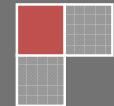


v1.0

Pemrograman Web dengan PHP dan MySQL

Achmad Solichin, S.Kom

Buku ini disusun berdasarkan materi perkuliahan Pemrograman Web sesuai dengan kurikulum standar di perguruan tinggi komputer. Di dalam buku ini tidak terlalu dibahas mengenai teori dan algoritma pemrograman, namun banyak berisi contoh-contoh program yang dapat dipraktekkan secara langsung serta disertai dengan penjelasannya. Dengan demikian, buku ini sangat cocok bagi Anda yang baru belajar pemrograman web dan haus akan contoh-contoh program. Buku ini juga sesuai dijadikan sebagai referensi penyusunan praktikum perkuliahan bagi mahasiswa maupun siswa.



*...ku persembahkan untuk Indonesia tanah air tercinta, dan untuk Chotimatul
Musyarofah istri tersayang serta Muhammad Lintang putra tercinta kami...*

Lisensi Dokumen

Seluruh isi dalam dokumen ini dapat digunakan, dimodifikasi dan disebarluaskan secara bebas untuk tujuan pendidikan, pembelajaran dan bukan komersial (*non profit*), dengan syarat tidak menghilangkan, menghapus atau mengubah atribut penulis dokumen ini dan pernyataan dalam lisensi dokumen yang disertakan di setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang atau mengkomersialkan buku ini kecuali mendapatkan ijin terlebih dahulu dari penulis.

Kata Pengantar

Internet merupakan salah satu fenomena paling berpengaruh dalam beberapa dekade terakhir ini, baik bagi perkembangan teknologi itu sendiri maupun bagi perkembangan disiplin ilmu lain, seperti ekonomi, politik, sosial dan budaya. Perkembangan internet juga berpengaruh langsung terhadap perkembangan situs web. Di awal kemunculannya, internet maupun situs web belum begitu populer karena layanannya yang masih terbatas. Namun lihatlah saat ini, situs web sudah demikian populer dan layanannya pun beragam, mulai dari situs berita seperti Detik dan Okezone, situs penyedia email seperti Yahoo dan Gmail, sampai situs pertemanan (*social networking*) seperti Friendster dan Facebook.

Dalam kaitannya dengan situs web dan layanan di dalamnya tidak lepas dari teknologi di baliknya, yaitu pemrograman web, desain web dan database. Salah satu bahasa pemrograman yang banyak digunakan untuk membangun suatu situs web adalah PHP (<http://www.php.net>). Penggunaannya saat ini bahkan sudah mencapai 25 juta situs web. Sedangkan dari sisi database, penggunaan MySQL (<http://www.mysql.com>) masih juga mendominasi dalam aplikasi berbasis web.

Melalui buku ini, penulis berusaha mengenalkan dasar-dasar dari PHP dan MySQL untuk membangun suatu situs web. Buku ini sebagian besar menyajikan berbagai contoh program yang disusun secara terstruktur dari yang mudah sampai yang cukup kompleks. Dengan adanya contoh-contoh program tersebut, diharapkan pembaca dapat mempraktekkannya secara langsung (*learning by doing*) dan dapat menyimpulkan sendiri maksud dari setiap perintah dalam program dengan cara melihat hasil yang ditampilkan di layar (*browser*). Di akhir buku ini, disajikan contoh yang lebih nyata, yaitu aplikasi situs berita sederhana yang dapat dikembangkan lebih lanjut.

Buku ini cocok bagi pembaca yang baru mulai belajar PHP dan MySQL secara mandiri karena contoh-contoh dalam buku ini disajikan sesederhana mungkin. Buku ini juga cocok digunakan sebagai panduan praktikum pemrograman web di perguruan tinggi, sekolah maupun kursus-kursus.

Akhirnya penulis berharap agar buku ini bermanfaat bagi perkembangan ilmu dan pengetahuan di Indonesia, khususnya dalam hal pemrograman web dan internet. Saran dan kritik untuk perbaikan buku ini sangat penulis harapkan. Saran, kritik dan masukan mengenai buku ini dapat disampaikan melalui email ke penulis di achmatim@gmail.com atau melalui situs penulis di <http://achmatim.net>.

Penulis
Achmad Solichin

Daftar Isi

I.	Pengenalan Web Server dan Server Side Scripting	7
1.	Web Server.....	7
2.	Server Side Scripting	8
3.	Test Instalasi XAMPP di Browser.....	13
II.	Struktur Dasar PHP	14
1.	Sebelum Belajar PHP	14
2.	Mari Mengenal PHP	14
3.	Say Hello to PHP	15
4.	Variabel	15
5.	Tipe Data	16
6.	Konstanta	17
7.	Operator dalam PHP	17
8.	Komentar Program.....	19
III.	Struktur Kondisi dan Perulangan.....	21
1.	Struktur Kondisi.....	21
2.	Struktur Perulangan	25
3.	Struktur Break dan Continue	29
IV.	Penanganan Form	30
1.	Berbagai Cara Penanganan Form.....	30
2.	Form Input Type TEXT dan PASSWORD	33
3.	Form Input Type RADIO	36
4.	Form Input Type CHECK BOX	38
5.	Form Input Type COMBO BOX	39
6.	Form Input Type TEXTAREA	41
V.	Array dan Fungsi	43
1.	Pengantar Array	43
2.	Fungsi-fungsi Array dalam PHP	47
3.	Fungsi dalam PHP.....	50
VI.	Penanganan String dan Tanggal	52
1.	Pengantar String.....	52
2.	Fungsi-fungsi String dalam PHP	53
3.	Fungsi-fungsi Operasi Tanggal dalam PHP	60
VII.	File dan Direktori	62
1.	Penanganan File	62

2.	Membuat dan Menghapus Direktori	66
3.	Manipulasi File	68
4.	Upload File	69
VIII.	Session dan Cookies.....	72
1.	Session.....	72
2.	Cookies.....	75
IX.	Pemrograman Berorientasi Objek dalam PHP	77
1.	Pengantar Pemrograman Berorientasi Objek.....	77
2.	Object dan Class.....	77
3.	Properties dan Method.....	78
4.	Mendefinisikan Class.....	78
5.	Manfaatkan Class yang Tersedia.....	82
X.	Pengenalan DBMS dan MySQL	84
1.	Tentang Database, DBMS dan RDBMS.....	84
2.	Tentang MySQL	85
3.	Instalasi MySQL di Windows	85
4.	Koneksi ke Server MySQL dengan MySQL Client	95
5.	Berbagai MySQL Client untuk Administrasi Server MySQL.....	96
XI.	Pengenalan Structured Query Language.....	99
1.	Apa Itu SQL ?	99
2.	Membuat, Mengubah dan Menghapus <i>Table</i>	101
3.	Menambah Record dengan INSERT	105
4.	Mengedit Record dengan UPDATE	106
5.	Menghapus Record dengan DELETE	107
XII.	Aplikasi Berita dengan PHP dan MySQL	113
1.	Merancang Struktur Database	113
2.	Membuat File Koneksi Database.....	114
3.	Membuat Halaman Input Berita	114
4.	Menampilkan Berita Terbaru di Halaman Depan	115
5.	Membuat Halaman Arsip Berita	117
6.	Membuat Halaman Edit Berita	118
7.	Membuat Halaman Delete Berita	120
8.	CSS sebagai Pemanis Tampilan	120

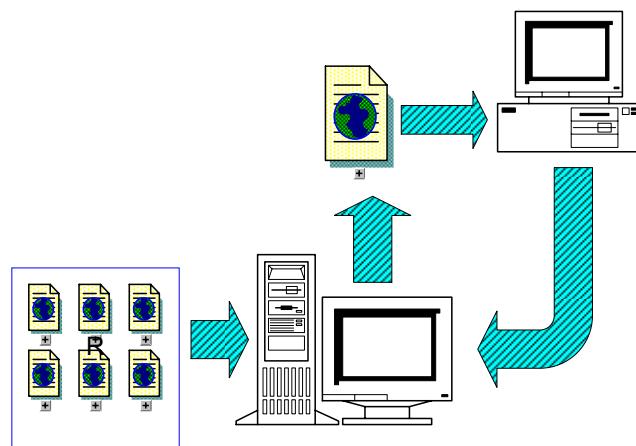
Bab 1

Pengenalan Web Server dan Server Side Scripting

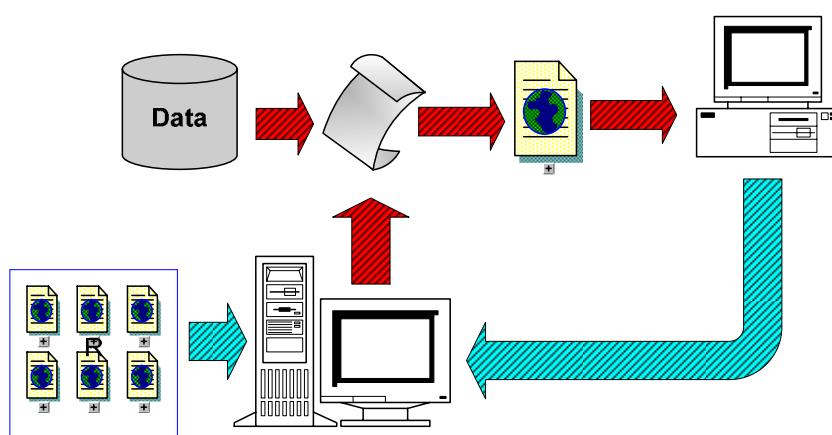
- ❖ Pengenalan Web Server
- ❖ Instalasi dan Konfigurasi Web Server
- ❖ Instalasi dan Konfigurasi PHP
- ❖ Testing Web Server dan PHP

Web Server

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



Gambar 1.1. Standar Web Architecture



Gambar 1.2. Dynamic Web Architecture

Beberapa Web Server yang banyak digunakan di internet antara lain :

1. Apache Web Server (<http://www.apache.org>)
2. Internet Information Service, IIS (<http://www.microsoft.com/iis>)

3. Xitami Web Server (<http://www.xitami.com>)
4. Sun Java System Web Server
(http://www.sun.com/software/products/web_srver/home_web_srver.xml)

Server Side Scripting

Server Side Scripting merupakan sebuah teknologi *scripting* atau pemrograman web dimana *script* (program) dikompilasi atau diterjemahkan di *server*. Dengan *server side scripting*, memungkinkan untuk menghasilkan halaman web yang dinamis.

Beberapa contoh Server Side Scripting (Programming) :

5. ASP (Active Server Page) dan ASP.NET
6. ColdFusion (<http://www.macromedia.com/software/coldfusion>)
7. Java Server Pages (<http://java.sun.com/products/jsp/>)
8. Perl (<http://www.perl.org>)
9. Phyton (<http://www.python.org>)
10. **PHP** (<http://www.php.net>)

Keistimewaan PHP

- ✓ Cepat
- ✓ Free
- ✓ Mudah dipelajari
- ✓ Multi-platform
- ✓ Dukungan *technical-support*
- ✓ Banyaknya komunitas PHP
- ✓ Aman

Instalasi Apache, PHP dan MySQL dengan XAMPP

Proses instalasi Apache, PHP dan MySQL seringkali menjadi kendala terutama bagi pemula yang baru belajar pemrograman web dengan PHP. Hal tersebut disebabkan karena *software* tersebut harus diinstall dan dikonfigurasikan satu per satu. Bagi belum mengerti benar, tentu sering mengalami kesulitan, apalagi bagi kita yang belajar secara mandiri.

Namun jangan berkecil hati, saat ini banyak tersedia aplikasi paket yang menyatukan ketiga software tersebut (Apache, PHP dan MySQL) ke dalam satu *installer*. Proses instalasinya pun dapat dilakukan dengan mudah dan cepat, hanya perlu *next, next, and finish*.

Beberapa aplikasi paket tersebut antara lain:

1. XAMPP (versi Windows) dan LAMPP (versi Linux) yang dapat didownload di <http://apachefriends.org>.
2. WAMP Server.
3. APPServ
4. PHPTriad.

Penulis sendiri menyarankan pemakaian XAMPP atau WAMP Server karena versi *software* di dalamnya menggunakan versi terbaru.

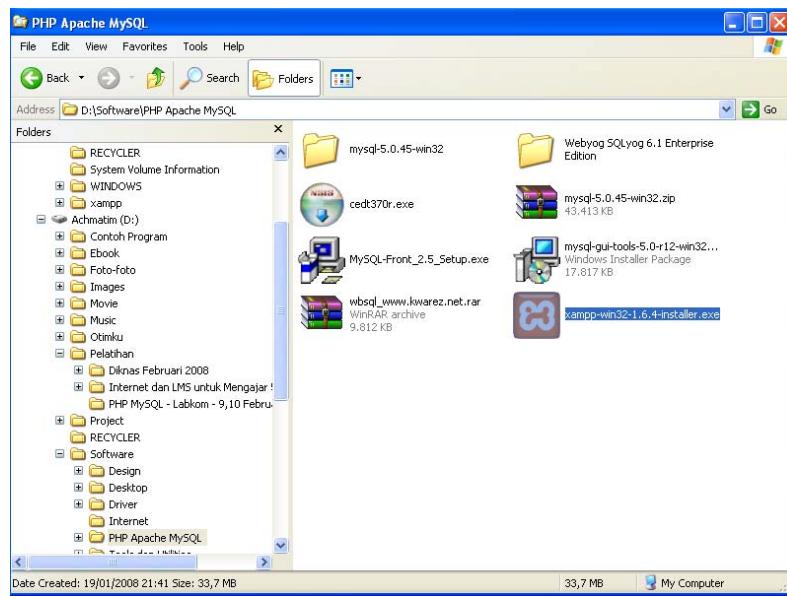
Persiapan Instalasi

1. Pastikan komputer Anda belum terinstall web server lain seperti IIS atau PWS karena dapat menyebabkan bentrok dengan web server Apache. Namun demikian, jika Anda masih tetap mempertahankan program tersebut, setelah instalasi selesai Anda dapat mengkonfigurasikan Apache secara manual dengan mengganti nomor port yang digunakan.

2. Download Source XAMPP versi stabil terbaru di <http://apachefriends.org>. Pilih versi sesuai dengan sistem operasi yang Anda gunakan (tersedia versi untuk Windows, Linux dan Mac).
3. Pastikan komputer yang Anda gunakan berjalan dengan baik dengan kapasitas memori dan hardisk yang masih mencukupi.

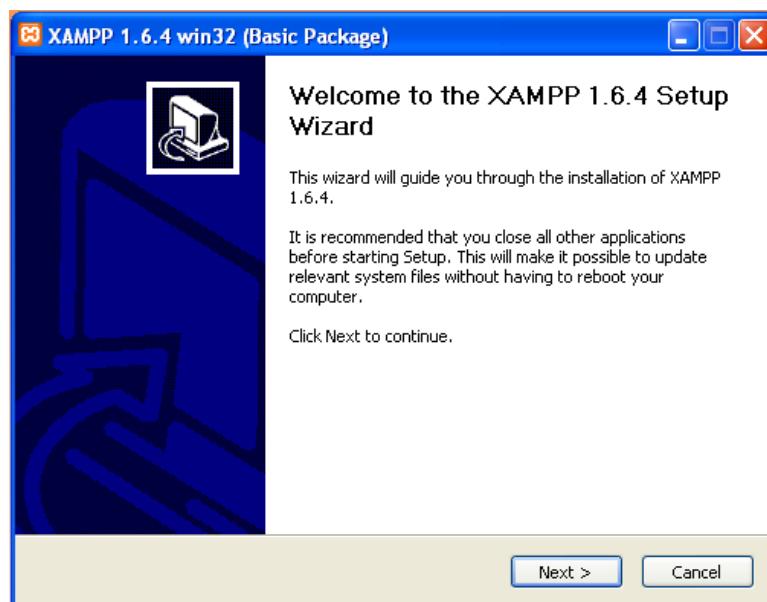
Proses Instalasi XAMPP

1. Jalankan file installer XAMPP **xampp-win32-x.x.x-installer.exe**.



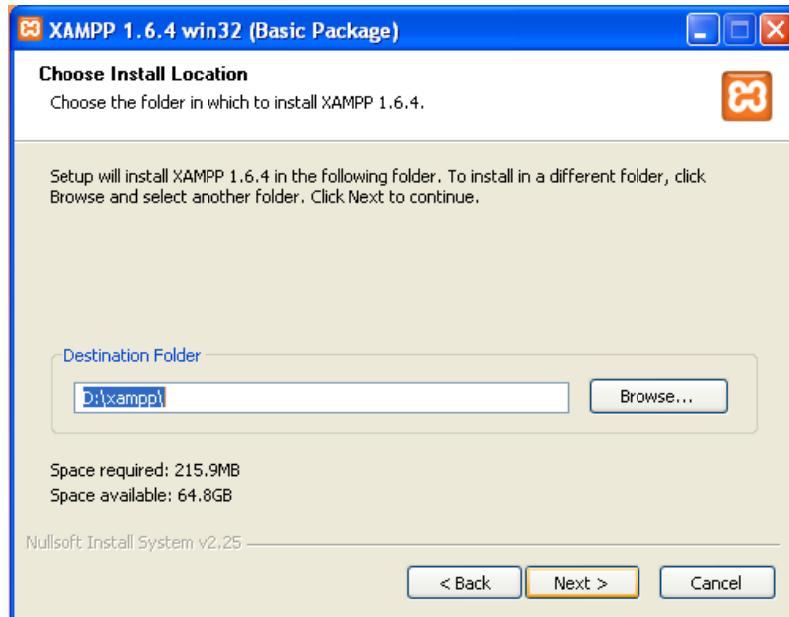
Gambar 1.1. File installer XAMPP

2. Akan ditampilkan window instalasi XAMPP. Pilih **Next >** untuk memulai proses instalasi.



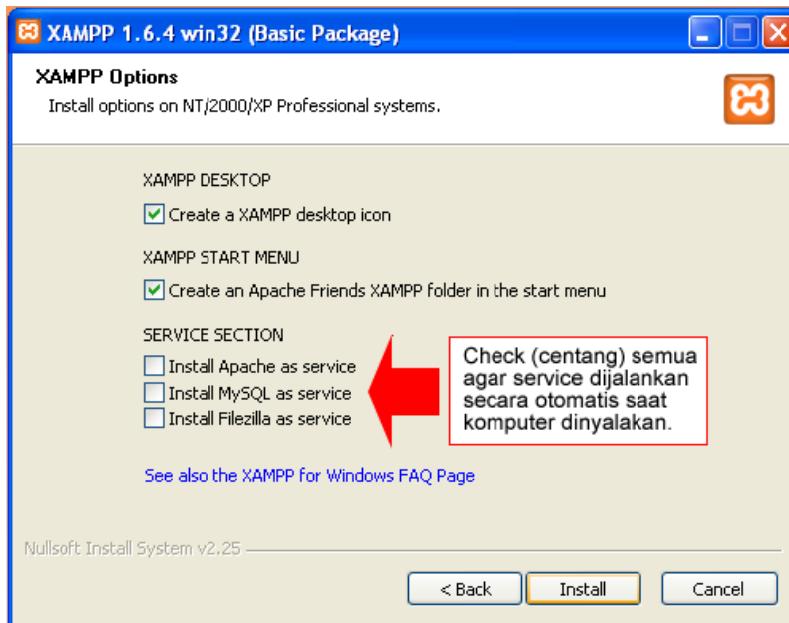
Gambar 1.2. Window Instalasi XAMPP

3. Akan ditampilkan window lokasi tujuan instalasi (*destination folder*), tentukan lokasi (folder) dimana XAMPP akan diinstall dengan menekan tombol **Browse..**. Pilih tombol **Next >** untuk melanjutkan proses instalasi.



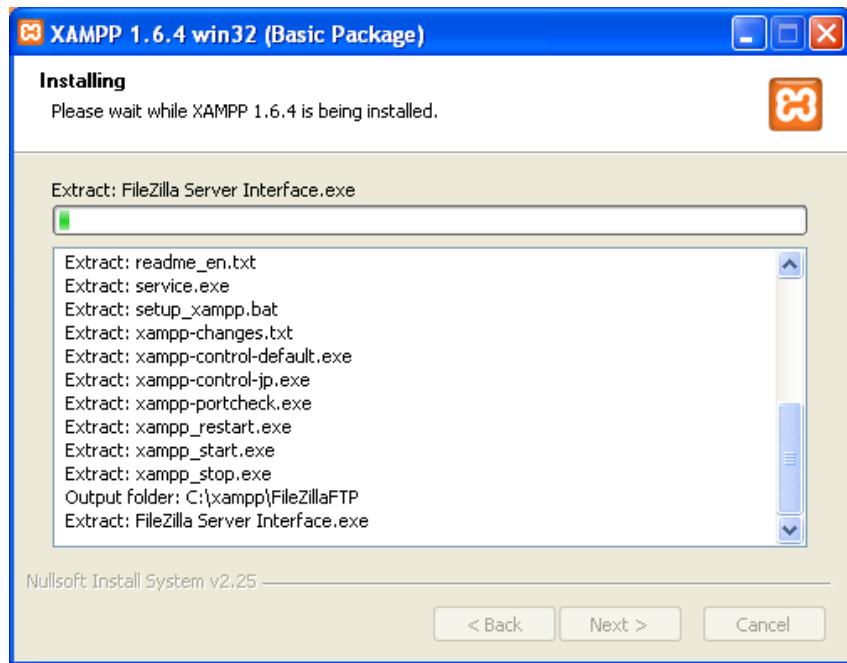
Gambar 1.3. Folder tujuan instalasi XAMPP

4. Selanjutnya akan ditampilkan window pilihan untuk kemudahan mengakses XAMPP nantinya. Tekan tombol **Install** untuk melanjutkan proses instalasi.



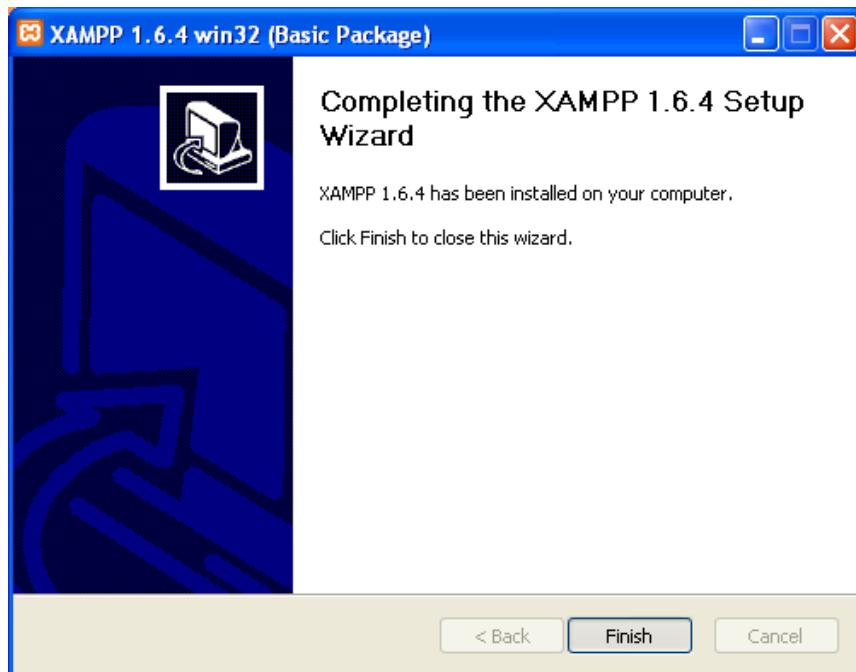
Gambar 1.4. Window pilihan XAMPP

5. Proses instalasi dimulai.



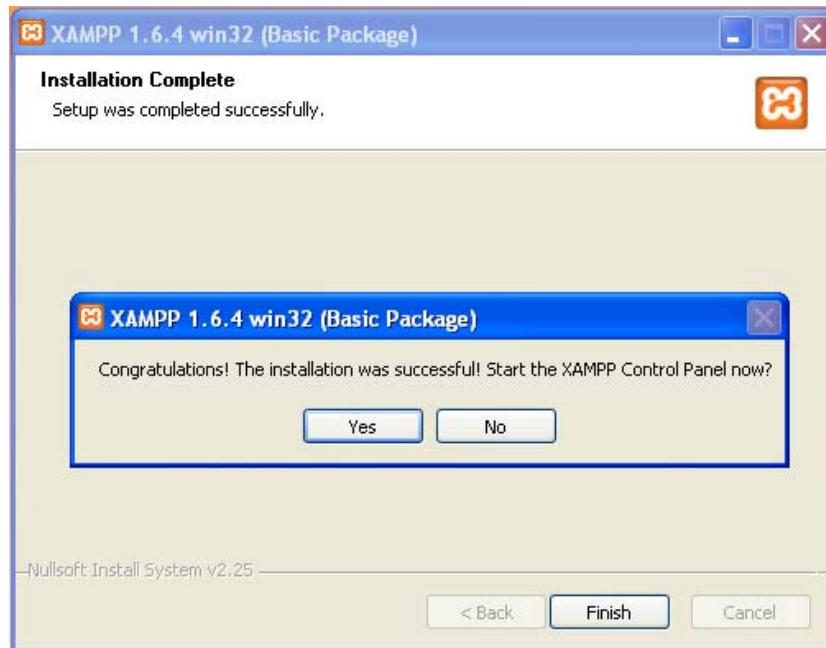
Gambar 1.5. Proses instalasi dimulai

6. Proses instalasi selesai. Klik **Finish** untuk menutup proses instalasi.



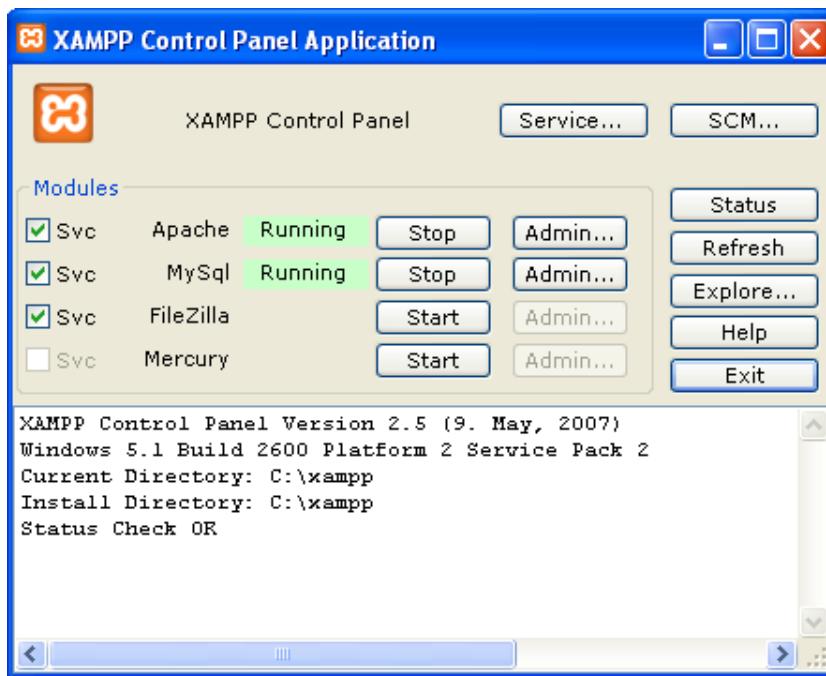
Gambar 1.6. Proses instalasi selesai

7. Jika proses instalasi berjalan dengan lancar maka akan ditampilkan window konfirmasi.



Gambar 1.7. Konfirmasi Instalasi Berhasil

8. Kita dapat mengatur XAMPP dengan mengakses XAMPP Control Panel yang dapat diakses melalui menu di Program Files atau icon di taskbar.



