

# MODUL PEMROGRAMAN WEB LANJUT



## DAFTAR ISI

Database Operation .....	1
1. SQL .....	1
1.1 Data Definition Language (DDL) .....	1
1.2 Data Manipulation Language (DML) .....	2
2. Integrasi PHP dan MySQL .....	2
2.1 Koneksi PHP dan MySQL .....	2
2.2 Menampilkan Data .....	3
2.2.1 Data Tunggal .....	3
2.2.2 Data Jamak .....	5
2.3 Menambah Data .....	9
2.4 Update Data .....	11
2.3 Menghapus Data .....	13
3. Integrasi Antar Program .....	15
Daftar Pustaka .....	18

# DATABASE OPERATION

Ketika sampai pada bab ini, anda dianggap sudah mengetahui semua hal yang menjadi dasar pemrograman web menggunakan PHP. Sekarang saatnya kita dapat memulai membangun website yang berdaya guna dan *intuitive*.

Pada bab ini, kita akan mengintegrasikan aplikasi yang telah kita buat dengan software basis data. MySQL dipilih sebagai software basis data yang akan kita gunakan karena bersifat open source, mendukung banyak platform sistem operasi, mempunyai dukungan komunitas yang melimpah dan yang tak kalah penting adalah kehandalannya. Beberapa hal penting berkaitan dengan SQL sebagai bahasa yang digunakan untuk mengolah basis data juga akan dibahas pada bab ini.

## 1. SQL

SQL (*Structure Query Language*) adalah sebuah bahasa pemrograman aras tinggi yang menjadi standar untuk pengolahan data pada sebagian besar DBMS. Secara umum, terdapat dua kelompok penggunaan perintah SQL :

- Data Definition Language
- Data Manipulation Language

### 1.1 Data Definition Language (DDL)

Data Definition Language adalah perintah SQL yang digunakan untuk mendefinisikan sebuah objek berkaitan dengan basis data. Misal : CREATE, DROP dan ALTER.

06.latihan\_01.sql

```
CREATE DATABASE db_pw;
```

06.latihan\_02.sql

```
CREATE TABLE biodata (  
    id INT(5) NOT NULL PRIMARY KEY AUTO_INCREMENT,  
    nama VARCHAR(25) NOT NULL,  
    alamat TEXT NULL  
);
```

## 1.2 Data Manipulation Language (DML)

Data Manipulation Language adalah perintah SQL yang digunakan untuk melakukan manipulasi terhadap object basis data. Misal : SELECT, INSERT, UPDATE dan DELETE.

06.latihan\_03.sql

```
INSERT INTO biodata (nama, alamat) VALUES
('Sugeng Riyadi', 'Jl. Mangkubumi No. 145 Yogyakarta'),
('Siti Nurhalimah', 'Jl. Jangkar Bumi No. 80 Sleman'),
('Febrianto Prabowo', 'RT 01 RW 04 Taman Siswa Yogyakarta'),
('Evi Susilowati', 'Jl. Mawar No. 56 Baciro Yogyakarta');
```

06.latihan\_04.sql

```
SELECT * FROM biodata;
```

id	nama	alamat
1	Sugeng Riyadi	Jl. Mangkubumi No. 145 Yogyakarta
2	Siti Nurhalimah	Jl. Jangkar Bumi No. 80 Sleman
3	Febrianto Prabowo	RT 01 RW 04 Taman Siswa Yogyakarta
4	Evi Susilowati	Jl. Mawar No. 56 Baciro Yogyakarta

## 2. Integrasi PHP dan MySQL

### 2.1 Koneksi PHP dan MySQL

Untuk melakukan koneksi terhadap sebuah server basis data, kita membutuhkan alamat server dan hak akses kepada server basis data yang terdiri dari username dan password.

Berikut ini adalah contoh penerapan koneksi server basis data melalui PHP.

Nama Server      Username      Password

```
mysql_connect("localhost", "root", "amikom0k3");
```

Setelah melakukan koneksi terhadap server basis data, langkah selanjutnya yaitu memilih basis data yang akan kita operasikan. Jika mengacu pada contoh latihan SQL diatas, maka basis data yang akan kita pilih adalah **db\_pw**. Berikut ini contoh penerapannya:

Database Name  
  
`mysql_select_db("db_pw");`

Pada contoh diatas fungsi `mysql_select_db()` digunakan untuk memilih database yang akan dioperasikan yaitu `db_pw`. Sehingga secara lengkap kedua contoh diatas dapat ditulis seperti latihan dibawah ini:

06.latihan\_04.php atau simpan dengan nama koneksi.php

```
<?php

mysql_connect("localhost","root","amikom0k3") or exit("Gagal Koneksi DB.");
mysql_select_db("db_pw") or exit("Gagal Memilih DB.");

?>
```

## 2.2 Menampilkan Data

Setelah melakukan koneksi terhadap basis data yang akan dioperasikan, sekarang kita bisa memulai untuk melakukan pengolahan dan manipulasi terhadap data yang terkandung dalam basis data tersebut. Operasi yang akan kita lakukan kali ini adalah menampilkan data.

### 2.2.1 Data Tunggal

06.latihan\_05.php

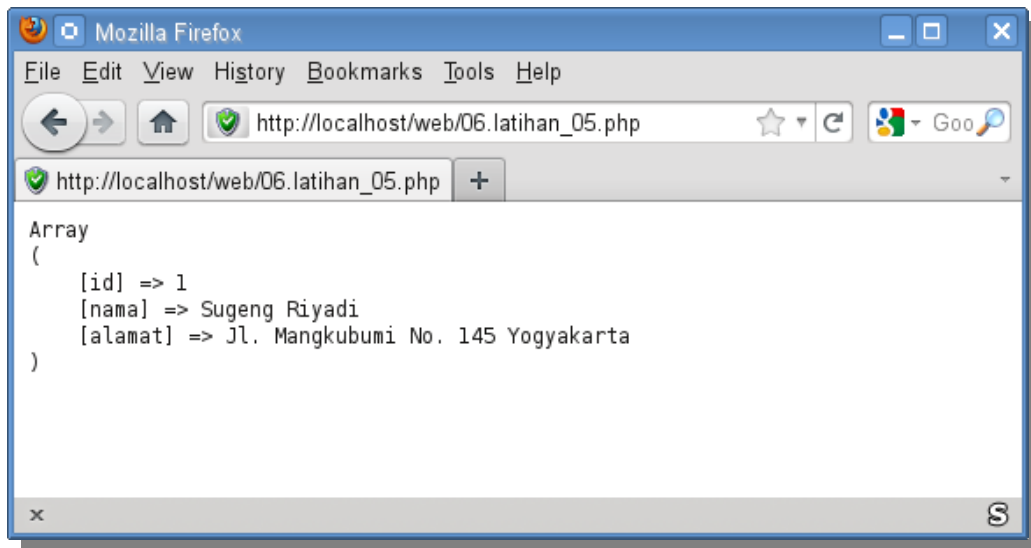
```
<?php

include("koneksi.php");

$sql = "SELECT id,nama,alamat FROM biodata WHERE id = 1";
$hasil = mysql_query($sql) or exit("Error query : <b>".$sql."</b>.");
$data = mysql_fetch_assoc($hasil);

echo "<pre>";
print_r($data);
echo "</pre>";

?>
```



