

Contact : doni.el.tobing@gmail.com

Website : www.donitobing.com

PANDUAN PEMROGRAMAN BERBASIS WEB

MENGUNAKAN HTML, PHP, DAN DATABASE MYSQL (PART 2)

By: Doni L. Tobing,S.Kom



Menggunakan Proses Save, Edit, dan Delete Data pada Database MySQL dengan PHP

Setelah sebelumnya (Pada Part 1) telah dijelaskan mengenai perintah dasar HTML, PHP, MySQL serta cara koneksi PHP dengan MySQL, selanjutnya di sini kita akan belajar tentang proses Save, Edit, dan Delete Data menggunakan PHP.

Proses Save, Edit, Delete data atau lebih sering dikenal dengan CRUD merupakan suatu proses dasar yang harus dikuasai oleh seorang programmer baik itu programmer web ataupun desktop. Seperti pada Part 1 yang telah dijelaskan mengenai perintah dasar MySQL (INSERT, DELETE, UPDATE), di sini kita juga akan menggunakan perintah itu untuk disisipkan pada PHP.

Pertama yang harus dilakukan adalah, kita harus membuat database (di sini penulis tidak lagi memberitahu bagaimana menggunakan web server dan mengaktifkannya, karena pembaca dianggap telah mengetahuinya). Pembuatan database mysql di sini, penulis menggunakan editor bantu phpmyadmin.

Misal kita beri nama databasenya **db_kepegawaian**.

Selanjutnya buat sebuah table dengan nama **pegawai**.

Pada table pegawai tersebut masukkan 3 field, yaitu nip, nama, dan jekel

nip = No. Induk Pegawai

nama = Nama Pegawai

jekel = Jenis Kelamin Pegawai

Berikut Querynya :

```
CREATE table pegawai (  
nip varchar(10) NOT NULL primary key,  
nama varchar(30) NOT NULL,  
jekel varchar(10) NOT NULL  
)
```

Setelah database berhasil dibuat, selanjutnya kita masuk ke koding PHPnya, di sini penulis menggunakan editor Notepad++.

Perintah SAVE

Alur dan prosedur pembuatannya :

1. Buat sebuah form standar seperti di bawah ini (dengan notepad atau notepad++), simpan dengan nama pegawai.php:

Form Data Pegawai

No. Induk Pegawai

Nama

Jenis Kelamin

Berikut kodenya :

```
<html>
<head>
<title>Form Save</title>
</head>
<body>
<h3>Form Data Pegawai</h3>
    <form method='POST' action=''>
        No. Induk Pegawai <br/>
        <input type='text' name='nip'><br/>
        Nama <br/>
        <input type='text' name='nama'><br/>
        Jenis Kelamin <br/>
        <select name='jkel'>
            <option>Pria</option>
            <option>Wanita</option>
        </select><br/>
        <input type='submit' value='Simpan'>
    </form>
</body>
</html>
```

Penjelasan :

1. Perintah yang diberi warna biru merupakan tag html dasar yang berfungsi membuka dan menutup halaman html.
2. Perintah `<h3>Form Data Pegawai</h3>` , memberi judul besar yang bertuliskan Form Data Pegawai.
3. Perintah berwarna merah yang diawali dengan `<form>` dan diakhiri dengan `</form>` merupakan perintah yang digunakan untuk membuka suatu form yang di dalamnya terdapat satu atau beberapa field yang harus diisi.
4. `<form method='POST' action=''>` mengindikasikan, bahwa metode pengiriman data yang digunakan adalah POST, dan saat data dikirimkan, perintah akan membaca *action*. Action bisa diisikan untuk merujuk pada halaman proses , misal `action='simpan.php'` .
Jika `action=''` , maka perintah / proses yang dijalankan tetap pada halaman yang sama.
Untuk itu nantinya pada halaman tersebut kita juga menambahkan beberapa perintah proses.
5. Perintah `<input type='text' name='nip'>` , menampilkan sebuah objek input bertipe teks/ textbox, dengan nama variable nip, begitu juga dengan `<input type='text' name='nama'>
`
Dan perintah

```
<select name='jkel'>
    <option>Pria</option>
    <option>Wanita</option>
</select><br/>
```

Menampilkan combo box dengan 2 nilai yaitu Pria dan Wanita.

