

BUKU AJAR & PANDUAN PRAKTIKUM



# PEMROGRAMAN WEB

M. Miftakul Amin, S. Kom., M. Eng.



**Jurusan Teknik Komputer  
Program D3 Teknik Komputer  
Politeknik Negeri Sriwijaya Palembang**

## DAFTAR ISI

<b>KATA PENGANTAR.....</b>	<b>ii</b>
<b>DAFTAR ISI.....</b>	<b>iii</b>
<b>BAB 1 Pengenalan PHP .....</b>	<b>1</b>
1. 1 Apa Itu PHP .....	1
1. 2 Apa yang Bisa Dilakukan PHP .....	1
1. 3 Modal Dasar Mempelajari PHP .....	2
1. 4 Aplikasi Yang Diperlukan .....	2
1. 5 Pengenalan Web Server .....	2
1. 6 Server Side Scripting .....	4
1. 7 Membuat Direktory Kerja.....	4
<b>BAB 2 Syntax Penulisan .....</b>	<b>7</b>
2. 1 Memulai Script PHP .....	7
2. 2 Menyimpan File PHP .....	8
2. 3 Tanda Semicolon.....	9
2. 4 Pindah Spasi .....	10
2. 5 Variabel .....	10
2. 6 Aturan Penamaan Variabel .....	11
2. 7 Perintah echo.....	11
2. 8 Menampilkan Nilai Variabel dengan echo .....	13
2. 9 Komentar dalam PHP.....	14

<b>BAB 3 Menggunakan Operator .....</b>	<b>16</b>
3. 1 Operator Aritmatika .....	16
3. 2 Operator Perbandingan .....	17
3. 3 Kombinasi Operator Artimatika dan Assignmen	18
3. 4 Operator Increment/Decrement.....	19
<b>BAB 4 Modularitas.....</b>	<b>20</b>
4. 1 Fungsi Include.....	21
4. 2 Fungsi Require .....	22
<b>BAB 5 Array .....</b>	<b>23</b>
5. 1 Array Berindeks Numerik .....	23
5. 2 Array Assosiatif .....	24
<b>BAB 6 Statement Control .....</b>	<b>26</b>
6. 1 Statement If .....	26
6. 2 Statement Switch.....	31
6. 3 Statement While .....	34
6. 4 Statement For .....	35
6. 5 Statement Foreach.....	37
6. 6 Statemetn Do...while .....	38
<b>BAB 7 Bekerja dengan Form .....</b>	<b>41</b>
7. 1 Objek Form .....	41
7. 2 Method Get dan Post.....	42
<b>BAB 8 Function .....</b>	<b>45</b>
8. 1 Pengertian Fungsi.....	45
8. 2 Fungsi dengan Parameter .....	47
8. 3 Pengembalian Nilai .....	49

<b>BAB 9 PHP dan MySQL .....</b>	<b>51</b>
9. 1 Koneksi PHP dan MySQL .....	51
9. 2 Memilih Database .....	52
9. 3 Memberikan Query Ke MySQL .....	53
9. 4 Mengambil Data dari MySQL .....	54
9. 5	
<b>BAB 10 Project PHP dan MySQL .....</b>	<b>57</b>
10. 1 Membuat Database dengan PHPMYADMIN .....	57
10. 2 Aplikasi Tampil Data .....	65
10. 3 Aplikasi Insert Data.....	67
10. 4 Aplikasi Edit Data.....	73
10. 5 Aplikasi Delete.....	79

---

---

# Bab 1

## Pengenalan PHP

### 1.1 Apa Itu PHP?

Berdasarkan informasi dari situs resmi PHP, “PHP.net”, PHP (PHP: Hypertext Preprocessor) merupakan bahasa pemrograman web yang dapat disisipkan dalam script HTML. Banyak sintaks di dalamnya yang mirip dengan bahasa C, Java dan Perl. Tujuan dari bahasa ini adalah membantu para pengembang web untuk membuat web dinamis dengan cepat.

Ketika seseorang mengunjungi web berbasis PHP, web server akan memproses kode-kode PHP. Beberapa perintah atau kode dari PHP tersebut selanjutnya ada yang diterjemahkan ke dalam HTML dan beberapa ada yang disembunyikan (misalnya proses kalkulasi dan operasi). Setelah diterjemahkan ke dalam HTML, web server akan mengirim kembali ke web browser pengunjung tersebut.

### 1.2 Apa yang Bisa Dilakukan dengan PHP?

- Mengurangi waktu untuk membuat web berskala besar
- Mampu menciptakan web interaktif
- Menciptakan berbagai tool untuk keperluan online
- Mendukung e-commerce (shopping carts)

### 1.3 Modal Dasar Mempelajari PHP

Sebelum mempelajari PHP, Anda harus menguasai

- HTML,
- Dasar-dasar pemrograman (C/C++ lebih baik)

### 1.4 Aplikasi Yang Diperlukan

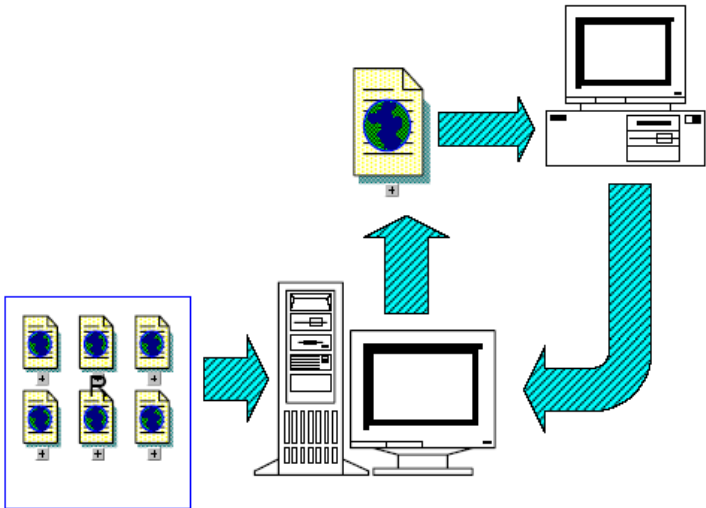
Untuk dapat bekerja dengan PHP, berikut ini adalah beberapa aplikasi yang diperlukan:

- Web server (Apache, IIS, Personal Web Server/PWS)
- PHP server (dapat didownload di PHP.net)
- Database server (MySQL, Interbase, MS SQL, dll)
- Web Editor (Dreamweaver, Frontpage, dll)

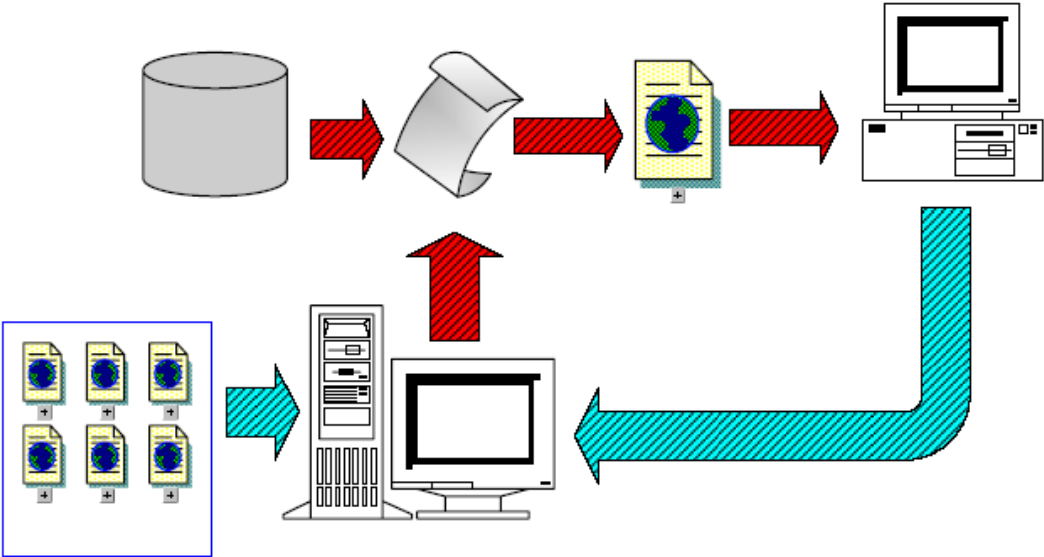
Anda dapat pula menggunakan tool aplikasi yang di dalamnya sudah terdapat web server (Apache), PHP server, dan MySQL yang terintegrasi menjadi satu. Tool tersebut dapat diinstal di PC sebagai sarana belajar PHP. Beberapa contoh tool tersebut diantaranya adalah Easyphp (Easyphp.org), PHPTriad, AppServe, dll. PHP server dapat berjalan dengan baik di beberapa OS seperti Windows, Linux, dan Macintosh.

### 1.5 Pengenalan Web Server

Web Server merupakan sebuah perangkat lunak dalam server yang berfungsi menerima permintaan (request) berupa halaman web melalui HTTP atau HTTPS dari client yang dikenal dengan browser web dan mengirimkan kembali (response) hasilnya dalam bentuk halaman-halaman web yang umumnya berbentuk HTML.



Gambar 1.1 Standar Arsitektur Web



Gambar 1.2 Arsitektur Web Dinamis

## 1.6 Server Side Scripting

Server side scripting merupakan sebuah teknologi scripting atau pemrograman web, dimana script atau program dikompilasi atau diterjemahkan di server dan hasilnya dikirimkan di browser.

Beberapa contoh server side scripting:

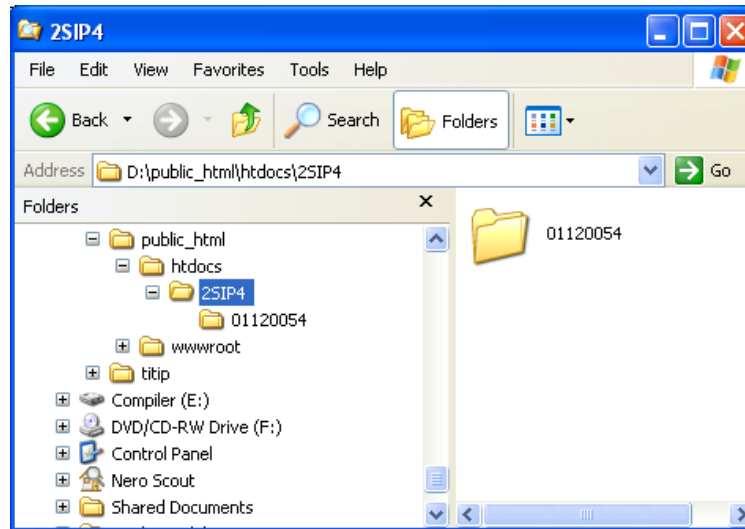
- Active Server Pages (ASP) dan ASP.NET
- Cold Fusion
- Java Server Pages
- Perl
- Python
- PHP

## 1.7 Membuat Direktori Kerja

Selama praktikum dilaksanakan, tempatkan file-file kerja (file .php) anda di dalam directory yang berfungsi sebagai **document root** sehingga dikenali oleh web server. Ikuti langkah-langkah berikut untuk membuat directory kerja.

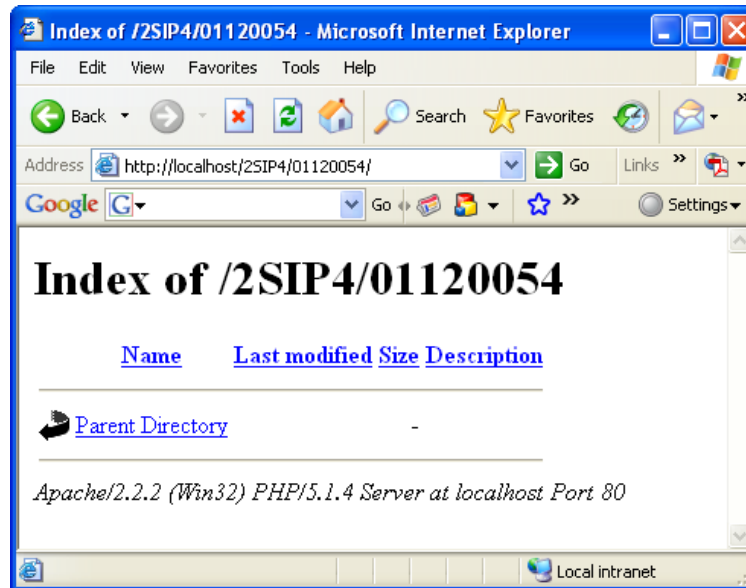
1. Buat folder di DRIVE **D:\public\_html\htdocs**, dengan nama kelas masing-masing dan dibawah folder kelas, buat folder NPM anda perhatikan gambar berikut.





Gambar 1.3 Membuat Direktori Kerja

2. Dari gambar di atas, misalkan anda ada di kelas 2SIP4 dan NPM anda 01120054. Selama praktikum dalam 1 semester usahakan semua file kerja disimpan di dalam folder yang telah anda buat.
3. Selanjutnya anda menguji apakah web server di dalam komputer anda sudah berjalan dengan baik. Buka web browser (internet explorer) kemudian ketikkan pada address bar <http://localhost/2SIP4/01120054>. Sehingga diperoleh tampilan seperti gambar berikut.



Gambar 1.4 Menguji Web Server

---

---

## BAB 2

### Syntax Penulisan PHP

#### 2.1 Memulai Script PHP

Untuk menuliskan script PHP anda dapat menuliskan script dengan notasi sebagai berikut ini:

```
<?php  
...  
...  
...  
?>
```

atau

```
<?  
...  
...  
...  
?>
```

Apabila Anda membuat kode PHP dan berencana akan mendistribusikan ke pihak/orang lain, maka usahakan untuk menggunakan sintaks **<?php ... ?>**. Hal ini dikarenakan untuk penggunaan kode yang menggunakan **<? ... ?>** terkadang tidak bisa dijalankan dalam server tertentu.

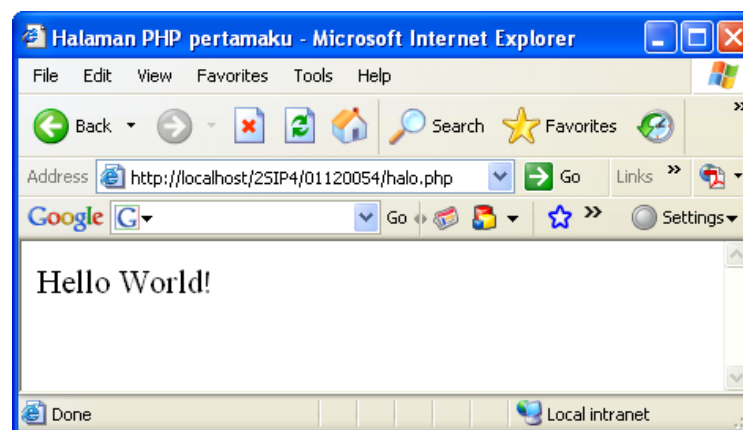
## 2.2 Menyimpan File PHP

Apabila Anda memiliki kode PHP yang disisipkan dalam HTML dalam suatu file dan menginginkan web server dapat menjalankannya, maka file tersebut harus disimpan dalam ekstensi **.php**. Apabila Anda menyimpannya dengan ekstensi **.html** atau **.htm**, maka kode PHP tersebut tidak akan diproses dan akan ditampilkan dalam web browser seperti apa adanya (berupa kode-kode).