Skip contoh latihan program diatas dapat dijelaskan sebagai berikut:

1. Pertama-tama, program akan menyertakan file `koneksi.php` melalui fungsi `include()`, dimana file tersebut bertugas untuk membuat sebuah koneksi ke server basis data, kemudian memilih basis data yang akan dioperasikan.
2. Query `"SELECT id,nama,alamat FROM biodata WHERE id = 1"` disimpan dalam variabel `$sql`, query tersebut dijalankan melalui fungsi `mysql_query()`. Yang mana hasil dari query tersebut akan disimpan pada variabel `$hasil`. Jika query gagal dijalankan maka program akan berhenti dan menampilkan pesan `Error query : <SQL_ERROR>`. Pesan kesalahan / error ini akan sangat kita butuhkan untuk melakukan *tracking* kesalahan. Karena dengan ditampilkannya pesan kesalahan, akan mudah bagi kita untuk menguji syntax SQL tersebut secara langsung melalui MySQL command line atau PHPMyAdmin.
3. Hasil query yang terkandung dalam variabel `$hasil` merupakan data mentah yang perlu diolah kembali. Kemudian program menjalankan fungsi `mysql_fetch_assoc()` yang berguna untuk mengolah data mentah tadi ke dalam bentuk array, data array ini disimpan ke variabel `$data`.
4. Oleh karena variabel `$data` berbentuk sebuah array, maka fungsi `print_r()` dapat digunakan untuk melihat deskripsi (*index* dan *value*) dari sebuah array.
5. Jika kita perhatikan, *index* asosiatif yang terdapat pada array `$data` adalah nama kolom yang sebelumnya didefinisikan pada query yaitu `id,nama,alamat`. Untuk lebih jelasnya, modifikasi contoh program sebelumnya sehingga menjadi seperti dibawah ini :

## 06.latihan\_06.php

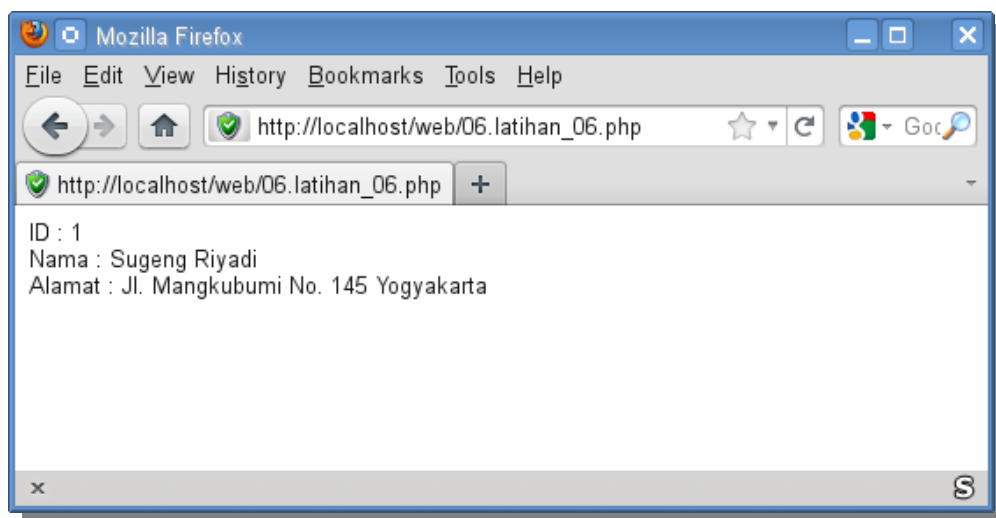
```
<?php

include("koneksi.php");

$sql = "SELECT * FROM biodata WHERE id = 1";
$hasil = mysql_query($sql) or exit("Error query : <b>".$sql."</b>.");
$data = mysql_fetch_assoc($hasil);

echo "ID : ".$data['id'];
echo "<br />";
echo "Nama : ".$data['nama'];
echo "<br />";
echo "Alamat : ".$data['alamat'];

?>
```



### 2.2.2 Data Jamak

PHP tidak menyediakan fungsi yang secara khusus digunakan untuk menangani pengambilan banyak data secara sekaligus. Ini terjadi karena PHP menampilkan data baris demi baris. Untuk lebih jelasnya perhatikan contoh syntax SQL berikut ini:

```
SELECT id,nama,alamat FROM biodata;
```

id	nama	alamat
1	Sugeng Riyadi	Jl. Mangkubumi No. 145 Yogyakarta
2	Siti Nurhalimah	Jl. Jangkar Bumi No. 80 Sleman
3	Febrianto Prabowo	RT 01 RW 04 Taman Siswa Yogyakarta
4	Evi Susilowati	Jl. Mawar No. 56 Baciro Yogyakarta

Kemudian kita akan menerapkan query tersebut diatas menggunakan melalui PHP:

#### 06.latihan\_07.php

```
<?php
include("koneksi.php");

$sql = "SELECT id,nama,alamat FROM biodata";
$hasil = mysql_query($sql) or exit("Error query : <b>".$sql."</b>.");

echo "<pre>";

$data = mysql_fetch_assoc($hasil);
print_r($data);

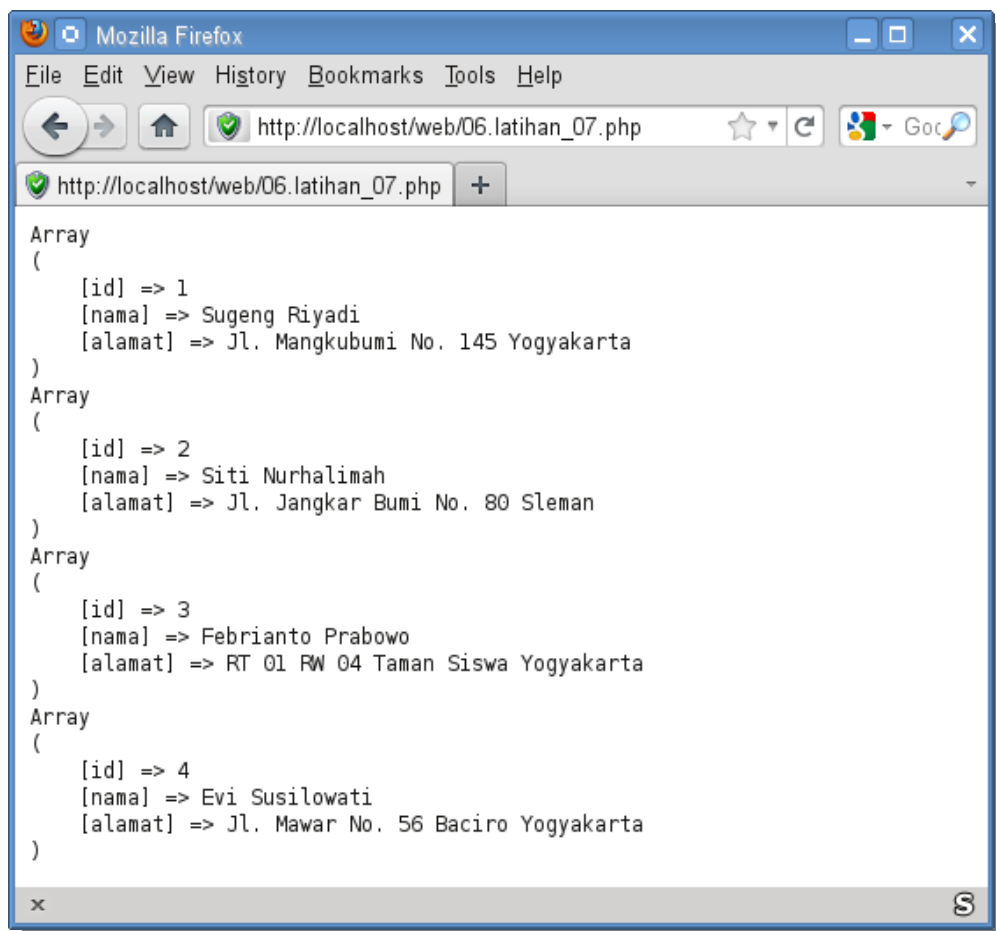
$data = mysql_fetch_assoc($hasil);
print_r($data);

$data = mysql_fetch_assoc($hasil);
print_r($data);

$data = mysql_fetch_assoc($hasil);
print_r($data);

echo "</pre>";

?>
```





Untuk menampilkan banyak data, digunakan fungsi `mysql_fetch_assoc($hasil)` yang dilakukan berulang-ulang sampai data yang diinginkan telah mencukupi atau sampai kondisi dimana data yang dihasilkan dari query telah habis. Oleh sebab itu, sebuah perulangan dapat digunakan untuk menggantikan pemanggilan fungsi tersebut secara berulang-ulang. Perhatikan contoh latihan skrip berikut ini:

#### 06.latihan\_08.php

```
<?php
include("koneksi.php");

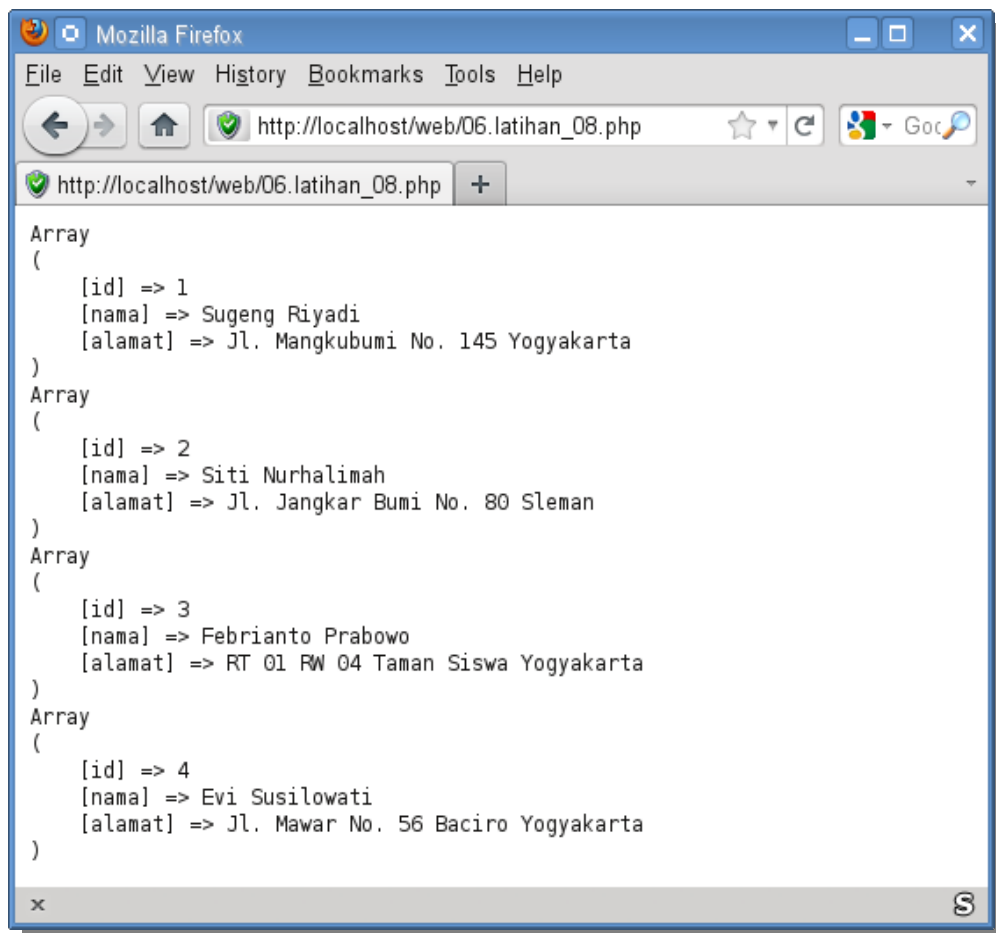
$sql = "SELECT id,nama,alamat FROM biodata";
$hasil = mysql_query($sql) or exit("Error query : <b>".$sql."</b>.");

echo "<pre>";

while($data = mysql_fetch_assoc($hasil)){
    print_r($data);
}

echo "</pre>";

?>
```