Setelah selesai membuat form, selanjutnya kita lanjutkan koding untuk proses penyimpanan datanya.

Lanjutkan koding berikut, letakkan di bawah `</form>`

```
<?php
```

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

```
$pilihdb=mysql_select_db("db_kepegawaian");
```

```
if($_POST){
```

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

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

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

```
    $simpan=mysql_query("insert into pegawai values('$nip','$nama','$jkel')") or die ("Gagal Menyimpan");
```

```

        if ($simpan){
            echo "Berhasil Disimpan..";
        }
    }
?>

```

Penjelasan :

1. Kode proses tersebut dapat diletakkan di halaman yang sama dengan form, dan dapat juga diletakkan di halaman lain. Namun di sini kita mencoba meletakkannya di halaman yang sama dengan form. Dengan ketentuan kita harus meletakkan koding untuk memPROSES data tersebut di dalam perintah

```

if ($_POST) {
    *letakkan koding disini*
}

```

2. Perintah **if(\$_POST) { }** berarti, jika suatu form dengan method POST di SEND (di SUBMIT) maka perintah yang dijalankan selanjutnya adalah perintah proses yang terdapat di antara tanda kurung kurawal yang ada pada if tersebut.
3. Perintah,

```

$konek=mysql_connect("localhost","root","");
$pilihdb=mysql_select_db("db_kepegawaian");

```

Telah dijelaskan di modul 1, berfungsi menghubungkan / koneksi ke database.

4. Selanjutnya perintah yang terdapat di antara kurung kurawal if :

```

$nip=$_POST['nip'];
$nama=$_POST['nama'];
$jekel=$_POST['jekel'];
$simpan=mysql_query("insert into pegawai values('$nip','$nama','$jekel')") or die ("Gagal Menyimpan");
if ($simpan){
    echo "Berhasil Disimpan..";
}

```

Penjelasan :

```

$nip=$_POST['nip'];

```



Maksudnya, kita memperkenalkan suatu variable dengan nama \$nip, yang diambil dari nama textbox/field bernama **nip** yang ada pada form...Jadi nip yang ada pada **\$_POST['nip']** harus sesuai dengan nama variable yang kita berikan pada textbox yang ada pada form. Begitu juga pada variable **\$nama** dan **\$jkel**.

Perintah

```
$simpan=mysql_query("insert into pegawai values('$nip','$nama','$jkel')") or die ("Gagal Menyimpan");
```

Merupakan perintah untuk menulis atau memasukkan data ke dalam database, di sini kita melihat terdapat Query INSERT INTO yang disisipkan untuk menginput data ke dalam table pegawai, dan jika proses input Gagal (perintah **or die**) maka akan tampil pesan **Gagal Menyimpan**.

Jadi kode lengkapnya adalah seperti di bawah ini (misal simpan dengan nama : **simpan.php**) :

```
<html>
<head>
<title>Form Save</title>
</head>
<body>
<h3>Form Data Pegawai</h3>
    <form method='POST' action=''>
        No. Induk Pegawai <br/>
        <input type='text' name='nip'><br/>
        Nama <br/>
        <input type='text' name='nama'><br/>
        Jenis Kelamin <br/>
        <select name='jkel'>
            <option>Pria</option>
            <option>Wanita</option>
        </select><br/>
        <input type='submit' value='Simpan'>
    </form>
</?php>
```

```

$konek=mysql_connect("localhost","root","");
$pilihdb=mysql_select_db("db_kepegawaian");
if($_POST){
    $nip=$_POST['nip'];
    $nama=$_POST['nama'];
    $jkel=$_POST['jkel'];
    $simpan=mysql_query("insert into pegawai values('$nip','$nama','$jkel')") or die ("Gagal
Menyimpan");
    if ($simpan){
        echo "Berhasil Disimpan..";
    }
}
?>
</body>
</html>

```

Selamat mencoba perintah simpan. Semoga berhasil.