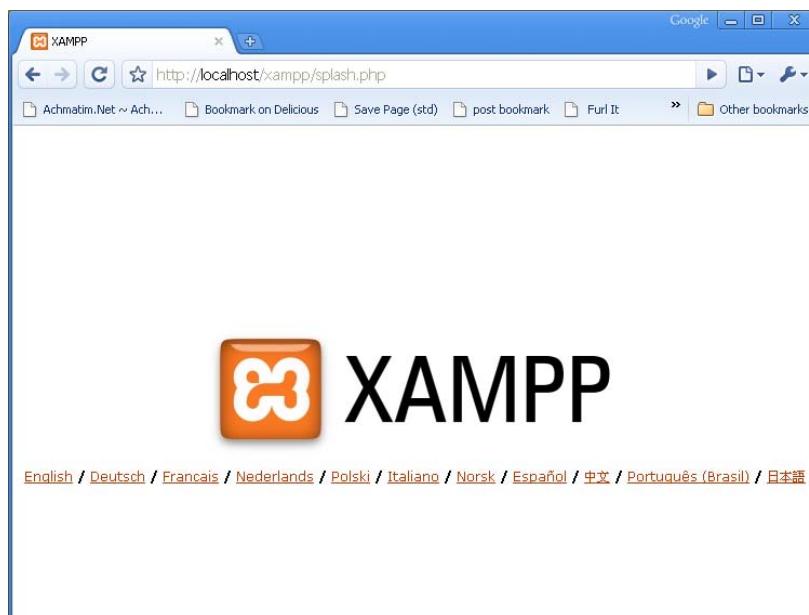
Gambar 1.8. XAMPP Control Panel

Test Instalasi XAMPP di Browser

Untuk memastikan bahwa XAMPP beserta PHP, Apache dan MySQL berjalan dengan baik, bukalah browser dan ketikkan URL sebagai berikut:

http://localhost

Jika semua sudah berjalan dengan baik, maka akan ditampilkan gambar sebagai berikut:



Gambar 1.9. Tampilan XAMPP di Browser

Pilih bahasa yang ingin digunakan dan akan ditampilkan halaman utama XAMPP sebagai berikut:



Gambar 1.9. Tampilan XAMPP di Browser

Bab 2

Struktur Dasar PHP

- ❖ Sebelum Belajar PHP
- ❖ Mari Mengenal PHP
- ❖ Say Hello to PHP
- ❖ Variabel
- ❖ Tipe Data
- ❖ Konstanta
- ❖ Operator dalam PHP
- ❖ Komentar Program

Sebelum Belajar PHP

1. Saya asumsikan Anda telah mengenal dengan cukup baik komputer Anda, dan juga bisa mengoperasikannya. Jadi Anda seharusnya sudah mengerti perbedaan file dan direktori, bagaimana menyimpan file atau document, bagaimana meng-copy file, dan seterusnya.
2. Saya asumsikan Anda telah mengerti bagaimana membuka dan menggunakan salah satu atau *salah semua* dari editor seperti Notepad, EditPlus, Crimson Editor, Dreamweaver, atau yang lainnya. Saya asumsikan juga, Anda telah mengerti bagaimana membuat, mengedit, dan menyimpan suatu file dengan editor tersebut.
3. Saya asumsikan bahwa Apache, PHP, dan MySQL di komputer yang Anda pakai sudah berjalan sebagaimana mestinya.
4. Saya asumsikan Anda telah mengerti bagaimana cara menggunakan Web Browser (IE, Mozilla, Opera, dsb)
5. Saya asumsikan Anda telah mengenal dan mengerti dengan cukup baik tentang tag-tag HTML dan juga bagaimana menggunakanannya. Anda tentu sudah mengambil matakuliah Pemrograman Web 1 atau Internet kan ?
6. Saya asumsikan bahwa penglihatan Anda masih normal sehingga bisa membedakan antara :
 - Tulisan **nama**, **Nama** dan **NAMA**
 - Koma (,), titik-koma (;) dan titik-dua (:)
 - Kutip-tunggal ('') dan kutip-ganda (")
 - Tulisan **namamhs**, **nama_mhs** dan **nama mhs**
7. Saya asumsikan Anda pernah menggunakan www.google.com

Mari Mengenal PHP

Tahukah Anda, PHP itu.....

1. Merupakan singkatan recursive dari **PHP** : **Hypertext Preprocessor**
2. Pertama kali dibuat oleh **Rasmus Lerdorf** pada tahun 1994.
3. Harus ditulis di antara tag :
 - <? dan ?>
 - <?php dan ?>
 - <script language="php"> dan </script>
 - <% dan %>
4. Setiap satu statement (perintah) biasanya diakhiri dengan titik-koma (;)
5. CASE SENSITIVE untuk nama identifier yang dibuat oleh user (berupa variable, konstanta, fungsi dll), namun TIDAK CASE SENSITIVE untuk identifier *built-in* dari PHP. Jadi :
 - \$nama ≠ \$Nama ≠ \$NAMA

- hitungLuas() ≠ HitungLuas()
 - echo = ECHO
 - while = WHILE
6. Mudah dipelajari.

Say Hello to PHP

1. Ketikkan perintah di bawah ini di editor :

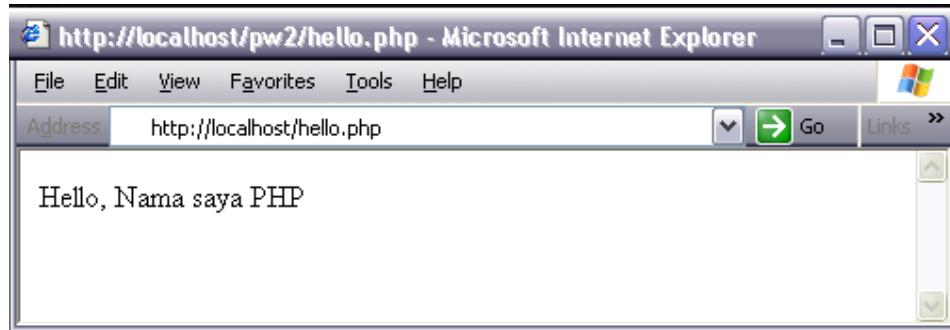
Program 2.1

Nama File : hello.php

Deskripsi : Program sederhana menampilkan sebuah string di layar.

```
1 <?php  
2 echo "Hello, Nama saya PHP ";  
3 ?>
```

2. Simpan dengan nama "**hello.php**" di *document-root* komputer Anda.
3. Buka browser dan ketikkan alamat <http://localhost/hello.php>
4. Lihat hasilnya di browser. Akan terlihat tampilan seperti pada gambar 2.1



Gambar 2.1. Tampilan Program 2.1

Variabel

- Digunakan untuk menyimpan sebuah value, data atau informasi
- Nama variabel diawali dengan tanda \$
- Panjang tidak terbatas
- Setelah tanda \$ diawali oleh huruf atau *under-score* (_). Karakter berikutnya bisa terdiri dari huruf, angka, dan karakter tertentu yang diperbolehkan (karakter ASCII dari 127 – 255).
- Bersifat case-sensitive.
- Tidak perlu dideklarasikan.
- Tidak boleh mengandung spasi.
- Contoh :

Benar

- \$_name
- \$first_name
- \$name3
- \$name_3

Salah

- \$3name
- \$name?
- \$first+name
- \$first.name
- \$first name

Program 2.2

Nama File : variabel.php

Deskripsi : Program mengisi dan menampilkan variabel di layar.

```
1 <?php
2 $nim = "0411500400";
3 $nama = 'Chotimatul Musyarofah';
4
5 echo "NIM : " . $nim . "<br>";
6 echo "Nama : $nama";
7 ?>
```



Gambar 2.2. Tampilan Program 2.2

Tipe Data

Pada PHP, tipe data variabel tidak didefinisikan oleh programmer, akan tetapi secara otomatis ditentukan oleh intepreter PHP. Namun demikian, PHP mendukung 8 (delapan) buah tipe data primitif, yaitu :

1. boolean
2. integer
3. float
4. string
5. array
6. object
7. resource
8. NULL

Program 2.3

Nama File : *tipe.php*

Deskripsi : Program contoh penanganan beberapa tipe data dasar dalam PHP.

```
1 <?php
2 $nim = "0411500400";
3 $nama = 'Chotimatul Musyarofah';
4 $umur = 23;
5 $nilai = 82.25;
6 $status = TRUE;
7
8 echo "NIM : " . $nim . "<br>";
9 echo "Nama : $nama<br>";
10 print "Umur : " . $umur; print "<br>";
11 printf ("Nilai : %.3f<br>", $nilai);
12 if ($status)
13     echo "Status : Aktif";
14 else
15     echo "Status : Tidak Aktif";
16 ?>
```



Gambar 2.3. Tampilan Program 2.3

Konstanta

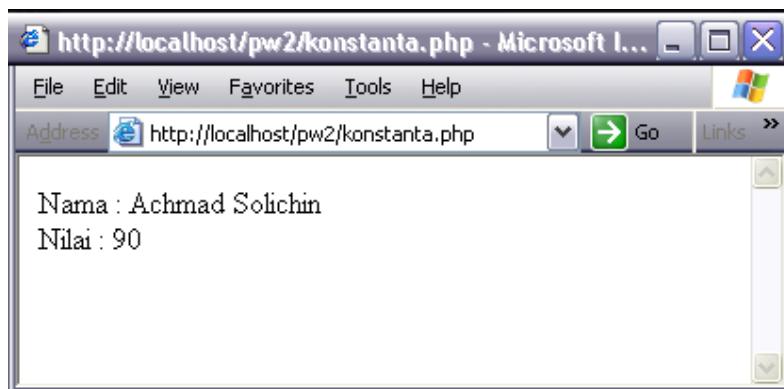
Konstanta merupakan variabel konstan yang nilainya tidak berubah-ubah. Untuk mendefinisikan konstanta dalam PHP, menggunakan fungsi define()

Program 2.4

Nama File : konstanta.php
Deskripsi : Program penggunaan konstanta dalam PHP.

```

1 <?
2 define ("NAMA", "Achmad Solichin");
3 define ("NILAI", 90);
4
5 //NAMA = "Muhammad"; //akan menyebabkan error
6 echo "Nama : " . NAMA;
7 echo "<br>Nilai : " . NILAI;
8 ?>
```



Gambar 2.4. Tampilan Program 2.4

Operator dalam PHP

Jenis Operator	Operator	Contoh	Keterangan
Aritmatika	+	\$a + \$b	Pertambahan
	-	\$a - \$b	Pengurangan
	*	\$a * \$b	Perkalian
	/	\$a / \$b	Pembagian
	%	\$a % \$b	Modulus, sisa pembagian

Penugasan	=	\$a = 4;	\$a diisi dengan 4
Bitwise	&	\$a & \$b	Bitwise AND
		\$a \$b	Bitwise OR
	^	\$a ^ \$b	Bitwise XOR
	~	~\$b	Bitwise NOT
	<<	\$a << \$b	Shift Left
	>>	\$a >> \$b	Shift Right
Perbandingan	==	\$a == \$b	Sama dengan
	==>	\$a ==> \$b	Identik
	!=	\$a != \$b	Tidak sama dengan
	<>	\$a <> \$b	Tidak sama dengan
	!==	\$a !== \$b	Tidak identik
	<	\$a < \$b	Kurang dari
	>	\$a > \$b	Lebih dari
	<=	\$a <= \$b	Kurang dari sama dengan
	>=	\$a >= \$b	Lebih dari sama dengan
Logika	and	\$a and \$b	TRUE jika \$a dan \$b TRUE
	&&	\$a && \$b	TRUE jika \$a dan \$b TRUE
	or	\$a or \$b	TRUE jika \$a atau \$b TRUE
		\$a \$b	TRUE jika \$a dan/atau \$b TRUE
	xor	\$a xor \$b	TRUE jika \$a atau \$b TRUE, tapi tidak keduanya
	!	!\$a	TRUE jika \$a FALSE
String	.	\$a . \$b	Penggabungan string \$a dan \$b

Program 2.5

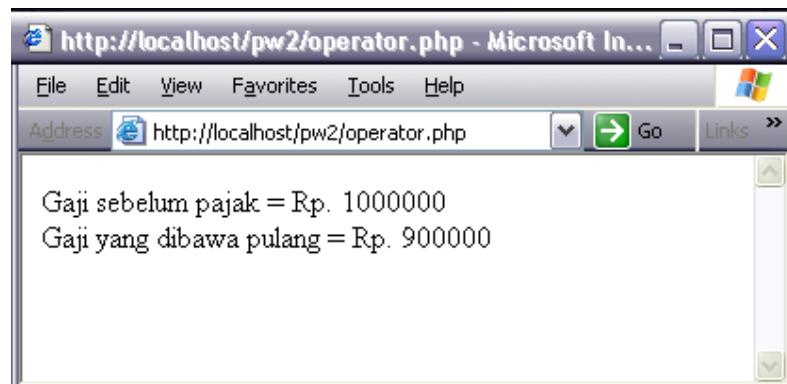
Nama File : operator.php

Deskripsi : Program beberapa Operator aritmatika dalam PHP.

```

1 <?
2 $gaji = 1000000;
3 $pajak = 0.1;
4 $thp = $gaji - ($gaji*$pajak);
5
6 echo "Gaji sebelum pajak = Rp. $gaji <br>";
7 echo "Gaji yang dibawa pulang = Rp. $thp";
8 ?>

```



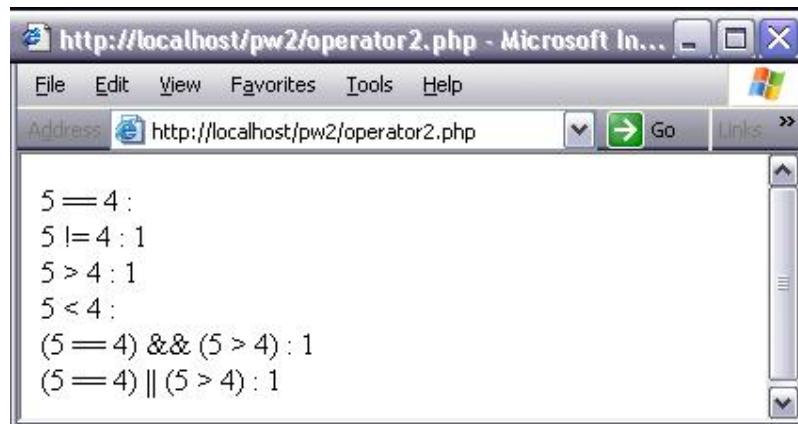
Gambar 2.5. Tampilan Program 2.5

Program 2.6

Nama File : operator2.php

Deskripsi : Program operator logika dan perbandingan dalam PHP.

```
1 <?
2 $a = 5;
3 $b = 4;
4
5 echo "$a == $b : ". ($a == $b);
6 echo "<br>$a != $b : ". ($a != $b);
7 echo "<br>$a > $b : ". ($a > $b);
8 echo "<br>$a < $b : ". ($a < $b);
9 echo "<br>($a == $b) && ($a > $b) : ". (($a != $b) && ($a > $b));
10 echo "<br>($a == $b) || ($a > $b) : ". (($a != $b) || ($a > $b));
11 ?>
```



Gambar 2.6. Tampilan Program 2.6

Komentar Program

Dalam PHP, komentar program bisa menggunakan :

- /* dan */
- // dan
- #

Program 2.7

Nama File : komentar.php

Deskripsi : Program cara memberikan komentar dalam program PHP.

```
1 <?
2 /* Ini komentar tidak akan tercetak di layar
3     yang ini juga komentar
4 */
5 // Nah kalo ini komentar satu baris
6
7 # Terus kalo ini juga komentar
8 echo "Ini akan tercetak di layar";
9
10 ?>
```



Gambar 2.7. Tampilan Program 2.7

Bab 3

Struktur Kondisi dan Perulangan

- ❖ Struktur Kondisi
 - + Struktur Kondisi If
 - + Struktur Kondisi If...Else
 - + Struktur Kondisi Khusus ? :
 - + Struktur Kondisi Switch...Case
- ❖ Struktur Perulangan
 - + Struktur Kondisi For
 - + Struktur Kondisi While
 - + Struktur Kondisi Do...While
 - + Struktur Kondisi Foreach
- ❖ Struktur Break dan Continue

Struktur Kondisi

Struktur Kondisi If

```
if (kondisi) {  
    statement-jika-kondisi-true;  
}
```

Keterangan :

- **kondisi** merupakan statemen atau variabel yang akan diperiksa TRUE atau FALSE-nya.

Struktur Kondisi If ... Else

```
if (kondisi) {  
    statement-jika-kondisi-true;  
} else {  
    statement-jika-kondisi-false;  
}
```

Keterangan :

- **kondisi** merupakan statemen atau variabel yang akan diperiksa TRUE atau FALSE-nya. Jika kondisinya TRUE maka statemen yang berada di blok if akan dieksekusi, sebaliknya jika kondisinya FALSE maka statemen yang berada di blok else yang akan dieksekusi.

Struktur Kondisi Khusus ? :

```
(kondisi) ? benar : salah;
```

Keterangan :

- **kondisi** merupakan statemen atau variabel yang akan diperiksa TRUE atau FALSE-nya. Statement pada blok benar dan salah hanya boleh satu statemen saja

Struktur Kondisi Switch ... Case

```
switch ($var) {  
    case '1' : statement-1; break;  
    case '2' : statement-2; break;  
    ....  
}
```

Keterangan :

- **\$var** merupakan variabel yang akan diperiksa isi atau nilainya. Tipe data variabel ini tidak dibatasi.
- Value pada **case** juga bisa berupa string, integer, boolean, bahkan bisa berupa conditional-statement. Boleh memakai kutip tunggal maupun kutip ganda.

Program 3.1

Nama File : if.php
Deskripsi : Program Struktur Kondisi If.

```
1 <?php
2 $nilai = 80;
3 if ($nilai >= 60) {
4     echo "Nilai Anda $nilai, Anda LULUS";
5 }
6 ?>
```

Penjelasan Program 3.1

Pada program di atas, terdapat pemeriksaan kondisi apakah isi variabel **\$nilai** lebih dari 60 (baris 3). Jika kondisi ini bernilai **TRUE** (variabel \$nilai berisi nilai yang lebih besar dari 60) maka *statement* pada baris ke-4 akan dijalankan. Sebaliknya jika kondisinya **FALSE**, maka *statement* pada baris ke-4 tidak akan dijalankan.



Gambar 3.1. Tampilan Program 3.1

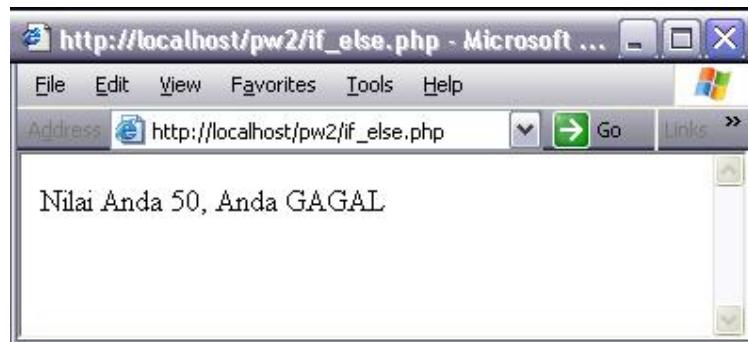
Program 3.2

Nama File : if_else.php
Deskripsi : Program Struktur Kondisi If..Else.

```
1 <?php
2 $nilai = 50;
3 if ($nilai >= 60) {
4     echo "Nilai Anda $nilai, Anda LULUS";
5 } else {
6     echo "Nilai Anda $nilai, Anda GAGAL";
7 }
8 ?>
```

Penjelasan Program 3.2

Pada program di atas, terdapat pemeriksaan kondisi apakah isi variabel **\$nilai** lebih dari 60 (baris 3). Jika kondisi ini bernilai **TRUE** (variabel \$nilai berisi nilai yang lebih besar dari 60) maka statement pada baris ke-4 akan dijalankan. Sebaliknya jika kondisinya **FALSE**, maka statement pada baris ke-6 yang akan dijalankan. Pada contoh program di atas, kondisi pada baris ke-3 bernilai **FALSE** karena isi variabel \$nilai adalah 50.



Gambar 3.2. Tampilan Program 3.2

Program 3.3

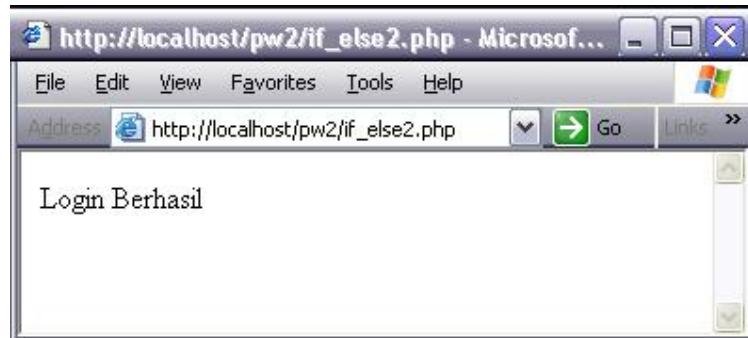
Nama File : if_else2.php

Deskripsi : Program Memeriksa username dan password dengan If..Else.

```
1 <?php
2 $user = "achmatim";
3 $pass = "123"
4 if ($user == "achmatim" && $pass == "123") {
5     echo "Login Berhasil";
6 } else {
7     echo "Login Gagal";
8 }
9 ?>
```

Penjelasan Program 3.3

Pada program di atas, terdapat pemeriksaan kondisi apakah isi variabel **\$nilai** sama dengan "achmatim" **dan** apakah isi variabel **\$pass** sama dengan "123" (baris 4). Jika kedua kondisi tersebut bernilai **TRUE** maka statement pada baris ke-5 akan dijalankan. Sebaliknya jika salah satunya bernilai **FALSE**, maka statement pada baris ke-7 yang akan dijalankan.