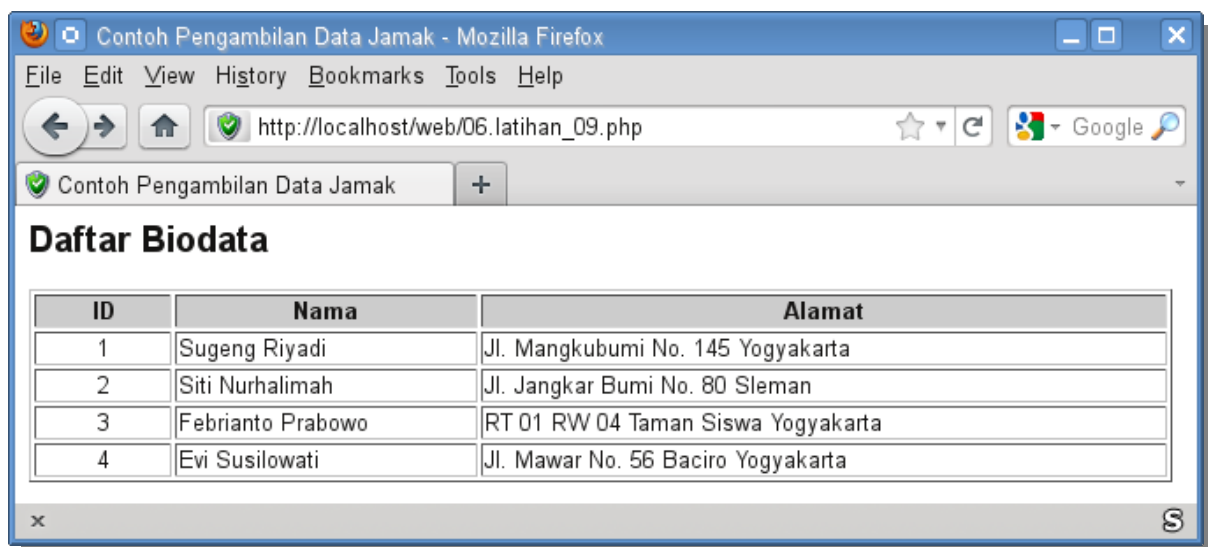
Pada contoh latihan diatas, perulangan **while** akan mengambil semua data hasil query secara bertahap sampai dengan data tersebut habis. Seperti kita ketahui, perulangan menggunakan **while** akan berhenti jika syarat / kondisi perulangan bernilai **false**. Jika data hasil query tersebut telah habis, maka syarat perulangan dianggap **false**.

06.latihan\_09.php

```
<html>
  <head>
    <title>Contoh Pengambilan Data Jamak</title>
  </head>
  <body>
    <h2>Daftar Biodata</h2>
    <table width="600px" border="1">
      <tr style="background:#ccc">
        <th width="10%">ID</th>
        <th width="35%">Nama</th>
        <th>Alamat</th>
      </tr>
      <?php
        include("koneksi.php");

        $sql = "SELECT id,nama,alamat FROM biodata";
        $hasil = mysql_query($sql) or exit("Error query: <b>".$sql."</b>.");

        while($data = mysql_fetch_assoc($hasil)){ ?>
          <tr>
            <td align="center"><?php echo $data['id']; ?></td>
            <td><?php echo $data['nama']; ?></td>
            <td><?php echo $data['alamat']; ?></td>
          </tr>
        <?php
        }
      ?>
    </table>
  </body>
</html>
```



## 2.3 Menambah Data

Pada modul sebelumnya, telah dijelaskan bagaimana cara mengirimkan data melalui sebuah form. Pada contoh kali ini, kita akan mengolah data yang telah dikirimkan melalui form tersebut untuk dimasukkan / ditambahkan ke dalam basis data.

06.latihan\_10.html

```
<html>
  <head>
    <title>Tambah Data</title>
  </head>
  <body>
    <form action="06.latihan_11.php" method="post">
      Nama : &nbsp;<input type="text" name="txtNama" />
      <br />
      Alamat : <textarea name="txtAlamat"></textarea>
      <br />
      <input type="submit" value="Submit" />
    </form>
  </body>
</html>
```