Contoh:

```
halo.php
<html>
<head>
<title>Halaman PHP pertamaku</title>
</head>
<body>
<?php
echo "Hello World!";
?>
</body>
</html>
```

Hasil dari script di atas diperoleh tampilan seperti gambar berikut.



**Gambar 2.1** Hasil Eksekusi Script halo.php

## 2.3 Tanda Semicolon (;)

Apabila Anda perhatikan contoh sebelumnya, maka terdapat tanda titik koma (semicolon) pada akhir perintah echo. Tanda semicolon merupakan penanda akhir dari statement PHP dan harus ada di setiap akhir baris.

Contoh:

```
halo2.php
<html>
<head>
<title>Halaman PHP pertamaku</title>
</head>
<body>
<?php
echo "Hello World! ";
echo "Hello World! ";
echo "Hello World! ";
echo "Hello World! ";
echo "Hello World! ";
?>
</body>
</html>
```

Kemudian jika script di atas dieksekusi maka akan diperoleh tampilan seperti gambar berikut.



Gambar 2.2 Hasil Eksekusi Script halo2.php

## 2.4 Pindah Spasi

Seperti halnya HTML, pergantian spasi dalam PHP tidak akan mempengaruhi tampilan hasilnya. Dengan kata lain, pergantian spasi akan diabaikan oleh PHP.

## 2.5 Variabel

Misalkan dalam PHP kita akan menyimpan suatu nilai berupa angka atau string dalam suatu variabel, caranya adalah membuat nama variabel terlebih dahulu kemudian diberikan suatu assignment pada nilai yang diinginkan. Perhatikan sintaks berikut ini

```
$nama_variabel = nilai;
```

Contoh:

```
Variabel1.php
<?php
$hello = "Hello World!";
$sebuah_bilangan = 4;
$bilanganYangLain = 8;
?>
```

Dari contoh di atas tampak bahwa dalam PHP, nama variabel tidak perlu dideklarasikan terlebih dahulu seperti halnya bahasa Pascal atau C/C++.

## 2.6 Aturan Penamaan Variabel

Berikut ini adalah beberapa aturan penulisan nama variabel:

- Nama variabel harus diawali dengan huruf atau underscore (`_`)
- Nama variabel hanya boleh dituliskan dengan alpha numeric a-z, A-Z, 0-9 dan underscore
- Nama variabel yang terdiri lebih dari satu kata, dapat dipisahkan dengan underscore

## 2.7 Perintah echo

Seperti yang Anda lihat pada contoh-contoh kode PHP sebelumnya, bahwa perintah echo digunakan untuk menampilkan teks ke dalam browser. Suatu teks atau string dapat dituliskan di browser dengan langsung dituliskan dalam echo yang diapit oleh dua tanda petik ganda (quotes) atau menyimpan string atau teks terlebih dahulu dalam suatu variabel kemudian dituliskan dalam echo. Berikut ini adalah contohnya:

Contoh:

```
love.php
<?php
$stringKu = "Hello!";
echo $stringKu;
echo "<h5>I love using PHP!</h5>";
?>
```

Hati-hati dalam penulisan suatu string yang di dalamnya terdapat tanda petik ganda (quotes) menggunakan echo. Dalam echo, tanda quotes merupakan penanda awal dan

akhir teks/string yang akan ditulis dengan echo, sehingga Anda harus memperhatikan hal-hal berikut ini

- Jangan menggunakan tanda quotes di dalam teks yang akan ditulis dengan echo.
- Apabila Anda tetap ingin menuliskan tanda quotes dalam teks yang akan ditulis dengan echo, maka berikan tanda slash “\” di depan quotes tersebut.
- Selain itu, dapat pula Anda gunakan tanda petik tunggal (apostrophes) untuk menggantikan tanda quotes pada teks.

Contoh:

**echoquotes.php**

```
<?php
echo "<font face="verdana" size="4">I love
using PHP!</font>";
?>

<?php
echo "<font face=\"verdana\" size=\"4\">I
love using PHP!</font>";
?>

<?php
echo "<font face='verdana' size='4'>I love
using PHP!</h5>";
?>
```

Pada contoh kode pertama di atas akan terjadi error karena dalam teks yang ditulis dalam echo terdapat tanda quotes. Sedangkan untuk kode kedua dan ketiga tidak terdapat error dan akan dihasilkan output yang sama di browser.



## 2.8 Menampilkan Nilai Variabel dengan Echo

Nilai variabel dapat dengan mudah ditampilkan dengan menggunakan echo, baik nilai yang berupa bilangan maupun string. Berikut ini adalah contoh-contoh di antaranya:

Contoh:

<b>variabel.php</b>
<pre>&lt;?php \$string_ku = "Hello.. Nama saya: "; \$bilangan_ku = 4; \$huruf_ku = "a"; echo \$string_ku; echo \$bilangan_ku; echo \$huruf_ku; ?&gt;</pre>

Catatan:

Untuk menampilkan nilai variabel dengan echo tanpa menggunakan tanda quotes. Berikut ini contoh menampilkan gabungan suatu nilai dari variabel yang berupa string.

Contoh:

<b>dot.php</b>
<pre>&lt;?php \$string_ku = "Hello. Nama saya: "; \$baris_baru = "&lt;br&gt;"; echo \$string_ku."Ari".\$baris_baru; echo "Hi, Nama saya Ari. Kamu siapa? ".\$string_ku.\$baris_baru;</pre>

```
echo "Hi, Nama saya Ari. Kamu siapa?  
".$string_ku."Amalia";  
?>
```

Untuk menggabungkan beberapa string menjadi satu digunakan operator dot (.)

## 2.9 Komentar dalam PHP

Seperti halnya bahasa pemrograman yang lain, komentar dalam suatu kode PHP tidak akan dieksekusi. Terdapat dua cara memberikan komentar dalam PHP, yaitu:

- Diberikan tanda // di depan teks komentar. Perintah ini hanya bisa berlaku untuk komentar dalam satu baris.
- Diberikan tanda /\* di depan teks komentar dan diakhiri dengan \*/. Perintah ini dapat digunakan untuk komentar yang terdiri lebih dari satu baris.

Contoh:

```
komentar.php  
  
<?php  
echo "Hello World!"; // Ini akan mencetak  
Hello World!  
echo "<br>Psst...You can't see my PHP  
comments!"; // echo "nothing";  
// echo "Namaku Faza!";  
?>  
  
<?php  
/* Berikut ini adalah perintah
```

```
untuk menuliskan Hello World */
echo "Hello World!";
/* echo "My name is Humperdinkle!";
echo "No way! My name is Ari-PHP
Programmer!";
*/
?>
```

## Bab 3

# Menggunakan Operator

Dalam bahasa pemrograman secara umum, operator digunakan untuk memanipulasi atau melakukan proses perhitungan pada suatu nilai. Sampai saat ini, anda telah mengenal operator “.” (menggabungkan string) dan “=” (proses assignment). Selain dua operator itu masih banyak jenis operator yang lain dalam PHP yaitu:

- Operator aritmatik
- Operator perbandingan
- Gabungan operator aritmatik dan assignment

### 3.1 Operator Aritmatik

Berikut ini adalah tabel operator aritmatik, makna dan contohnya:

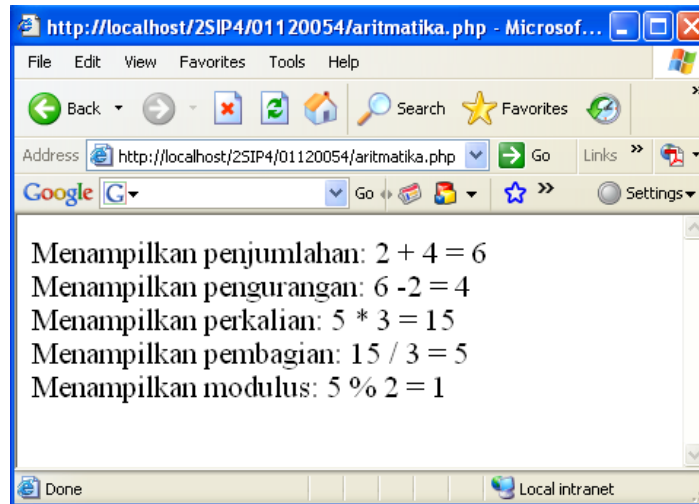
Operator	Makna	Contoh
+	Penjumlahan	2 + 4
-	Pengurangan	6 - 2
*	Perkalian	5 * 3
/	Pembagian	15 / 3
%	Modulus	43 % 10

Contoh:

```
aritmatika.php  
<?php  
$penjumlahan = 2 + 4;  
$pengurangan = 6 - 2;  
$perkalian = 5 * 3;  
$pembagian = 15 / 3;  
$modulus = 5 % 2;  
echo "Menampilkan penjumlahan: 2 + 4 = ".$penjumlahan."<br>";  
echo "Menampilkan pengurangan: 6 - 2 = ".$pengurangan."<br>";  
echo "Menampilkan perkalian: 5 * 3 = ".$perkalian."<br>";
```

```
echo "Menampilkan pembagian: 15 / 3 = ".$pembagian."<br>";  
echo "Menampilkan modulus: 5 % 2 = " . $modulus."<br>";  
?>
```

Selanjutnya jalankan script di atas, sehingga diperoleh tampilan seperti diperlihatkan pada gambar berikut.



Gambar 3.1 Eksekusi Script aritmatika.php

## 3. 2 Operator Perbandingan

Perbandingan digunakan untuk menguji hubungan antara nilai dan atau variabel. Operator ini digunakan dalam suatu statement bersyarat yang selalu menghasilkan nilai TRUE atau FALSE.

Misalkan:

```
$x = 4; $y = 5;
```

berikut ini adalah beberapa contoh penggunaan operator perbandingan dan hasilnya.

Operator	Makna	Contoh	Hasil
==	Sama dengan	<code>\$x == \$y</code>	FALSE
!=	Tidak sama dengan	<code>\$x != \$y</code>	TRUE
<	Lebih kecil dari	<code>\$x &lt; \$y</code>	TRUE
>	Lebih besar dari	<code>\$x &gt; \$y</code>	FALSE
<=	Lebih kecil atau sama dengan dari	<code>\$x &lt;= \$y</code>	TRUE
>=	Lebih besar atau sama dengan dari	<code>\$x &gt;= \$y</code>	FALSE

### 3.3 Kombinasi Operator Aritmatik dan Assignment

Dalam pemrograman seringkali dijumpai proses yang melibatkan proses increment. Misalkan kita menginginkan proses increment dengan tingkat kenaikan 1, maka perintah yang dituliskan dapat berupa

```
$counter = $counter + 1;
```

dalam PHP, perintah di atas dapat ditulis dalam satu perintah singkat sebagai

```
$counter += 1;
```

Dari contoh di atas tampak bahwa operator yang digunakan (`+=`) merupakan gabungan dari operator aritmatik dan assignment. Berikut ini adalah bentuk-bentuk operator lain jenis ini.

Operator	Contoh	Operasi yang ekuivalen
<code>+=</code>	<code>\$x += 2;</code>	<code>\$x = \$x + 2;</code>
<code>-=</code>	<code>\$x -= 4;</code>	<code>\$x = \$x - 4;</code>
<code>*=</code>	<code>\$x *= 3;</code>	<code>\$x = \$x * 3;</code>
<code>/=</code>	<code>\$x /= 2;</code>	<code>\$x = \$x / 2;</code>
<code>%=</code>	<code>\$x %= 5;</code>	<code>\$x = \$x % 5;</code>
<code>.=</code>	<code>\$my_str .= "hello";</code>	<code>\$my_str = \$my_str . "hello";</code>

### 3. 4 Operator Pre/Post Increment dan Decrement

Operator jenis ini merupakan pengembangan dari operator jenis sebelumnya. Operator ini hanya digunakan pada proses increment maupun decrement dengan tingkat 1.

Berikut ini adalah operator yang termasuk jenis ini:

- $\$x++$ ; ekuivalen dengan  $\$x += 1$ ; atau  $\$x = \$x + 1$ ;
- $\$x--$ ; ekuivalen dengan  $\$x -= 1$ ; atau  $\$x = \$x - 1$ ; Contoh:

<b>increment.php</b>
<pre>&lt;?php \$x = 4; \$x++; echo "\$x"; \$x = 4; \$x--; echo "\$x"; ?&gt;</pre>

---

---

## BAB 4

# MODULARITAS

Suatu pemrograman yang baik seharusnya program yang besar dipecah menjadi program-program yang kecil yang selanjutnya disebut modul. Modul-modul kecil tersebut dapat dipanggil sewaktu-waktu diperlukan. Dalam PHP juga mendukung konsep tersebut yang selanjutnya diberi nama modularitas. Kita dapat menyisipkan isi suatu file/modul lain ke dalam file/modul tertentu.

Terdapat 2 perintah/function untuk hal tersebut dalam PHP yaitu menggunakan include dan require.

### 4.1 Fungsi Include()

Untuk memudahkan pemahaman, diberikan contoh berikut. Misalkan kita akan membuat menu link sejumlah 4 buah yaitu index, about, links, dan contact pada setiap halaman web yang kita buat. Teknik yang digunakan adalah membuat menu link dalam suatu modul tersendiri kemudian modul tersebut dipanggil pada setiap halaman web yang diinginkan terdapat menu link di dalamnya.

<b>menu.php</b>
-----------------

<pre>&lt;html&gt; &lt;body&gt; &lt;a href="index.php"&gt;Home&lt;/a&gt; - &lt;a href="about.php"&gt;About Us&lt;/a&gt; - &lt;a href="links.php"&gt;Links&lt;/a&gt; - &lt;a href="contact.php"&gt;Contact Us&lt;/a&gt; &lt;br&gt;</pre>
--



---

---

**index.php**

```
<?php
include( "menu.php" );
?>

<p>Ini adalah halaman index</p>
</body>
</html>
```

**about.php**

```
<?php
include( "menu.php" );
?>

<p>Ini adalah halaman about</p>
</body>
</html>
```

Dari teknik di atas tampak adanya kemudahan dalam membuat halaman web. Dalam hal ini, kita tidak perlu membuat menu link di setiap halaman web yang ada. Bayangkan seandainya kita punya halaman web sejumlah 100 buah yang kesemuanya ingin diberi menu link tanpa menggunakan teknik di atas, tentu hal tersebut sangat merepotkan.

Meskipun secara teknis, kode pembangun web dipecah dalam modul-modul, namun ketika di browser akan terlihat menyatu. Berikut ini adalah kode HTML yang dihasilkan oleh browser ketika membuka halaman web `index.php`

```
<html>
<body>
<a href="index.php">Home</a> -
<a href="about.php">About Us</a> -
<a href="links.php">Links</a> -
<a href="contact.php">Contact Us</a> <br>
<p>Ini adalah halaman index</p>
</body>
</html>
```

## 4.2 Fungsi Require()

Cara penggunaan maupun fungsi dari `require()` sama dengan `include()`. Jadi apa perbedaannya? Sebaiknya mana yang kita gunakan? Perhatikan contoh berikut ini

```
<?php
include("noFileExistsHere.php");
echo "Hello World!";
?>
```

dengan asumsi bahwa file `noFileExistsHere.php` tidak ada.

## Bab 5

### Array

Dalam PHP, indeks untuk array dapat berupa numerik dan dapat pula berupa suatu nilai atau yang sering disebut dengan array asosiatif.