Gambar 3.3. Tampilan Program 3.3

Program 3.4

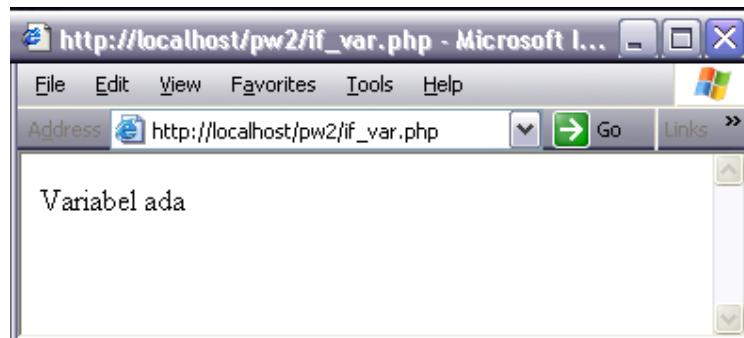
Nama File : if_var.php

Deskripsi : Program Memeriksa suatu variabel ada atau tidak.

```
1 <?php
2 $user="";
3 if (!isset($user)) {
4     echo "Variabel tidak ada/belum terbentuk";
5 } else {
6     echo "Variabel ada";
7 }
8 ?>
```

Penjelasan Program 3.4

Pada program di atas, terdapat fungsi **isset()** yang merupakan fungsi untuk memeriksa apakah suatu variabel sudah terbentuk (ada) atau belum. Fungsi akan bernilai **TRUE** jika variabelnya ada. Jadi jika baris ke-2 dari program di atas di-*remark* atau dihilangkan maka *statement* pada baris ke-4 yang akan dijalankan.



Gambar 3.4. Tampilan Program 3.4

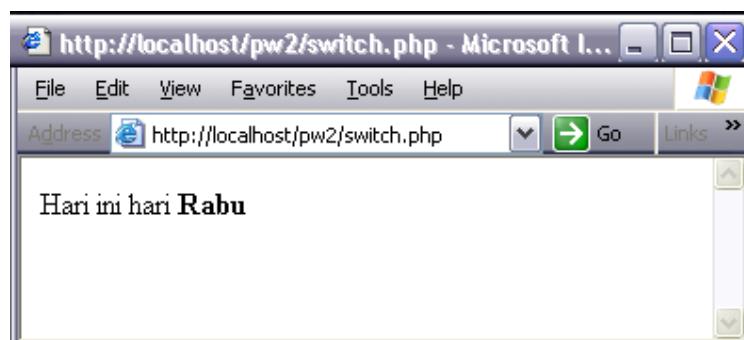
Program 3.5

Nama File : switch.php
Deskripsi : Program Struktur Switch..Case untuk menampilkan nama hari dalam bahasa Indonesia.

```
1 <?php
2 $day = date ("D");
3 switch ($day) {
4     case 'Sun' : $hari = "Minggu"; break;
5     case 'Mon' : $hari = "Senin"; break;
6     case 'Tue' : $hari = "Selasa"; break;
7     case 'Wed' : $hari = "Rabu"; break;
8     case 'Thu' : $hari = "Kamis"; break;
9     case 'Fri' : $hari = "Jum'at"; break;
10    case 'Sat' : $hari = "Sabtu"; break;
11    default : $hari = "Kiamat";
12 }
13 echo "Hari ini hari <b>$hari</b>";
14 ?>
```

Penjelasan Program 3.5

Pada program di atas, variabel **\$day** (baris ke-2) akan berisi 3 (tiga) digit pertama dari nama hari dalam bahasa Inggris. Dengan fungsi **date()**, kita akan memperoleh informasi tanggal, hari dan waktu sekarang. Fungsi ini akan diperlajari lebih lanjut pada Bab mendatang. Selanjutnya isi variabel **\$day** akan diperiksa dengan switch (baris ke-3), jika isinya adalah 'Sun' maka **\$hari** adalah "Minggu" dan seterusnya.



Gambar 3.5. Tampilan Program 3.5

Program 3.6

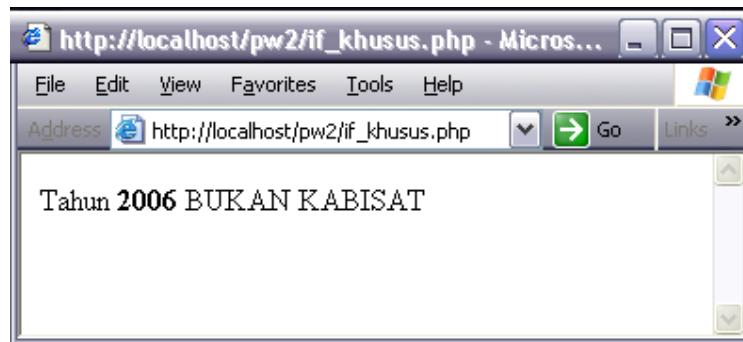
Nama File : if_khusus.php

Deskripsi : Program Struktur Kondisi Khusus ? untuk memeriksa tahun kabisat.

```
1 <?php
2 $tahun = date ("Y");
3 $kabisat = ($tahun%4 == 0) ? "KABISAT" : "BUKAN KABISAT";
4 echo "Tahun <b>$tahun</b> $kabisat";
5 ?>
```

Penjelasan Program 3.6

Pada program di atas, fungsi **date()** dengan parameter "Y" (baris ke-2) akan menghasilkan 4 digit tahun sekarang. Pada baris ke-3 akan diperiksa apakah isi variabel \$tahun jika dimodulus dengan 4 akan menghasilkan nilai 0. Jika **TRUE** maka \$kabisat akan berisi "KABISAT" dan jika **FALSE** \$kabisat akan berisi "BUKAN KABISAT"



Gambar 3.6. Tampilan Program 3.6

Struktur Perulangan

Struktur Perulangan For

```
for (init_awal, kondisi, counter) {
    statement-yang-diulang;
}
```

Keterangan :

- **init_awal** merupakan inisialisasi atau nilai awal variable.
- **kondisi** merupakan statemen kondisi yang akan membatasi perulangan.
- **counter** merupakan pertambahan atau pengurangan nilai variabel sehingga perulangan tetap berjalan.

Struktur Perulangan While

```
init_awal;
while (kondisi) {
    statement-yang-diulang;
    counter;
}
```

Keterangan :

- **init_awal** merupakan inisialisasi atau nilai awal variable.
- **kondisi** merupakan statemen kondisi yang akan membatasi perulangan.
- **counter** merupakan pertambahan atau pengurangan nilai variabel sehingga perulangan tetap berjalan.

Struktur Perulangan Do ... while

```
init_awal;
do {
    statement-yang-diulang;
    counter;
} while (kondisi);
```

Keterangan :

- **init_awal** merupakan inisialisasi atau nilai awal variable.
- **kondisi** merupakan statemen kondisi yang akan membatasi perulangan.
- **counter** merupakan pertambahan atau pengurangan nilai variabel sehingga perulangan tetap berjalan.
- Pada struktur do...while, pemeriksaan kondisi ada di bawah, sehingga statement yang berada dalam block do...while setidaknya akan dieksekusi sebanyak satu kali.

Struktur Perulangan Foreach

```
foreach (array_expression as $value)
statement;

foreach (array_expression as $key => $value)
statement;
```

Keterangan :

- Struktur foreach biasanya digunakan untuk melakukan perulangan berdasarkan isi suatu array. Perulangan akan berakhir jika isi array telah habis.

Program 3.7

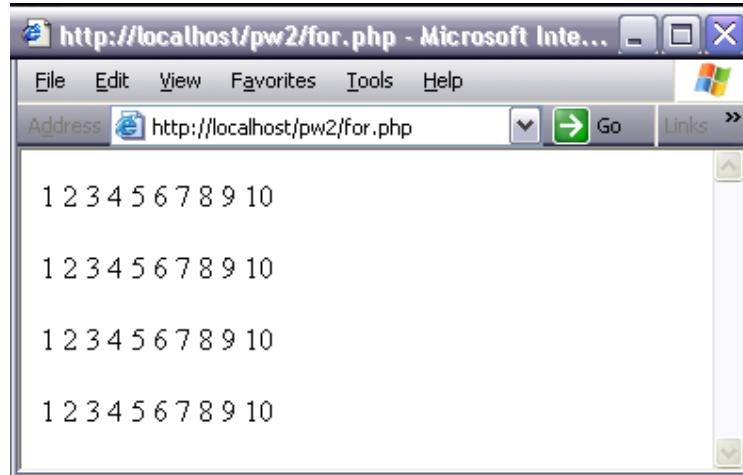
Nama File : for.php

Deskripsi : Program Struktur Perulangan For dan beberapa variasinya.

```
1 <?php
2
3 /* contoh 1 */
4 for ($i = 1; $i <= 10; $i++) {
5     echo "$i ";
6 }
7 echo "<br><br>";
8
9 /* contoh 2 */
10 for ($i = 1; ; $i++) {
11     if ($i > 10) {
12         break;
13     }
14     echo "$i ";
15 }
16 echo "<br><br>";
17
18 /* contoh 3 */
19 $i = 1;
20 for (; ; ) {
21     if ($i > 10) {
22         break;
23     }
24     echo "$i ";
25     $i++;
26 } echo "<br><br>";
27
28 /* contoh 4 */
29 for ($i = 1; $i <= 10; print "$i ", $i++);
30
31 ?>
```

Penjelasan Program 3.7

Program di atas merupakan bentuk-bentuk perulangan dengan menggunakan for. Contoh yang pertama (baris 4-6) merupakan bentuk yang paling umum. Pada contoh 2, batas akhir perulangan tidak disebutkan dalam for, tapi diatur dengan if dan **break** (baris 11-13).



Gambar 3.7. Tampilan Program 3.7

Program 3.8

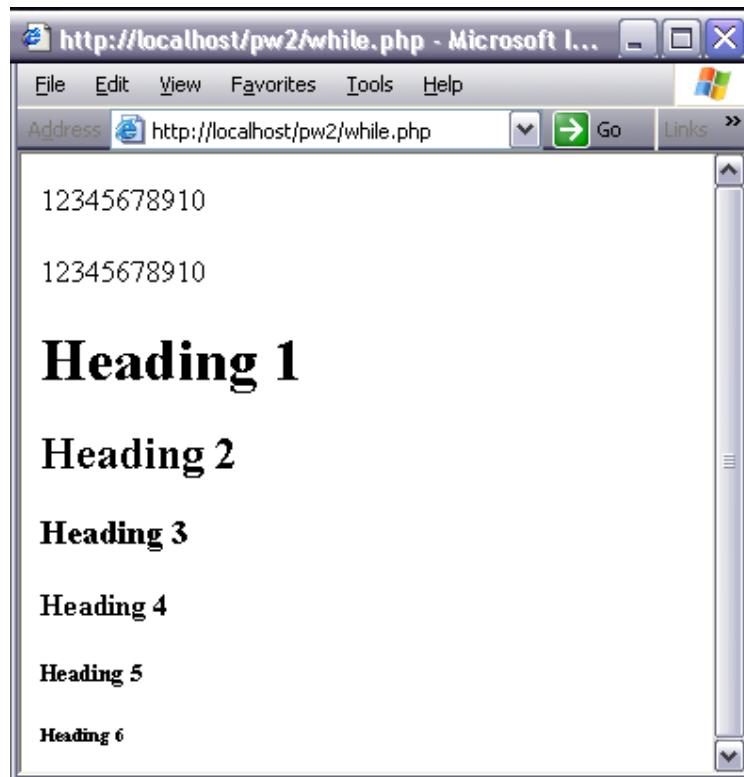
Nama File : while.php

Deskripsi : Program Struktur Perulangan While dan beberapa variasinya.

```
1 <?php
2 /* contoh 1 */
3 $i = 1;
4 while ($i <= 10) {
5     echo $i++;
6 }
7 echo "<br><br>";
8
9 /* contoh 2 */
10 $i = 1;
11 while ($i <= 10):
12     echo "$i";
13     $i++;
14 endwhile;
15 echo "<br><br>";
16
17 /* contoh 3 */
18 $i = 1;
19 while ($i <= 6) {
20     echo "<h$i>Heading $i</h$i>";
21     $i++;
22 }
23 ?>
```

Penjelasan Program 3.8

Program di atas merupakan beberapa variasi perulangan dengan while. Contoh yang pertama (baris 3-6) merupakan bentuk yang paling umum dipakai. Blok perintah (baris 5) akan dijalankan selama pemeriksaan kondisi pada baris ke-4 bernilai **TRUE**. Contoh kedua merupakan bentuk lain dari while, bentuk ini jarang digunakan. Pada contoh ketiga, penggunaan perulangan untuk menampilkan tulisan dengan format <H1> sampai <H6>



Gambar 3.8. Tampilan Program 3.8

Program 3.9

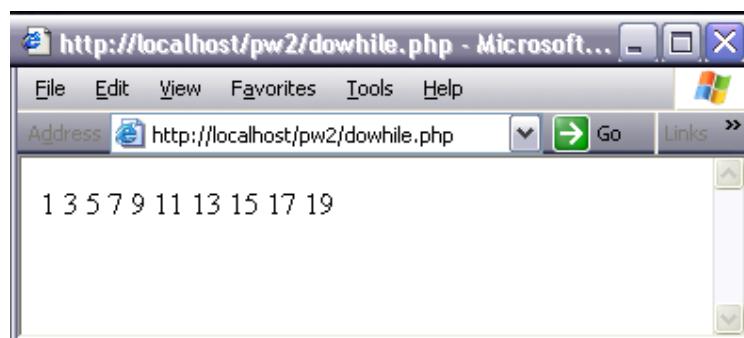
Nama File : dowhile.php

Deskripsi : Program Struktur Perulangan dengan Do...While.

```
1 <?php
2
3 $i = 1;
4 do {
5     echo "$i ";
6     $i+=2;
7 } while ($i <= 20);
8
9 ?>
```

Penjelasan Program 3.9

Program di atas akan menampilkan bilangan ganjil antara 1 sampai 20 menggunakan struktur perulangan **do..while**. Pada struktur perulangan do...while, pemeriksaan kondisi berada di bawah.



Gambar 3.9. Tampilan Program 3.9

Struktur Break dan Continue

Program 3.10

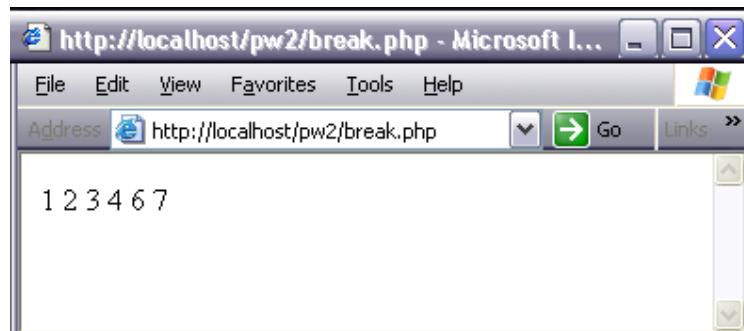
Nama File : break.php

Deskripsi : Program Struktur Break dan Continue.

```
1 <?php
2
3 for ($i=1; $i<10; $i++) {
4     if ($i == 5)
5         continue;
6     if ($i == 8)
7         break;
8     echo "$i ";
9 }
10 ?>
```

Penjelasan Program 3.10

Dari program di atas, dapat disimpulkan bahwa perintah **continue** akan melanjutkan proses perulangan tanpa melewati (meng-eksekusi) baris perintah di bawahnya. Jadi perintah pada baris 6-8 akan dilewati (5 tidak akan tercetak) Sedangkan dengan perintah **break** akan menyebabkan program menghentikan perulangan (langsung keluar dari perulangan)



Gambar 3.10. Tampilan Program 3.10

Bab 4

Penanganan Form

- ❖ Berbagai Cara Penanganan Form
- ❖ Form Input Type TEXT dan PASSWORD
- ❖ Form Input Type RADIO
- ❖ Form Input Type CHECKBOX
- ❖ Form Input Type COMBO BOX
- ❖ Form Input Type TEXTAREA

Form inputan dibuat dengan tag-tag HTML. Halaman yang mengandung form murni (tidak ada *script php*) tidak harus disimpan dalam bentuk php, bisa dalam bentuk html.

Untuk merancang sebuah form inputan, setidaknya ada 3 (tiga) hal penting, yi :

1. METHOD

Method dari sebuah form menentukan bagaimana data inputan form dikirim. Method ini ada dua macam, yaitu GET dan POST. Method ini menentukan bagaimana data inputan dikirim dan diproses oleh PHP.

2. ACTION

Action dari sebuah form menentukan dimana data inputan dari form diproses. Jika action ini dikosongkan, maka dianggap proses form terjadi di halaman yang sama. Jadi halaman form dan halaman proses bisa saja dipisah atau dijadikan satu.

3. SUBMIT BUTTON

Submit button merupakan sebuah tombol (pada umumnya) yang berfungsi sebagai *trigger* pengiriman data dari form inputan. Jika tombol ini ditekan, maka data form akan dikirimkan (diproses) di halaman yang sudah ditentukan pada atribut action.

Berbagai Cara Penanganan Form

Cara 1 : Menyatukan antara Form dan Proses

Proses pengolahan form dilakukan di halaman yang sama dengan form inputannya. Jika proses penanganan form berada di satu halaman, maka *value* atribut **action** pada tag **form** tidak perlu diisi (dikosongkan).

Program 4.1

Nama File : *input01.php*

Deskripsi : Program Contoh pengolahan form dimana antara form inputan dan proses pengolahan inputan berada dalam satu halaman.

```

1 <html>
2   <head><title>Pengolahan Form</title></head>
3   <body>
4     <FORM ACTION="" METHOD="POST" NAME="input">
5       Nama Anda : <input type="text" name="nama"><br>
6       <input type="submit" name="Input" value="Input">
7     </FORM>
8   </body>
9 </html>
10
11 <?php
12 if (isset($_POST['Input'])) {
13   $nama = $_POST['nama'];
14   echo "Nama Anda : <b>$nama</b>";
15 }
16 ?>
```

Penjelasan Program 4.1

Beberapa hal yang perlu diperhatikan dari program 4.1 di atas, di antaranya mengenai nama setiap komponen form karena nama ini akan menjadi index array dalam PHP. Pada program 4.1 di atas, *value* atribut **action** pada tag **form** tidak diisi (baris 4), ini berarti bahwa proses pengolahan form berada di halaman yang sama. Selanjutnya (baris 4) method yang digunakan dalam penanganan form adalah POST. Cara ini lebih disarankan dalam penanganan sebuah form inputan.

Mulai baris ke-11 hingga 16, terdapat script / program PHP yang akan menangani (mengolah) nilai yang diinputkan melalui form. Letak proses inputan ini boleh sebelum atau sesudah form, tergantung kebutuhan. Pada baris ke-12, terdapat pemeriksaan kondisi apakah tombol dengan nama "**Input**" (perhatikan index array **\$_POST** dan bandingkan dengan name tombol **submit** pada form) benar-benar ditekan atau tidak oleh user. Selanjutnya pada baris ke-13, nilai inputan dari form akan diambil dengan cara mengakses array **\$_POST** sesuai dengan komponen form yang akan diambil. Pada baris 13 ini, isi komponen inputan dengan nama "**nama**" akan diambil dan dimasukkan ke variabel **\$nama**. Index array **\$_POST** pada baris 13 harus sama dengan *value* atribut **name** pada baris ke-5.



Gambar 4.1. Tampilan Program 4.1

Cara 2 : Memisahkan antara Form dan Proses

Proses pengolahan form dilakukan di halaman yang terpisah dengan form inputannya. Jika proses penanganan form berada dilakukan di halaman yang berbeda, maka *value* atribut **action** pada tag **form** harus diisi dengan alamat halaman tempat proses pengolahan form.

Program 4.2

Nama File : *input02.php*

Deskripsi : Program menampilkan form inputan dengan method POST

```

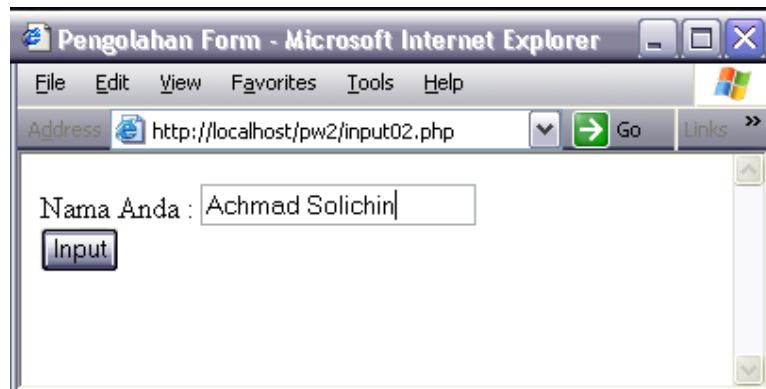
1 <html>
2   <head><title>Pengolahan Form</title></head>
3   <body>
4     <FORM ACTION="proses02.php" METHOD="POST" NAME="input">
5       Nama Anda : <input type="text" name="nama"><br>
6       <input type="submit" name="Input" value="Input">
7     </FORM>
8   </body>
9 </html>
  
```

Program 4.3

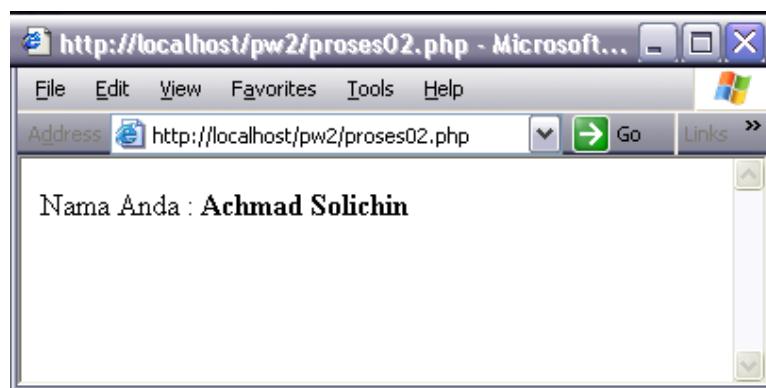
Nama File : proses02.php

Deskripsi : Program penanganan form inputan untuk Program 4.2.

```
1 <?php
2 if (isset($_POST['Input'])) {
3     $nama = $_POST['nama'];
4     echo "Nama Anda : <b>$nama</b>";
5 }
6 ?>
```



Gambar 4.2. Tampilan Program 4.2



Gambar 4.3. Tampilan Program 4.3

Program 4.4

Nama File : input03.php

Deskripsi : Program menampilkan form inputan dengan method GET

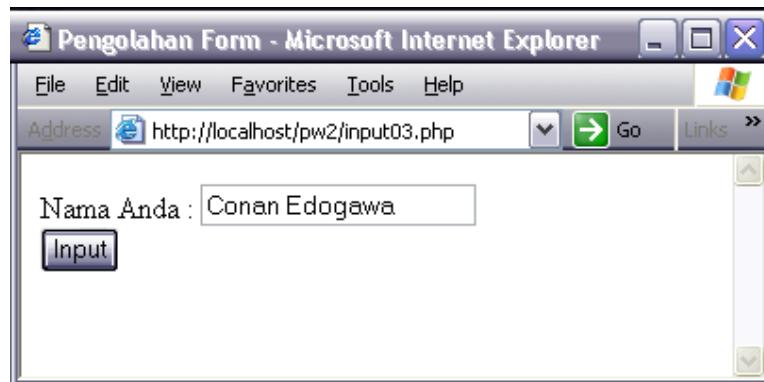
```
1 <html>
2     <head><title>Pengolahan Form</title></head>
3     <body>
4         <FORM ACTION="proses03.php" METHOD="GET" NAME="input">
5             Nama Anda : <input type="text" name="nama"><br>
6             <input type="submit" name="Input" value="Input">
7         </FORM>
8     </body>
9 </html>
```

Program 4.5

Nama File : proses03.php

Deskripsi : Program penanganan form inputan untuk Program 4.4.

```
1 <?php
2 if (isset($_GET['Input'])) {
3     $nama = $_GET['nama'];
4     echo "Nama Anda : <b>$nama</b>";
5 }
6 ?>
```



Gambar 4.4. Tampilan Program 4.4



Gambar 4.5. Tampilan Program 4.5

Penjelasan Program 4.2, 4.3, 4.4 dan 4.5

Program 4.2 dan program 4.3 pada prinsipnya sama dengan program 4.1. Hasilnya pun sama. Akan tetapi, proses penanganan form (PHP) terpisah dengan tampilan form inputannya (HTML). Pada program 4.2 dan 4.3, penanganan form menggunakan method POST, sedang pada program 4.4 dan 4.5, penanganan form dilakukan dengan method GET. Perhatikan, method ini menentukan bagaimana sebuah form diproses dan bagaimana variabel inputan diambil. Bandingkan baris ke-2 dan ke-3 pada program 4.3 dan program 4.5 !