Perintah EDIT dan DELETE

Sebelum masuk ke perintah edit dan delete, kita akan mencoba menampilkan isi table pada database terlebih dahulu. Untuk menampilkan isi table telah dijelaskan pada Modul 1, namun di sini kita menambahkan 2 kolom terakhir yaitu Aksi(Edit dan Delete) tampilannya seperti berikut (masih menggunakan database yang sama dengan perintah SAVE).

NIP	Nama	Jenis Kelamin	Aksi	
001	Susi	Wanita	Delete	Edit
002	Anto	Pria	Delete	Edit
003	Budi	Pria	Delete	Edit
004	Mia	Wanita	Delete	Edit

```

<html>
<head>
<title>Form Tampil</title>
</head>
<body>

<?php
$konek=mysql_connect("localhost","root","");
$pilihdb=mysql_select_db("db_kepegawaian");
echo "<table border='1'>";
echo "<tr>

        <td>NIP</td>
        <td>Nama</td>
        <td>Jenis Kelamin</td>
        <td colspan='2'>Aksi</td>
    </tr>";

    $stampil=mysql_query("select * from pegawai");
    while ($data=mysql_fetch_array($stampil)){
        $nip=$data['nip'];
        $nama=$data['nama'];
        $jkel=$data['jkel'];
        echo "<tr>

            <td>$nip</td>
            <td>$nama</td>
            <td>$jkel</td>
            <td><a href='delete_data.php?nip=$nip'>Delete</a></td>
            <td><a href='edit_data.php?nip=$nip'>Edit</a></td>
        </tr>";

    }
?>
</body>
</html>

```



Penjelasan :

1. Untuk proses menampilkan data pada table telah dijelaskan pada modul 1.
2. Yang membedakan di sini adalah, kita menambahkan 2 kolom tambahan Aksi, yaitu **Delete** dan **Edit** (Pada teks koding berwarna biru).

3. Perintah

```
<td><a href='delete_data.php?nip=$nip'>Delete</a></td>
```

Menampilkan link **Delete**, di mana jika link tersebut di klik maka akan langsung menuju ke halaman **delete_data.php**.

Sementara perintah *?nip=\$nip* Memberikan suatu variable yang akan dibawa oleh link tersebut, yaitu variable **nip** yang bernilai **\$nip**. Variabel tersebut nantinya akan dipanggil menggunakan perintah `$_GET` pada halaman **delete_data.php**.

Begitu juga dengan perintah

```
<td><a href='edit_data.php?nip=$nip'>Edit</a></td>
```

Perintah Delete

Selanjutnya, langsung saja salin kode **delete_data.php** berikut :

```
<?php
$konek=mysql_connect("localhost","root","");
$pilihdb=mysql_select_db("db_kepegawaian");
$nip=$_GET['nip'];
$delete=mysql_query("delete from pegawai where nip='$nip'") or die ("Gagal Menghapus");
header("Location:tampil.php");
?>
```

Penjelasan :

```
$nip=$_GET['nip'];
```

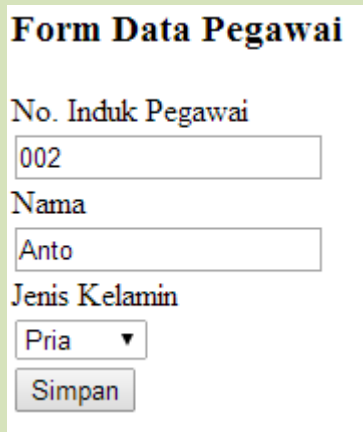
Maksudnya, memperkenalkan kembali variable nip menggunakan metode GET yang didapat dari link delete pada halaman tampil.php tadi.

```
$delete=mysql_query("delete from pegawai where nip='$nip'") or die ("Gagal Menghapus");
header("Location:tampil.php");
```

Merupakan perintah (query) untuk menghapus sebuah record dengan nip yang didapat dari variable \$nip yang telah diperkenalkan.

Perintah Edit

Untuk perintah edit, agak berbeda dengan delete, di mana di sini kita akan menampilkan form berisi nilai dari data yang akan diedit, seperti tampilan berikut :



Form Data Pegawai

No. Induk Pegawai

Nama

Jenis Kelamin

Terlihat bahwa, kita mencoba menampilkan data berdasarkan nip yang dipanggil.