06.latihan\_11.php

```
<?php

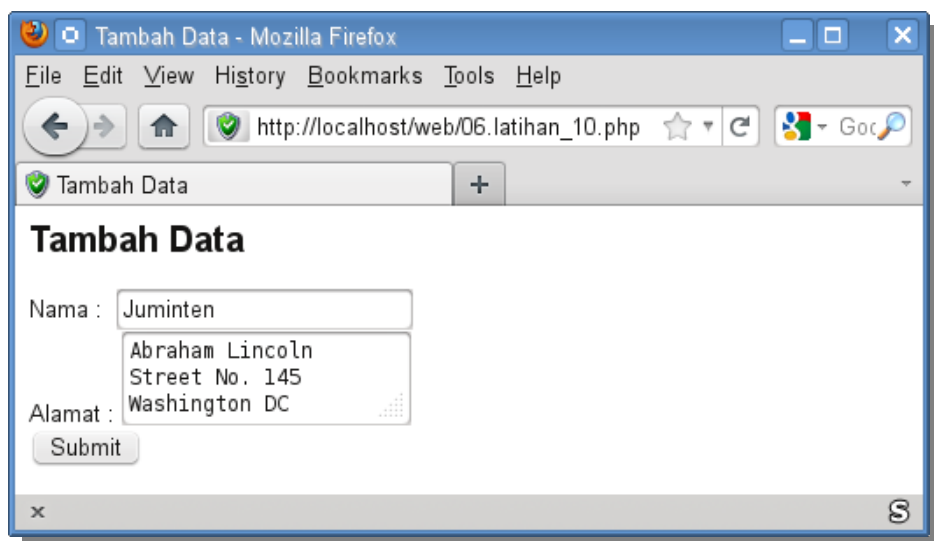
include("koneksi.php");

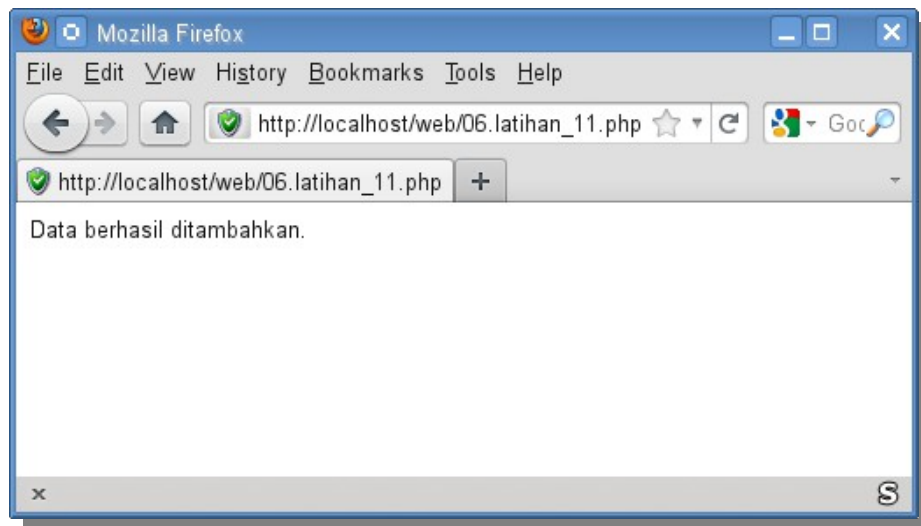
$sql = "INSERT INTO biodata(nama,alamat) ";
$sql.= "VALUES ('".$_POST['txtNama']. "','".$_POST['txtAlamat']. "')";

mysql_query($sql) or exit("Error query : <b>".$_sql."</b>.");

echo "Data berhasil ditambahkan.";

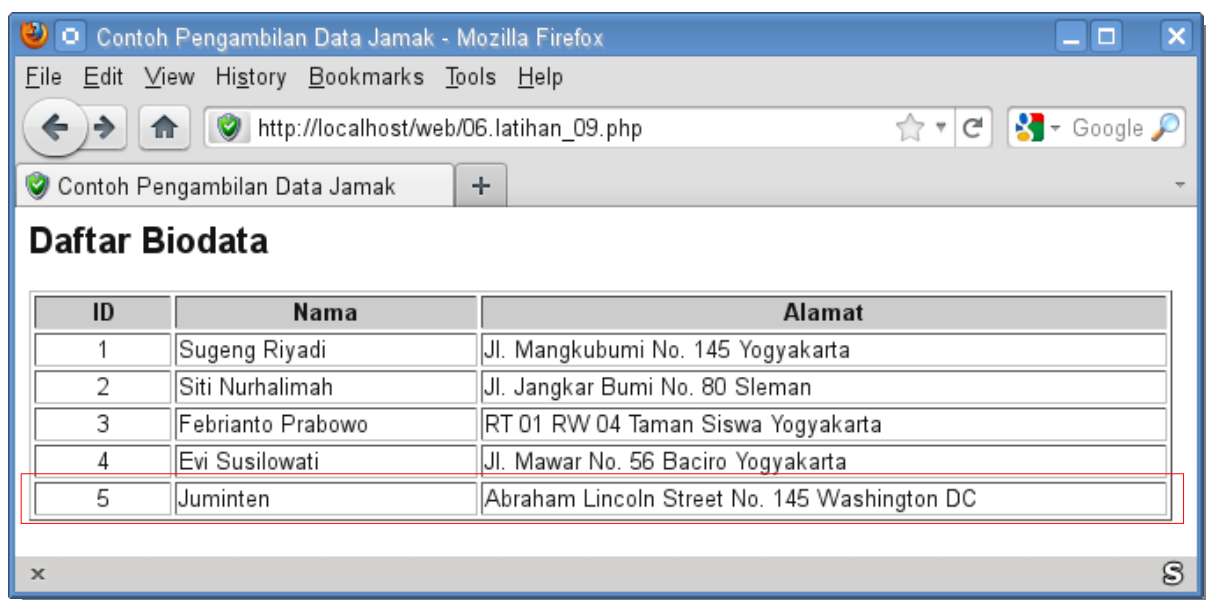
?>
```





Kedua contoh latihan program diatas dapat dijelaskan sebagai berikut :

1. Skrip `06.latihan_10.html` menyediakan form dengan elemen-elemen yang dijadikan sebagai media input data.
2. Setelah menekan tombol submit, data akan dikirimkan ke file `06.latihan_11.php` menggunakan metode POST.
3. Skrip `06.latihan_11.php` melakukan query INSERT data melalui fungsi `mysql_query()`. Jika query berhasil dijalankan, maka program akan menampilkan pesan **Data berhasil ditambahkan**. Untuk melihat efek dari dijelankannya program ini, jalankan kembali program `06.latihan_09.php` melalui browser.



## 2.4 Update Data

Untuk melakukan update terhadap suatu data, terlebih dahulu data tersebut perlu ditampilkan dalam sebuah form, agar pengguna dapat mengerti data mana saja yang ingin diubahnya.

06.latihan\_12.php

```
<html>
  <head>
    <title>Ubah Data</title>
  </head>
  <body>
    <?php

      include("koneksi.php");

      $sql = "SELECT id,nama,alamat FROM biodata ";
      $sql.= "WHERE id = 1";

      $hasil = mysql_query($sql) or exit("Error query : <b>".$sql."</b>.");
      $data = mysql_fetch_assoc($hasil);

    ?>
    <h2>Ubah Data</h2>
    <form action="06.latihan_13.php" method="post">
      Nama :&nbsp;
      <input type="text" name="txtNama" value="<?php echo $data['nama'];
      ?>" />
      <br />
      Alamat :
      <textarea name="txtAlamat"><?php echo $data['alamat']; ?></textarea>
      <br />
      <input type="hidden" name="txtID" value="<?php echo $data['id'];
      ?>" />
      <input type="submit" value="Submit" />
    </form>
  </body>
</html>
```