Form Input Type TEXT dan PASSWORD

Program 4.6

Nama File : input04.php

Deskripsi : Program menampilkan form inputan text dalam jumlah banyak.

```
1 <html>
2     <head><title>Pengolahan Form ~ Text</title></head>
```

```
3 <body>
4     <FORM ACTION="proses04.php" METHOD="POST" NAME="input">
5         Sahabat-sahabat Dekatku<br>
6         <input type="text" name="nama1"><br>
7         <input type="text" name="nama2"><br>
8         <input type="text" name="nama3"><br>
9         <input type="text" name="nama4"><br>
10        <input type="submit" name="Input" value="Input">
11    </FORM>
12  </body>
13 </html>
```

Program 4.7

Nama File : proses04.php

Deskripsi : Program penanganan form inputan untuk Program 4.6.

```
1 <?php
2 if (isset($_POST['Input'])) {
3     $nama1 = $_POST['nama1'];
4     $nama2 = $_POST['nama2'];
5     $nama3 = $_POST['nama3'];
6     $nama4 = $_POST['nama4'];
7     echo "<b>Nama Sahabat-sahabat Dekatku :</b> <br>";
8     echo $nama1. "<br>";
9     echo $nama2. "<br>";
10    echo $nama3. "<br>";
11    echo $nama4. "<br>";
12 }
13 ?>
```

Penjelasan Program 4.6 dan 4.7

Baris 3-6 program 4.7 merupakan proses pengambilan nilai dari masing-masing form inputan text yang terdapat pada program 4.6. Selanjutnya variabel ini ditampilkan di layar (baris 8-11, program 4.7).



Gambar 4.6. Tampilan Program 4.6



Gambar 4.7. Tampilan Program 4.7

Program 4.8

Nama File : *input05.php*

Deskripsi : Program menampilkan form login (inputan text dan password).

```
1 <html>
2   <head><title>Login Here</title></head>
3   <body>
4     <FORM ACTION="proses05.php" METHOD="POST" NAME="input">
5       <h2>Login Here...</h2>
6       Username : <input type="text" name="username"><br>
7       Password : <input type="password" name="password"><br>
8       <input type="submit" name="Login" value="Login">
9       <input type="reset" name="reset" value="Reset">
10      </FORM>
11    </body>
12  </html>
```

Program 4.9

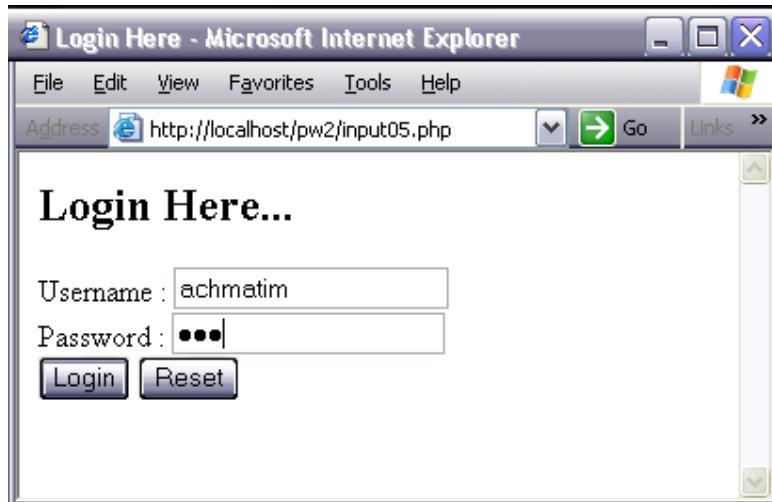
Nama File : *proses05.php*

Deskripsi : Program sederhana untuk memeriksa inputan username dan password pada program 4.8

```
1 <?php
2 if (isset($_POST['Login'])) {
3   $user = $_POST['username'];
4   $pass = $_POST['password'];
5   if ($user == "achmatim" && $pass == "123") {
6     echo "<h2>Login Berhasil</h2>";
7   } else {
8     echo "<h2>Login Gagal</h2>";
9   }
10 }
11 ?>
```

Penjelasan Program 4.8 dan 4.9

Program 4.8 akan menampilkan form login sederhana yang terdiri dari inputan **username** dan **password**. Selanjutnya nilai yang diinput akan diproses di program 4.9. Jika username dan password diinput dengan benar maka akan ditampilkan pesan berhasil (lihat gambar) dan jika login salah maka akan ditampilkan pesan kesalahan (lihat gambar).



Gambar 4.8. Tampilan Program 4.8



Gambar 4.9. Tampilan Program 4.9 Jika Login Berhasil



Gambar 4.10. Tampilan Program 4.9 Jika Login Gagal

Form Input Type RADIO

Pada inputan jenis radio button, user hanya bisa memilih satu pilihan di antara beberapa pilihan.

Program 4.10

Nama File : input06.php

Deskripsi : Program menampilkan form pilihan jurusan dengan radio button

```
1 <html>
2   <head><title>Pilih Jurusan</title></head>
3   <body>
4     <FORM ACTION="proses06.php" METHOD="POST" NAME="input">
5       <h2>Pilih Jurusan Anda :</h2>
```

```
6      <input type="radio" name="jurusan" value="TI" checked>
7      Teknik Informatika<br>
8      <input type="radio" name="jurusan" value="SI"> Sistem
9      Informasi<br>
10     <input type="radio" name="jurusan" value="SK"> Sistem
11     Komputer<br>
12     <input type="radio" name="jurusan" value="KA">
13     Komputerisasi Akuntansi<br>
14     <input type="submit" name="Pilih" value="Pilih">
15   </FORM>
16 </body>
17 </html>
```

Program 4.11

Nama File : proses06.php
Deskripsi : Program untuk mengambil dan menampilkan jurusan yang dipilih pada program 4.10

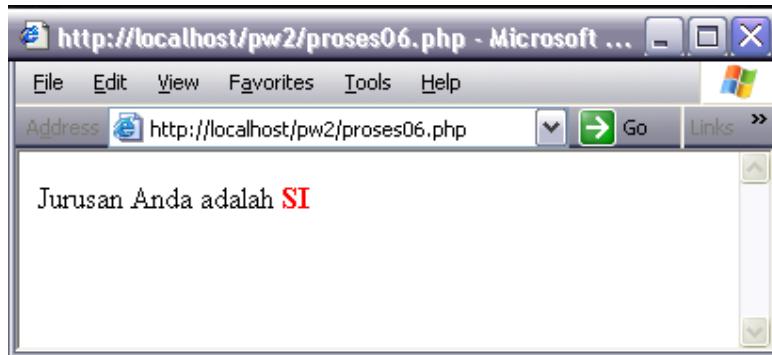
```
1 <?php
2 if (isset($_POST['Pilih'])) {
3     $jurusan = $_POST['jurusan'];
4     echo "Jurusan Anda adalah
5         <b><font color='red'>$jurusan</font></b>";
6 }
7 ?>
```

Penjelasan Program 4.10 dan 4.11

Program 4.10 akan menampilkan form pilihan inputan jurusan dengan radio button (lihat gambar). Pada form inputan jenis radio button, name dari masing-masing radio button harus sama, akan tetapi value-nya harus dibedakan. Perhatikan program 4.10 baris 6-9 ! Untuk mengambil nilai (value) dari form jenis radio, bisa langsung dengan mengakses name dari form tersebut. Perhatikan program 4.11 baris ke-3 !



Gambar 4.11. Tampilan Program 4.10



Gambar 4.12. Tampilan Program 4.11

Form Input Type CHECK BOX

Pada form inputan jenis check box, user dimungkinkan memilih lebih dari satu pilihan.

Program 4.12

Nama File : *input07.php*
Deskripsi : *Program menampilkan form inputan nama band favorit dengan check box.*

```
1 <html>
2   <head><title>Band Favorit ~ Inputan Checkbox</title></head>
3   <body>
4     <FORM ACTION="proses07.php" METHOD="POST" NAME="input">
5       <h2>Pilih Band Favorit Anda :</h2>
6       <input type="checkbox" name="band01" value="Padi" checked> Padi<br>
7       <input type="checkbox" name="band02" value="Sheila On 7"> Sheila On 7<br>
8       <input type="checkbox" name="band03" value="Dewa 19"> Dewa 19<br>
9       <input type="checkbox" name="band04" value="Ungu"> Ungu<br>
10      <input type="submit" name="Pilih" value="Pilih">
11    </FORM>
12  </body>
13 </html>
```

Program 4.13

Nama File : *proses07.php*
Deskripsi : *Program untuk menampilkan nama band favorit sesuai dengan inputan pada program 4.12*

```
1 <?php
2 if (isset($_POST['Pilih'])) {
3   echo "Band Favorit Anda adalah :<br>";
4   if (isset($_POST['band01'])) {
5     echo "+" . $_POST['band01'] . "<br>";
6   }
7   if (isset($_POST['band02'])) {
8     echo "+" . $_POST['band02'] . "<br>";
9   }
10  if (isset($_POST['band03'])) {
11    echo "+" . $_POST['band03'] . "<br>";
12  }
13  if (isset($_POST['band04'])) {
14    echo "+" . $_POST['band04'] . "<br>";
```

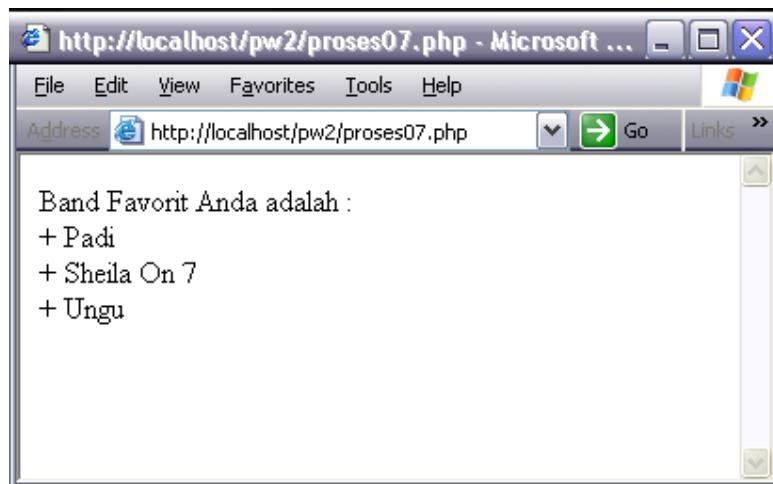
```
15 }  
16 }  
17 ?>
```

Penjelasan Program 4.12 dan 4.13

Program 4.12 akan menampilkan form pilihan inputan check box band favorit (lihat gambar). Pada form inputan jenis check box, name dari masing-masing check box harus dibedakan. User dapat memilih lebih dari satu pilihan. Perhatikan program 4.12 baris 6-9 ! Untuk mengambil nilai (value) dari form jenis check box, sebaiknya diperiksa terlebih dahulu apakah check box dipilih atau tidak, dengan menggunakan fungsi **isset()**. Perhatikan program 4.13 baris ke-4. Jika check box di-cek (dipilih) maka ambil *value* dari check box tersebut (baris ke-5)



Gambar 4.13. Tampilan Program 4.12



Gambar 4.14. Tampilan Program 4.13

Form Input Type COMBO BOX

Program 4.14

Nama File : input08.php
Deskripsi : Program menampilkan form inputan film kartun favorit dengan combo box.

```

1 <html>
2   <head><title>Film Kartun Favorit ~ Inputan Combo
3   box</title></head>
4   <body>
5     <FORM ACTION="proses08.php" METHOD="POST" NAME="input">
6       <h2>Pilih Film Kartun Favorit Anda :</h2>
7       <select name="kartun">
8         <option value="Sponge Bob">Sponge Bob</option>
9         <option value="Sinchan">Sinchan</option>
10        <option value="Conan">Conan</option>
11        <option value="Doraemon">Doraemon</option>
12        <option value="Dragon Ball">Dragon Ball</option>
13        <option value="Naruto">Naruto</option>
14      </select>
15      <input type="submit" name="Pilih" value="Pilih">
16    </body>
17 </html>

```

Program 4.15

Nama File : proses08.php

Deskripsi : Program untuk menampilkan nama film kartun favorit sesuai dengan inputan pada program 4.14

```

1 <?php
2 if (isset($_POST['Pilih'])) {
3   $film = $_POST['kartun'];
4   echo "Film Kartun Favorit Anda adalah :
5       <font color=blue><b>$film</b></font>";
6 }
7 ?>

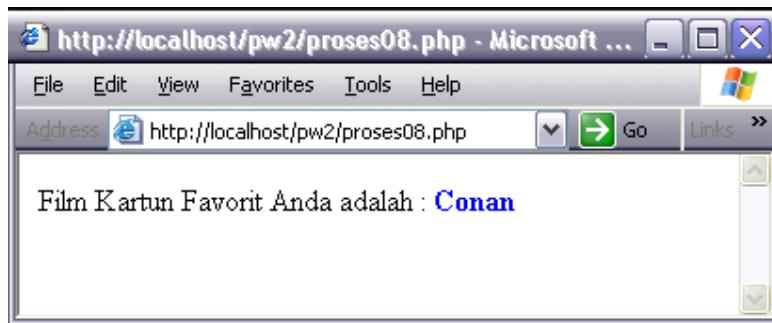
```

Penjelasan Program 4.14 dan 4.15

Program 4.14 akan menampilkan form pilihan inputan *combo box* film kartun favorit (lihat gambar). Untuk membuat inputan jenis *combo box*, bisa menggunakan tag *<select>* dan *<option>*. Pada form inputan jenis *check box*, *name* diletakkan pada tag *<select>*. User hanya dapat memilih satu pilihan dari sejumlah pilihan yang ditampilkan dalam bentuk *drop down list*. Lihat gambar. Untuk mengambil nilai (*value*) dari form jenis *combo box*, dapat langsung mengaksesnya sesuai dengan *name*-nya. Perhatikan program 4.15 baris ke-3 !



Gambar 4.15. Tampilan Program 4.14



Gambar 4.16. Tampilan Program 4.15

Form Input Type TEXTAREA

Program 4.16

*Nama File : input09.php
Deskripsi : Program menampilkan form inputan kritik dan saran dengan text area.*

```
1 <html>
2   <head><title>Kritik dan Saran ~ Inputan
3 Textarea</title></head>
4   <body>
5     <FORM ACTION="proses09.php" METHOD="POST" NAME="input">
6       <h2>Input Kritik / Saran :</h2>
7       <textarea name="saran" cols="40"
8 rows="5"></textarea><br>
9       <input type="submit" name="Proses" value="Input
10 Saran">
11     </FORM>
12   </body>
13 </html>
```

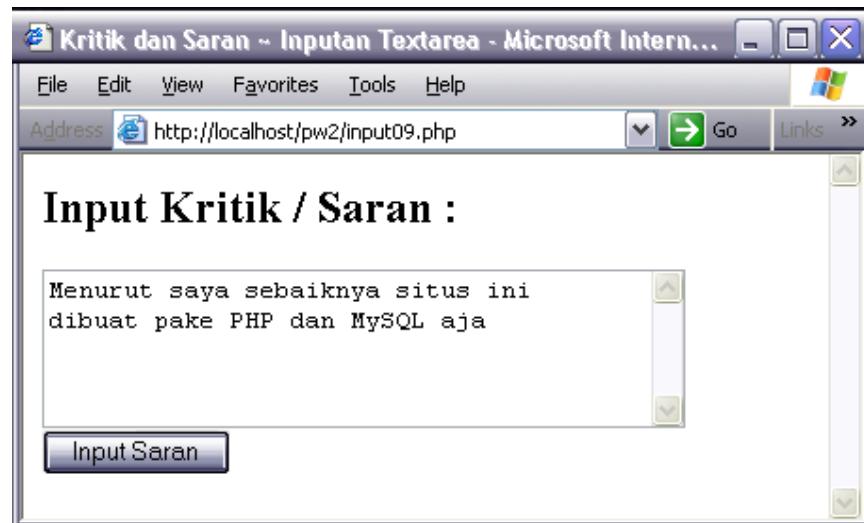
Program 4.17

*Nama File : proses09.php
Deskripsi : Program untuk menampilkan isi kritik / saran sesuai dengan inputan text area pada program 4.16*

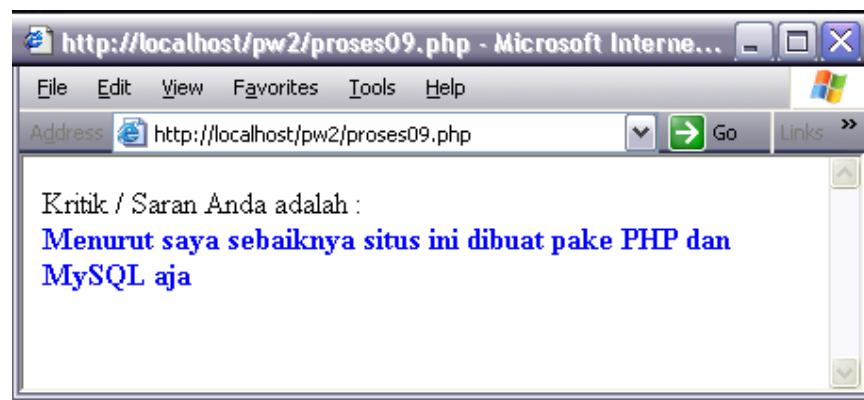
```
1 <?php
2 if (isset($_POST['Proses'])) {
3   $saran = nl2br($_POST['saran']);
4   echo "Kritik / Saran Anda adalah : <br>";
5   echo "<font color=blue><b>$saran</b></font>";
6 }
7 ?>
```

Penjelasan Program 4.16 dan 4.17

Program 4.16 akan menampilkan form kritik saran menggunakan text area (lihat gambar). Untuk membuat inputan jenis combo box, bisa menggunakan tag <textarea>. Berbeda dengan inputan type text yang hanya bisa diinput satu baris, pada text area, bisa diinput lebih dari satu baris. Lihat gambar. Untuk mengambil nilai (value) dari form jenis textarea, dapat langsung mengaksesnya sesuai dengan name-nya. Perhatikan program 4.17 baris ke-3 !



Gambar 4.17. Tampilan Program 4.16



Gambar 4.18. Tampilan Program 4.17

Bab 5

Array dan Fungsi

- ❖ Pengantar Array
- ❖ Mendeklarasikan dan Menampilkan Array
- ❖ Fungsi-fungsi Array dalam PHP
- ❖ Fungsi dalam PHP

Pengantar Array

Array merupakan tipe data terstruktur yang berguna untuk menyimpan sejumlah data yang bertipe sama. Bagian yang menyusun array disebut elemen array, yang masing-masing elemen dapat diakses tersendiri melalui index array. Index array dapat berupa bilangan integer atau string.

Mendeklarasikan dan Menampilkan Array

Program 5.1

Nama File : array01.php

Deskripsi : Program mendeklarasikan array dengan index numeric.

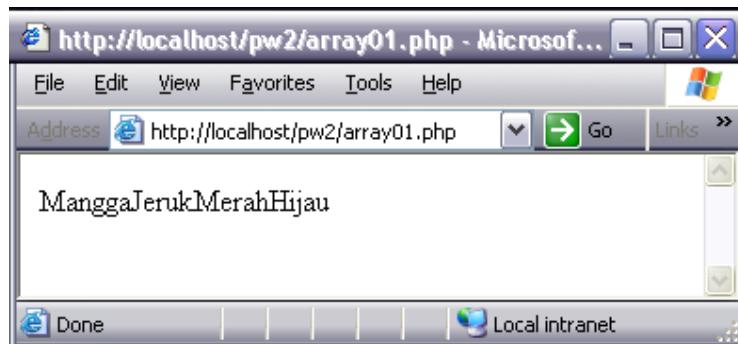
```

1 <?php
2
3 $arrBuah = array ("Mangga", "Apel", "Pisang", "Jeruk");
4 echo $arrBuah[0]; //Mangga
5 echo $arrBuah[3]; //Jeruk
6
7 $arrwarna = array();
8 $arrwarna[] = "Merah";
9 $arrwarna[] = "Biru";
10 $arrwarna[] = "Hijau";
11 $arrwarna[] = "Putih";
12 echo $arrwarna[0]; //Merah
13 echo $arrwarna[2]; //Hijau
14
15 ?>

```

Penjelasan Program 5.1

Untuk mendeklarasikan atau mendefinisikan sebuah array di PHP bisa menggunakan keyword **array()**. Pada program 5.1, terdapat pendeklarasian dan pendefinisian array pada baris ke-3. Jumlah elemen array tidak perlu disebutkan saat deklarasi. Sedangkan untuk menampilkan isi array pada elemen tertentu, cukup dengan menyebutkan nama array beserta index array-nya (lihat baris ke-4 dan ke-5). Pendeklarasian dan pengisian array juga bisa dilakukan seperti pada baris ke-7 sampai baris ke-11. Dengan pendefinisian array seperti pada dua cara di atas, index (key) array secara otomatis akan diisi dengan bilangan bulat integer yang dimulai dari 0.



Gambar 5.1. Tampilan Program 5.1

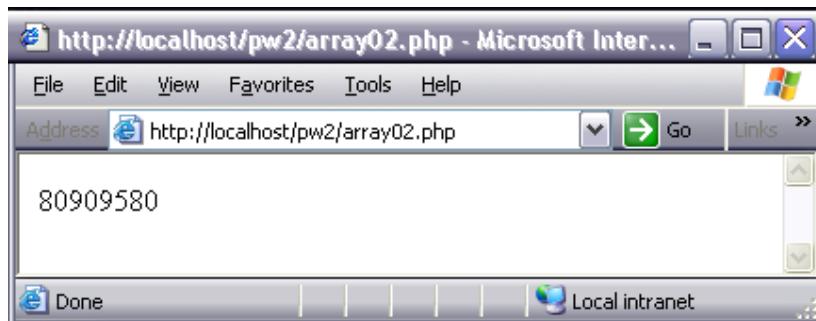
Program 5.2

Nama File : array02.php
Deskripsi : Program mendeklarasikan array dengan index string (array assosiatif).

```
1 <?php
2
3 $arrNilai = array ("Ani" => 80, "Otim" => 90, "Ana" => 75,
4 "Budi" => 85);
5 echo $arrNilai['Ani'];           //80
6 echo $arrNilai['Otim'];         //90
7
8 $arrNilai = array();
9 $arrNilai['Ami'] = 80;
10 $arrNilai['Asma'] = 95;
11 $arrNilai['Sri'] = 77;
12 echo $arrNilai['Asma'];        //95
13 echo $arrNilai['Ami'];         //80
14 ?>
```

Penjelasan Program 5.2

Untuk mendeklarasikan atau mendefinisikan sebuah array *assosiatif* (array yang menggunakan index selain integer), dapat dengan cara menyebutkan indexnya terlebih dahulu diikuti operator => dan diikuti value atau nilai elemennya. Lihat program baris ke-3 ! Sedangkan untuk menampilkan value atau nilai dari suatu array, bisa dengan menyebutkan nama array yang diikuti index arraynya (lihat baris ke-4 dan ke-5). Pendefinisan array *assosiatif* juga bisa dilakukan seperti pada baris ke-7 sampai baris ke-10



Gambar 5.2. Tampilan Program 5.2

Program 5.3

Nama File : array03.php
Deskripsi : Program menampilkan seluruh isi array dengan index numerik.