#### 5.1 Array Berindeks Numerik

Sintaks umum untuk menyatakan suatu array berindeks numerik beserta nilainya adalah.

```
$nama_array[$key] = value;
```

Dalam hal ini \$key berupa bilangan bulat mulai dari 0, 1, 2, ...

Contoh:

```
$karyawan[0] = "Bob";  
$karyawan[1] = "Sally";  
$karyawan[2] = "Charlie";  
$karyawan[3] = "Clare";
```

Sedangkan berikut ini adalah contoh untuk menampilkan nilai dari suatu array berindeks numerik.

**Array1.php**

```
<?
$karyawan[0] = "Bob";
$karyawan[1] = "Sally";
$karyawan[2] = "Charlie";
$karyawan[3] = "Clare";

echo "Berikut ini adalah 2 orang karyawan saya,
yaitu "
. $karyawan[0] . " & " . $karyawan[1];
echo "<br>Dua orang karyawan saya yang lain adalah "
. $karyawan[2] . " & " . $karyawan[3];
?>
```

## 5.2 Array Asosiatif

Untuk array asosiatif, sintaksnya sama dengan array berindeks numerik namun perbedaannya adalah pada \$key. Pada array asosiatif, \$key dapat berupa suatu string. Berikut ini adalah contohnya.

```
$gaji["Bob"] = 2000;
$gaji["Sally"] = 4000;
$gaji["Charlie"] = 600;
$gaji["Clare"] = 0;
```

dan berikut ini adalah contoh kode untuk menampilkan nilai dari array asosiatif

**Array2.php**

```
<?
$gaji["Bob"] = 2000;
$gaji["Sally"] = 4000;
$gaji["Charlie"] = 600;
$gaji["Clare"] = 0;

echo "Bob digaji -$" . $gaji["Bob"] . "<br>";
echo "Sally digaji -$" . $gaji["Sally"] . "<br>";
echo "Charlie digaji -$" . $gaji["Charlie"] .
"<br>";
echo "dan Clare digaji -$" . $gaji["Clare"];
?>
```

## Bab 6

### Statement Control

#### 6.1 Statement IF ...

Sintaks dari statement tersebut dalam PHP adalah

```
if (syarat){  
    statement1;  
    statement2;  
    .  
    .  
}
```

Untuk menyatakan syarat, biasanya digunakan operator perbandingan seperti yang telah dibahas sebelumnya. Apabila syarat bernilai TRUE maka statement-statement yang diapit dengan tanda kurung kurawal akan dijalankan. Bentuk lain dari sintaks IF adalah

```
if (syarat) {  
  
    statement1;  
    statement2;  
    .  
    .  
}
```

```
} else {  
  
    statement3;  
    statement4;  
    .  
    .  
}
```

Untuk sintaks kedua di atas, `statement3`, `statement4`, dst akan dijalankan apabila syarat bernilai `FALSE`.

Contoh:

```
<?  
$my_name = "nada";  
if ($my_name == "nada")  
{  
    echo "Your name is ".$my_name."<br>";  
}  
echo "Welcome to my homepage!";  
?>
```

Contoh:

```
<?  
$number = 3;  
if ($number == 4)  
{  
    echo "Benar";  
}
```

```
}  
else  
{  
  
echo "Salah";  
}  
?>
```

Terdapat pula bentuk sintaks berikutnya dari IF ... yaitu dengan ditambahkan elseif

```
if (syarat1)  
{  
    statement1;  
    statement2;  
    .  
    .  
}  
elseif (syarat2)  
{  
    statement21;  
    statement22;  
    .  
    .  
}  
else  
{  
    statement1;  
    statement2;
```



```
    .  
    .  
}
```

Jika syarat1 bernilai TRUE, maka statement11, statement12 dst akan dijalankan. Sedangkan jika syarat1 FALSE maka selanjutnya akan dicek untuk syarat2. Jika syarat2 TRUE maka statemen21, statement22, dst akan dijalankan, sedangkan jika syarat2 FALSE akan dicek syarat berikutnya (jika masih ada). Statement1, statement2, dst baru akan dijalankan apabila semua syarat sebelumnya bernilai FALSE.

Contoh:

```
<?  
$karyawan = "Bob";  
if($karyawan == "Tanner")  
{  
echo "Hello Tanner!";  
}  
elseif($karyawan == "Bob")  
{  
echo "Hello Bob!";  
}  
else  
{  
echo "Hello!";  
}
```

}

## 6.2 Statement SWITCH

Sintaks dari statement ini adalah

```
switch (variabel)

{
    case option1:
        statement11;
        statement12;
        .
        .
        break;
    case option2:
        statement21;
        statement22;
        .
        .
        break;
    .
    .

    default:
        statementdefault1;
        statementdefault2;
        .
        .
        break;
}
```

Pada sintaks di atas, nilai dari variabel akan dicek pada setiap option yang ada (terletak di bagian case). Jika ada option yang sama dengan nilai variabel, maka statement-statement di bawah option tersebutlah yang akan dijalankan. Bagian default adalah optional (boleh ada, boleh tidak).

Contoh:

```
<?php
$tujuan = "Tokyo";
echo "Biaya Perjalanan Menuju $tujuan adalah ";
switch ($tujuan){
case "Las Vegas":
echo " $500";
break;
case "Amsterdam":
echo " $1500";
break;
case "Egypt":
echo " $1750";
break;
case "Tokyo":
echo " $900";
break;
case "Caribbean Islands":
echo " $700";
break;
}
?>
```

Contoh:

```
<?php
$tujuan = "New York";
echo "Biaya Perjalanan Menuju $tujuan adalah ";
switch ($tujuan){
case "Las Vegas":
echo " $500";
break;
case "Amsterdam":
echo " $1500";
break;
case "Egypt":
echo " $1750";
break;
case "Tokyo":
echo " $900";
break;
case "Caribbean Islands":
echo " $700";
break;
default:
echo " $100";
break;
}
?>
```

## 6.3 Statement WHILE

Statement ini digunakan untuk mengerjakan suatu statement secara berulang-ulang sampai suatu syarat dipenuhi. Sintaksnya adalah

```
while (syarat){  
    statement;  
    statement;  
    .  
    .  
}
```

Pada sintaks di atas, selama syarat bernilai TRUE maka statement-statement di dalam while akan terus dijalankan secara berulang-ulang. Perulangan baru akan berhenti apabila syarat bernilai FALSE. Sebelum statement yang diulang-ulang dilakukan, terlebih dahulu akan dicek syarat nya apakah bernilai TRUE atau FALSE. Apabila TRUE maka statement akan dijalankan. Sedangkan apabila FALSE, perulangan akan langsung berhenti. Dengan kata lain, statement dalam WHILE bisa jadi tidak akan pernah dilakukan, yaitu apabila syaratnya langsung bernilai FALSE.

Contoh:

```
<?  
$harga_sikat = 1500;  
$jumlah_sikat = 10;  
  
echo "<table border=\"1\" align=\"center\">";  
echo "<tr><td><b>Jumlah Sikat</b></td>";  
echo "<td><b>Harga</b></tr></td>";
```

```
while ( $jumlah_sikat <= 100 )

{
echo "<tr><td>";
echo $jumlah_sikat;
echo "</td><td>";
echo "Rp. ".$harga_sikat * $jumlah_sikat;
echo "</td></tr>";
$jumlah_sikat = $jumlah_sikat + 10;
}
echo "</table>";
?>
```

Kode di atas akan menampilkan hasil di browser berupa tabel yang berisi jumlah sikat dan harganya, dengan asumsi harga sebuah sikat adalah Rp. 1.500. Jumlah sikat yang ditampilkan adalah kelipatan 10 dengan batas sampai 100 buah.

## 6.4 Statement FOR

Statement FOR mirip dengan WHILE yang memiliki sintaks berikut ini

```
for (inisialisasi counter; syarat; increment/decrement
counter)

{
statement;
.
.
}
```

Untuk memperjelas pemahaman tentang FOR, berikut ini adalah contoh kode dengan for untuk menghasilkan tampilan yang sama dengan contoh while sebelumnya (tentang jumlah sikat dan harganya). Coba bandingkan dengan kode contoh while sebelumnya.

Contoh:

```
<?
$harga_sikat = 1500;

echo "<table border=\"1\" align=\"center\">";
echo "<tr><td><b>Jumlah Sikat</b></td>";
echo "<td><b>Harga</b></td></tr>";

for ($jumlah_sikat = 10; $jumlah_sikat <= 100;
    $jumlah_sikat+=10)

{
echo "<tr><td>";
echo $jumlah_sikat;
echo "</td><td>";
echo "Rp. ".$harga_sikat * $jumlah_sikat;
echo "</td></tr>";

}

echo "</table>";
?>
```



## 6.5 Statement Foreach

Misalkan Anda punya data berupa array assosiatif yang akan diproses secara berulang-ulang, maka PHP menyediakan statement foreach yang mudah digunakan.

Sintaksnya adalah:

```
foreach(variabelarray as kunci => value)

{
statement;
.
.
}
```

Sebagai contoh, misalkan Anda memiliki 5 orang karyawan dengan usianya masing-masing yang ditulis dalam kode PHP sebagai berikut

```
$UsiaKaryawan["Lisa"] = "28";
$UsiaKaryawan["Jack"] = "16";
$UsiaKaryawan["Ryan"] = "35";
$UsiaKaryawan["Rachel"] = "46";
$UsiaKaryawan["Grace"] = "34";
```

Berikut ini adalah contoh kode PHP yang akan menampilkan semua karyawan beserta usianya dengan menggunakan foreach.

```
<?
$UsiaKaryawan["Lisa"] = "28";
$UsiaKaryawan["Jack"] = "16";
$UsiaKaryawan["Ryan"] = "35";
$UsiaKaryawan["Rachel"] = "46";
$UsiaKaryawan["Grace"] = "34";

foreach($UsiaKaryawan as $Nama => $umur)
{

echo "Nama Karyawan: $Nama, Usia: $umur." th <br>";
}
?>
```

## 6.6 Statement DO WHILE

Statement ini merupakan bentuk modifikasi dari WHILE. Sintaksnya adalah sebagai berikut

```
do

{
statement;
.
.

}
while (syarat);
```

Coba bandingkan dengan sintaks WHILE sebelumnya. Dilihat dari posisi statement yang diulang, posisi statement yang diulang pada DO WHILE terletak di atas syarat. Dengan demikian, sebelum syarat dicek TRUE atau FALSE nya, statement akan dikerjakan terlebih dahulu. Sedangkan pada WHILE, sebelum statement yang diulang dikerjakan, terlebih dahulu syarat akan dicek.

Prinsip kerja DO WHILE sama dengan WHILE yaitu statement akan terus dikerjakan selama syarat bernilai TRUE dan perulangan akan berhenti apabila FALSE.

Perhatikan contoh berikut ini yang membandingkan DO WHILE dengan WHILE

Contoh:

```
<?
$ kue = 0;
while($ kue > 1)
{

echo "Mmmmm...Aku suka kue! Nyam..nyam..nyam..";
}
?>

<?
$ kue = 0;
do
{
echo "Mmmmm... Aku suka kue! Nyam..nyam..nyam..";
} while ($ kue > 1);
?>
```

Pada kode WHILE, teks "Mmmm.... " dst tidak akan ditampilkan karena syaratnya langsung bernilai FALSE (perulangan berhenti). Sedangkan pada DO WHILE, teks akan ditampilkan dahulu kemudian perulangan berhenti (syarat bernilai FALSE).

---

---

## Bab 7

# Bekerja dengan Form

### 7.1 Objek Form

Setelah kita belajar dasar-dasar perintah PHP, sekarang saatnya mengaplikasikannya pada aplikasi web. Biasanya PHP digunakan sebagai pengolah data yang diinputkan melalui form yang dibuat dengan HTML. Sebagai contoh, andaikan Anda memiliki toko virtual dalam web yang menjual alat-alat tulis seperti pensil, buku tulis, dan ballpoint. Berikut ini adalah salah satu bentuk kode HTML yang digunakan untuk membuat form pemesanan pembelian barang-barang tersebut.

```
<html><body>
<h2>Toko Alat Tulis Amalia</h2>
<form action="proses.php" method="post">
<select name="barangpesanan">
<option>Pensil</option>
<option>Buku Tulis</option>
<option>Ballpoint</option>
</select>
Jumlah pesanan: <input name="jumlah" type="text">
<input type="submit" value="Submit">
</form>
</body></html>
```

Seperti yang Anda lihat pada kode HTML di atas, perintah `action="proses.php"` digunakan untuk mengarahkan ke file PHP yang digunakan untuk memproses barang pembelian ketika tombol submit ditekan. Dalam form yang dihasilkan dari kode di

atas terdapat 2 buah komponen input yaitu berbentuk combobox dan textbox. Untuk combobox, diberi nama “barangpesanan” (perhatikan perintah `<select name="barangpesanan">`) dan textbox diberi nama “jumlah” (perhatikan perintah `<input name="jumlah" type="text">`).

Penjelasan mengenai `method="post"` akan dijelaskan pada bab berikutnya. Sedangkan berikut ini adalah salah satu contoh kode PHP untuk memproses input dari form di atas. Kode PHP ini disimpan dengan nama `proses.php`.

```
<html><body>
<?php
$jumlah = $_POST['jumlah'];
$barangpesanan = $_POST['barangpesanan'];

echo "Anda memesan ". $jumlah . " " . $barangpesanan .
".<br>";
echo "Terima kasih atas kesediaan Anda memesan barang
dari kami!";
?>
</body></html>
```

Kalau Anda perhatikan, terdapat keterkaitan perintah `$_POST['xxx'];` dengan `"xxx"` pada `name = "xxx"` (nama komponen input).

Disusun oleh: Rosihan Ari Yuana 22

## 7.2 Method Get dan Post

Pada contoh sebelumnya, kita mengirim data input dari form menuju ke file PHP untuk diproses menggunakan metode `post`. Selain metode tersebut, terdapat pula

metode get. Lantas perbedaannya apa? Kapan kita gunakan metode post atau get? Itulah yang akan dibahas pada bab ini.

Untuk melihat perbedaan post dan get, kita akan sedikit mengubah file HTML form dan file PHP proses.php sebelumnya.

```
<html><body>
<h2>Toko Alat Tulis Amalia</h2>
<form action="proses.php" method="get">
<select name="barangpesanan">
<option>Pensil</option>
<option>Buku Tulis</option>
<option>Ballpoint</option>
</select>
Jumlah pesanan: <input name="jumlah" type="text">
<input type="submit" value="Submit">
</form>
</body></html>
```

dan isi proses.php nya adalah

```
<html><body>
<?php
$jumlah = $_GET['jumlah'];
$barangpesanan =
$_GET['barangpesanan'];
echo "Anda memesan ". $jumlah
. " " . $barangpesanan .
".<br>";
echo "Terima kasih atas
```

```
kesediaan Anda memesan barang  
dari kami!";  
?>  
</body></html>
```

Perbedaan kode HTML dan PHP di atas dengan sebelumnya adalah yang dicetak merah. Metode pengiriman data input dari form menggunakan get, dan dalam proses.php \$\_POST diganti dengan \$\_GET.

Apabila aplikasi di atas dijalankan, maka secara sekilas hasil yang tampak sama dengan ketika digunakan metode post. Namun, coba perhatikan URL yang tampak ketika proses.php muncul. Pada URL tersebut terdapat tambahan ?barangpesanan=...&jumlah=... setelah nama file (proses.php). Titik-titik tersebut akan diisi dengan data sesuai dengan yang diinputkan pada form.