Untuk pembuatan kodingnya hampir sama dengan tampil.php (dalam pembuatan form), hanya sedikit perubahan saat pemanggilan datanya.

Berikut kodingnya (simpan dengan nama **edit_data.php** / Perubahannya dengan tampil.php pada teks koding berwarna biru)

```
<html>
<head>
<title>Form Edit</title>
</head>
<body>
<?
$koneksi=mysql_connect("localhost","root","");
$pilihdb=mysql_select_db("db_kepegawaian");
$nip=$_GET['nip'];
$tampil=mysql_query("select * from pegawai where nip='$nip'");
$data=mysql_fetch_array($tampil);
$nip=$data['nip'];
$nama=$data['nama'];
$jenis_kelamin=$data['jenis_kelamin'];
?>
```

<h3>Form Data Pegawai</h3>

<form method='POST' action=''>

No. Induk Pegawai

<input type='text' name='nip' value= '<? echo "\$nip"; ?>'>

Nama

<input type='text' name='nama' value= '<? echo "\$nama"; ?>'>

Jenis Kelamin

<select name='jkel'>

<option <?if (\$jkel=='Pria') { echo "selected";}?>>Pria</option>

<option <?if (\$jkel=='Wanita') { echo "selected";}?>>Wanita</option>

</select>

<input type='submit' value='Simpan'>

</form>

<?php

if(\$_POST){

\$nip=\$_POST['nip'];

\$nama=\$_POST['nama'];

\$jkel=\$_POST['jkel'];

\$simpan=mysql_query("update pegawai set nama='\$nama', jkel='\$jkel' where nip='\$nip'") or die ("Gagal Mengupdate");

if (\$simpan){

echo "Berhasil Diedit..Kembali ke Data";

}

}

?>

</body>

</html>

Penjelasan :

```
$nip=$_GET['nip'];  
$tampil=mysql_query("select * from pegawai where nip='$nip'");  
$data=mysql_fetch_array($tampil);  
$nip=$data['nip'];  
$nama=$data['nama'];  
$jkel=$data['jkel'];
```

Merupakan script untuk memanggil data yang memiliki nip yang dipanggil dengan metode GET.

Perintah `value= '<? echo "$nip"; ?>'` memberikan nilai nip pada textbox NIP, begitu juga dengan NAMA yang menggunakan perintah `value= '<? echo "$nama"; ?>'`.

```
<select name='jkel'>  
  <option <?if ($jkel=='Pria') { echo "selected";}?>>Pria</option>  
  <option <?if ($jkel=='Wanita') { echo "selected";}?>>Wanita</option>  
</select><br/>
```

Merupakan perintah untuk membuat **selected item combo box** sesuai dengan pemanggilan data dari database (jika data jkel nya Wanita, maka option Wanita akan menjadi **selected**, begitu juga sebaliknya).

```
$simpan=mysql_query("update pegawai set nama='$nama', jkel='$jkel' where nip='$nip'") or die ("Gagal  
Mengupdate");  
if ($simpan){  
  echo "Berhasil Diedit..<a href='tampil.php'>Kembali ke Data</a>";
```

Merupakan perintah untuk update database (hampir sama dengan Query saat penyimpanan data).

Dengan demikian Proses Save, Edit, dan Delete Data telah lengkap.

Pada tutorial kali ini kita berhasil membuat totalnya ada 3 file PHP.

Yaitu :

pegawai.php	→ Menyimpan data ke database
tampil.php	→ Membaca data dari database (Menampilkan data)
delete_data.php	→ Menghapus data dari database
edit_data.php	→ Mengubah/edit data dari database

Dari proses Save, Edit , Delete tersebut kita dapat mengembangkannya menjadi aplikasi yang lebih kompleks, sekarang tergantung pembaca ingin mengembangkannya menjadi seperti apa.

-----Selamat Mencoba-----

www.donitobing.com





TENTANG PENULIS

Nama : Doni L. Tobing , S.Kom
Tempat/Tgl Lahir : Padang, 19 Januari 1992
Pendidikan : S1 Sistem Informasi Universitas Putra Indonesia
"YPTK" Padang

CONTACT

Facebook : www.facebook.com/deathaphrodite
Email : doni.el.tobing@gmail.com
Website : www.donitobing.com