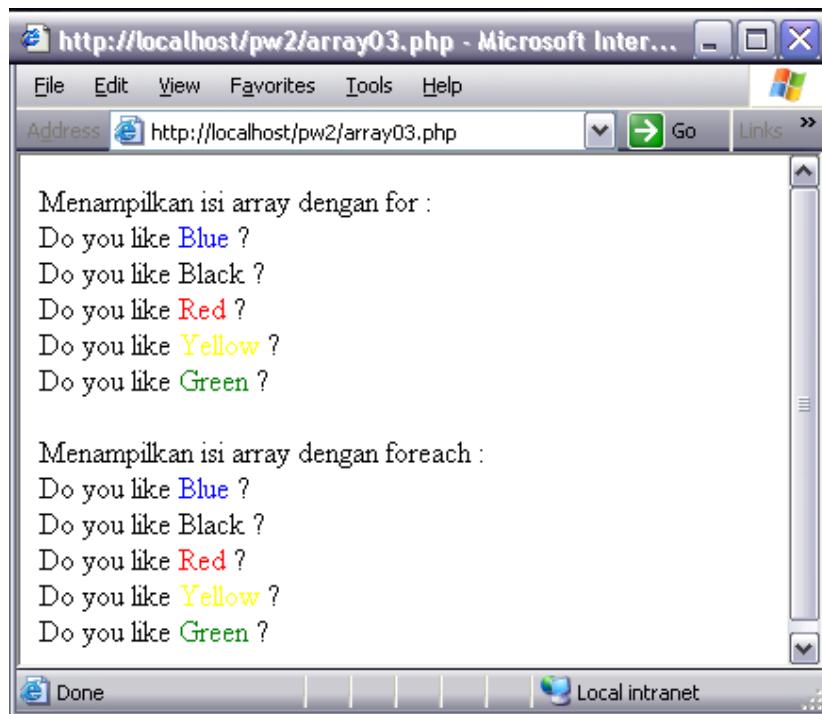
```

1 <?php
2 $arrwarna = array ("Blue","Black","Red","Yellow","Green");
3
4 echo "Menampilkan isi array dengan for : <br>";
5 for ($i=0; $i<count($arrwarna); $i++) {
6     echo "Do you like <font color=$arrwarna[$i]>". $arrwarna[$i]
7     . "</font> ?<br>";
8 }
9 echo "<br>Menampilkan isi array dengan foreach : <br>";
10 foreach ($arrwarna as $warna) {
11     echo "Do you like <font color=$warna>". $warna . "</font>
12 ?<br>";
13 ?>

```

Penjelasan Program 5.3

Untuk menampilkan keseluruhan isi array bisa dilakukan dengan perulangan for, while, do..while atau foreach. Baris 5 – 7 program di atas merupakan contoh menampilkan array dengan perulangan for. Fungsi **count()** pada baris ke-5 berfungsi untuk mencari jumlah elemen dari array \$assWarna. Untuk menampilkan array juga bisa menggunakan struktur **foreach()**. Struktur foreach akan mengambil elemen array mulai dari posisi pointer sampai akhir pointer. Pada contoh program di atas (baris ke-10) bisa dinyatakan dengan pernyataan bahwa “untuk setiap elemen array \$arrWarna, masukkan ke variabel \$warna”. Tampilan program 5.3 bisa dilihat pada gambar 5.3



Gambar 5.3. Tampilan Program 5.3

Program 5.4

Nama File : array04.php

Deskripsi : Program menampilkan seluruh isi array assosiatif.

```

1 <?php
2 $arrNilai = array ("Ani" => 80, "Otim" => 90, "Ana" => 75,
3 "Budi" => 85);

```

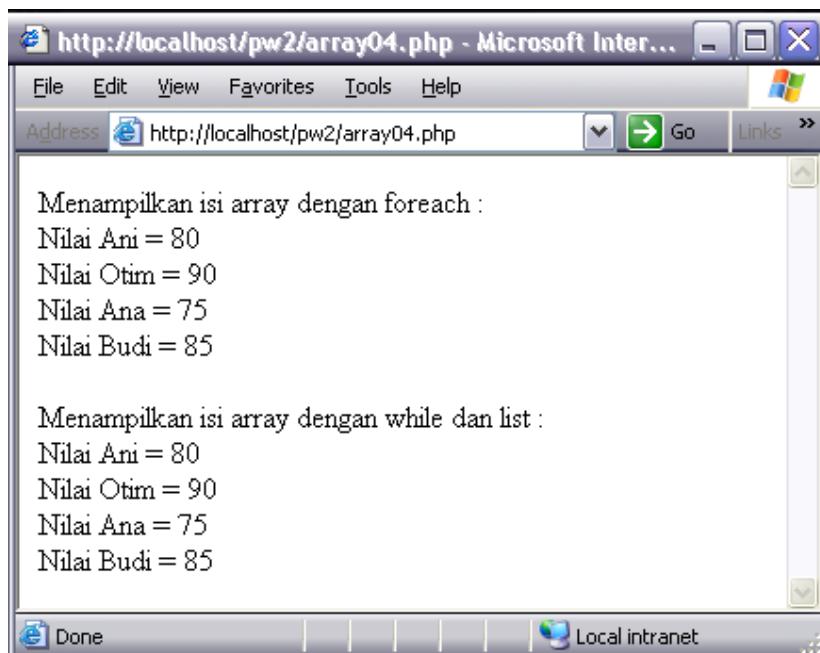
```

3 echo "Menampilkan isi array dengan foreach : <br>";
4 foreach ($arrNilai as $nama => $nilai) {
5     echo "Nilai $nama = $nilai<br>";
6 }
7 reset ($arrNilai);
8 echo "<br>Menampilkan isi array dengan while dan list : <br>";
9 while (list ($nama, $nilai) = each($arrNilai)) {
10    echo "Nilai $nama = $nilai<br>";
11 }
12
13
14 ?>

```

Penjelasan Program 5.4

Untuk menampilkan array *assosiatif* bisa menggunakan perulangan foreach dan while dengan fungsi list() dan each(). Baris ke-5 dari program 5.4. di atas bisa dinyatakan dengan pernyataan "untuk setiap elemen array \$arrNilai, masukkan key/index-nya ke variabel \$nama dan masukkan value-nya ke variabel \$nilai". Pada baris ke-8 terdapat fungsi **reset()** yang berfungsi mengembalikan pointer array ke elemen pertama. Pada baris ke-10, array assosiatif ditampilkan dengan struktur while yang menggunakan fungsi **list()** dan **each()**.



Gambar 5.4. Tampilan Program 5.4

Program 5.5

Nama File : array05.php

Deskripsi : Program mencetak struktur array.

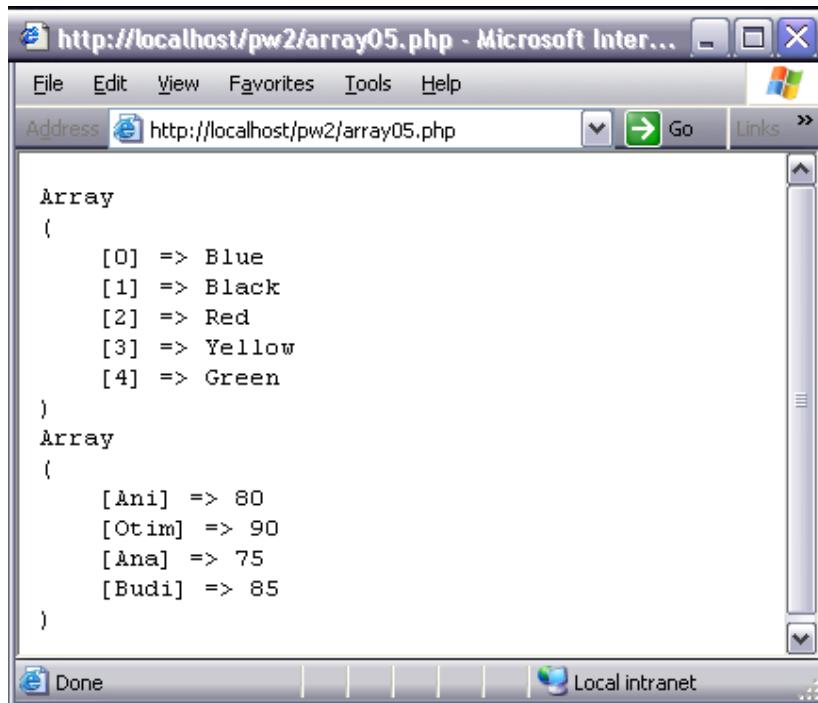
```

1 <?
2 $arrWarna = array ("Blue","Black","Red","Yellow","Green");
3 $arrNilai = array ("Ani" => 80, "Otim" => 90, "Ana" => 75,
4 "Budi" => 85);
5 echo "<pre>";
6 print_r ($arrwarna);
7 echo "<br>";
8 print_r ($arrNilai);
9 echo "</pre>";
?>

```

Penjelasan Program 5.5

Pada program 5.5 di atas terdapat fungsi **print_r()** (baris ke-5) yang berfungsi untuk menampilkan struktur dari array. Tampilan program 5.5 bisa dilihat pada gambar 5.5



The screenshot shows a Microsoft Internet Explorer window with the title bar "http://localhost/pw2/array05.php - Microsoft Internet Explorer". The address bar contains "http://localhost/pw2/array05.php". The main content area displays the output of the print_r() function:

```
Array
(
    [0] => Blue
    [1] => Black
    [2] => Red
    [3] => Yellow
    [4] => Green
)
Array
(
    [Ani] => 80
    [Otim] => 90
    [Ana] => 75
    [Budi] => 85
)
```

Gambar 5.5. Tampilan Program 5.5

Fungsi-fungsi Array dalam PHP

PHP menyediakan lebih dari 70 fungsi untuk manipulasi array. Fungsi-fungsi array dalam PHP bisa dilihat di alamat <http://ca.php.net/manual/en/ref.array.php>.

Fungsi Pengurutan Array

- **arsort()** – Pengurutan array berdasarkan **value** secara **descending**
- **asort()** – Pengurutan array berdasarkan **value** secara **ascending**
- **krsort()** - Pengurutan array berdasarkan **index/key** secara **descending**
- **ksort()** - Pengurutan array berdasarkan **index/key** secara **ascending**
- **rsort()** - Pengurutan array berdasarkan **value** secara **descending** dengan mengubah **index/key**
- **sort()** - Pengurutan array berdasarkan **value** secara **ascending** dengan mengubah **index/key**
- **shuffle()** – Random pengurutan array

Fungsi Pengaturan Pointer Array

- **current()** – Mendapatkan elemen array yang ditunjuk oleh pointer
- **end()** – Pointer menunjuk pada elemen array terakhir
- **key()** – Mendapatkan key yang ditunjuk oleh pointer
- **next()** – Pointer menunjuk pada elemen selanjutnya
- **prev()** – Pointer menunjuk pada elemen sebelumnya
- **reset()** – Memindahkan pointer array ke awal (elemen pertama)
- **count()** – Menghitung jumlah elemen array

Fungsi Pencarian pada Array

- **array_search()** – Mencari posisi (key) dari suatu value dalam array
- **array_key_exists()** – Memeriksa suatu key ada dalam array atau tidak
- **in_array()** – Memeriksa suatu elemen ada dalam array atau tidak

Program 5.6

Nama File : array06.php

Deskripsi : Program mengurutkan array dengan sort() dan rsort().

```
1 <?php
2 $arrNilai = array ("Ani" => 80, "Otim" => 90, "Sri" => 75,
3 "Budi" => 85);
4 echo "<b>Array sebelum pengurutan</b>";
5 echo "<pre>";
6 print_r($arrNilai);
7 echo "</pre>";
8 sort($arrNilai);
9 reset($arrNilai);
10 echo "<b>Array setelah pengurutan dengan sort()</b>";
11 echo "<pre>";
12 print_r($arrNilai);
13 echo "</pre>";
14
15 rsort($arrNilai);
16 reset($arrNilai);
17 echo "<b>Array setelah pengurutan dengan rsort()</b>";
18 echo "<pre>";
19 print_r($arrNilai);
20 echo "</pre>";
21 ?>
```

Program 5.7

Nama File : array07.php

Deskripsi : Program mengurutkan array dengan asort() dan arsort().

```
1 <?php
2 $arrNilai = array ("Ani" => 80, "Otim" => 90, "Sri" => 75,
3 "Budi" => 85);
4 echo "<b>Array sebelum pengurutan</b>";
5 echo "<pre>";
6 print_r($arrNilai);
7 echo "</pre>";
8 asort($arrNilai);
9 reset($arrNilai);
10 echo "<b>Array setelah pengurutan dengan asort()</b>";
11 echo "<pre>";
12 print_r($arrNilai);
13 echo "</pre>";
14
15 arsort($arrNilai);
16 reset($arrNilai);
17 echo "<b>Array setelah pengurutan dengan arsort()</b>";
18 echo "<pre>";
19 print_r($arrNilai);
20 echo "</pre>";
21 ?>
```

Program 5.8

Nama File : array08.php

Deskripsi : Program mengurutkan array dengan ksort() dan krsort().

```
1 <?php
2 $arrNilai = array ("Ani"=>80, "Otim"=>90, "Sri"=>75,
3 "Budi"=>85);
4 echo "<b>Array sebelum pengurutan</b>";
5 echo "<pre>";
6 print_r($arrNilai);
7 echo "</pre>";
8 ksort($arrNilai);
9 reset($arrNilai);
10 echo "<b>Array setelah pengurutan dengan ksort()</b>";
11 echo "<pre>";
12 print_r($arrNilai);
13 echo "</pre>";
14 krsort($arrNilai);
15 reset($arrNilai);
16 echo "<b>Array setelah pengurutan dengan krsort()</b>";
17 echo "<pre>";
18 print_r($arrNilai);
19 echo "</pre>";
20 echo "</pre>";
21 ?>
```

Program 5.9

Nama File : array09.php

Deskripsi : Program mengatur posisi pointer dalam array.

```
1 <?php
2 $transport = array('foot', 'bike', 'car', 'plane');
3 echo "<pre>";
4 print_r ($transport);
5 echo "</pre>";
6
7 $mode = current($transport);
8 echo $mode."<br>"; // $mode = 'foot';
9 $mode = next($transport);
10 echo $mode."<br>"; // $mode = 'bike';
11 $mode = current($transport);
12 echo $mode."<br>"; // $mode = 'bike';
13 $mode = prev($transport);
14 echo $mode."<br>"; // $mode = 'foot';
15 $mode = end($transport);
16 echo $mode."<br>"; // $mode = 'plane';
17 $mode = current($transport);
18 echo $mode."<br>"; // $mode = 'plane';
19 ?>
```

Program 5.10

Nama File : array10.php

Deskripsi : Program mencari elemen array.

```
1 <?php
2 $arrBuah = array ("Mangga", "Apel", "Pisang", "Kedondong",
3 "Jeruk");
4 if (in_array ("Kedondong", $arrBuah)) {
5     echo "Ada buah Kedondong di sini";
6 } else {
7     echo "Tidak ada buah Kedondong di sini";
8 }
9 ?>
```

Fungsi dalam PHP

Bentuk umum pendefinisian fungsi dalam PHP

```
function nama_fungsi(parameter1,...,n) {  
    statement2;  
}
```

Program 5.11

Nama File : fungsi01.php

Deskripsi : Program sederhana mendefinisikan dan memanggil fungsi tanpa parameter dan tanpa return value.

```
1 <?php  
2 function cetak_ganjil () {  
3     for ($i=0; $i<100; $i++) {  
4         if ($i%2 == 1) {  
5             echo "$i ";  
6         }  
7     }  
8 } //pemanggilan fungsi  
9 cetak_ganjil();  
10 ?>
```

Program 5.12

Nama File : array02.php

Deskripsi : Program fungsi dengan parameter.

```
1 <?php  
2 function cetak_ganjil ($awal, $akhir) {  
3     for ($i=$awal; $i<$akhir; $i++) {  
4         if ($i%2 == 1) {  
5             echo "$i ";  
6         }  
7     }  
8 } //pemanggilan fungsi  
9 $a = 10;  
10 $b = 50;  
11 echo "<b>Bilangan ganjil dari $a sampai $b : </b><br>";  
12 cetak_ganjil($a, $b);  
13 ?>
```

Program 5.13

Nama File : fungsi03.php

Deskripsi : Program fungsi yang mengembalikan nilai.

```
1 <?php  
2 function luas_lingkaran ($jari) {  
3     return 3.14 * $jari * $jari;  
4 } //pemanggilan fungsi  
5 $r = 10;  
6 echo "Luas lingkaran dengan jari-jari $r = ";  
7 echo luas_lingkaran($r);  
8 ?>
```

Program 5.14

Nama File : fungsi04.php

Deskripsi : Program passing by value dalam fungsi.

```
1 <?php
2 function tambah_string ($str) {
3     $str = $str . ", Jakarta";
4     return $str;
5 }
6 //
7 $str = "Universitas Budi Luhur";
8 echo "\$str = $str<br>";
9 echo tambah_string ($str). "<br>";
10 echo "\$str = $str<br>";
11 ?>
```

Program 5.15

Nama File : fungsi05.php

Deskripsi : Program passing by reference dalam fungsi.

```
1 <?php
2 function tambah_string (&$str) {
3     $str = $str . ", Jakarta";
4     return $str;
5 }
6 //
7 $str = "Universitas Budi Luhur";
8 echo "\$str = $str<br>";
9 echo tambah_string ($str). "<br>";
10 echo "\$str = $str<br>";
11 ?>
```

Program 5.16

Nama File : fungsi06.php

Deskripsi : Program menampilkan defined-function yang didukung oleh PHP.

```
1 <?php
2 function luas_lingkaran ($jari) {
3     return 3.14 * $jari * $jari;
4 }
5
6 $arr = get_defined_functions();
7 echo "<pre>";
8 print_r($arr);
9 echo "</pre>";
10 ?>
```

Program 5.17

Nama File : fungsi07.php

Deskripsi : Program memeriksa suatu fungsi tersedia di PHP atau tidak.

```
1 <?php
2 if (function_exists('exif_read_data')) {
3     echo "Fungsi exif_read_data() ada di PHP.<br />\n";
4 } else {
5     echo "Fungsi exif_read_data() tidak ada di PHP.<br />\n";
6 }
7 ?>
```

Bab 6

Penanganan String dan Tanggal

- ❖ Pengantar String
- ❖ Fungsi-fungsi String dalam PHP
- ❖ Fungsi-fungsi Operasi Tanggal dalam PHP

Pengantar String

String merupakan kumpulan dari karakter. Dalam PHP, karakter sama dengan byte, dimana terdapat 256 karakter. PHP tidak mendukung native-unicode.

Untuk menuliskan sebuah string dalam PHP, bisa menggunakan 3 (tiga) cara, yaitu dengan :

1. single quoted – kutip tunggal (')
2. double quoted – kutip ganda (")
3. heredoc sintaks

Program 6.1

Nama File : *string01.php*
 Deskripsi : Program string dengan single quoted (').

```

1 <?php
2 echo 'this is a simple string';
3
4 echo 'You can also have embedded newlines in
5 strings this way as it is
6 okay to do';
7
8 // Outputs: Arnold once said: "I'll be back"
9 echo 'Arnold once said: "I\'ll be back"';
10
11 // Outputs: You deleted C:\*.*?
12 echo 'You deleted C:\\\*.*?';
13
14 // Outputs: You deleted C:\*.*?
15 echo 'You deleted C:\*.*?';
16
17 // Outputs: This will not expand: \n a newline
18 echo 'This will not expand: \n a newline';
19
20 // Outputs: Variables do not $expand $either
21 echo 'Variables do not $expand $either';
22 ?>

```

Program 6.2

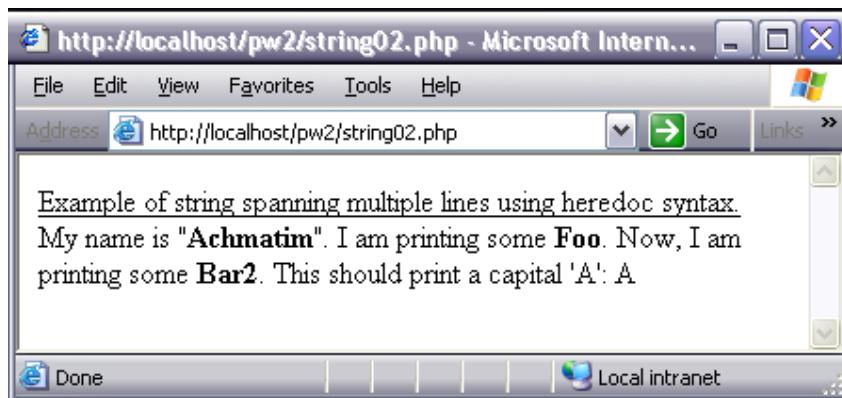
Nama File : *string02.php*
 Deskripsi : Program string dengan heredoc sintaks.

```

1 <?php
2 $str = <<<EOD
3 Example of string
4 spanning multiple lines
5 using heredoc syntax.
6 EOD;
7
8 /* More complex example, with variables. */
9 class foo
{
    var $foo;

```

```
12     var $bar;
13
14     function foo()
15     {
16         $this->foo = 'Foo';
17         $this->bar = array('Bar1', 'Bar2', 'Bar3');
18     }
19 }
20
21 $foo = new foo();
22 $name = 'Achmatim';
23
24 echo <<<EOT
25 <u>$str</u><br>
26 My name is "<b>$name</b>". I am printing some <b>$foo->foo</b>.
27 Now, I am printing some <b>{$foo->bar[1]}</b>.
28 This should print a capital 'A': \x41
29 EOT;
30 ?>
```



Gambar 6.1. Tampilan Program 6.2

Fungsi-fungsi String dalam PHP

PHP menyediakan lebih dari 90 fungsi untuk manipulasi string. Fungsi-fungsi string dalam PHP bisa dilihat di <http://ca.php.net/manual/en/ref.strings.php>.

Beberapa fungsi manipulasi string yang sering digunakan, antara lain :

- **addslashes()** – menambahkan backslashes (\) di setiap tanda kutip (quote) dalam string.
- **chr()** – menghasilkan karakter dari bilangan ASCII
- **crypt()** – menghasilkan string terenkripsi searah.
- **echo()** – menampilkan satu atau beberapa string.
- **explode()** – memecah string berdasarkan suatu delimiter (separator).
- **htmlentities ()** – mengubah semua tag html menjadi HTML entities.
- **htmlspecialchars()** – mengubah semua karakter khusus menjadi HTML entities
- **implode()** – menggabungkan elemen array menjadi string dengan suatu delimiter (separator).
- **join()** – sama dengan implode()
- **ltrim()** – menghapus karakter tertentu (" ", \t, \n, \r, \0, \x0B) di awal string.
- **md5()** – menghasilkan string terenkripsi searah.
- **nl2br()** – menambahkan HTML line break (
) sebelum semua baris baru dalam string.
- **number_format()** – mengatur format angka.
- **ord()** – menghasilkan kode ASCII dari suatu karakter.

- **rtrim()** - menghapus karakter tertentu (" ", \t, \n, \r, \0, \x0B) di akhir string.
- **split()** - sama dengan explode
- **str_repeat()** - mengulang string
- **str_replace()** - mengganti semua string dalam pola menjadi suatu string.
- **strip_tags()** - mengabaikan HTML dan PHP tag dalam string.
- **stripslashes()** - menghilangkan backslashes (\) dalam string.
- **strlen()** - menghitung panjang string.
- **strpos()** - mencari posisi pertama sebuah string dalam string.
- **strrchr()** - mencari posisi terakhir sebuah karakter dalam string.
- **strrpos()** - mencari posisi terakhir sebuah string dalam string.
- **strrev()** - membalik string.
- **strstr()** - mencari posisi pertama sebuah karakter dalam string.
- **strtolower()** - mengubah string menjadi huruf kecil (lower-case).
- **strtoupper()** - mengubah string menjadi huruf kapital (upper-case)
- **substr()** - memotong string
- **trim()** - menghapus karakter tertentu (" ", \t, \n, \r, \0, \x0B) di akhir dan awal string.
- **ucfirst()** - mengubah huruf pertama dari semua string menjadi Upper-case.
- **ucwords()** - mengubah huruf pertama tiap kata dalam string menjadi upper-case.
- **wordwrap()** - memotong sejumlah karakter dalam string dengan *string-break-character*.

Program 6.3

Nama File : string03.php

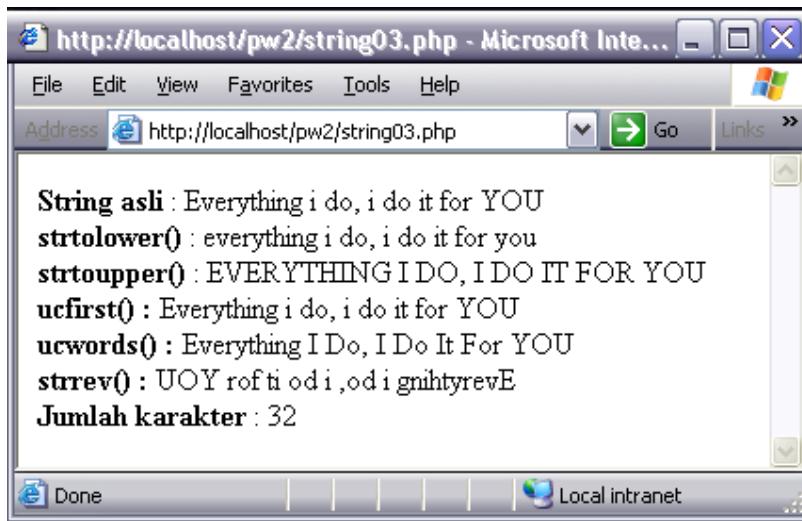
Deskripsi : Program penggunaan fungsi strtolower, strtoupper, ucfirst, ucwords, strrev, dan strlen dalam string.

```
1 <?php
2 $str = "Everything i do, i do it for YOU";
3
4 echo "<b>String asli</b> : $str";
5 echo "<br><b>strtolower()</b> : ". strtolower($str);
6 echo "<br><b>strtoupper()</b> : ". strtoupper($str);
7 echo "<br><b>ucfirst()</b> : ". ucfirst($str);
8 echo "<br><b>ucwords()</b> : ". ucwords($str);
9 echo "<br><b>strrev()</b> : ". strrev($str);
10 echo "<br><b>Jumlah karakter</b> : ". strlen($str);
11 ?>
```

Penjelasan Program 6.3

Beberapa fungsi yang digunakan dalam program 6.3 di atas antara lain :

- strtolower, untuk mengubah string menjadi lower-case.
- strtoupper, untuk mengubah string menjadi upper-case.
- ucfirst, untuk mengubah huruf pertama string menjadi upper-case.
- ucwords, untuk mengubah huruf pertama tiap kata menjadi upper-case.
- strrev, untuk membalik string.
- strlen, untuk menghitung panjang string.