Coba bandingkan dengan URL ketika digunakan metode post. Data isian pada form tidak ditampilkan pada URL. Sehingga inilah perbedaan antara keduanya.

Dengan demikian, hendaknya kita jangan menggunakan metode get ketika akan memproses data input melalui form. Bayangkan seandainya form tersebut digunakan untuk login atau untuk keperluan yang menyangkut privasi. Apabila Anda gunakan metode get, maka semua input data

akan ditampilkan pada URL. Bisa-bisa password Anda akan kelihatan di URL (jika terdapat input password ketika login). Untuk metode get, biasanya digunakan untuk input data melalui link (bukan melalui form). Untuk contoh aplikasinya dapat dilihat pada contoh-contoh aplikasi pada bab-bab berikutnya.



## Bab 8

# Function

### 8.1 Pengertian Fungsi

Sebuah function merupakan sebuah nama yang kita berikan pada suatu blok program yang sewaktu-waktu dapat kita panggil dan gunakan. Sebuah function dapat diletakkan di bagian manapun, bisa di awal, tengah, dan akhir dari keseluruhan bagian kode PHP.

Berikut ini adalah contoh membuat sebuah function sederhana yang di dalamnya tidak ada statementnya.

Contoh:

```
<?php

function myCompanyMotto()
{
}

myCompanyMotto();

?>
```

Pada contoh di atas, myCompanyMotto merupakan nama function. Nama function inilah yang dapat dipanggil sewaktu-waktu diperlukan. Aturan membuat nama function sama dengan ketika membuat nama variabel (lihat di Bab II).

Statement/perintah dari function dituliskan di dalam kurung kurawal {}. Sedangkan perintah myCompanyMotto(); bagian paling bawah dari kode di atas merupakan cara memanggil function. Perhatikan contoh berikutnya:

Contoh:

```
<?php
function myCompanyMotto()
{

echo "Sabar adalah bagian dari keimanan";
}
myCompanyMotto();
?>
```

Pada contoh tersebut, terdapat perintah echo di dalam function. Sehingga begitu nama function dipanggil, PHP akan menampilkan teks yang di-echo-kan tersebut. Sebuah function dapat dipanggil berulang-ulang, seperti pada contoh berikut.

```
<?php
function myCompanyMotto()
{

echo "Sabar adalah bagian dari keimanan";
}

echo "Selamat datang di PT. Nada Corp. <br>";
```

```
myCompanyMotto();

echo "Terima kasih atas kunjungan Anda<br>";
echo "dan ingatlah selalu... <br>";

myCompanyMotto();

?>
```

## 8.2 Fungsi dengan Parameter

Contoh function sebelumnya tidak menggunakan parameter. Peran parameter adalah sebagai input untuk function yang selanjutnya diolah oleh function tersebut. Berikut ini contoh penggunaan parameter pada function.

```
<?php
function UcapanSalam($nama)
{
echo "Hallo ". $nama . "!<br>";
}
?>
```

Pada contoh di atas, variabel \$nama merupakan parameter dari function. Nilai dari variabel tersebut akan ditambahkan pada string yang di-echo-kan.

Selanjutnya akan diberikan contoh penggunaan function dengan parameter.

```
<?php
function UcapanSalam($nama)
{

echo "Hallo ". $nama . "!<br>";
}
UcapanSalam("Agus");
UcapanSalam("Ahmad");
UcapanSalam("Budi");
UcapanSalam("Fauzan");
?>
```

Jumlah parameter dari function boleh lebih dari satu. Untuk memisahkan antar parameter digunakan tanda koma. Berikut ini contohnya.

```
<?php
function UcapanSalam($kepada, $dari)
{

echo $dari . " mengucapkan salam kepada ". $kepada .
"<br>";
}
UcapanSalam("Ari", "Amalia");
UcapanSalam("Amalia", "Nada");
UcapanSalam("Nada", "Faza");
UcapanSalam("Fauzan", "Ari");
?>
```

### 8.3 Pengembalian Nilai (Return Value)

Sebuah function juga dapat mengembalikan suatu nilai. Function hanya dapat mengembalikan sebuah nilai saja. Nilai yang dikembalikan dapat berupa suatu bilangan (bulat, real), string, maupun array, dll.

Berikut ini adalah contoh penggunaan function yang mengembalikan nilai.

```
<?php

function Jumlahkan($x, $y){

$hasil = $x + $y;

return $hasil;

}

$bil = 0;

echo "Nilai bil mula-mula adalah ". $bil ."<br>";

$bil = Jumlahkan(3, 4);

echo "Nilai bil setelah memanggil function adalah " .
$bil ."<br>";

?>
```

Function Jumlahkan() di atas mengembalikan nilai dari variabel \$hasil yang merupakan hasil penjumlahan dari nilai \$x dan \$y. Sedangkan perintah \$bil = Jumlahkan(3, 4); bermakna nilai yang dikembalikan function Jumlahkan(3, 4) disimpan pada variabel \$bil (dalam hal ini nilai \$bil adalah 7).

## **Bab 9**

# **PHP dan MySQL**

MySQL merupakan salah satu DBMS open source yang paling populer pada saat ini. Meskipun dahulu MySQL pernah dikritisi karena tidak memiliki beberapa fitur yang ada dalam DBMS pada umumnya, namun saat ini MySQL sudah banyak dikembangkan.

### **9.1 Koneksi PHP ke MySQL**

Sebelum kita melakukan koneksi ke MySQL ada beberapa parameter yang harus kita ketahui terlebih dahulu. Untuk melakukan koneksi, dibutuhkan:

- Server name, merupakan nama server atau no. IP server dimana MySQL tersebut diinstall -Username, merupakan nama user yang diberikan wewenang untuk mengakses database
- dalam MySQL -Password, merupakan password yang dimiliki username dalam rangka autentifikasi. -Database name, merupakan nama database dalam MySQL yang ingin kita akses.
- Untuk memperoleh informasi parameter di atas, dapat menghubungi server administrator.

Sedangkan perintah PHP untuk melakukan koneksi ke MySQL adalah

```
<?php
mysql_connect("nama server", "username", "password") or
die(mysql_error());
echo "Koneksi ke MySQL Sukses<br>";
?>
```

Perintah di atas akan menampilkan Koneksi ke MySQL sukses apabila koneksi telah berhasil, sedangkan apabila gagal akan menampilkan pesan kesalahan.

## 9.2 Memilih Database

Setelah koneksi berhasil, selanjutnya kita dapat memilih database yang kita inginkan. Adapun perintahnya menggunakan `mysql_select_db()`.

Contoh:

```
<?php
mysql_connect("localhost", "admin", "ladmin") or
die(mysql_error());
echo "Connected to MySQL<br />";
mysql_select_db("test") or die(mysql_error());
echo "Connected to Database";
?>
```

Contoh di atas menggambarkan bagaimana cara melakukan koneksi ke MySQL dengan nama servernya localhost, username : admin, dan password: ladmin. Selanjutnya memilih database test.



### 9.3 Memberikan Query ke MySQL

Setelah kita memilih database dimana kita akan bekerja, selanjutnya kita dapat memberikan perintah query seperti SELECT, DELETE, CREATE, UPDATE. Berikut ini contoh script PHP untuk membuat tabel example dalam database test.

```
<?php
mysql_connect("localhost", "admin", "ladmin") or
die(mysql_error());
mysql_select_db("test") or die(mysql_error());

mysql_query("CREATE TABLE example(
id INT NOT NULL AUTO_INCREMENT,
nama VARCHAR(30),
umur INT)
PRIMARY KEY(id)")

or die(mysql_error());

echo "Tabel sudah dibuat";

?>
```

Pada contoh di atas, dapat dilihat bahwa perintah PHP untuk menuliskan query ke MySQL adalah

```
mysql_query("query");
```

Sedangkan berikut ini contoh script untuk menyisipkan 2 buah record/data ke tabel example.

```
<?php
mysql_connect("localhost", "admin", "ladmin") or
die(mysql_error());
mysql_select_db("test") or die(mysql_error());

mysql_query("INSERT INTO example(nama, umur)
VALUES('budiman', 20)");

mysql_query("INSERT INTO example(nama, umur)
VALUES('surti', 30)");

echo "Data sudah dimasukkan";

?>
```

## 9.4 Mengambil Data dari MySQL

Mengambil data di sini terkait dengan penggunaan query SELECT. Berikut ini contoh untuk menampilkan record pertama dari tabel example.

```
<?php
mysql_connect("localhost", "admin", "ladmin") or
die(mysql_error());
mysql_select_db("test") or die(mysql_error());
```

```
$result = mysql_query("SELECT * FROM example")

or die(mysql_error());

// menyimpan record ke dalam variabel $data
$record = mysql_fetch_array( $result );
// menampilkan data dari $record untuk setiap field
echo "Namanya: ".$record['nama']. "<br>";
echo "Umurnya: ".$record['umur'];
?>
```

Output script di atas adalah:

Namanya: budiman

Umurnya: 20

Perintah di atas hanya akan menampilkan record pertama dari tabel example. Lantas, bagaimana caranya untuk menampilkan record yang lebih dari satu? Untuk menampilkan record yang lebih dari satu, kita gunakan looping. Perhatikan contoh berikut ini.

```
<?php
mysql_connect("localhost", "admin", "ladmin") or
die(mysql_error());
mysql_select_db("test") or die(mysql_error());

$hasil = mysql_query("SELECT * FROM example")
```

```
or die(mysql_error());

while ($record = mysql_fetch_array($hasil))

{
echo "Namanya: ".$record['nama']. "<br>";
echo "Umurnya: ".$record['umur']. "<br><br>";
}
?>
```

Looping while di atas akan terus berjalan selama record masih ada untuk dibaca.

Hasil dari script di atas adalah

Namanya: budiman

Umurnya: 20

Namanya: surti

Umurnya: 30

Catatan:

Untuk lebih memudahkan Anda dalam administrasi dan mengatur database dalam MySQL, sangat dianjurkan untuk menginstall phpMyAdmin yang dapat diunduh melalui situs resminya di <http://phpmyadmin.sourceforge.net>

## **Bab 10**

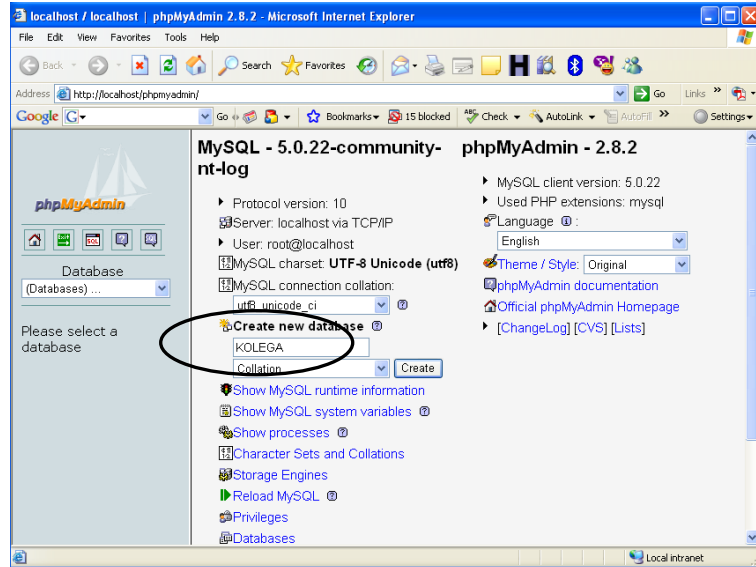
### **Project PHP dan MySQL**

Pada bab ini akan membahas koneksi PHP dan MySQL untuk membuat aplikasi manipulasi data, berisi materi tentang bagaimana menghubungkan antara database mysql dengan php, bagaimana menampilkan isi database, update database, hapus database.

Pertama kita akan membuat database di mysql dan menampilkannya di browser tentunya masih menggunakan PHP. Untuk membuat database di mysql sebaiknya gunakan PhpMyadmin. Jika anda menggunakan phptriad untuk mengakses phpmyadmin, pada browser ketikkan <http://localhost/phpmyadmin/>.

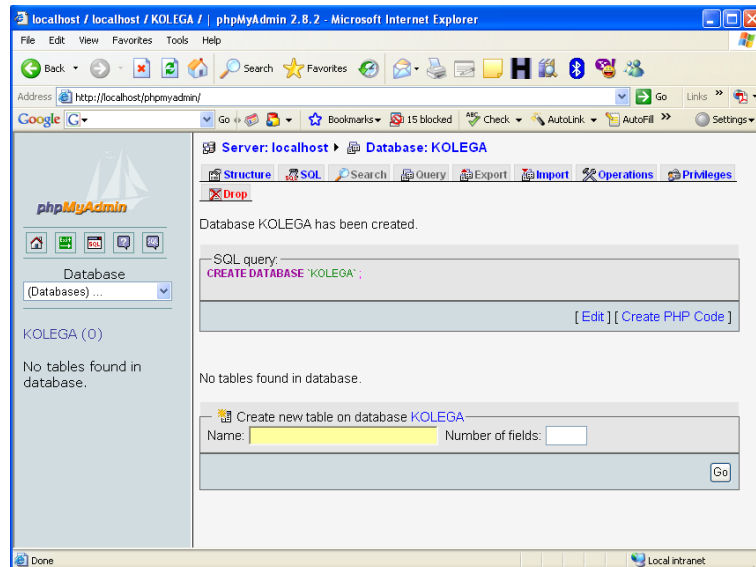
#### **10.1 Membuat Database dengan PHPMYADMIN**

Pertama kita akan membuat database di mysql dan menampilkannya di browser tentunya masih menggunakan Php. Untuk membuat database di mysql sebaiknya gunakan PhpMyadmin. Jika anda menggunakan phptriad untuk mengakses phpmyadmin, pada browser ketikkan <http://localhost/phpmyadmin/>.



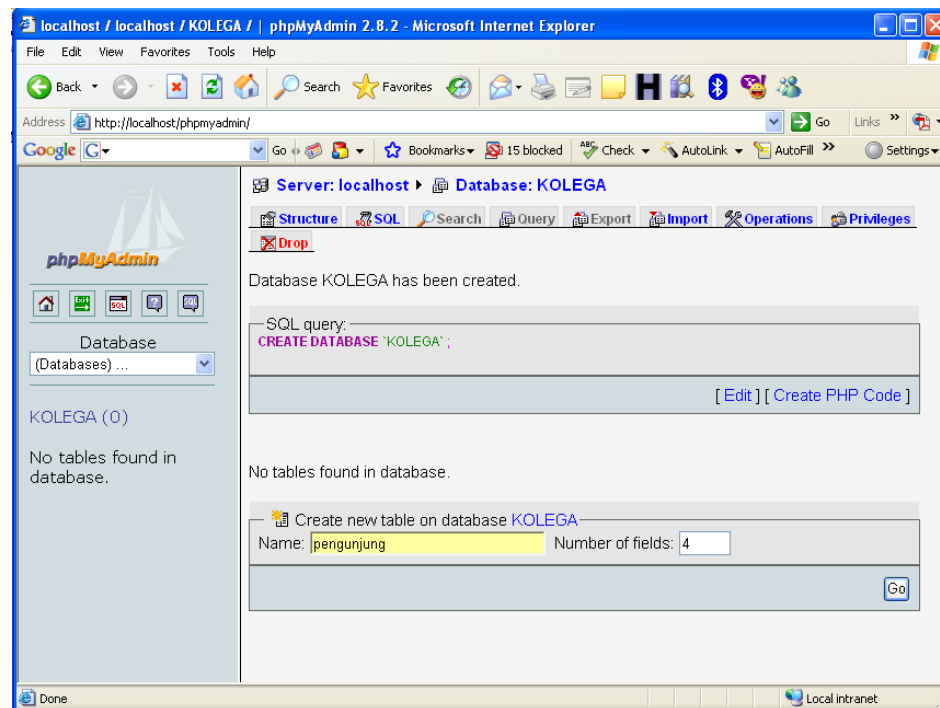
Gambar 10.1 Membuat Database

Setelah itu klik **create**. Database baru sudah terbentuk, sekarang kita tinggal membuat tabelnya. Contoh pertama kita buat tabel sederhana yaitu tabel pengunjung pada sebuah halaman web.