06.latihan\_13.php

```
<?php

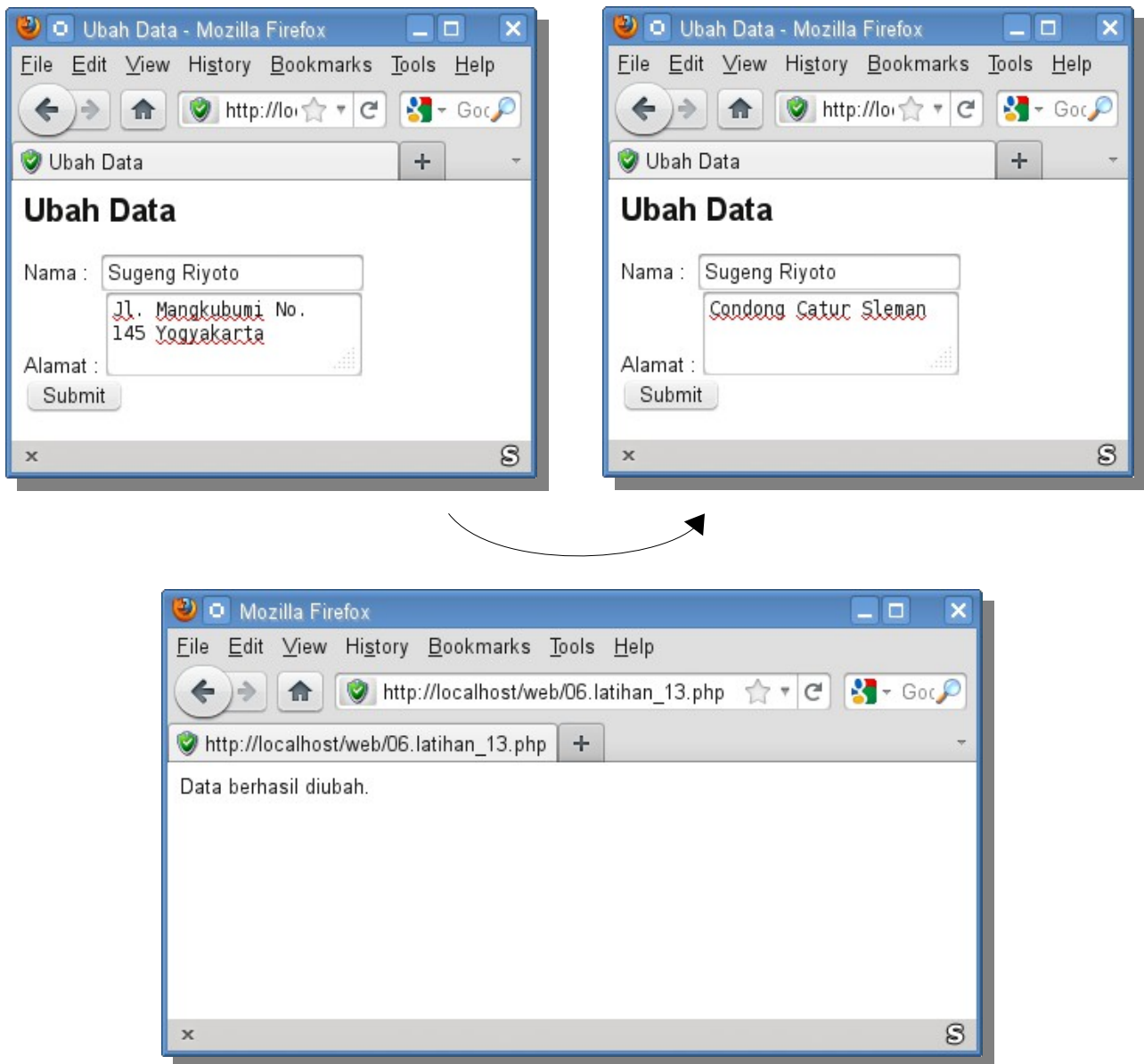
include("koneksi.php");

$sql = "UPDATE biodata SET ";
$sql.= "nama='".$$_POST['txtNama'].','alamat='".$$_POST['txtAlamat']. "' ";
$sql.= "WHERE id = '".$_POST['txtID'];

mysql_query($sql) or exit("Error query : <b>".$sql."</b>.");

echo "Data berhasil diubah.";

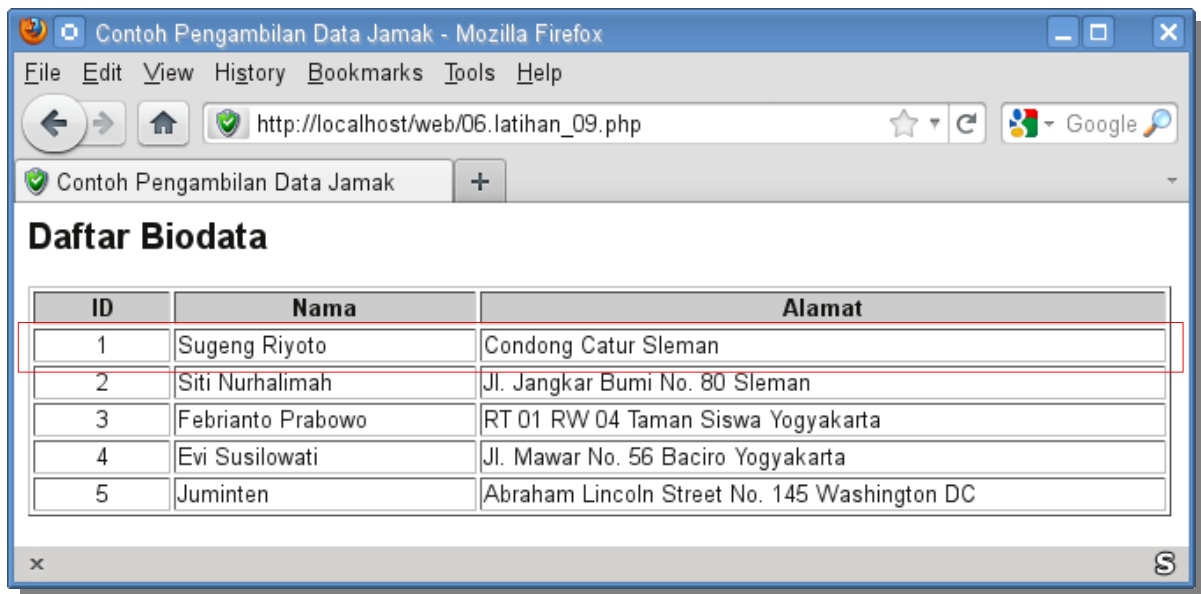
?>
```



Kedua skrip latihan program diatas dapat dijelaskan sebagai berikut :

1. Program 06.latihan\_12.php bertugas menyediakan elemen input yang digunakan untuk melakukan pengubahan pada data. Sebelum data diubah, terlebih dahulu data asli diambil dari basis data dan hasilnya akan ditampilkan / diisikan ke elemen input yang telah disediakan.
2. Pada program 6.latihan\_12.php terdapat satu elemen input dengan tipe *hidden*. Jika dijalankan, data yang terdapat pada elemen ini tidak akan terlihat secara langsung pada browser, kecuali menggunakan mode *view source* pada browser. Elemen ini digunakan untuk menampung data ID, yang mana data ID ini menjadi kondisi prasyarat pengubahan data pada query UPDATE.