Gambar 6.2 Tampilan Program 6.3

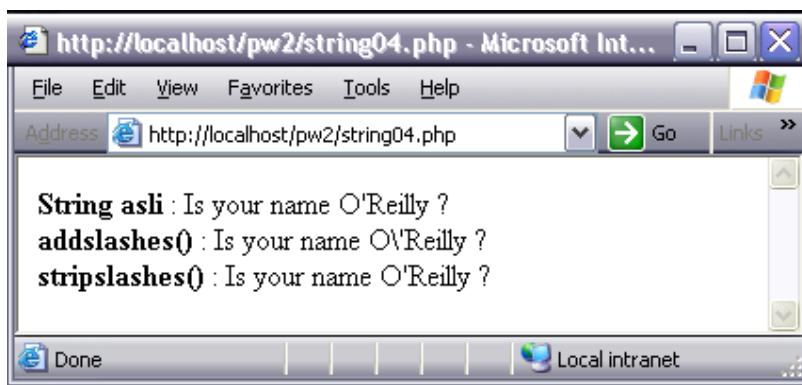
Program 6.4

Nama File : string04.php
Deskripsi : Program penggunaan fungsi addslashes dan stripslashes pada string.

```
1 <?php
2 $str = "Is your name O'Reilly ?";
3 $str2 = addslashes ($str);
4 $str3 = stripslashes ($str2);
5
6 echo "<b>String asli</b> : $str";
7 echo "<br><b>addslashes()</b> : $str2";
8 echo "<br><b>stripslashes()</b> : $str3";
9 ?>
```

Penjelasan Program 6.4

Pada program 6.4 di atas terdapat fungsi addslashes() yang akan menambahkan *backslashes* (\) pada setiap quote dalam string, dan sebaliknya fungsi stripslashes akan menghilangkan *backslashes* pada string.



Gambar 6.3. Tampilan Program 6.4

Program 6.5

Nama File : string05.php
Deskripsi : Program penggunaan fungsi chr untuk menampilkan 256 karakter ASCII.

```
1 <?php
2 echo "Menampilkan bilangan ASCII";
3 for ($i=1; $i<=256; $i++) {
4     echo "<br>$i.\t". chr($i);
5 }
6 ?>
```

Penjelasan Program 6.5

Program 6.5 di atas akan menampilkan 256 karakter ASCII. Pada baris 4, terdapat fungsi `chr()` yang akan mengubah nilai `$i` (integer) menjadi karakter ASCII.

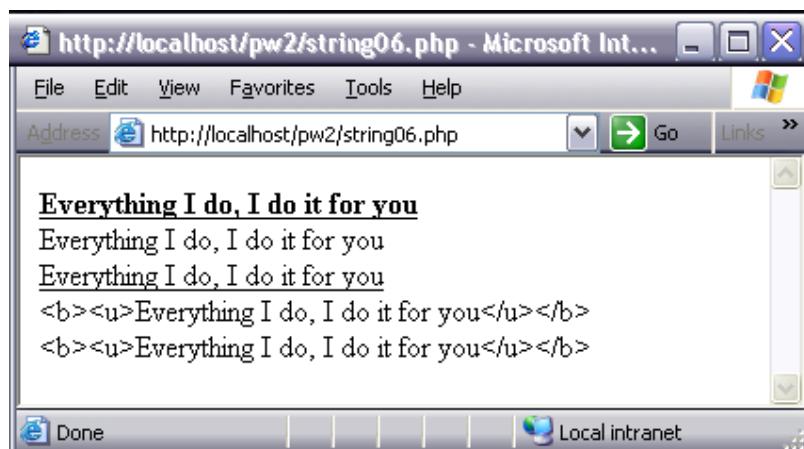
Program 6.6

*Nama File : string06.php
Deskripsi : Program penggunaan fungsi strip_tags, htmlspecialchars, dan htmlentities untuk mengatur tag HTML.*

```
1 <?php
2 $str = "<b><u>Everything I do, I do it for you</u></b>";
3
4 echo $str."<br>";
5 //no tag html
6 echo strip_tags ($str). "<br>";
7 //allow tag <u>
8 echo strip_tags ($str,"<u>")."<br>";
9 //tampil apa adanya
10 echo htmlspecialchars ($str). "<br>";
11 //tampil apa adanya
12 echo htmlentities ($str);
13 ?>
```

Penjelasan Program 6.6

Program 6.6 di atas merupakan contoh penggunaan fungsi `strip_tags`, `htmlspecialchars` dan `htmlentities` untuk menangani tag-tag HTML yang ada dalam string. Pada baris ke-6, fungsi `strip_tags` akan menghilangkan semua tag HTML. Pada baris ke-8, fungsi `strip_tags` dengan tambahan parameter "`<u>`" akan menghilangkan semua tag HTML kecuali tag `<u>`. Pada baris ke-10 dan ke-12, fungsi `htmlspecialchars` dan `htmlentities` akan mengubah semua tag HTML menjadi HTML special character, sehingga tag HTML akan ditampilkan apa adanya



Gambar 6.4. Tampilan Program 6.6

Program 6.7

*Nama File : string07.php
Deskripsi : Program penggunaan fungsi number_format untuk mengatur format tampilan angka.*

```

1 <?php
2 $number = 1234.56;
3
4 // english notation (default)
5 $english_format_number = number_format($number);
6 echo "<br>".$english_format_number; // 1,235
7
8 // French notation
9 $nombre_format_francais = number_format($number, 2, ',', ' ');
10 echo "<br>".$nombre_format_francais; // 1 234,56
11
12 //indonesian notation
13 $format_indonesia = number_format ($number, 2, ',', '.');
14 echo "<br>".$format_indonesia; //1.234,56
15
16 $number = 1234.5678;
17 // english notation without thousands seperator
18 $english_format_number = number_format($number, 2, '.', '');
19 echo "<br>". $english_format_number; // 1234.57
20
21 ?>

```

Penjelasan Program 6.7

Program 6.7 di atas merupakan contoh penggunaan fungsi `number_format` untuk mengatur tampilan angka. Secara default, fungsi `number_format` akan memformat angka dengan model Inggris yaitu setiap tiga digit dipisahkan dengan koma (,) dan tanpa pecahan. Pada baris ke-9, fungsi `number_format` akan mengubah format angka dengan 2 digit pecahan (parameter kedua), pecahan dipisahkan dengan koma (parameter ketiga) dan setiap tiga digit dipisahkan dengan spasi (parameter keempat).

Program 6.8

*Nama File : string08.php
Deskripsi : Program penggunaan fungsi explode untuk memecah string menjadi array.*

```

1 <?php
2 // Contoh 1
3 $buah = "mangga pisang rambutan apel jeruk kedondong";
4 $buahan = explode(" ", $buah);
5 echo $buahan[0]."<br>"; // mangga
6 echo $buahan[3]."<br>"; // apel
7
8 // Contoh 2
9 $data = "foo:*:1023:1000::/home/foo:/bin/sh";
10 list($user, $pass, $uid, $gid, $gecos, $home, $shell) =
explode(":", $data);
11 echo $user."<br>"; // foo
12 echo $pass; // *
13
14 ?>

```

Penjelasan Program 6.8

Program 6.8 di atas merupakan contoh penggunaan fungsi `explode` untuk memecah string berdasarkan pola (aturan) tertentu. Pada baris ke-4, dengan fungsi `explode()` akan memecah string `$buah` berdasarkan spasi dan masing-masing akan menjadi elemen array `$buahan`.

Program 6.9

*Nama File : string09.php
Deskripsi : Program penggunaan fungsi implode untuk menggabungkan array menjadi string.*

```
1 <?php
2 $buahan = array('mangga','jeruk','rambutan','apel','nanas');
3 $buah = implode(", ", $buahan);
4
5 echo "Saya suka buah ". $buah;
// mangga,jeruk,rambutan,apel,nanas
6 ?>
```

Penjelasan Program 6.9

Program 6.9 di atas merupakan contoh penggunaan fungsi implode() untuk menggabungkan semua isi array menjadi satu string. Fungsi implode merupakan kebalikan dari fungsi explode. Pada baris ke-3, dengan fungsi implode() akan menggabungkan setiap elemen array \$buahan menjadi string \$buah dengan pemisah koma di antara tiap elemen.

Program 6.10

Nama File : string10.php

Deskripsi : Program penggunaan fungsi strstr, strchr dan strrchr untuk mengambil sebagian string berdasarkan pola (karakter) tertentu.

```
1 <?php
2 $file = "test.this.txt";
3 $ext1 = strstr($file, ".");
4 $ext2 = strchr($file, ".");
5 $ext3 = strrchr($file, ".");
6
7 echo $ext1. "<br>"; // .this.txt
8 echo $ext2. "<br>"; // .this.txt
9 echo $ext3; // .txt
10 ?>
```

Penjelasan Program 6.10

Program 6.10 di atas merupakan contoh pengambilan sebagian string berdasarkan pola (karakter) tertentu. Fungsi strstr dan strchr pada baris ke-3 dan ke-4 akan mengambil string setelah karakter titik (.) dimana pencarian dilakukan dari awal string. Sedangkan fungsi strrchr, akan mengambil string setelah karakter titik (.) yang mana pencarian dilakukan dari akhir string.

Program 6.11

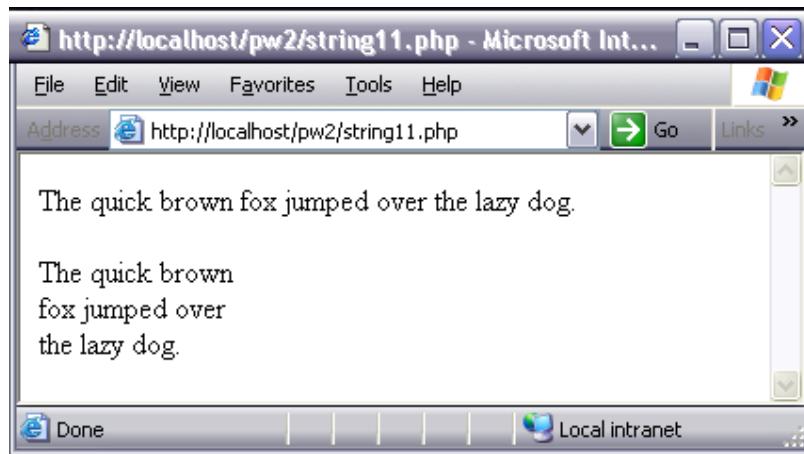
Nama File : string11.php

Deskripsi : Program penggunaan fungsi wordwrap untuk memotong sejumlah karakter dari string.

```
1 <?php
2 $text = "The quick brown fox jumped over the lazy dog.";
3 $newtext = wordwrap($text, 15, "<br>");
4
5 echo $text."<br><br>";
6 echo $newtext;
7 ?>
```

Penjelasan Program 6.11

Program 6.11 di atas merupakan contoh penggunaan fungsi wordwrap untuk memotong sejumlah karakter dari string. Pada baris ke-3, fungsi wordwrap akan memotong string \$text per-15 karakter dan dipisahkan (ditambahkan) dengan string "
"



Gambar 6.5. Tampilan Program 6.11

Program 6.12

*Nama File : string12.php
Deskripsi : Program penggunaan fungsi nl2br dalam string.*

```
1 <form action="" method="post">
2   Input teks di sini :<br>
3   <textarea name="input" cols="40" rows="4"></textarea><br>
4   <input type="submit" name="Submit" value="Proses">
5 </form>
6
7 <?php
8 if (isset($_POST['Submit'])) {
9   $txt = $_POST['input'];
10  echo "<u>Tanpa nl2br() :</u> <br>$txt<br><br>";
11  echo "<u>Dengan nl2br :</u> <br>". nl2br ($txt);
12 }
14 ?>
```

Penjelasan Program 6.12

Program 6.12 di atas merupakan contoh penggunaan fungsi nl2br. Fungsi nl2br() akan menambahkan karakter pindah baris (
) di setiap awal baris string inputan. Untuk mencoba program di atas, inputkan string sebagai berikut :

```
bila yang tertulis untukku
adalah yang terbaik untukku
kan kujadikan kau kenangan
yang terindah dalam hidupku
namun tak kan mudah bagiku
meninggalkan jejak hidupku
yang telah terukir abadi
sebagai kenangan yang terindah
```

Program 6.13

*Nama File : string13.php
Deskripsi : Program penggunaan fungsi substr untuk memotong string.*

```
1 <form action="" method="post">
2   NIM Anda :
3   <input type="text" name="txtnim"><br>
4   <input type="submit" name="Submit" value="Proses">
5 </form>
6
7 <?php
8 if (isset($_POST['Submit'])) {
9   $nim = $_POST['txtnim'];
```

```

10 $a = substr ($nim, 2, 2);
11 switch($a) {
12     case '11' : $jurusan = "Teknik Informatika";
13         break;
14     case '22' : $jurusan = "Sistem Informasi";
15         break;
16     case '33' : $jurusan = "Sistem Komputer";
17         break;
18     case '44' : $jurusan = "Komputerisasi Akuntansi";
19         break;
20     default    : $jurusan = "Salah jurusan";
21 }
22 echo "Nim Anda : $nim<br>";
23 echo "Jurusan Anda : $jurusan";
24 }
25 ?>

```

Penjelasan Program 6.13

Program 6.13 di atas merupakan contoh penggunaan fungsi substr (baris ke-10) untuk memotong string. Pada baris ke-10 tersebut, string \$nim akan dipotong mulai karakter index ke-2 (index dimulai dari 0) sepanjang 2 karakter.

Fungsi-fungsi Operasi Tanggal dalam PHP

Fungsi dalam Operasi tanggal dalam PHP yang utama dan sering digunakan adalah fungsi date(). Fungsi ini akan menghasilkan tanggal dan waktu server sekarang. Beberapa pilihan parameter dari fungsi date() ini dapat dilihat pada tabel berikut ini :

parameter	Keterangan	Contoh nilai
Hari		
d	Tanggal, 2 digit dengan nol	01 s/d 31
D	Tiga digit nama hari dalam seminggu	Mon s/d Sun
j	Tanggal tanpa nol	1 s/d 31
I (/lowercase 'L')	Nama hari lengkap dalam seminggu	Sunday through Saturday
N	Urutan hari dalam seminggu	1 (untuk Monday) s/d 7 (untuk Sunday)
S	Akhiran angka English untuk tanggal, 2 karakter	st, nd, rd atau th.
w	Urutan hari dalam seminggu	0 (untuk Sunday) s/d 6 (untuk Saturday)
z	Urutan hari dalam setahun	0 s/d 365
Minggu		
W	Urutan minggu dalam setahun	Contoh: 42 (minggu ke-42 dalam tahun ini)
Bulan		
F	Nama bulan lengkap	January s/d December
m	Urutan bulan dalam setahun dengan nol	01 s/d 12
M	Tiga digit nama bulan dalam setahun	Jan s/d Dec
n	Urutan bulan dalam setahun,	1 s/d 12

parameter	Keterangan	Contoh nilai
	tanpa nol	
t	Jumlah hari dalam tiap bulan	28 s/d 31
Tahun		
Y	4 (empat) digit tahun	Contoh: 1999 atau 2006
y	2 (dua) digit tahun	Contoh: 99 or 06
Waktu		
a	Lowercase Ante meridiem dan Post meridiem	am atau pm
A	Uppercase Ante meridiem dan Post meridiem	AM atau PM
g	Jam format 12 tanpa nol	1 s/d 12
G	Jam format 24 tanpa nol	0 s/d 23
h	Jam format 12 dengan nol	01 s/d 12
H	Jam format 24 dengan nol	00 s/d 23
i	Menit dengan nol	00 s/d 59
s	Detik dengan nol	00 s/d 59

Program 6.14

Nama File : date01.php

Deskripsi : Program menampilkan date/time dengan berbagai parameter.

```

1 <?php
2 // Sekarang tanggal 13 April 2006 09:43:49
3
4 echo "<br>". date("d/m/Y H:i:s");      // 13/04/2006 09:43:03
5 echo "<br>". date("F j, Y, g:i a");    // April 13, 2006, 9:43 am
6 echo "<br>". date("d.m.y");           // 13.04.06
7 echo "<br>". date("Ymd");            // 20060413
8 // 13-04-06, 4330 4349 4 Thuam06 102
9 echo "<br>". date('j-m-y, it is w Day z ');
10 // it is the 13th day.
11 echo "<br>". date('\i\t \i\s \t\h\e js \d\a\y.');
12 // Thu Apr 13 9:43:49 KRAST 2006
13 echo "<br>". date("D M j G:i:s T Y");
14 echo "<br>". date("H:i:s");          // 17:16:17
15
16 ?>

```

Program 6.15

Nama File : date02.php

Deskripsi : Program menampilkan nama hari dalam bahasa Indonesia.

```

1 <?php
2 // Sekarang tanggal 13 April 2006 09:43:49
3 $arrDay = array("Minggu", "Senin", "Selasa", "Rabu", "Kamis",
4 "Jum'at", "Sabtu");
5 $day = date ("w");      //0 - 6 of day
6 echo "Hari ini hari : <b>" . $arrDay[$day]."</b>";
7 ?>

```

Bab 7

File dan Direktori

- ❖ Membuat, Menulis dan Membaca Isi File
 - + Membuka dan Membuat File
 - + Menulis ke File
 - + Membaca Isi File
 - + Menutup File
- ❖ Membuat dan Menghapus Direktori
- ❖ Manipulasi File
 - + Mendapatkan Informasi File
 - + Copy, Rename dan Delete File
- ❖ Upload File

Penanganan File

Dalam management file dan direktori, PHP menyediakan lebih dari 70 fungsi. Beberapa fungsi utama yang berhubungan dengan management file (create, write, append, dan delete), antara lain :

Membuka dan Membuat File

```
fopen ($namafайл, $mode);
```

Keterangan :

\$namafайл merupakan nama file yang akan dibuat, sedangkan \$mode merupakan mode akses file. Mode akses file yang bisa digunakan yaitu :

Mode	Keterangan
r	Hanya untuk baca file, pointer berada di awal file
r+	Untuk baca dan tulis file, pointer berada di awal file
w	Hanya untuk tulis file, isi file lama dihapus, jika file belum ada maka akan di-create
w+	Untuk baca dan tulis file, isi file lama dihapus, jika file belum ada maka akan di-create
a	Hanya untuk menambahkan isi file, pointer berada di akhir file, jika file belum ada maka di-create
a+	Untuk membaca dan menambahkan isi file, pointer berada di akhir file, jika file belum ada maka di-create

Program 7.1

Nama File : file01.php

Deskripsi : Program mengakses (membuka) file dengan mode r.

```

1 <?php
2 $namafайл = "data.txt";
3 $handle = fopen ($namafайл, "r");
4 if (!$handle) {
5   echo "<b>File tidak dapat dibuka atau belum ada</b>";
6 } else {
7   echo "<b>File berhasil dibuka</b>";
8 }
9 fclose($handle);
10 ?>

```



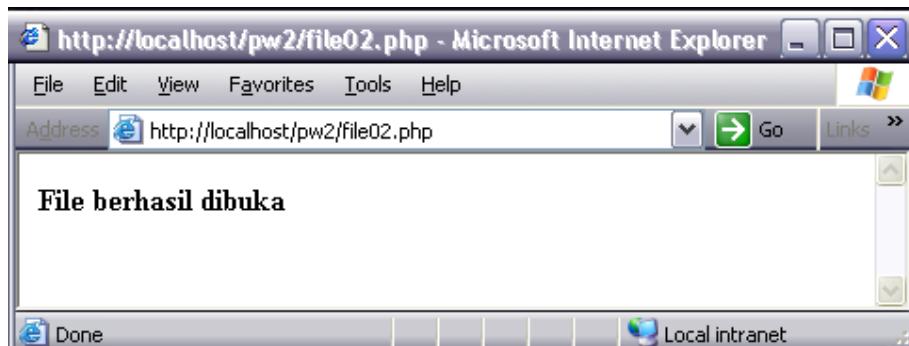
Gambar 7.1 Tampilan Program 7.1

Program 7.2

Nama File : file02.php

Deskripsi : Program mengakses (membuka) file dengan mode w.

```
1 <?php
2 $namafile = "data.txt";
3 $handle = fopen ($namafile, "w");
4 if (!$handle) {
5     echo "<b>File tidak dapat dibuka atau belum ada</b>";
6 } else {
7     echo "<b>File berhasil dibuka</b>";
8 }
9 fclose($handle);
10 ?>
```



Gambar 7.2 Tampilan Program 7.2

Menulis ke File

```
fwrite ($handle, $isi);
fputs ($handle, $isi);
```

Program 7.3

Nama File : file03.php

Deskripsi : Program menulis ke file.

```
1 <?php
2 $namafile = "data.txt";
3 $handle = fopen ($namafile, "w");
```

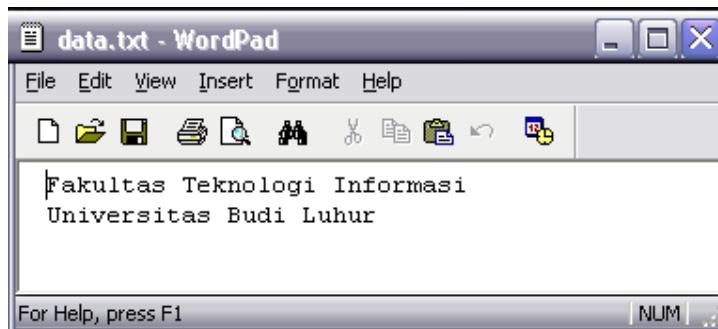
```

4 if (!$handle) {
5     echo "<b>File tidak dapat dibuka atau belum ada</b>";
6 } else {
7     fwrite ($handle, "Fakultas Teknologi Informasi\n");
8     fputs ($handle, "Universitas Budi Luhur\n");
9     //file_put_contents ($namafайл, "Jakarta");
10    echo "<b>File berhasil ditulis</b>";
11 }
12 fclose($handle);
13 ?>

```

Penjelasan Program 7.3

Untuk menulis ke file, dapat menggunakan fungsi **fwrite()** dan **fputs()**. Variabel \$handle merupakan mode pengaksesan file, dimana mode yang digunakan harus bisa menulis ke file. Contohnya mode w dan a. Selain itu, kita juga bisa menggunakan fungsi **file_puts_contents()** dengan parameter berupa nama file dan isi yang akan ditambahkan. Jika program di atas dijalankan, maka isi file data.txt akan berisi seperti pada gambar 7.3 (buka dengan WordPad).



Gambar 7.3. File data.txt dibuka dengan WordPad

Membaca Isi File

```

fread ($handle [, $panjang]);
fgets ($handle [, $panjang]);

```

Program 7.4

Nama File : file04.php

Deskripsi : Program membaca dan menampilkan isi file.

```

1 <?php
2 $namafile = "data.txt";
3 $handle = fopen ($namafile, "r");
4 if (!$handle) {
5     echo "<b>File tidak dapat dibuka atau belum ada</b>";
6 } else {
7     $isi = fgets ($handle, 2048);
8     // $isi2 = fread ($handle, 20);
9     echo "Isi 1 : $isi<br>";
10    //echo "Isi 2 : $isi2<br>";
11 }
12 fclose($handle);
13 ?>

```

Penjelasan Program 7.4

Untuk membaca isi file, bisa menggunakan fungsi **fgets()** dan **fread()**. Fungsi **fgets()** akan membaca isi file tiap baris. Sedangkan fungsi **fread()** akan membaca isi file setiap jumlah byte tertentu. Pada program di atas, baris ke-8 akan membaca file setiap 20 byte atau 20 karakter.

Program 7.5

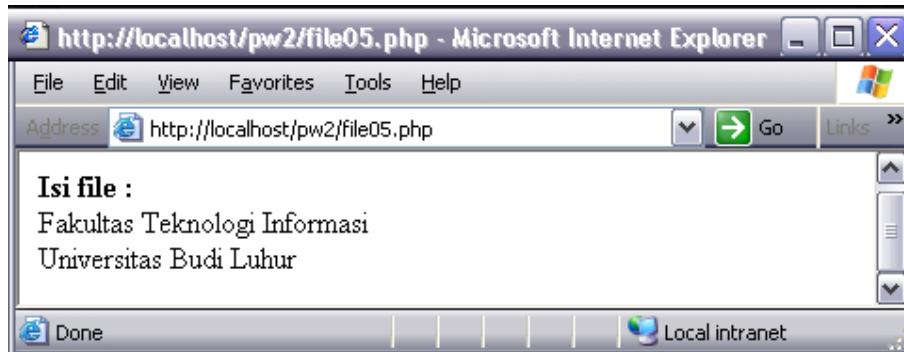
Nama File : file05.php

Deskripsi : Program membaca isi file baris tiap baris.

```
1 <?php
2 $namafile = "data.txt";
3 $handle = fopen ($namafile, "r");
4 if (!$handle) {
5     echo "<b>File tidak dapat dibuka atau belum ada</b>";
6 } else {
7     echo "<b>Isi file : </b><br>";
8     while ($isi = fgets ($handle, 2048)) {
9         echo "$isi<br>";
10    }
11 }
12 fclose($handle);
13 ?>
```

Penjelasan Program 7.5

Program 7.5 di atas akan menampilkan keseluruhan isi file, dimana isi file akan dibaca tiap baris. Pembacaan dilakukan dari baris pertama sampai baris terakhir.