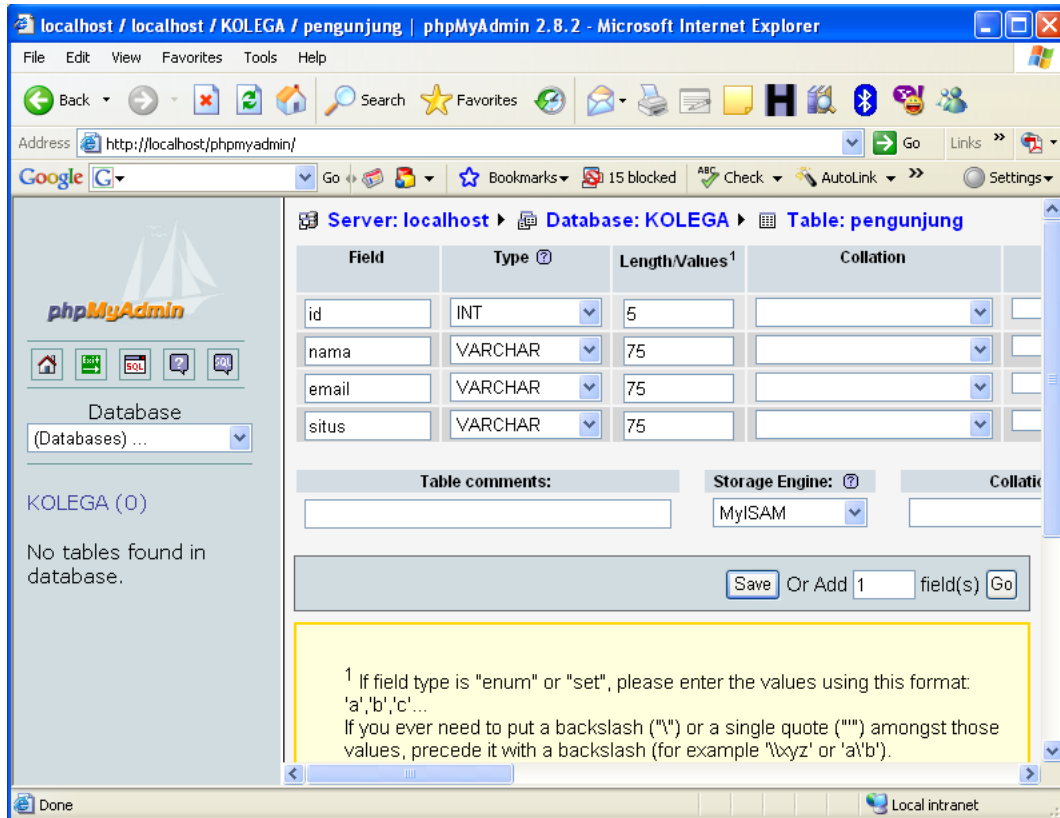
Gambar 10.2 Database Telah Terbentuk

Karena kita akan membuat tabel dengan nama **pengunjung** yang mempunyai field sejumlah 4 yaitu : id, nama, email dan situs. Maka masukkan **pengunjung** pada name dan **4** pada **fields**. Klik **Go**.



Gambar 10.3 Mengisi Nama Tabel dan Ukuran Fields

Selanjutnya akan ditampilkan window untuk mendefinisikan table, seperti tampilan berikut.



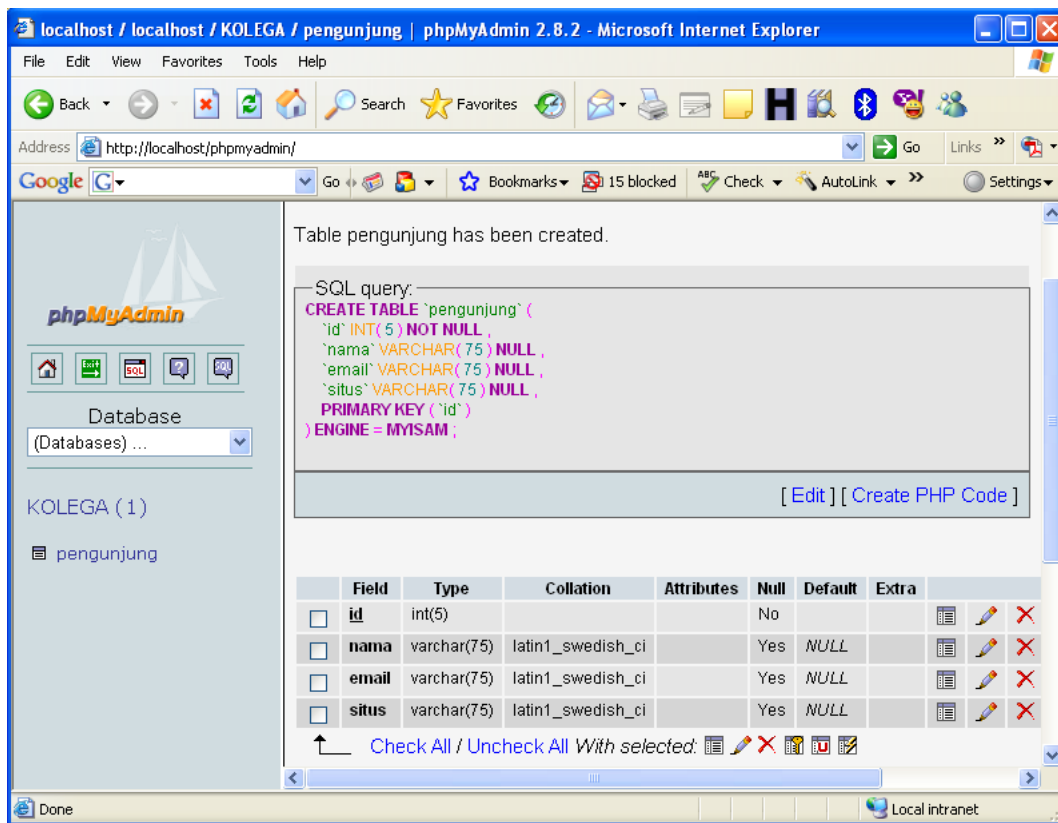
Gambar 10.4 Window Definisi Tabel

Field	Type	Length	Attirbut	Null	Default	extra	PIU
id	Int	5		Not null		Auto_increment	P
nama	Varchar	75		Not null			
email	Varchar	75		Not null			
situs	Varchar	75		Not null			

P : Primary  
 I : Index  
 U : Unique



Jangan lupa bahwa php *case sensitif* jadi sebaiknya isikan semua field dengan huruf kecil termasuk **id**. Pada gambar 4 **id** di setting auto\_increement dan Primary. Auto\_increement artinya setiap pengisian record (data ) otomatis nilai id akan di naikkan . Primary artinya memberi kunci yang unik yang dapat di jadikan pengenalan dalam tabel (dalam hal ini tabel pengunjung). Angka lebih baik di jadikan kunci utama dari sebuah tabel. Karena angka itu unik dan penggunaannya pun bisa di urutkan sehingga tidak akan ada duplikasi data. Dengan kata lain hanya dengan **id** kita nantinya akan dapat melakukan editing, deleteing, update pada record. Jika sudah klik **save**. Nantinya hasil tabel pengunjung seperti gambar 5 :

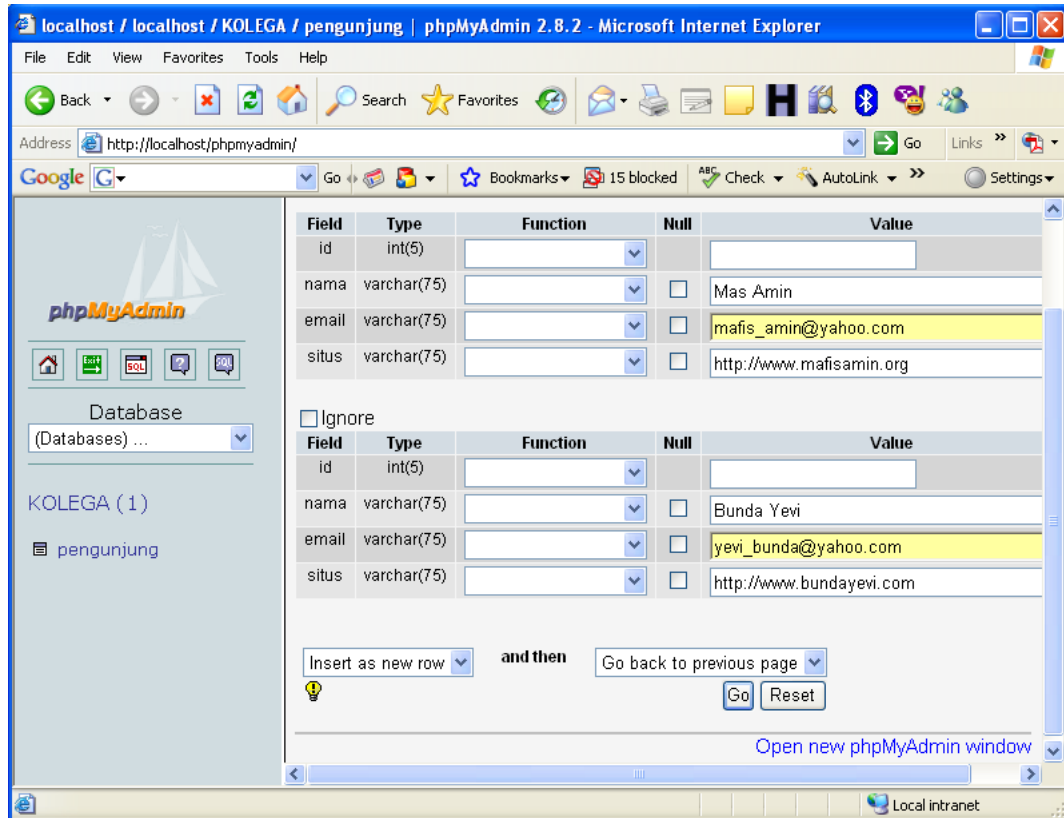


Gambar 10.5 Proses Pembuatan Tabel Berhasil

Berikut beberapa fungsionalitas yang dapat anda gunakan untuk manipulasi data di dalam table.

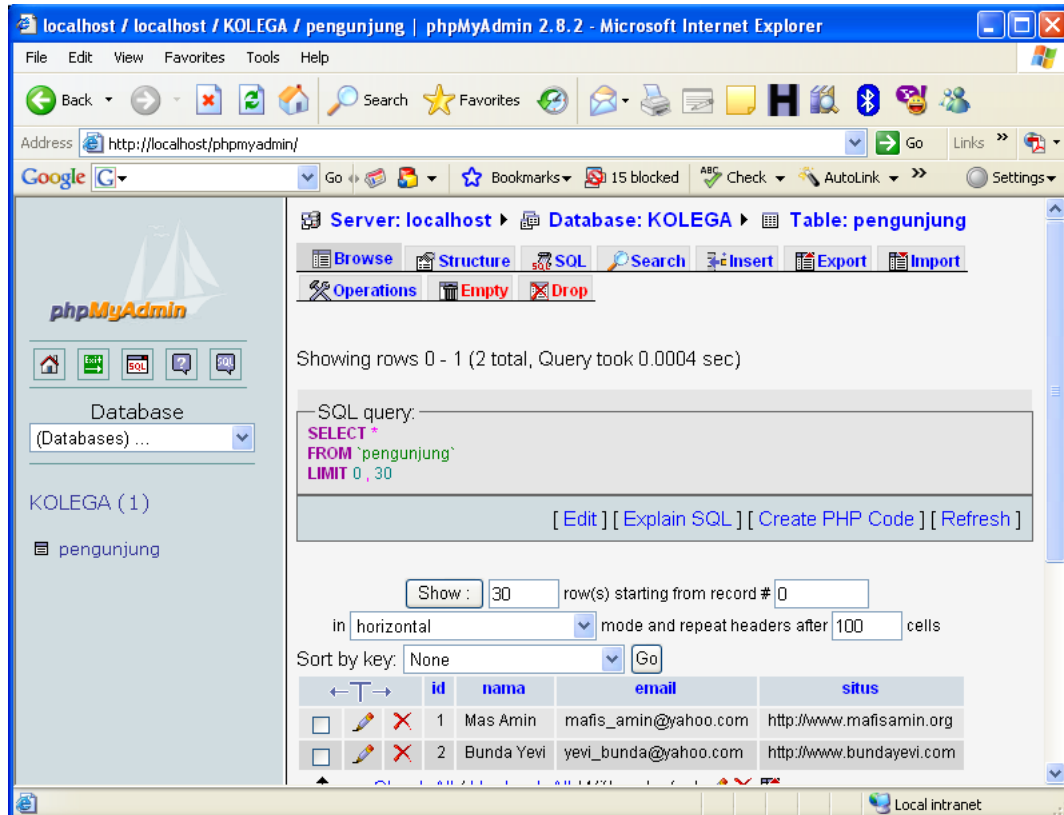
- Browse**                      untuk melihat isi table (record)
- Insert**                      untuk mengisi table
- Drop**                      Menghapus field
- Primary, Index, Unique**    Menjadikan primary, index atau unique

Isi tabel tersebut dengan beberapa records, klik **Insert** untuk mengisinya, selanjutnya klik tombol Go :



Gambar 10.6 Mengisi Record Tabel

Karena id **auto\_increment** maka id tidak perlu diisi. Jika sudah klik Save. Untuk melihat record klik Browse :



Gambar 10.7 Daftar Record Tabel

Apabila anda tidak menggunakan PhpMyadmin untuk administrasi database MySQL maka script yang dapat anda ketikkan di mysql atau paste di jendela query:

```
CREATE TABLE `pengunjung` (`id` INT(5) not null AUTO_INCREMENT, `nama`  
VARCHAR(75) not null, `email` VARCHAR(75) not null, `situs` VARCHAR(75) not null ,  
PRIMARY KEY (`id`))
```

