```

```

<td width="110">Gaji Karyawan</td>
<td><input name="gaji_karyawan" type="text" id="gaji_karyawan"></td>
</tr>
<tr>
<td width="110"> </td>
<td> </td>
</tr>
<tr>
<td width="110"> </td>
<td>
<input name="tambah" type="submit" id="tambah" value="Tambah Karyawan">
</td>
</tr>
</table>
</form>
<?php
}
?>
</body>
</html>

```

Cara Menampilkan Data di Tabel Database

Data dapat diambil dari tabel MySQL di PHP dengan mengeksekusi pernyataan SQL **SELECT** melalui fungsi `mysql_query`. Kita memiliki beberapa pilihan untuk mengambil data dari MySQL, dan akan kita bahas satu persatu.

Untuk menarik baris data dalam tabel, pilihan yang paling sering digunakan adalah dengan menggunakan fungsi `mysql_fetch_array ()`. Fungsi ini mengembalikan baris data sebagai array asosiatif, array numerik, atau keduanya. Fungsi ini mengembalikan **FALSE** jika tidak ada lagi baris yang ditemukan.

Di bawah ini adalah contoh skrip sederhana untuk mengambil data dari tabel karyawan.

```
<?php
$dbhost = 'localhost';
$dbuser = 'root';
$dbpass = '';
$koneksi = mysql_connect($dbhost, $dbuser, $dbpass);
if(! $koneksi )
{
    die('Gagal Koneksi: ' . mysql_error());
}
$sql = 'SELECT id_karyawan, nama_karyawan, gaji_karyawan FROM karyawan';

mysql_select_db('test_db');
$ambildata = mysql_query( $sql, $koneksi);
if(! $ambildata )
{
    die('Gagal ambil data: ' . mysql_error());
}
while($row = mysql_fetch_array($ambildata, MYSQL_ASSOC))
{
    echo "ID KARYAWAN :{$row['id_karyawan']} <br> ".
        "NAMA KARYAWAN : {$row['nama_karyawan']} <br> ".
        "GAJI KARYAWAN : {$row['gaji_karyawan']} <br> ".
        "-----<br>";
}
echo "Berhasil ambil data\n";
mysql_close($koneksi);
?>
```


Cara 2 Menampilkan Data di Tabel Database

PHP menyediakan fungsi lain yang disebut `mysql_fetch_assoc ()` yang juga mengembalikan baris sebagai array asosiatif. Coba contoh skrip berikut untuk menampilkan semua data dari tabel karyawan dengan menggunakan fungsi `mysql_fetch_assoc ()`.

```
<?php
$dbhost = 'localhost';
$dbuser = 'root';
$dbpass = '';
$koneksi = mysql_connect($dbhost, $dbuser, $dbpass);
if(! $koneksi )
{
    die('Gagal Koneksi: ' . mysql_error());
}
$sql = 'SELECT id_karyawan, nama_karyawan, gaji_karyawan FROM karyawan';

mysql_select_db('test_db');
$ambildata = mysql_query( $sql, $koneksi);
if(! $ambildata )
{
    die('Gagal ambil data: ' . mysql_error());
}
while($row = mysql_fetch_assoc($ambildata))
{
    echo "ID KARYAWAN :{$row['id_karyawan']} <br> ".
        "NAMA KARYAWAN : {$row['nama_karyawan']} <br> ".
        "GAJI KARYAWAN : {$row['gaji_karyawan']} <br> ".
        "-----<br>";
}
echo "Berhasil ambil data\n";
```

```
mysql_close($koneksi);  
?>
```

Cara 3 Menampilkan Data di Tabel Database

Kita juga dapat menggunakan MYSQL_NUM, sebagai argumen kedua untuk `mysql_fetch_array ()`. Hal ini akan menyebabkan fungsi akan mengembalikan array dengan indeks numerik.

Coba contoh skrip berikut untuk menampilkan semua data dari tabel karyawan menggunakan argumen MYSQL_NUM.

```
<?php  
$dbhost = 'localhost';  
$dbuser = 'root';  
$dbpass = '';  
$koneksi = mysql_connect($dbhost, $dbuser, $dbpass);  
if(! $koneksi )  
{  
    die('Gagal Koneksi: ' . mysql_error());  
}  
$sql = 'SELECT id_karyawan, nama_karyawan, gaji_karyawan FROM karyawan';  
  
mysql_select_db('test_db');  
$ambildata = mysql_query( $sql, $koneksi);  
if(! $ambildata )  
{  
    die('Gagal ambil data: ' . mysql_error());  
}  
while($row = mysql_fetch_array($ambildata, MYSQL_NUM))
```