Gambar 7.4. Tampilan Program 7.5

Program 7.6

Nama File : file06.php

Deskripsi : Program membaca isi file baris tiap baris.

```
1 <?php
2 $namafile = "data.txt";
3 $handle = @fopen($namafile, "r");
4 if ($handle) {
5     while (!feof($handle)) {
6         $buffer = fgets($handle, 4096);
7         echo $buffer."<br>";
8     }
9     fclose($handle);
10 }
11 ?>
```

Penjelasan Program 7.6

Program 7.6 di atas akan menampilkan keseluruhan isi file, dimana isi file akan dibaca tiap baris. Pembacaan dilakukan selama pointer file belum sampai di end-of-file. Fungsi feof() pada baris ke-5 berfungsi untuk memeriksa pointer file sudah berada di akhir file atau belum.

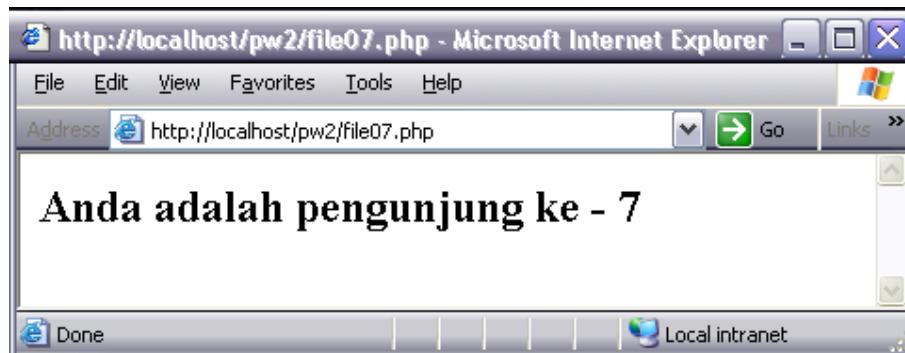
Menutup File

```
fclose ($handle);
```

Program 7.7

Nama File : file07.php
Deskripsi : Program aplikasi user counter dengan file.

```
1 <?php
2 $counter_file="counter.txt";
3 if (!file_exists ($counter_file)) {
4     fopen ($counter_file, "w");
5 }
6 $file = fopen($counter_file,"r");
7
8 $counter = fread($file,10);
9 fclose($file);
10
11 $counter++;
12
13 echo "<h2>Anda adalah pengunjung ke - $counter</h2>";
14 $file = fopen($counter_file, "w");
15 fwrite($file,$counter);
16 fclose($file);
17 ?>
```



Gambar 7.5. Tampilan Program 7.7

Membuat dan Menghapus Direktori

Membuat Direktori Baru

Program 7.8

Nama File : file08.php
Deskripsi : Program membuat direktori baru.

```
1 <?php
2 $dir = "images"; //nama direktori
3 $cek = mkdir ($dir);
4 if ($cek) {
5     echo "Direktori <b>$dir</b> berhasil dibuat";
6 } else {
7     echo "Direktori <b>$dir</b> gagal dibuat";
8 }
9 ?>
```

Penjelasan Program 7.8

Program 7.8 di atas akan membuat direktori bernama "images" di direktori tempat program 7.8 disimpan. Fungsi untuk membuat direktori baru dalam PHP adalah **mkdir()**. Jika direktori yang akan dibuat berada di luar tempat program disimpan, maka sertakan *path* direktori secara lengkap.

Menghapus Direktori

Program 7.9

Nama File : file09.php
Deskripsi : Program menghapus direktori.

```
1 <?php
2 //buat direktori
3 $dir = "include";           //nama direktori
4 $cek = mkdir ($dir);
5 if ($cek) {
6     echo "Direktori <b>$dir</b> berhasil dibuat";
7 } else {
8     echo "Direktori <b>$dir</b> gagal dibuat";
9 }
10 //hapus direktori
11 $del = rmdir ($dir);
12 if ($del) {
13     echo "<br>Direktori <b>$dir</b> berhasil dihapus";
14 } else {
15     echo "<br>Direktori <b>$dir</b> gagal dihapus";
16 }
17 ?>
```

Penjelasan Program 7.9

Program 7.9 di atas akan membuat direktori baru bernama "include" (lihat baris ke-4). Setelah itu, direktori akan dihapus dengan fungsi **rmdir()** pada baris ke-10. Fungsi rmdir() hanya bisa menghapus direktori kosong. Jika direktori ada isinya maka, gunakan fungsi rekursif untuk menghapus isi direktori. Lihat contoh di PHP Manual.

Menampilkan Isi Direktori

Program 7.10

Nama File : file10.php
Deskripsi : Program menampilkan file dan direktori dalam suatu direktori.

```
1 <?php
2 $dir = "images";
3 if ($handle = opendir($dir)) {
4     while (false !== ($file = readdir($handle))) {
5         if ($file != "." && $file != "..") {
6             echo "$file<br>";
7         }
8     }
9     closedir($handle);
10 }
11 ?>
```

Penjelasan Program 7.10

Program 7.10 di atas akan menampilkan di layar, semua file dan direktori yang berada dalam direktori "images". Fungsi **opendir()** pada baris ke-3 adalah untuk membuka direktori. Sedangkan fungsi **readdir()** akan membaca file dalam direktori satu per satu secara urut. Fungsi **closedir()** pada baris ke-9 adalah untuk menutup pembukaan direktori.

Manipulasi File

Mendapatkan Informasi File

Beberapa Fungsi yang berhubungan dengan informasi dan manipulasi file, terdapat dalam tabel sebagai berikut :

Fungsi	Keterangan
file_exists(\$file)	Memeriksa apakah \$file ada atau tidak
is_file(\$file)	Memeriksa apakah \$file adalah file atau bukan.
is_dir(\$file)	Memeriksa apakah \$file adalah direktori atau bukan.
is_executable(\$file)	Memeriksa apakah \$file termasuk file yang bisa dijalankan secara langsung atau bukan.
is_writable(\$file)	Memeriksa apakah \$file termasuk file yang bisa ditulis atau diedit.
is_readable(\$file)	Memeriksa apakah \$file termasuk file yang bisa dibaca atau tidak.
filectime(\$file)	Menghasilkan waktu akses terakhir file (unix timestamp).
filemtime(\$file)	Menghasilkan waktu pembuatan file (unix timestamp).
filesize(\$file)	Menghasilkan waktu modifikasi terakhir file (dalam unix timestamp).
filetype(\$file)	Menghasilkan ukuran besar file (dalam byte).

Program 7.11

Nama File : file11.php

Deskripsi : Program memeriksa keberadaan suatu file.

```

1 <?php
2 $filename = "data.txt";
3
4 if (file_exists($filename)) {
5     echo "File $filename ADA";
6 } else {
7     echo "File $filename TIDAK ADA";
8 }
?>
```

Program 7.12

Nama File : file12.php

Deskripsi : Program mendapatkan beberapa informasi file.

```

1 <?php
2 $file = "data.txt";
3
4 if (is_file($file)) {
5     echo "File <b>$file</b> adalah FILE<br>";
6     //
7     if (is_executable ($file)) {
8         echo "File <b>$file</b> bisa dijalankan secara
9 langsung (executable)<br>";
10    } else {
11        echo "File <b>$file</b> TIDAK executable<br>";
12    }
13    //
14    if (is_writable ($file)) {
15        echo "File <b>$file</b> bisa ditulis / diedit<br>";
16    } else {
17        echo "File <b>$file</b> TIDAK bisa ditulis /
diedit<br>";
18    }
?>
```

```
18 //  
19 if (is_readable ($file)) {  
20     echo "File <b>$file</b> bisa dibaca<br>";  
21 } else {  
22     echo "File <b>$file</b> TIDAK bisa dibaca<br>";  
23 }  
24 //  
25 echo "Akses terakhir file <b>$file</b> = ". date("d-m-Y  
H:i:s.", filemtime($file)). "<br>";  
26 echo "File <b>$file</b> dibuat = ". date("d-m-Y H:i:s.",  
filectime($file)). "<br>";  
27 echo "Modifikasi terakhir file <b>$file</b> = ". date("d-  
m-Y H:i:s.", filemtime($file)). "<br>";  
28 echo "Ukuran file <b>$file</b> = ". filesize($file). "bytes<br>";  
29 echo "Jenis file <b>$file</b> = ". filetype($file). "<br>";  
30 } else if (is_dir($file)){  
31     echo "File <b>$file</b> adalah DIREKTORI<br>";  
32 } else {  
33     echo "File <b>$file</b> tidak dikenal<br>";  
34 }  
35 ?>
```

Copy, Rename dan Delete File

Program 7.13

Nama File : file13.php

Deskripsi : Program meng-copy, rename dan delete file.

```
1 <?php  
2 $file = "data.txt";  
3 $filebaru = "databaru.txt";  
4 $filebaru2 = "databaru2.txt";  
5 if (copy ($file, $filebaru)) {  
6     echo "File <b>$file</b> berhasil dicopy menjadi  
<b>$filebaru</b>. <br>";  
7 }  
8 if (rename ($filebaru, $filebaru2)) {  
9     echo "File <b>$filebaru</b> berhasil direname menjadi  
<b>$filebaru2</b>. <br>";  
10 }  
11 if (unlink ($filebaru2)) {  
12     echo "File <b>$filebaru2</b> berhasil dihapus. <br>";  
13 }  
14 ?>
```

Penjelasan Program 7.13

Program 7.13 di atas terdapat beberapa fungsi untuk meng-copy, rename dan delete file. Fungsi **copy()** pada baris ke-5 akan menduplikasi file \$file menjadi file \$filebaru. Selanjutnya fungsi **rename()** akan mengganti nama file \$filebaru menjadi \$filebaru2. Fungsi **unlink()** pada baris ke-11 akan menghapus file \$filebaru2.

Upload File

PHP, sebagai server-side-scripting, sangat memungkinkan untuk menangani upload file ke server. Ada beberapa hal yang perlu diperhatikan dalam upload file ini, yaitu :

1. Pada Form HTML harus ditambahkan atribut :
`ENCTYPE="multipart/form-data"`

2. Form inputan upload file dapat menggunakan tag <input> dengan value atribut TYPE="FILE".
3. Untuk menangani inputan, PHP menyediakan sebuah variabel array global yaitu \$_FILES. Index dari variabel ini antara lain :
 - \$_FILES['file']['name'] : Nama asli dari file yang diupload
 - \$_FILES['file']['tmp_name'] : Nama temporary file yang diupload
 - \$_FILES['file']['size'] : Ukuran file asli (dalam byte)
 - \$_FILES['file']['type'] : MIME type file yang diupload
4. *Destination folder* file upload harus bisa writable (bisa diakses), biasanya dengan permission 777 atau 775.

Program 7.14

Nama File : form_upload.php
Deskripsi : Program menampilkan form upload file.

```
1 <html>
2   <head><title>Upload File</title></head>
3   <body>
4     <form action="upload.php" method="POST"
ENCTYPE="multipart/form-data">
5       Upload File : <input type="file" name="file"><br>
6       <input type="submit" name="Upload" value="Upload">
7     </form>
8   </body>
9 </html>
```

Program 7.15

Nama File : upload.php
Deskripsi : Program proses upload file.

```
1 <?php
2 if (isset($_POST['Upload'])) {
3   $dir_upload = "images/";
4   $nama_file = $_FILES['file']['name'];
5   //
6   if (is_uploaded_file($_FILES['file']['tmp_name'])) {
7     $cek = move_uploaded_file($_FILES['file']['tmp_name'],
$dir_upload.$nama_file);
8     if ($cek) {
9       die ("File berhasil diupload");
10    } else {
11      die ("File gagal diupload");
12    }
13  }
14 }
15 ?>
```

Penjelasan Program 7.15

Program 7.15 di atas merupakan program sederhana penanganan upload file. Fungsi **is_uploaded_file()** pada baris ke-6 akan mengupload file yang telah dipilih melalui form pada program 7.14 ke folder temporary (sementara). Selanjutnya pada baris ke-7, file yang sudah diupload ke folder temporary akan dipindahkan (move) ke direktori yang diinginkan menggunakan fungsi **move_uploaded_file()**. Lihat baris ke-7!



Gambar 7.6. Tampilan Program 7.14

Bab 9

Session dan Cookies

- ❖ Session
- ❖ Cookies

Session

Session merupakan hal yang cukup penting dalam aplikasi berbasis web. Dengan session memungkinkan programmer menyimpan informasi user secara semi-permanen, artinya selama masa tertentu informasi akan tersimpan. Penyimpanan isi variabel session berada di server, jadi tidak bisa diakses secara langsung oleh client.

Dalam aplikasi berbasis web, session banyak digunakan sebagai autentifikasi login. Dengan session memungkinkan programmer mengatur siapa saja yang bisa mengakses suatu halaman. Misalnya saja, untuk melihat halaman kotak surat pada email, kita harus login terlebih dahulu. Dalam proses login antara lain akan terjadi pembuatan suatu session yang akan dibawa oleh user di setiap halaman. Di halaman kotak surat, session tersebut akan diperiksa. Jika session benar maka user dipersilahkan membuka halaman kotak surat, namun jika salah maka user tidak bisa membuka halaman kotak surat dan biasanya akan diminta untuk login terlebih dahulu. Itulah sebabnya, user tidak bisa mengakses halaman kotak surat secara langsung tanpa melakukan login.

Dalam penanganan session terdapat beberapa proses yang perlu diperhatikan :

- Proses pembuatan session
- Proses pemeriksaan session
- Proses penghapusan session

Selanjutnya bagaimana session itu sendiri dijalankan? Agar proses penyimpanan dalam session berjalan, PHP melakukan beberapa hal berikut ini :

1. PHP meng-generate (membentuk) sebuah ID session.
ID session ini merupakan sejumlah deret angka random yang unik untuk setiap user dan hampir tidak mungkin bisa ditebak. ID session disimpan oleh PHP di dalam variabel sistem PHP dengan nama PHPSESSID
2. PHP menyimpan nilai yang akan Anda simpan dalam session di dalam file yang berada di server.
Nama file tempat penyimpanan session tersebut sesuai (sama) dengan ID session. File disimpan dalam suatu direktori yang ditunjukkan oleh session.save_path dalam file php.ini.
3. PHP melempar ID session ke setiap halaman.
4. PHP mengambil nilai session dari file session untuk setiap halaman session.

Program 9.1

*Nama File : session01.php
Deskripsi : Program menciptakan session.*

```

1 <?php
2 /*********************************************************************
3 Nama file : session01.php
4 Halaman ini merupakan halaman contoh penciptaan session.
5 Perintah session_start() harus ditaruh di perintah pertama
6 tanpa spasi di depannya. Perintah session_start() harus ada
7 pada setiap halaman yang berhubungan dengan session
8 *****/
9 session_start();
10 if (isset ($_POST['Login'])) {
    $user = $_POST['user'];

```

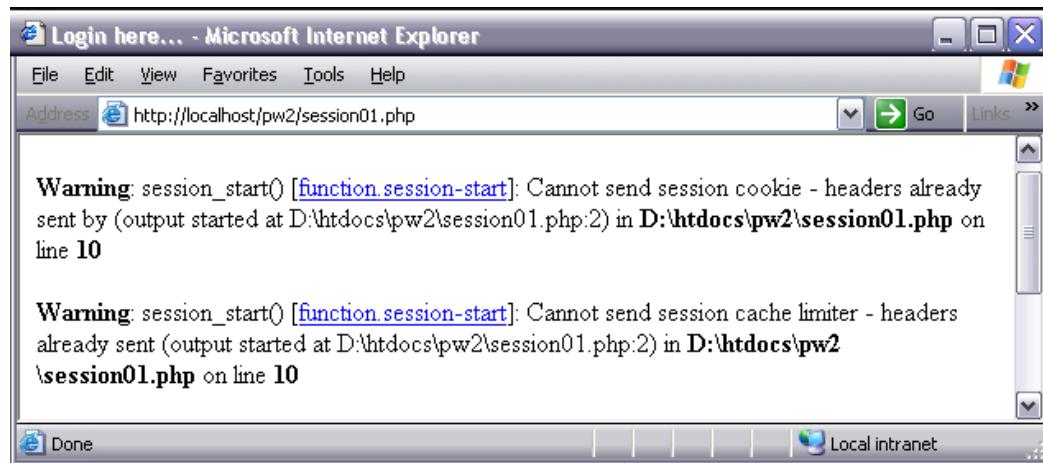
```

12     $pass = $_POST['pass'];
13     //periksa login
14     if ($user == "achmatim" && $pass = "123") {
15         //menciptakan session
16         $_SESSION['login'] = $user;
17         //menuju ke halaman pemeriksaan session
18         echo "<h1>Anda berhasil LOGIN</h1>";
19         echo "<h2>Klik <a href='session02.php'>di sini
(session02.php)</a>
untuk menuju ke halaman pemeriksaan session";
20     }
21 } else {
22 ?>
23 <html>
24     <head>
25         <title>Login here...</title>
26     </head>
27     <body>
28         <form action="" method="post">
29             <h2>Login Here...</h2>
30             Username : <input type="text" name="user"><br>
31             Password : <input type="password" name="pass"><br>
32             <input type="submit" name="Login" value="Log In">
33         </form>
34     </body>
35 </html>
36 <? } ?>
37

```

Penjelasan Program 9.1.

Pada program di atas terdapat fungsi **session_start()** yang berada pada baris ke-9. Fungsi session_start() berfungsi untuk memulai sebuah session. Fungsi harus dipanggil pertama kali dalam suatu halaman PHP, artinya fungsi session_start() harus dipanggil sebelum sesuatu tampil di layar, walaupun hal itu hanya berupa spasi. Jika sebelum session_start() terdapat output yang ditampilkan maka akan terjadi error seperti pada gambar 9.1 berikut ini :



Gambar 9.1. Error yang terjadi karena sebelum session_start() sudah ada output

Selanjutnya, program 9.1 akan menampilkan sebuah form login sederhana di layar. Form terdiri dari form inputan username, password dan sebuah tombol login. Jika tombol login ditekan maka pemeriksaan kondisi pada baris 10 akan bernilai TRUE dan perintah yang ada di blok **if** akan dieksekusi. Baris 11 dan 12 merupakan perintah untuk mengambil nilai pada form inputan username dan password. Nilai username dan password tersebut diperiksa pada baris 14. Pemeriksaan username dan password ini, untuk aplikasi web yang sudah

menggunakan database, bisa diganti dengan pemeriksaan username dan password ke tabel database.

Pada baris 16, terdapat perintah sebagai berikut :

```
$_SESSION['login'] = $user;
```

Baris perintah tersebut merupakan perintah untuk membuat session baru (*create session*) dimana nama session adalah "login" dan isi dari session adalah \$user. \$_SESSION merupakan sebuah variabel array global yang didefinisikan oleh PHP, jadi variabel ini harus ditulis dengan huruf kapital (*upper-case*).

Selanjutnya, pada baris 18-20 akan ditampilkan pesan bahwa login berhasil dan juga sebuah link menuju ke halaman pemeriksaan session (session02.php, program 9.2).

Program 9.2

Nama File : session02.php

Deskripsi : Program pemeriksaan session.

```
1 <?php
2 /*********************************************************************
3 Halaman ini merupakan contoh halaman pemeriksaan session. Peme-
4 riksaan session biasanya dilakukan jika suatu halaman memiliki
5 akses terbatas. Misalnya harus login terlebih dahulu.
6 *****/
7 session_start();
8
9 //pemeriksaan session
10 if (isset($_SESSION['login'])) { //jika sudah login
11     //menampilkan isi session
12     echo "<h1>Selamat Datang ". $_SESSION['login'] . "</h1>";
13     echo "<h2>Halaman ini hanya bisa diakses jika Anda sudah
14 login</h2>";
15     echo "<h2>klik <a href='session03.php'>di sini
16 (session03.php)</a> untuk LOGOUT</h2>";
17 } else {
18     //session belum ada artinya belum login
19     die ("Anda belum login! Anda tidak berhak masuk ke halaman
ini.Silahkan login <a href='session01.php'>di sini</a>");
20 }
21 ?>
```

Penjelasan Program 9.2

Program 9.2 di atas diawali dengan pemanggilan fungsi **session_start()**. Hal ini berarti dalam halaman program ini terdapat operasi session. Pada baris 10 program di atas, terdapat pemeriksaan keberadaan variabel \$_SESSION['login'] dengan menggunakan fungsi **isset()**. Pemeriksaan kondisi ini akan bernilai TRUE jika \$_SESSION['login'] sudah terbentuk. Jika bernilai TRUE maka blok program baris 11-14 yang akan dieksekusi, dan sebaliknya, jika FALSE maka blok program baris 16-17 yang akan dijalankan. Blok program baris 11-14 hanya bisa diakses saat user sudah login dengan benar. Dengan kata lain, bagian tersebut merupakan bagian yang bersifat private (terbatas).

Untuk menampilkan isi variabel session dapat dilakukan seperti pada baris 12 program di atas. Cukup dengan meng-echo variabel \$_SESSION['login'].

Jika user belum login (tidak melewati halaman login) atau langsung mengakses halaman ini, maka akan ditampilkan pesan bahwa user tidak berhak masuk ke halaman ini. Fungsi **die()** pada baris 17 digunakan untuk menampilkan suatu pesan sekaligus men-terminate program, artinya perintah selain die() akan diabaikan.

Program 9.3

Nama File : session03.php
Deskripsi : Program menghapus session.

```
1 <?php
2 ****
3 Halaman ini merupakan halaman logout, dimana kita menghapus
4 session yang ada.
5 ****
6 session_start();
7 if (isset($_SESSION['login'])) {
8     unset ($_SESSION);
9     session_destroy();
10    //
11    echo "<h1>Anda sudah berhasil LOGOUT</h1>";
12    echo "<h2>Klik <a href='session01.php'>di sini</a> untuk
13 LOGIN kembali</h2>";
14    echo "<h2>Anda sekarang tidak bisa masuk ke halaman
15 <a href='session02.php'>session02.php</a> lagi</h2>";
16 }?
17 ?>
```

Penjelasan Program 9.3.

Program 9.3 merupakan contoh program penghapusan session. Penghapusan session biasanya digunakan ketika user melakukan logout. Proses penghapusan session dilakukan dengan memanggil fungsi **unset()** dan fungsi **session_destroy()**. Fungsi **unset()** digunakan untuk menghapus atau menghilangkan suatu variabel. Sedangkan **session_destroy()** digunakan untuk menghapus semua data yang berhubungan dengan session tersebut.

Cookies

Seperti halnya session, cookies juga merupakan sebuah konsep penyimpanan informasi user. Hanya saja, jika session tempat penyimpanan berada di server, cookies berada di client. Oleh karena itu, konsep cookies sebaiknya jangan digunakan untuk menyimpan informasi login user seperti username, password dsb. Selain user bisa melihat informasi yang disimpan, user juga bisa men-disable cookies itu sendiri. Jika cookies di-disable, maka program yang memanfaatkan cookies tentunya tidak akan berjalan dengan baik.

Cookies sendiri biasanya dipakai dalam aplikasi shopping cart. Biasa digunakan untuk menyimpan sementara, produk-produk yang dipilih oleh pengunjung pada saat berbelanja.

Dalam penanganan cookies juga terdapat beberapa proses yang perlu diperhatikan :

- Proses pembuatan cookies
- Proses pemeriksaan cookies
- Proses penghapusan cookies

Program 9.4

Nama File : cookie01.php
Deskripsi : Program membuat cookie.

```
1 <?php
2 $value = 'achmatim';
3 $value2 = 'Achmad Solichin';
4
5 setcookie("username", $value);
6 setcookie("namalengkap", $value2, time() + 3600); /* expire in 1
hour */
7
8 echo "<h1>Ini halaman pengesetan cookie</h1>";
9 ?>
```

```
10 echo "<h2>Klik <a href='cookie02.php'>di sini</a> untuk  
11 pemeriksaan cookies</h2>";  
12 ?>
```

Program 9.5

Nama File : cookie02.php

Deskripsi : Program pemeriksaan cookie.

```
1 <?php  
2  
3 if(isset($_COOKIE['username'])) {  
4     echo "<h1>Cookie 'username' ada. Isinya : " .  
5     $_COOKIE['username'];  
6 } else {  
7     echo "<h1>Cookie 'username' TIDAK ada.</h1>";  
8 }  
9 if(isset($_COOKIE['namalengkap'])) {  
10    echo "<h1>Cookie 'namalengkap' ada. Isinya : " .  
11    $_COOKIE['namalengkap'];  
12 } else {  
13    echo "<h1>Cookie 'namalengkap' TIDAK ada.</h1>";  
14 }  
15 echo "<h2>Klik <a href='cookie01.php'>di sini</a> untuk  
16 penciptaan cookies</h2>";  
17 echo "<h2>Klik <a href='cookie03.php'>di sini</a> untuk  
18 penghapusan cookies</h2>";  
19 ?>
```

Program 9.6

Nama File : cookie03.php

Deskripsi : Program penghapusan cookie.

```
1 <?php  
2 // set the expiration date to one hour ago  
3 setcookie ("username", "", time() - 3600);  
4 setcookie ("namalengkap", "", time() - 3600);  
5 echo "<h1>Cookie Berhasil dihapus.</h1>";  
6 echo "<h2>Klik <a href='cookie01.php'>di sini</a> untuk  
7 penciptaan cookies</h2>";  
8 echo "<h2>Klik <a href='cookie02.php'>di sini</a> untuk  
9 pemeriksaan cookies</h2>";  
10 ?>
```

Bab 10

Pemrograman Berorientasi Objek dalam PHP

- ❖ Pengantar Pemrograman Berorientasi Objek
- ❖ Object dan Class
- ❖ Properties dan Method
- ❖ Mendefinisikan Class

Pengantar Pemrograman Berorientasi Objek

PHP pada awalnya hanyalah kumpulan script sederhana. Dalam perkembangannya, selanjutnya ditambahkan berbagai fitur pemrograman berorientasi objek. Hal ini dimulai sejak PHP 4. Dengan lahirnya PHP 5, fitur-fitur pemrograman berorientasi objek semakin mantap dan semakin cepat. Dengan PHP 5, script yang menggunakan konsep *object-oriented* akan lebih cepat dan lebih efisien.