## 10.2 Aplikasi Tampil Data

Berikut diberikan contoh sebuah script untuk menampilkan data atau isi record yang berasal dari table pengunjung yang baru saja dibuat. Ketik script di bawah ini menggunakan script editor.

```
view.php
<?
    $host = "localhost"; // alamat atau host
    $user = "root"; // username
    $pass = "sa"; // password

    $koneksi = mysql_connect("$host", "$user", $pass);
    mysql_select_db("KOLEGA", $koneksi);

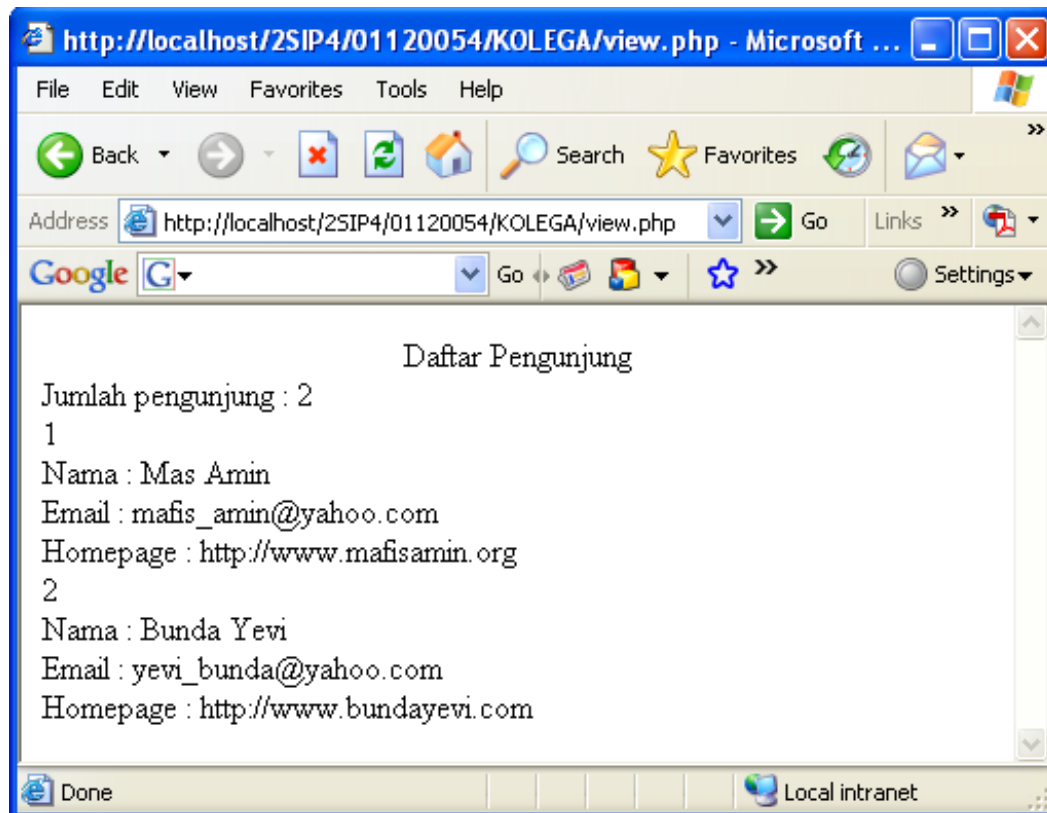
    $query = mysql_query("select * from pengunjung", $koneksi);
    $jumlah = mysql_num_rows($query);

    echo "<center>Daftar Pengunjung</center>";
    echo "Jumlah pengunjung : $jumlah";

while($baris=mysql_fetch_array($query)){
    echo "<br>";
    echo $baris[0];
    echo "<br>";
    echo "Nama : ";
    echo $baris[1];
    echo "<br>";
    echo "Email : ";
```

```
echo $baris[2];  
echo "<br>";  
echo "Homepage : ";  
echo $baris[3];  
}  
?>
```

Selanjutnya anda dapat memanggil script yang telah dibuat sebelumnya melalui web browser. Sehingga diperoleh tampilan seperti gambar berikut.



Gambar 10.8 Hasil Running view.php

**Kesimpulan :**

Untuk menampilkan database mysql di Browser dengan Php langkah-langkahnya:

- Hubungkan dulu Php dengan mysql (menggunakan fungsi `mysql_connect()`)
- Pilih database (menggunakan fungsi `mysql_select_db`)
- Query (menggunakan fungsi `mysql_query()`)
- Tampilkan (menggunakan fungsi `mysql_fetch_array`)

### 10.3 Aplikasi Insert Data

File `view.php` tidak memungkinkan untuk seseorang memasukkan record dikarenakan file tersebut hanya menampilkan record. Untuk dapat membuat sebuah halaman yang dapat memasukkan sebuah inputan maka perlu dibentuk sebuah *form*. Form tersebut memuat variabel-variabel yang dibutuhkan yang kemudian setelah di *submit* akan di proses dan dimasukkan dalam database. Maka dibutuhkanlah dua file yaitu `input.htm` dan file pemroses `input.php`.

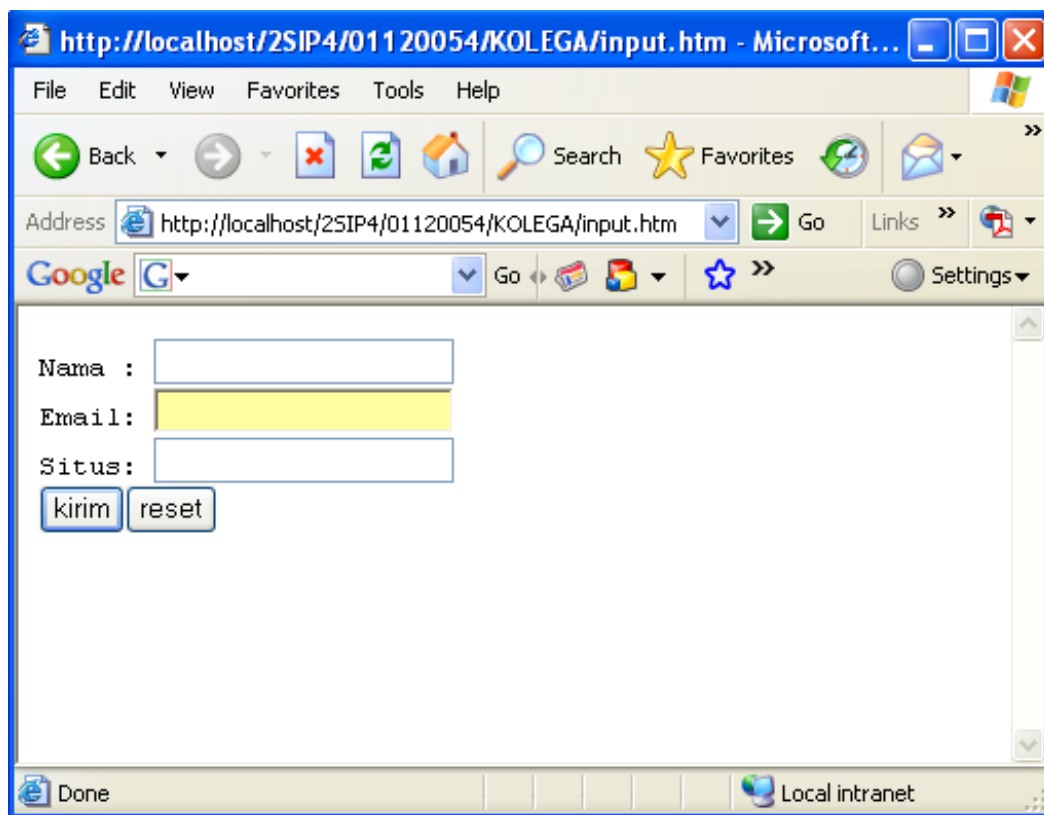
**input.htm**

```
<html>
<head></head>
<body>
<form method="post" action="input.php">
<pre>
Nama : <input type="text" name="nama">
Email: <input type="text" name="email">
Situs: <input type="text" name="situs">
<input type="submit" name="submit" value="kirim"><input
type="reset"
name="reset" value="reset">
```

```
</pre>  
</form>  
</body>  
</html>
```

<input type="text" name="nama">

**name="nama"** artinya bahwa apapun yang di inputkan dalam text box tersebut menjadi variable nama.



Gambar 10.9 Hasil Eksekusi Form



Selanjutnya script berikut digunakan untuk melakukan proses input data ke dalam table pengunjung.

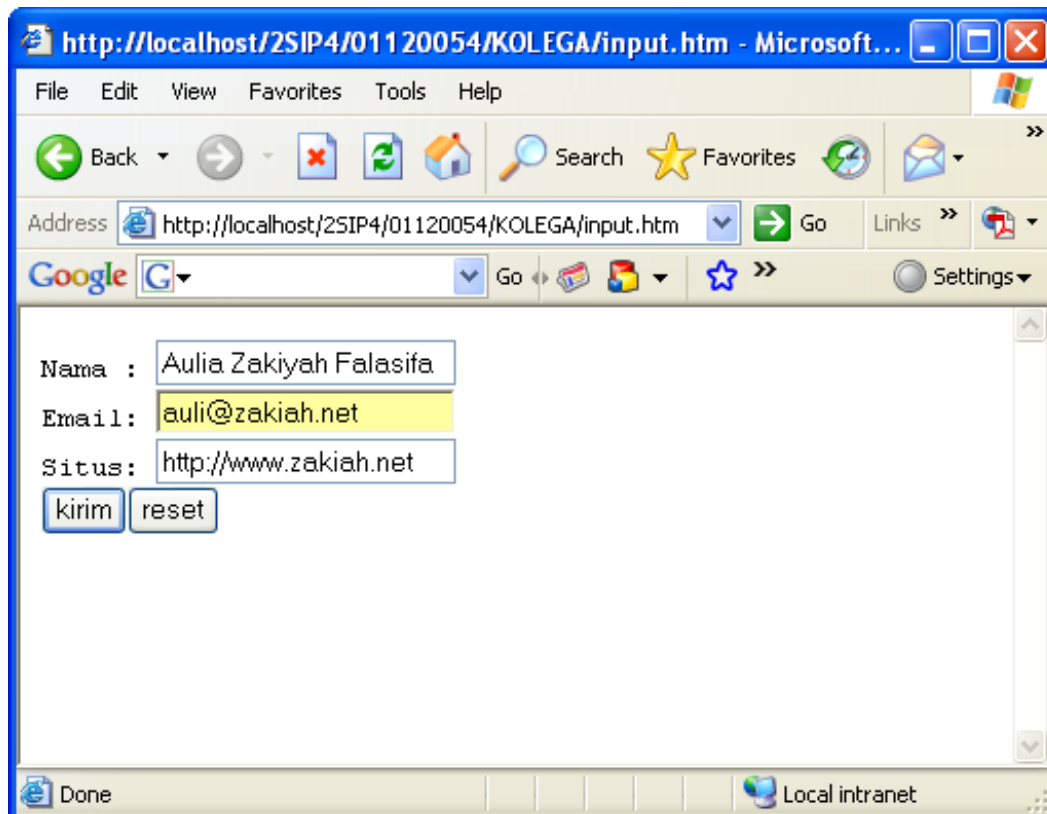
```
input.php
<?
$koneksi = mysql_connect("localhost","root","sa");
mysql_select_db("KOLEGA", $koneksi);
mysql_query("insert into pengunjung (nama, email, situs)
values ('$nama', '$email', '$situs')",$koneksi);
echo "Data telah dimasukkan";
echo "<br>";
echo "<a href=view.php>Lihat Data</a>";
?>
```

Pada source code input.htm terdapat 3 variabel yaitu nama, email dan situs. Ketiga variable tersebut nantinya akan diproses dalam file input.php. Cara pemrosesan atau pemasukan data dalam database mysql ialah menggunakan perintah mysql yaitu perintah **insert**.

```
mysql_query("insert into pengunjung (nama, email, situs)
values ('$nama', '$email', '$situs')",$koneksi);
```

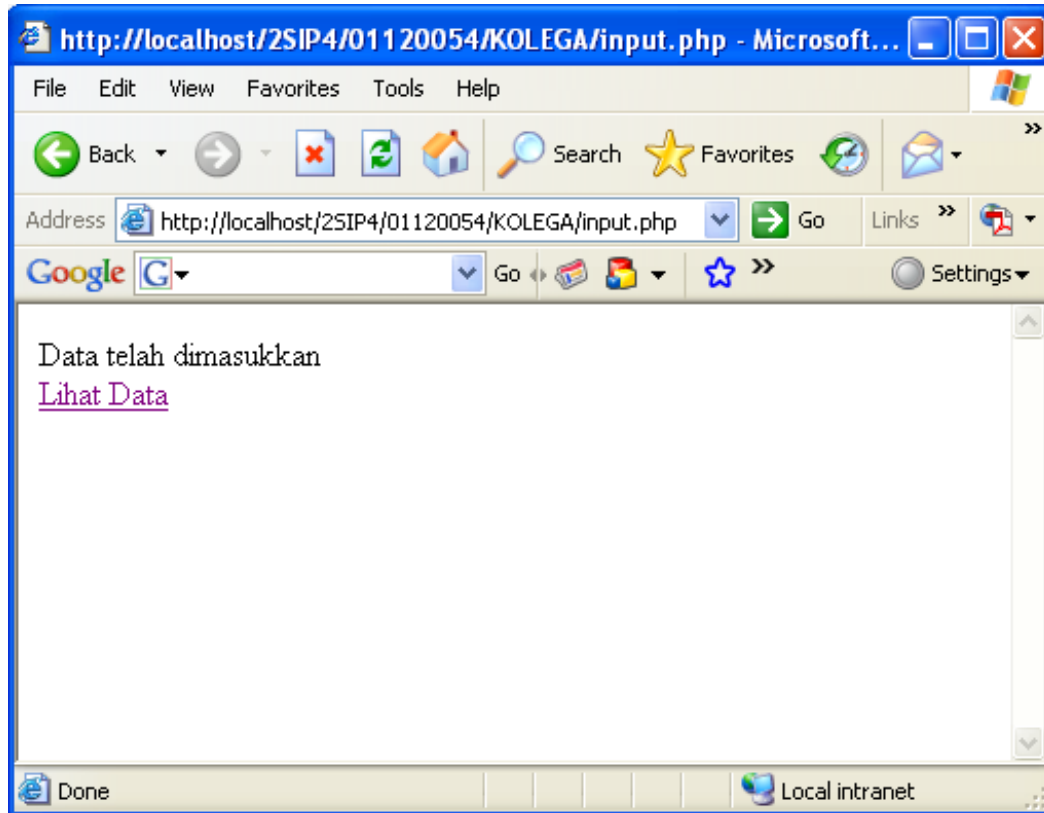
Sintak tersebut memasukkan data yang terdapat pada variable nama, variable email, variable situs untuk di masukkan ke dalam tabel pengunjung dengan urutan field nya nama, email dan situs. **Id** tidak dimasukkan karena sudah **auto\_increment**. Memasukkan dalam database harusurut dengan fieldnya. Pada tabel pengunjung urutannya id, nama, email, situs. Maka pada pengisiannya harus di urutkan.