3. Setelah data diubah dan tombol submit ditekan, data akan dikirimkan ke file 06.latihan\_13.php menggunakan metode POST untuk diolah lebih lanjut.
4. Skrip 06.latihan\_13.php melakukan query UPDATE data melalui fungsi `mysql_query()`. Jika query berhasil dijalankan, maka program akan menampilkan pesan **Data berhasil diubah**. Untuk melihat efek dari dijalkannya program ini, jalankan kembali program 06.latihan\_09.php melalui browser.



## 2.5 Menghapus Data

06.latihan\_14.php

```
<?php

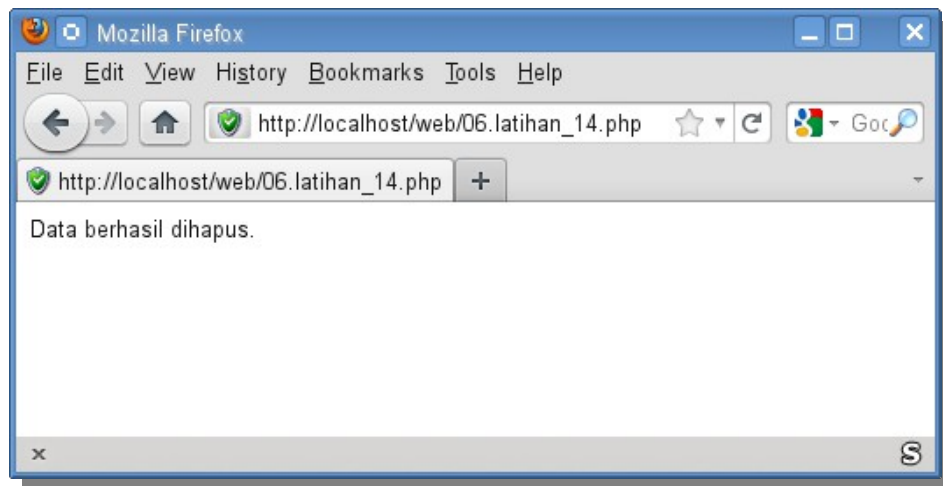
include("koneksi.php");

$sql = "DELETE FROM biodata WHERE id = 1";

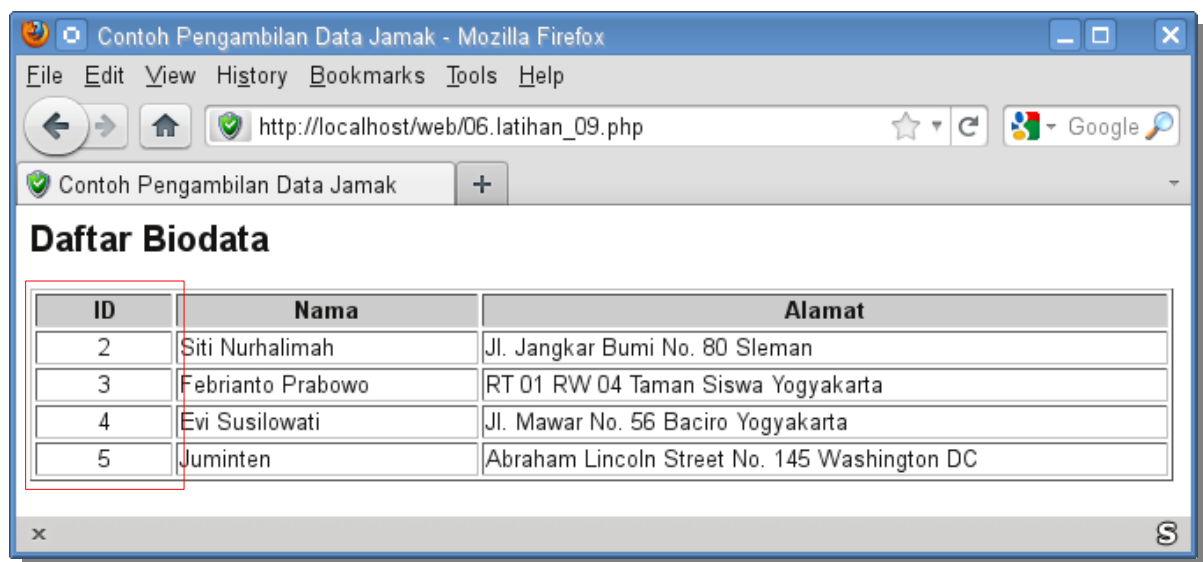
mysql_query($sql) or exit("Error query : <b>".$sql."</b>.");

echo "Data berhasil dihapus.";

?>
```



Program tersebut diatas bertugas untuk melakukan penghapusan data tertentu dari basis data. Data yang akan dihapus dari basis data adalah data dengan ID = 1. Untuk melihat efek dari dijalkannya program ini, jalankan kembali program 06.latihan\_09.php melalui browser.



Dari screenshot diatas, dapat kita lihat bahwa data yang memiliki ID = 1 sudah tidak nampak dalam daftar biodata. Ini karena data tersebut telah dihapus melalui program 06.latihan\_14.php.