```
{
    echo "ID KARYAWAN :{$row[0]} <br> ".
        "NAMA KARYAWAN : {$row[1]} <br> ".
        "GAJI KARYAWAN : {$row[2]} <br> ".
        "-----<br>";
}
echo "Berhasil ambil data\n";
mysql_close($koneksi);
?>
```

Cara Update Data Tabel MySQL di PHP

Di bawah ini adalah contoh skrip sederhana untuk memperbarui data dalam tabel MySQL karyawan. Untuk memperbarui data dalam tabel MySQL, diperlukan satu kolom data dalam row untuk menemukan row data yang akan di-update dengan menggunakan klausa bersyarat.

Cara update data MySQL melalui PHP bisa dibilang cukup mudah karena kita hanya butuh mengeksekusi pernyataan UPDATE SQL melalui fungsi PHP `mysql_query`.

Pada contoh di bawah, data kolom `id_karyawan` yang dijadikan klausa bersyarat telah di format menjadi primary key untuk menjaga keunikan tiap row. Silahkan anda coba contoh skrip berikut untuk memahami operasi update. Kita menggunakan ID karyawan untuk memperbarui gaji karyawan, dan ingat untuk memastikan server apache dalam kondisi hidup.

```
<html>
<head>
    <title>Update data dalam database MySQL</title>
</head>
<body>
```

```
<?php
if(isset($_POST['update']))
{
$dbhost = 'localhost';
$dbuser = 'root';
$dbpass = '';
$conn = mysql_connect($dbhost, $dbuser, $dbpass);
if(! $conn )
{
    die("Tidak bisa Koneksi: ' . mysql_error());
}

$id = $_POST['id_karyawan'];
$gaji = $_POST['gaji_karyawan'];

$sql = "UPDATE karyawan ".
    "SET gaji_karyawan = $gaji ".
    "WHERE id_karyawan = $id" ;

mysql_select_db('db_karyawan');
$retval = mysql_query( $sql, $conn );
if(! $retval )
{
    die("Tidak bisa update data: ' . mysql_error());
}
echo "Updated data sukses\n";
mysql_close($conn);
}
else
```

```
{
?>
<form method="post" action="index.php">
<table width="500" border="0" cellspacing="1" cellpadding="2">
<tr>
<td width="100">ID Karyawan</td>
<td><input name="id_karyawan" type="text" id="id_karyawan"></td>
</tr>
<tr>
<td width="100">Gaji Karyawan</td>
<td><input name="gaji_karyawan" type="text" id="gaji_karyawan"></td>
</tr>
<tr>
<td width="100"></td>
<td></td>
</tr>
<tr>
<td width="100"></td>
<td>
<input name="update" type="submit" id="update" value="Update">
</td>
</tr>
</table>
</form>
<?php
}
?>
</body>
</html>
```

Cara hapus data Tabel di Database MySQL PHP

Setelah kita membuat form untuk menambah data, saatnya kita untuk belajar cara hapus data tabel di database MySQL dengan skrip PHP. Pada tutorial ini kita akan belajar membuat formulir untuk operasi hapus data dari tabel MySQL.

Data dapat dihapus dari tabel MySQL dengan mengeksekusi SQL DELETE melalui fungsi PHP `mysql_query`.

Di bawah ini adalah contoh skrip sederhana untuk menghapus data dari tabel karyawan. Untuk menghapus data di setiap tabel, diperlukan satu kolom data yang akan digunakan sebagai klausa bersyarat. Contoh di bawah ini menggunakan data pada kolom `id_karyawan` yang telah disetting menjadi primary key untuk mencocokkan baris data dalam tabel karyawan.

```
<html>
<head>
<title>Menghapus data MySQL - Nyekrip</title>
</head>
<body>

<?php
if(isset($_POST['hapus']))
{
$dbhost = 'localhost';
$dbuser = 'root';
$dbpass = '';
$koneksi = mysql_connect($dbhost, $dbuser, $dbpass);
if(! $koneksi )
{
    die('Gagal Koneksi: ' . mysql_error());
}
```

```
$id_karyawan = $_POST['id_karyawan'];
```

```
$sql = "DELETE FROM karyawan ".
```

```
"WHERE id_karyawan = $id_karyawan" ;
```

```
mysql_select_db('test_db');
```

```
$hapusdata = mysql_query( $sql, $koneksi );
```

```
if(! $hapusdata )
```

```
{
```

```
    die('Gagal hapus data: ' . mysql_error());
```

```
}
```

```
echo "Berhasil hapus data\n";
```

```
mysql_close($koneksi);
```

```
}
```

```
else
```

```
{
```

```
?>
```

```
<form method="post" action="index.php">
```

```
<table width="500" border="0" cellspacing="1" cellpadding="2">
```

```
<tr>
```

```
<td width="100">ID KARYAWAN</td>
```

```
<td><input name="id_karyawan" type="text" id="id_karyawan"></td>
```

```
</tr>
```

```
<tr>
```

```
<td width="100"> </td>
```

```
<td> </td>
```

```
</tr>
```

```
<tr>
```

```
<td width="100"> </td>
```

```
<td>
```

```
<input name="hapus" type="submit" id="hapus" value="Delete">
</td>
</tr>
</table>
</form>
<?php
}
?>
</body>
</html>
```

DAFTAR PUSTAKA

<http://www.nyekrip.com/tutorial-lengkap-belajar-php-mysql-untuk-pemula/>