Pengunjung (nama, email, situs) = nama field dalam tabel pengunjung. Sekarang akan kita coba memasukkan data. Buka file input.htm di browser :



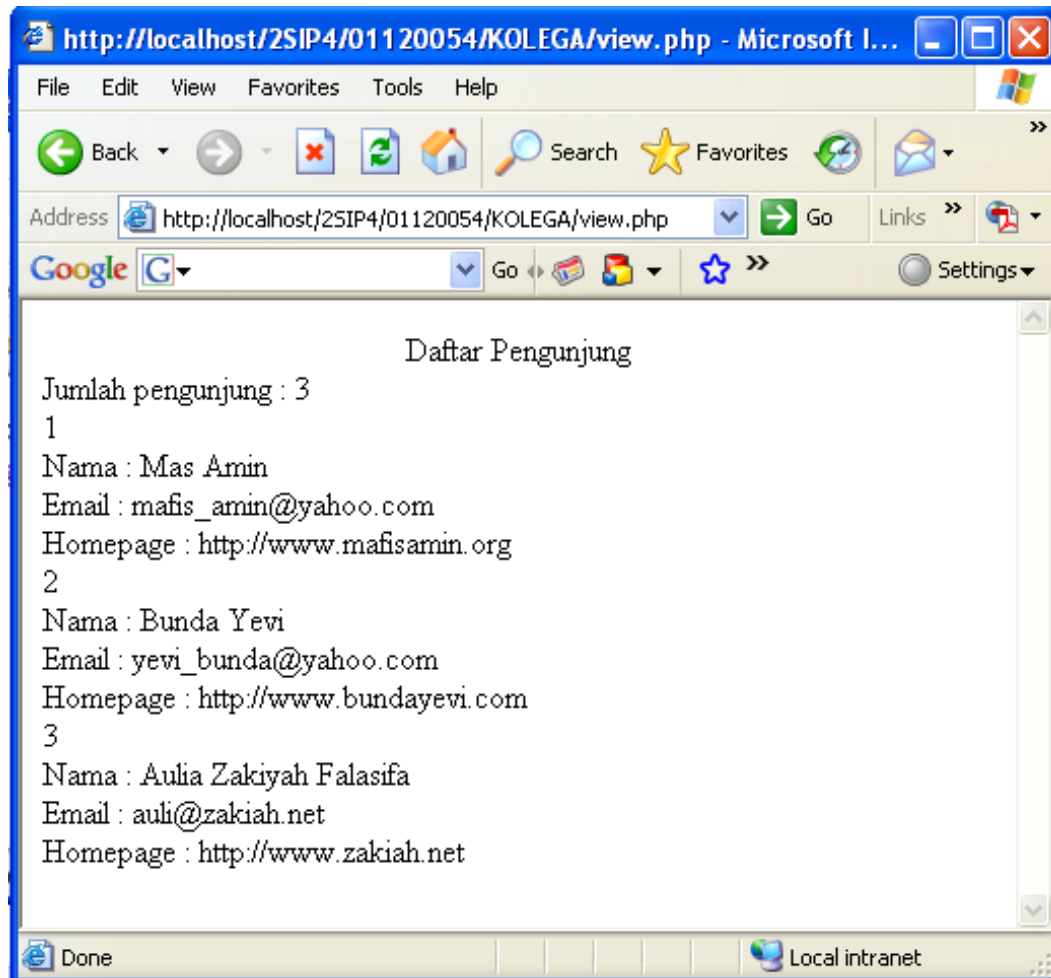
Gambar 10.10 Mengisi Record Baru

Klik kirim, maka selanjutnya akan diproses di input.php



Gambar 10.11 Proses Pengisian Record Ke Tabel

Untuk melihat data, klik link Lihat Data. Sehingga diperoleh tampilan seperti gambar berikut.



Gambar 10.12 Daftar Pengunjung Setelah Ditambah Record

Bisa anda lihat bahwa sekarang jumlah record ada 3. **id** yang tidak kita isikan dan masukkan dalam input.php maka di isi otomatis dan nilainya di naikan satu (+1) menjadi 3. Inilah yang dinamakan `auto_increment`. Dari sini anda bisa membuat sebuah form untuk input data. Program diatas juga dapat anda tambahkan yaitu : jika ada nama atau email atau situs yang dikosongi maka data tidak dapat di masukkan ke dalam database karena kosong.

Coba anda modifikasi kembali script input.php anda untuk menghalangi proses, ketika teks isian dalam keadaan kosong.

<b>input.php</b>
<pre>&lt;? if((\$nama!="") or (\$email!="") or (\$situs!="")) { \$koneksi = mysql_connect("localhost","root","sa"); mysql_select_db("KOLEGA", \$koneksi); mysql_query("insert into pengunjung (nama, email, situs) values ('\$nama', '\$email', '\$situs')",\$koneksi); echo "Data telah dimasukkan"; echo "&lt;br&gt;"; echo "&lt;a href=view.php&gt;Lihat Data&lt;/a&gt;"; }else{ echo "Cek kembali form anda"; echo "&lt;br&gt;"; echo "&lt;a href=input.htm&gt;Isi Data&lt;/a&gt;"; } ?&gt;</pre>

## 10.4 Aplikasi Edit Data

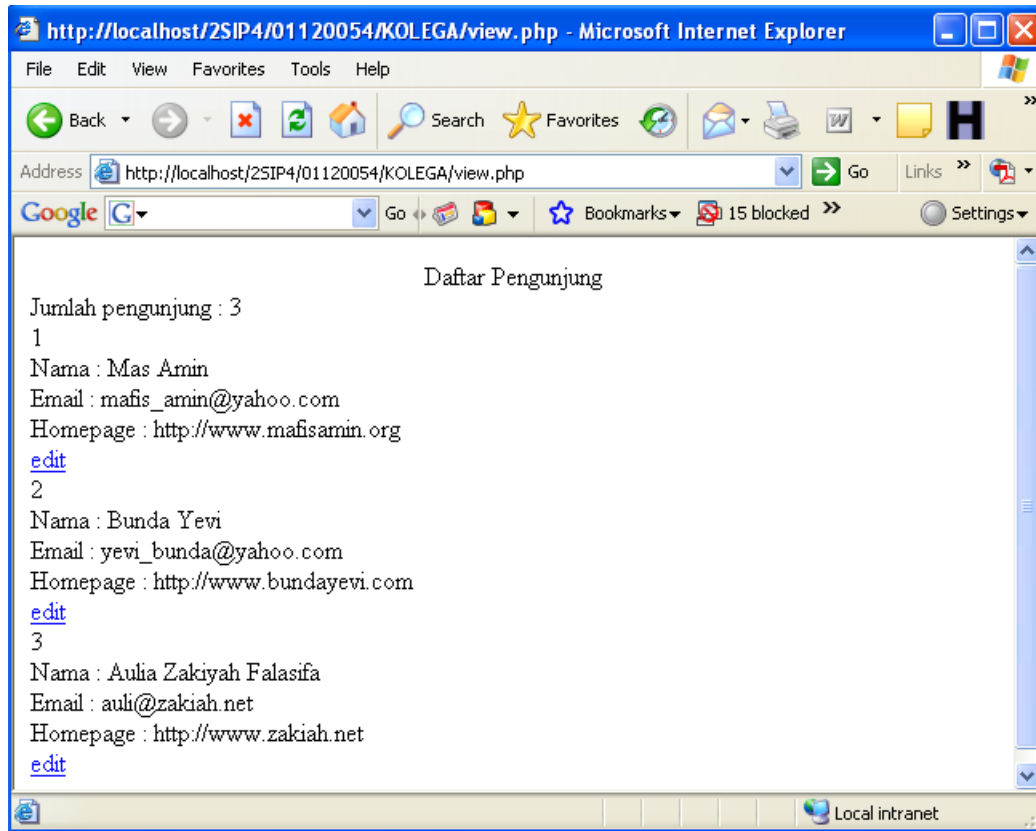
Sebuah database tentunya terdapat revisi-revisi atau perubahan isi tabel atau record. Maka harusnya terdapat sebuah form yang mampu mengedit isi yang terdapat dalam database. Jika pada file input.php kita sudah dapat memasukkan data ke dalam database maka kali ini kita akan mengedit data yang telah di masukkan. Langkah pertama kita harus memodifikasi file view.php agar dapat mempunyai *link* untuk menuju sebuah form

peng-editan yang selanjutnya dengan form tersebut akan di proses dan di update ke dalam database.

**view.php**

```
<?
    $host = "localhost";// alamat atau host
    $user = "root"; // username
    $pass = "sa"; // password
    $koneksi = mysql_connect("$host", "$user", $pass);
    mysql_select_db("KOLEGA", $koneksi);
    $query = mysql_query("select * from pengunjung", $koneksi);
    $jumlah = mysql_num_rows($query);
    echo "<center>Daftar Pengunjung</center>";
    echo "Jumlah pengunjung : $jumlah";
while($baris=mysql_fetch_array($query)){
    echo "<br>";
    echo $baris[0];
    echo "<br>";
    echo "Nama : ";
    echo $baris[1];
    echo "<br>";
    echo "Email : ";
    echo $baris[2];
    echo "<br>";
    echo "Homepage : ";
    echo $baris[3];
    echo "<br><a href=edit.php?id=$baris[0]>edit</a>";
}
?>
```

Script `view.php` mendapat tambahan pada baris ke-3 dari akhir script. Dimana dibuat sebuah link yang memanfaatkan variable `id` untuk di proses pada tahap selanjutnya.



Gambar 10.13 Hasil Eksekusi Script `View.php`

Jika cursor diarahkan pada link `edit` maka link tersebut akan menggunakan `id` dari record untuk dijadikan sebagai variable pemroses di file `edit.php`.

<http://localhost/2SIP4/01120054/KOLEGA/edit.php?id=1>

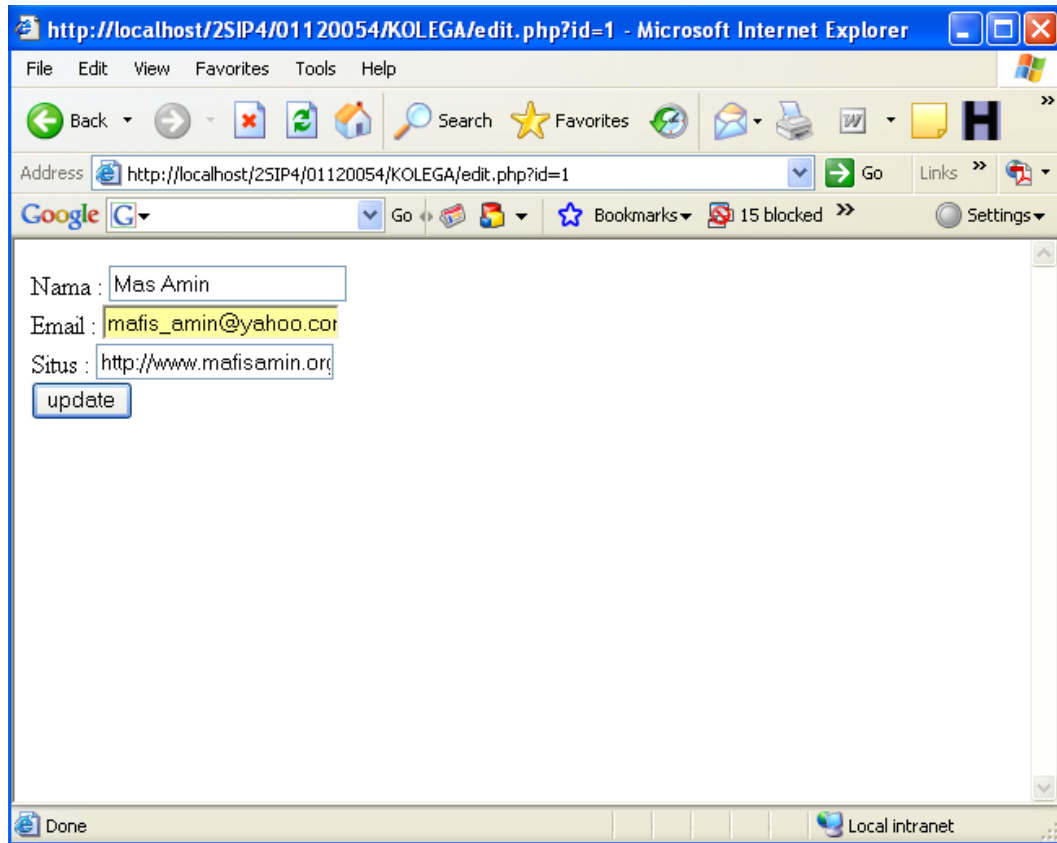
Ketika cursor diarahkan pada record pertama maka link mempunyai nilai id=1. Nilai 1 diambil dari hasil query dari \$baris[0]. Begitu seterusnya. Sehingga dengan mengambil nilai dari **id** maka pada file edit.php nilai **id** tersebut akan di jadikan alat bantu untuk editing. Kenapa harus menggunakan **id** ? karena **id** telah di set primary yang artinya tidak ada *redundancy id*. File edit.php membawa sebuah variable **id**. Dengan variable ini akan di lakukan query yaitu memilih semua field yang mempunyai id = 1 (nilai 1 diambil dari edit.php?id=1); dalam hal ini di contohkan id=1.

```
edit.php
<?
    $koneksi = mysql_connect("localhost","root","sa");
    mysql_select_db("KOLEGA",$koneksi);
    $query = mysql_query("select * from pengunjung where
id='$id'",$koneksi);
    while ($baris = mysql_fetch_array($query)){
        echo "<form method=\"post\"
action=\"update.php\">";
        echo "Nama : <input type=\"text\" name=\"nama\"
value=\"$baris[1]\">";
        echo "<br>";
        echo "Email : <input type=\"text\" name=\"email\"
value=\"$baris[2]\">";
        echo "<br>";
        echo "Situs : <input type=\"text\" name=\"situs\"
value=\"$baris[3]\">";
        echo "<br>";
        echo "<input type=\"submit\" name=\"submit\" value=\"update\">";
        echo "<input type=\"hidden\" name=\"id\" value=\"$baris[0]\">";
        echo "</form>";
    }
}
```



```
}  
?>
```

Selanjutnya anda dapat menjalankan script di atas, sehingga diperoleh tampilan seperti gambar berikut.

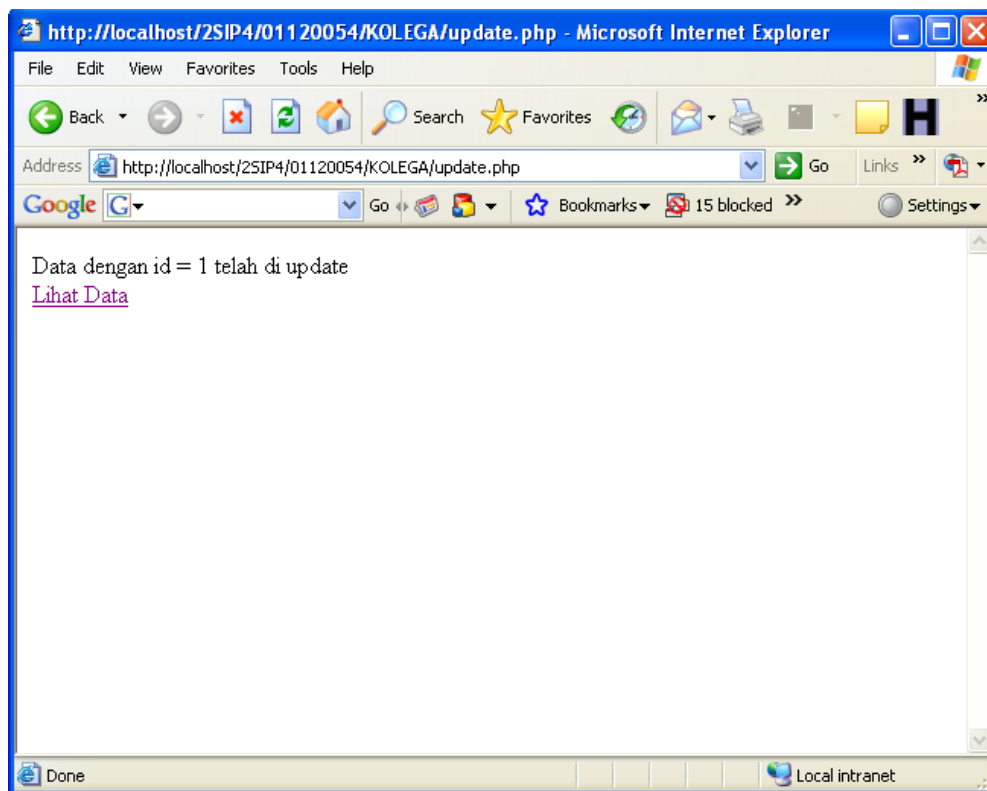


Gambar 10.14 Mengubah Data Record

Bisa anda lihat bahwa dengan modal variable id saja record tersebut dapat di edit. Logika dari edit.php ialah hasil query di tampung dalam text box. Sehingga nilai (value) yang ada dalam text box hasil query. Silahkan merubah nya jika sudah klik **update**.