### 3. Integrasi Antar Program

Agar program anda berjalan dinamis (dapat memanipulasi data manapun) dan saling terintegrasi satu sama lain, ubah skrip latihan program-program diatas menjadi seperti dibawah ini:

06.latihan\_09.php

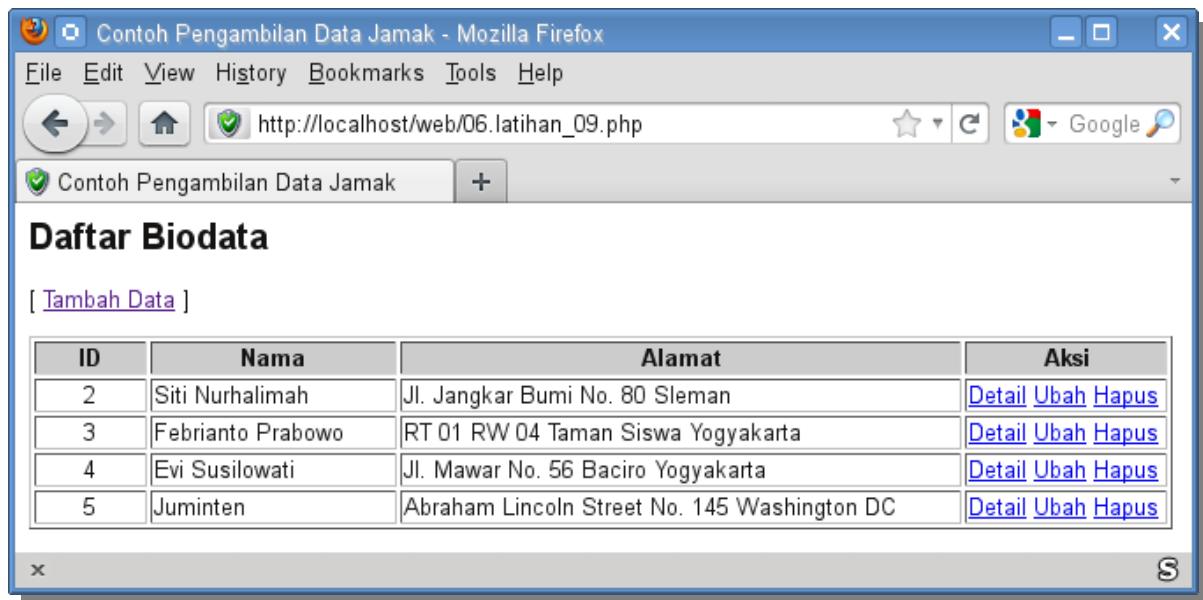
```
<html>
  <head>
    <title>Contoh Pengambilan Data Jamak</title>
  </head>
  <body>
    <h2>Daftar Biodata</h2>
    <p>[ <a href="06.latihan_10.php">Tambah Data</a> ] </p>
    <table width="650px" border="1">
      <tr style="background:#ccc">
        <th width="10%">ID</th>
        <th width="22%">Nama</th>
        <th width="50%">Alamat</th>
        <th>Aksi</th>
      </tr>
      <?php

        include("koneksi.php");

        $sql = "SELECT id,nama,alamat FROM biodata";
        $hasil = mysql_query($sql) or exit("Error query: <b>".$sql."</b>.");

        while($data = mysql_fetch_assoc($hasil)){
          ?>
          <tr>
            <td align="center"><?php echo $data['id']; ?></td>
            <td><?php echo $data['nama']; ?></td>
            <td><?php echo $data['alamat']; ?></td>
            <td>
              <a href="06.latihan_06.php?id=<?php echo $data['id']; ?>">
                Detail
              </a>
              <a href="06.latihan_13.php?id=<?php echo $data['id']; ?>">
                Ubah
              </a>
              <a href="06.latihan_15.php?id=<?php echo $data['id']; ?>">
                Hapus
              </a>
            </td>
          </tr>
        </tr>
      <?php
    }

    ?>
  </table>
</body>
</html>
```



Sekarang kita dapat dengan mudah melakukan manipulasi terhadap data manapun yang kita inginkan. Karena program `06.latihan_09.php` telah dilengkapi dengan menu “Tambah, Detail, Ubah dan Hapus Data” secara dinamis dan terintegrasi.

#### 06.latihan\_06.php

```
<?php
include("koneksi.php");

$sql = "SELECT * FROM biodata WHERE id = ".$_GET['id'];
$hasil = mysql_query($sql);
$data = mysql_fetch_assoc($hasil);

echo "ID : ".$data['id'];
echo "<br />";
echo "Nama : ".$data['nama'];
echo "<br />";
echo "Alamat : ".$data['alamat'];

?>
```

## 06.latihan\_12.php

```
<html>
  <head>
    <title>Ubah Data</title>
  </head>
  <body>
    <?php

      include("koneksi.php");

      $sql = "SELECT id,nama,alamat FROM biodata ";
      $sql.= "WHERE id = ".$_GET['id'];

      $hasil = mysql_query($sql) or exit("Error query : <b>".$sql."</b>.");
      $data = mysql_fetch_assoc($hasil);

    ?>
    <h2>Ubah Data</h2>
    <form action="06.latihan_13.php" method="post">
      Nama : &nbsp;
      <input type="text" name="txtNama" value="<?php echo $data['nama'];
      ?>" />
      <br />
      Alamat :
      <textarea name="txtAlamat"><?php echo $data['alamat']; ?></textarea>
      <br />
      <input type="hidden" name="txtID" value="<?php echo $data['id'];
      ?>" />
      <input type="submit" value="Submit" />
    </form>
  </body>
</html>
```

## 06.latihan\_14.php

```
<?php

include("koneksi.php");

$sql = "DELETE FROM biodata WHERE id = ".$_GET['id'];

mysql_query($sql) or exit("Error query : <b>".$sql."</b>.");

echo "Data berhasil dihapus.";

?>
```

Jika kita amati pada skrip [06.latihan\\_06.php](#), [06.latihan\\_12.php](#) dan [06.latihan\\_14.php](#), data yang akan dimanipulasi / dioperasikan pada query berasal dari variabel GET, yaitu `$_GET['id']`. Ini akan membantu kita untuk mengubah data manapun yang kita inginkan, karena variabel `$_GET['id']` bersifat dinamis yang mengacu pada data yang akan diubah.

## DAFTAR PUSTAKA

1. Buzton, Toby. 2002. *PHP By Example*. Indianapolis, Indiana: Que.
2. Muhardin, Endy. 2003. *PHP Programming Fundamental dan MySQL Fundamental*. IlmuKomputer.Com
3. Rasmus, Lerdorf., Dkk. 2003. *PHP 5 Manual*. PHP Documentation Group.