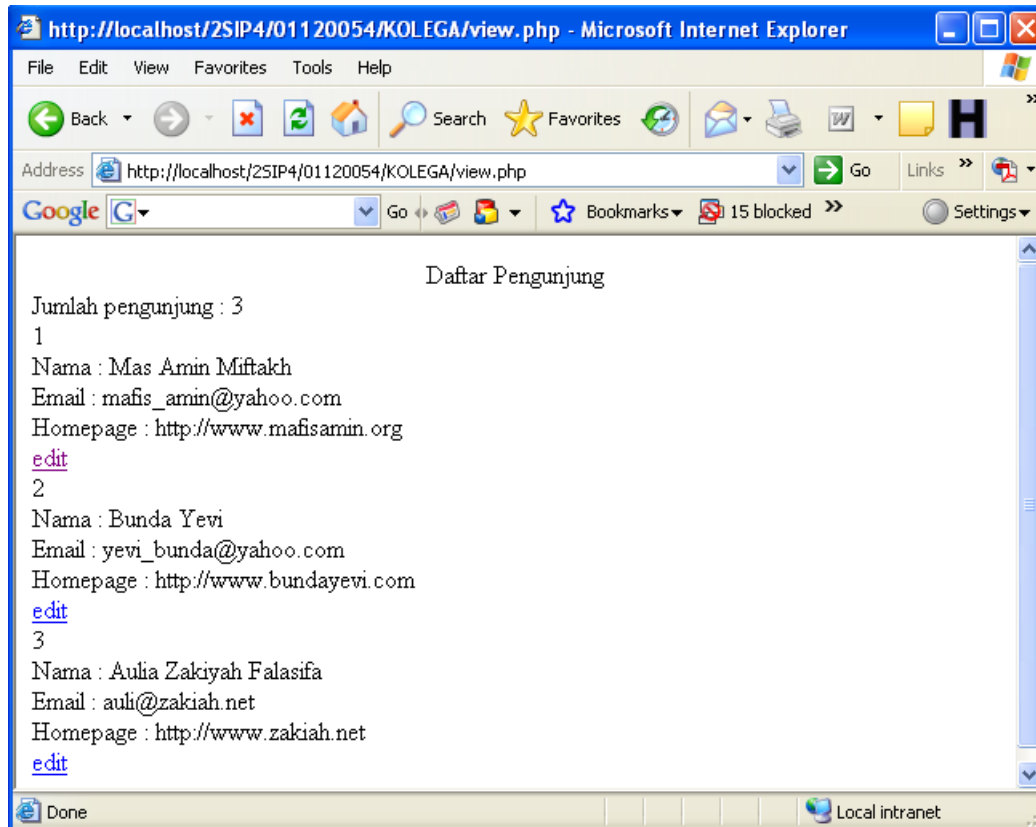
```
update.php  
<?  
    $koneksi = mysql_connect("localhost","root","sa");  
    mysql_select_db("KOLEGA",$koneksi);  
    $query = mysql_query("update pengunjung set id='$id',  
nama='$nama', email='$email', situs = '$situs' where id='$id'");  
    echo "Data dengan id = $id telah di update";  
    echo "<br>";  
    echo "<a href=view.php>Lihat Data</a>";  
?>
```

Jalankan script di atas, sehingga diperoleh tampilan seperti berikut.



Gambar 10.15 Hasil Update Data

Proses update telah selesai. Anda bisa meliha isi data pada file view.php.



Gambar 10.16 Daftar Pengunjung Setelah Diedit

## 10.5 Aplikasi Delete

Record tidak selamanya akan disimpan jika database yang digunakan bersifat dinamis. Dimana setiap waktu atau detik database akan di perbaharui dan data-data lama juga nantinya akan di hapus jikalau sudah tidak sesuai dengan fakta yang ada. Memang seharusnya data-data lama menjadi arsip tapi data-data lama yang ternyata sudah tidak

sesuai dengan kebijakan yang baru inilah yang akan di hapus. Menghapus database MySQL dengan php tentunya sangat mudah. Logika bekerjanya sama dengan edit.php dimana mengambil variable **id** dan di proses di file delete.php. Perintah yang digunakan untuk menghapus adalah perintah MySQL yaitu **delete**. File view.php harus kita modifikasi terlebih dahulu.

**view.php**

```
<?
    $host = "localhost";// alamat atau host
    $user = "root"; // username
    $pass = "sa"; // password

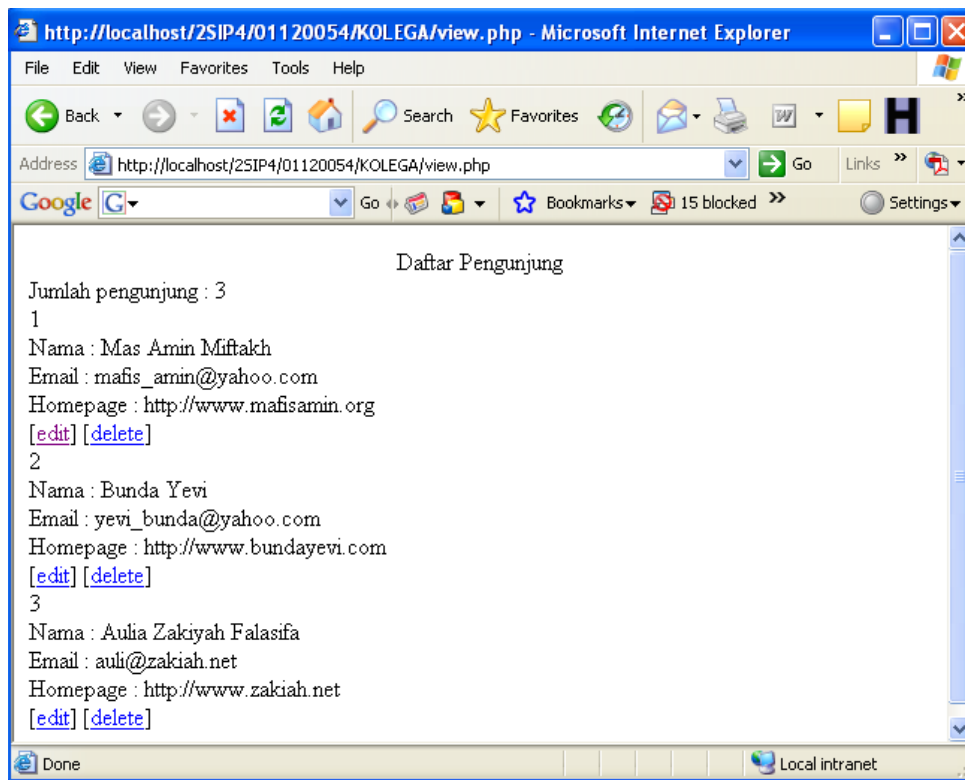
    $koneksi = mysql_connect("$host", "$user", $pass);
    mysql_select_db("KOLEGA", $koneksi);

    $query = mysql_query("select * from pengunjung", $koneksi);
    $jumlah = mysql_num_rows($query);
    echo "<center>Daftar Pengunjung</center>";
    echo "Jumlah pengunjung : $jumlah";

while($baris=mysql_fetch_array($query)){
    echo "<br>";
    echo $baris[0];
    echo "<br>";
    echo "Nama : ";
    echo $baris[1];
    echo "<br>";
    echo "Email : ";
    echo $baris[2];
```

```
echo "<br>";  
echo "Homepage : ";  
echo $baris[3];  
echo "<br>";  
echo "[<a href=edit.php?id=$baris[0]>edit</a>]  
[<a href=delete.php?id=$baris[0]>delete</a>]";  
}  
?>
```

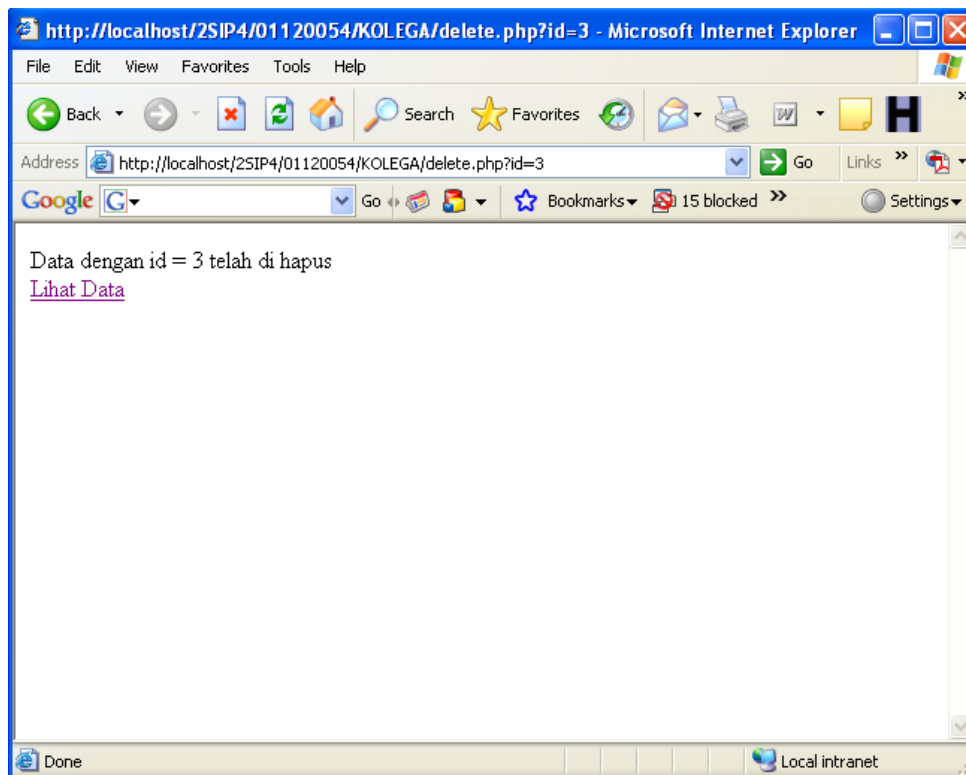
Selanjutnya jalankan script di atas, sehingga diperoleh tampilan seperti gambar berikut.



Gambar 10.17 Eksekusi Script View.php

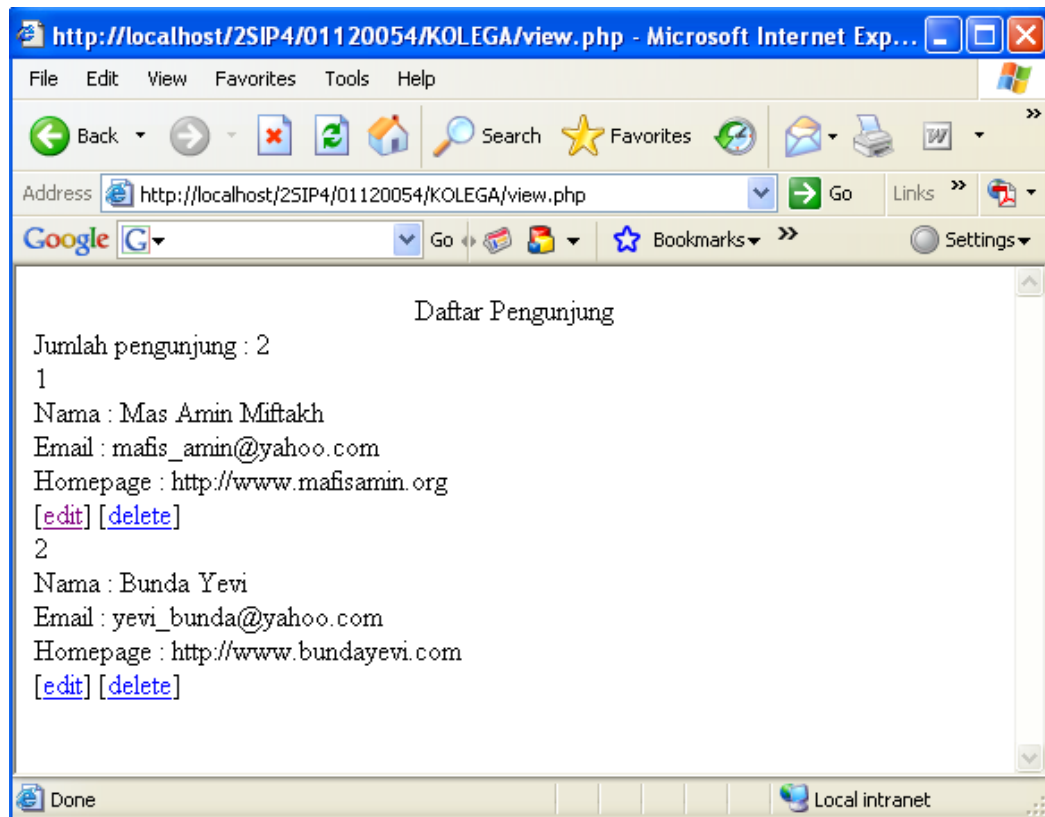
```
delete.php  
  
<?  
    $koneksi = mysql_connect("localhost","root","sa");  
    mysql_select_db("KOLEGA",$koneksi);  
    mysql_query("delete from pengunjung where id='$id'", $koneksi);  
    echo "Data dengan id = $id telah di hapus";  
    echo "<br>";  
    echo "<a href=view.php>Lihat Data</a>";  
?>
```

Jalankan script sehingga diperoleh tampilan seperti berikut.



Gambar 10.18 Eksekusi Script Hapus Data

Selanjutnya anda dapat melihat perubahan yang terjadi setelah anda melakukan proses hapus data. Seperti tampak pada gambar berikut.



Gambar 10.19 Data Setelah Dihapus

## Tentang Penulis



**M. Miftakul Amin** lahir di Raman Aji, Lampung Timur. Menyelesaikan jenjang Strata 1 di Jurusan Teknik Informatika, Universitas Teknologi Yogyakarta (UTY). Kemudian melanjutkan studi Strata 2 di Jurusan Teknik Elektro dan Teknologi Informasi Universitas Gadjah Mada (UGM) dengan mengambil konsentrasi *Computer and Information System (CIS)*. Aktifitas saat ini adalah sebagai dosen tetap di jurusan teknik informatika/komputer Politeknik Negeri Sriwijaya Palembang. Selain sibuk mengajar penulis juga seorang *programmer* yang cukup berpengalaman mengembangkan sistem informasi yang telah banyak digunakan baik oleh instansi swasta maupun pemerintah.

### Informasi kontak :

homepage : <http://mafisamin.web.ugm.ac.id>  
e-mail : [mafis\\_amin@yahoo.com](mailto:mafis_amin@yahoo.com)  
mobile : 0813 790 18 202  
y-messenger : mafis\_amin  
software Dev. : <http://www.masaminsoft.com>

---