Pemrograman berorientasi objek atau *object-oriented programming* (OOP) merupakan suatu pendekatan pemrograman yang menggunakan object dan class. Saat ini konsep OOP sudah semakin berkembang. Hampir setiap perguruan tinggi di dunia mengajarkan konsep OOP ini pada mahasiswanya. Pemrograman yang banyak dipakai dalam penerapan konsep OOP adalah Java dan C++.

OOP bukanlah sekedar cara penulisan sintaks program yang berbeda, namun lebih dari itu, OOP merupakan cara pandang dalam menganalisa sistem dan permasalahan pemrograman. Dalam OOP, setiap bagian dari program adalah *object*. Sebuah *object* mewakili suatu bagian program yang akan diselesaikan.

Beberapa konsep OOP dasar, antara lain :

1. *Encapsulation* (*Class* dan *Object*)
2. *Inheritance* (Penurunan sifat), dan
3. *Polymorphism*

PHP khususnya PHP 5 sudah mendukung beberapa konsep OOP. Akan tetapi PHP 5 tidak mendukung konsep *Multiple-inheritance* dan *polymorphism*.

Object dan Class

Bagian dasar dari sebuah program yang berorientasi objek adalah **objects**. Secara mudah kita dapat memahami mengenai *object* ini. Sebagai contoh, sebuah **mobil** adalah objek. Sebuah mobil mempunyai *properties* atau bagian-bagian di dalamnya, seperti warna, mesin, roda, pintu dsb. Sebuah mobil juga dapat melakukan sesuatu (ada sesuatu yang bisa dilakukan dengan mobil), seperti mengisi bensin, menyalakan mesin, berjalan, mengerem dsb.

Biasanya *object* adalah sebuah kata benda. **Orang** adalah *object*. Demikian juga mobil, pohon, bunga, komputer, TV, buku dsb. Namun, *object* tidak selamanya sebuah objek fisik. Bisa saja sebuah benda abstrak, seperti *account* bank, sebuah file di komputer, database, pesan email, acara TV, dsb.

Class merupakan penjelasan atau deskripsi dari *object*. Di dalam class, terdapat penjelasan tentang suatu *object* termasuk *properties* yang dimilikinya serta kelakuan atau *method* yang bisa dilakukan oleh *object*. Sebagai contoh, *class* **Orang**. Class Orang tentu setidaknya memiliki beberapa bagian seperti tangan, kaki, mata, telinga dsb. Class Orang juga setidaknya harus bisa jalan, bisa loncat, bisa lari, bisa melihat, bisa bicara dsb.

Salah satu keuntungan program didefinisikan dengan konsep OOP adalah adanya pengkapsulan (*encapsulation*) program dalam *class* dan *object*, dimana programmer yang menggunakan *class* tidak perlu mengetahui isi dan jalannya *class* secara detail, hanya perlu tahu bagaimana cara menggunakannya. Sama halnya dengan sebuah **mobil** misalnya. Seorang pemilik mobil tentunya tidak

perlu mengetahui bagian-bagian mobil secara menyeluruh. Dia tidak perlu mengetahui bagaimana mesin mobil melakukan pembakaran dan bagaimana mesin mobil bisa menggerakkan roda, dsb. Dia hanya perlu tahu bagaimana cara menjalankan mobil, bagaimana menghentikan mobil, dan fungsi mobil lainnya.

Properties dan Method

Setiap class memiliki *properties* yang kadang disebut juga *attributes*. *Properties* dari sebuah mobil misalnya warna, ukuran, harga dsb. Di dalam class, *properties* dinyatakan dengan sebuah variabel. Misalnya \$warna, \$harga, dsb.

Method merupakan sesuatu yang bisa dilakukan oleh *object*. Method dalam PHP sama artinya dengan sebuah fungsi. Method yang mungkin dipunyai dari sebuah mobil misalnya, method untuk menghidupkan mobil, menjalankan mobil, menghentikan mobil, dsb.

Penamaan *properties* dan *method* memiliki aturan yang sama dengan penamaan sebuah variabel atau fungsi. Akan tetapi berdasarkan kesepakatan (*convention*), penamaan *properties* dan *method* harus menggunakan *camel Caps*, dimana tiap kata diawali dengan huruf besar kecuali kata pertama, setiap kata digabung tanpa spasi atau *under-score* (_).

Mendefinisikan Class

Bentuk umum mendefinisikan sebuah *class* adalah sbb :

```
class namaClass
{
    Deklarasikan dan definisikan properties di sini
    Definisikan semua method di sini
}
```

Penamaan **namaClass** pada dasarnya sama dengan penamaan variabel. Penamaan bebas, boleh apa saja, kecuali **stdClass**. PHP sudah menggunakan nama **stdClass** sebagai nama *class built-in*. Isi tubuh class terletak di antara tanda kurung kurawal buka ({) dan kurawal tutup (}). Di tubuh class terdapat pendefinisian properties (variabel) dan method-method class.

Menambahkan Properties (Variabel)

Perhatikan contoh pendefinisian class beserta propertiesnya, sbb :

```
1 class Mobil
2 {
3     var $warna;
4     var $merk;
5     var $harga;
6
7     //Tambahkan pendefinisian method di sini
8 }
```

Pendeklarasian variabel dalam class seperti pada contoh di atas, bukanlah suatu keharusan, karena dalam PHP variabel tidak perlu dideklarasikan, cukup digunakan saja. Namun demikian, pendeklarasian variabel yang digunakan sangat dianjurkan untuk kemudahan pembacaan dan pemahaman program.

Variabel class juga dapat langsung diinisialisasi dengan sebuah nilai. Namun inisialisasi variabel tidak boleh mengandung operasi aritmatika maupun operasi yang lainnya. Perhatikan contoh sebagai berikut :

```
1 class Mobil
2 {
3     var $warna = "Biru";
4     var $merk = "BMW";
5     var $harga = "10000000";
6 }
```

```

7 } //Tambahkan pendefinisian method di sini
8 }
```

Menambahkan Method

Untuk menambahkan method, tinggal mendefinisikan method seperti halnya fungsi biasa. Misalkan kita akan menambahkan fungsi atau method untuk mengganti warna mobil dan untuk menampilkan warna mobil. Nama method pada dasarnya terserah (mengikuti aturan nama variabel). Namun demikian, jangan menggunakan nama method yang diawali dengan dua buah *under-score* (_) yaitu **__construct()**, **__destruct()** dan **__clone()** karena ketiga fungsi tersebut mempunyai arti tersendiri di PHP.

Lihat contoh menambahkan method sbb :

```

1 class Mobil
2 {
3     var $warna = "Biru";
4     var $merk = "BMW";
5     var $harga = "10000000";
6
7     function gantiwarna ($warnaBaru)
8     {
9         $this->warna = $warnaBaru;
10    }
11
12    function tampilwarna ()
13    {
14        echo "warna mobilnya : " . $this->warna;
15    }
16 }
```

Pada listing program di atas sudah ditambahkan fungsi atau method **gantiWarna()** dan **tampilWarna()**. Untuk mengakses *properties* (variabel) bisa menggunakan keyword **\$this**. Keyword ini me-refer pada *class* tempatnya berada. Method gantiWarna() mempunyai satu parameter, yaitu \$warnaBaru.

Menambahkan Konstruktur

Konstruktur merupakan sebuah *method* khusus yang akan secara otomatis dijalankan saat *object* terbentuk. Konstruktur tidak harus ada, namun dalam satu *class* hanya boleh ada satu konstruktur. Method konstruktur biasanya berisi pemberian nilai *default* dari masing-masing *properties* (variabel).

Untuk membuat konstruktur, cukup dengan mendefinisikan suatu fungsi dengan nama **__construct()**. Perhatikan contoh sebagai berikut :

```

1 class Mobil
2 {
3     var $warna;
4     var $merk;
5     var $harga;
6
7     function __construct()
8     {
9         $this->warna = "Biru";
10        $this->merk = "BMW";
11        $this->harga = "10000000";
12    }
13
14     function gantiwarna ($warnaBaru)
15     {
16         $this->warna = $warnaBaru;
17     }
18
19     function tampilwarna ()
20     {
21         echo "warna mobilnya : " . $this->warna;
```

```
22 }  
23 }
```

Membentuk Objek Class

Untuk menggunakan sebuah objek, harus dibentuk objek dari classnya. Dari sebuah class bisa dibentuk beberapa objek sekaligus. Bentuk umum pembentukan objek adalah sbb :

```
$namaObjek = new namaClass();
```

Sedangkan untuk memanggil anggota (member) dari class dapat dengan format sbb :

```
$namaObjek->variabel;  
$namaObjek->namaMethod();
```

Program 10.1

Nama File : pbo01.php

Deskripsi : Program sederhana pendefinisian class dan pemanggilan class.

```
1 <?  
2 class Mobil  
3 {  
4     var $warna;  
5     var $merk;  
6     var $harga;  
7  
8     function __construct()  
9     {  
10        $this->warna = "Biru";  
11        $this->merk = "BMW";  
12        $this->harga = "10000000";  
13    }  
14    function gantiwarna ($warnaBaru)  
15    {  
16        $this->warna = $warnaBaru;  
17    }  
18    function tampilwarna ()  
19    {  
20        echo "warna mobilnya : " . $this->warna;  
21    }  
22}  
23  
24  
25 $a = new Mobil();  
26 $b = new Mobil();  
27 echo "<b>Mobil pertama</b><br>";  
28 $a->tampilwarna();  
29 echo "<br><b>Mobil pertama ganti warna<br>";  
30 $a->gantiwarna("Merah");  
31 $a->tampilwarna();  
32 //  
33 echo "<br><b>Mobil kedua</b><br>";  
34 $b->gantiwarna("Hijau");  
35 $b->tampilwarna();  
36 ?>
```

Program 10.2

Nama File : pbo02.inc.php

Deskripsi : Program class untuk membuat sebuah form inputan sederhana.

```
1 <?php  
2 /*  
3 Nama Class : Form
```

```

4  Deskripsi : Class untuk membuat form inputan text sederhana
5  */
6
7  class Form
8  {
9      var $fields = array();
10     var $action;
11     var $submit = "Submit Form";
12     var $jumlahField = 0;
13
14     function __construct($action, $submit)
15     {
16         $this->action = $action;
17         $this->submit = $submit;
18     }
19     function displayForm()
20     {
21         echo "<form action='".$this->action."' method='POST'>";
22         echo "<table width='100%'>";
23         for ($j=0; $j<count($this->fields); $j++) {
24             echo "<tr><td align='right'>".$this->
25 fields[$j]['label']."</td>";
26             echo "<td><input type='text' name='".$this->
27 fields[$j]['name']."'></td></tr>";
28         }
29         echo "<tr><td colspan='2'>";
30         echo "<input type='submit' value='".$this->submit.
31 "'></td></tr>";
32         echo "</table>";
33     }
34     function addField($name, $label)
35     {
36         $this->fields [$this->jumlahField]['name'] = $name;
37         $this->fields [$this->jumlahField]['label'] = $label;
38         $this->jumlahField++;
39     }
40 }
41 ?>
```

Program 10.3

Nama File : pbo03.php

Deskripsi : Program memanfaatkan Program 10.2 untuk membuat form inputan sederhana.

```

1 <?php
2 include "pbo02.inc.php";
3 echo "<html><head><title>Mahasiswa</title></head><body>";
4 $form = new Form ("", "Input Form");
5 $form->addField ("txtnim", "Nim");
6 $form->addField ("txtnama", "Nama");
7 $form->addField ("txtalamat", "Alamat");
8 echo "<h3>Silahkan isi form berikut ini :</h3>";
9 $form->displayForm();
10 echo "</body></html>";
11 ?>
```

Penjelasan Program 10.3

Program 10.3 di atas merupakan contoh pemanggilan class yang didefinisikan pada Program 10.2 (pbo02.inc.php). Untuk itu pertama kali program **pbo02.inc.php** harus di-include-kan dengan menggunakan fungsi include(). Perhatikan program baris ke-2. Untuk menambahkan form inputan text box, hanya perlu dengan memanggil fungsi **addField()**. Hasil tampilan dari program di atas dapat dilihat pada gambar 10.1



Gambar 10.1. Tampilan Program 10.3

Memanfaatkan Class yang Tersedia

Dengan adanya konsep pemrograman berorientasi objek, pada programmer PHP di seluruh dunia dapat lebih mudah berbagi teknik programming. Kita bisa membuat suatu class dan programmer lain dapat dengan mudah menggunakan tanpa perlu mengetahui proses jalannya class tersebut. Just use it! Di internet saat ini bertebaran begitu banyak class dengan fungsinya masing-masing yang tersedia secara cuma-cuma dan dapat dimodifikasi secara bebas. Situs yang berisi kumpulan-kumpulan class PHP yang cukup terkenal antara lain <http://www.phpclasses.org>. Silahkan berkunjung dan ambil class yang Anda inginkan. Selain itu, Anda juga bisa mendapatkan banyak class PHP di kumpulan *repository* PHP dengan nama PECL (*PHP Extention Community Library*). PECL dapat diakses di <http://pecl.php.net>.

Berikut ini beberapa aplikasi PHP yang dibuat dengan konsep OOP yang banyak digunakan dalam aplikasi berbasis web :

1. PEAR

PEAR merupakan singkatan dari PHP Extension and Application Repository. PEAR berisi kumpulan class PHP yang dapat digunakan sewaktu-waktu. Class yang cukup banyak digunakan dari PEAR adalah **PEAR::DB** yang merupakan *abstraction layer* antara aplikasi PHP dengan database. Dengan PEAR::DB program menjadi lebih fleksibel dan secara mudah dapat berganti jenis database apapun (MySQL, PosgreSQL, Oracle dsb). PEAR dapat di-download dan dilihat di situs <http://pear.php.net>

2. Smarty

Smarty merupakan sebuah *template-engine* untuk PHP. Dengan Smarty memungkinkan terjadi pemisahan antara logika program (PHP) dan tampilan halaman web. Jadi programmer dan *web-desainer* dapat mengerjakan sebuah *project* aplikasi secara paralel (bersamaan). Smarty merupakan salah satu *template-engine* yang banyak digunakan saat ini. Selain memisahkan antara logika program dan layout program, smarty juga dapat mempercepat akses ke halaman web karena smarty juga menggunakan konsep *cache* halaman web. Beberapa situs yang menggunakan Smarty antara lain www.bl.ac.id, www.blogsome.com, www.tikiwiki.org dan sebagainya. Smarty dapat di-download di situs <http://smarty.php.net>

3. FPDF

FPDF merupakan class PHP yang memungkinkan untuk membuat (generate) file PDF dengan PHP murni, tanpa menggunakan *library* PDFlib. F dari **FPDF** merupakan singkatan dari **Free**: jadi Anda bisa mendapatkannya secara

cuma-cuma dan juga dapat memodifikasinya. Informasi mengenai FPDF bisa didapat di situsnya <http://www.fpdf.org>

4. JpGraph

JpGraph merupakan library PHP untuk pembuatan grafik, baik dalam bentuk grafik batang, pie, chart, line dsb. JpGraph membutuhkan *extension* PHP berupa PHP GD. JpGraph dapat didownload secara gratis di <http://www.aditus.nu/jpgraph/>

Bab 11

Pengenalan DBMS dan MySQL

- ❖ Tentang Database, DBMS, dan RDBMS
- ❖ Tentang MySQL
- ❖ Instalasi MySQL di Windows
- ❖ Koneksi ke Server MySQL dengan MySQL Client
- ❖ Berbagai Tools Administrasi Server MySQL

Tentang Database, DBMS dan RDBMS

Basis data (atau **database**) adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut (<http://id.wikipedia.org/wiki/Database>). Database digunakan untuk menyimpan informasi atau data yang terintegrasi dengan baik di dalam komputer.

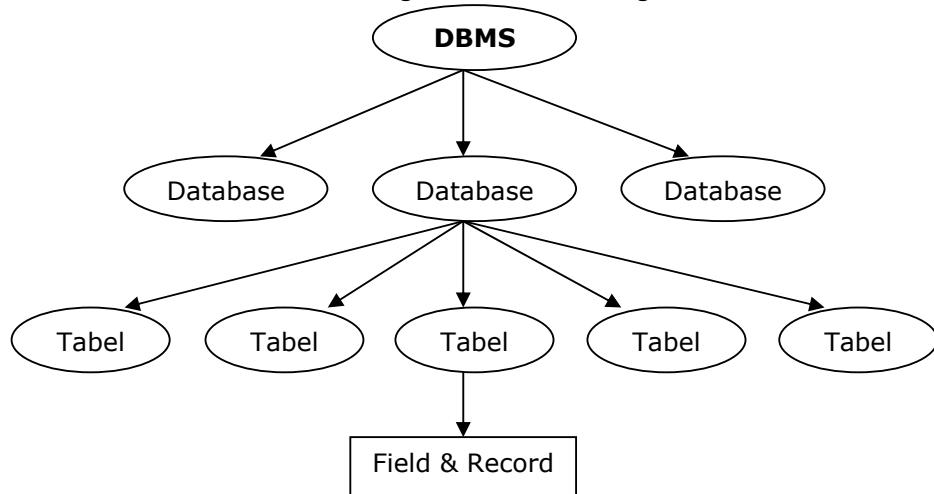
Untuk mengelola *database* diperlukan suatu perangkat lunak yang disebut **DBMS** (*Database Management System*). DBMS merupakan suatu sistem perangkat lunak yang memungkinkan user (pengguna) untuk membuat, memelihara, mengontrol, dan mengakses *database* secara praktis dan efisien. Dengan DBMS, user akan lebih mudah mengontrol dan memanipulasi data yang ada.

Sedangkan **RDBMS** atau *Relationship Database Management System* merupakan salah satu jenis DBMS yang mendukung adanya *relationship* atau hubungan antar tabel. Di samping RDBMS, terdapat jenis DBMS lain, misalnya *Hierarchy DBMS*, *Object Oriented DBMS*, dsb.

Beberapa software atau perangkat lunak DBMS yang sering digunakan dalam aplikasi program antara lain :

- DB2 - <http://www-306.ibm.com/software/data/db2/>
- Microsoft SQL Server - <http://www.microsoft.com/sql/>
- Oracle - <http://www.oracle.com>
- Sybase - <http://www.sybase.com/>
- Interbase - <http://www.borland.com/interbase>
- Teradata - <http://www.teradata.com/>
- Firebird - <http://www.firebirdsql.org/>
- MySQL - <http://www.mysql.com>
- PostgreSQL - <http://www.postgresql.org/>

Dalam konsep database, urutan atau hierarki database sangatlah penting. Urutan atau hierarki database digambarkan dalam gambar sbb :



Gambar 11.1. Hierarki Database

Tentang MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: *database management system*) atau DBMS yang multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis di bawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.

Tidak seperti Apache yang merupakan software yang dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia yaitu **MySQL AB**. MySQL AB memegang penuh hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson, dan Michael "Monty" Widenius.

Beberapa kelebihan MySQL antara lain :

- Free (bebas didownload)
- Stabil dan tangguh
- Fleksibel dengan berbagai pemrograman
- Security yang baik
- Dukungan dari banyak komunitas
- Kemudahan *management database*.
- Mendukung transaksi
- Perkembangan software yang cukup cepat.

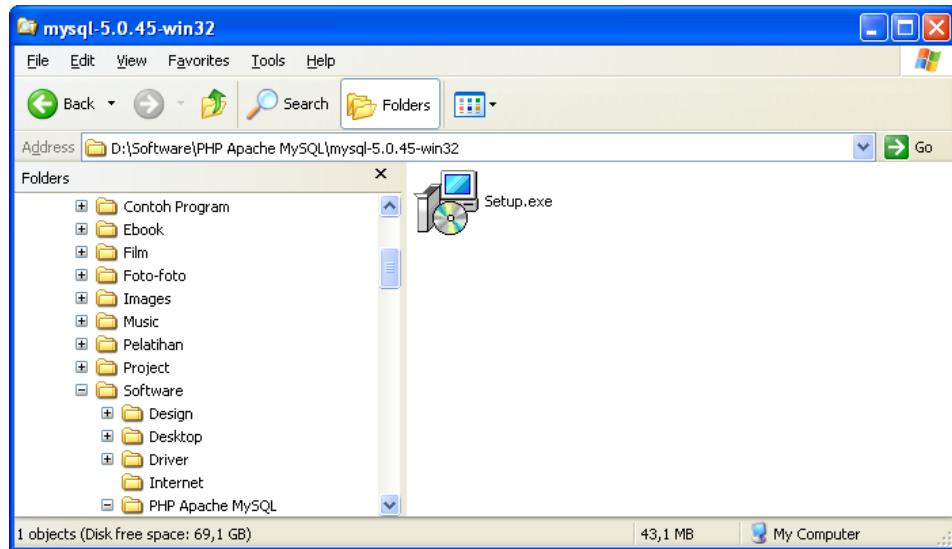
Instalasi MySQL di Windows

Persiapan

1. Download Source MySQL di <http://www.mysql.com/downloads/>
MySQL versi terakhir saat materi ini dibuat adalah MySQL 5.0.45. Silahkan Anda download versi terakhir tersebut dan simpan di komputer Anda. Pada dasarnya, instalasi untuk setiap versi MySQL tidak jauh berbeda.

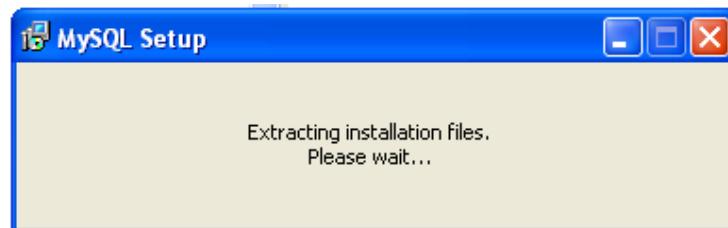
Proses Instalasi MySQL

1. Setelah Anda mendapatkan source MySQL, selanjutnya Anda perlu meng-ekstrak file tersebut ke komputer Anda.
2. Jalankan file **Setup.exe** yang ada di dalam folder source MySQL. Lihat gambar berikut ini !



Gambar 2.1. File Setup.exe

3. MySQL Setup akan mengekstrak file instalasi MySQL seperti pada gambar berikut ini.



Gambar 2.2. Proses Instalasi Dimulai

4. Selanjutnya akan ditampilkan window **MySQL Server 5.0 Setup Wizard for MySQL**. Klik tombol **Next** untuk memulai proses instalasi.



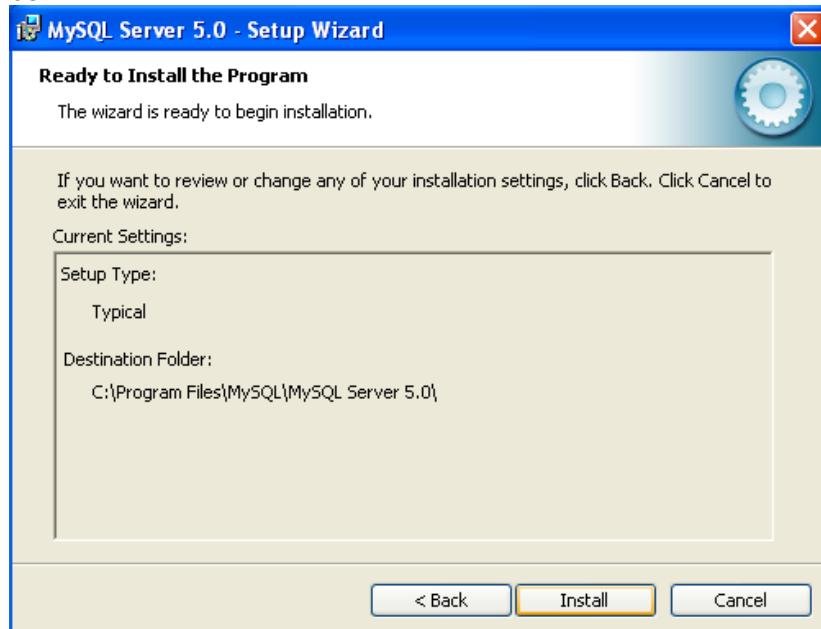
Gambar 2.3. Memulai Proses Instalasi

5. Selanjutnya akan ditampilkan pilihan untuk memilih cara instalasi. Pilih **Typical** jika kita ingin menginstall MySQL yang umumnya digunakan.



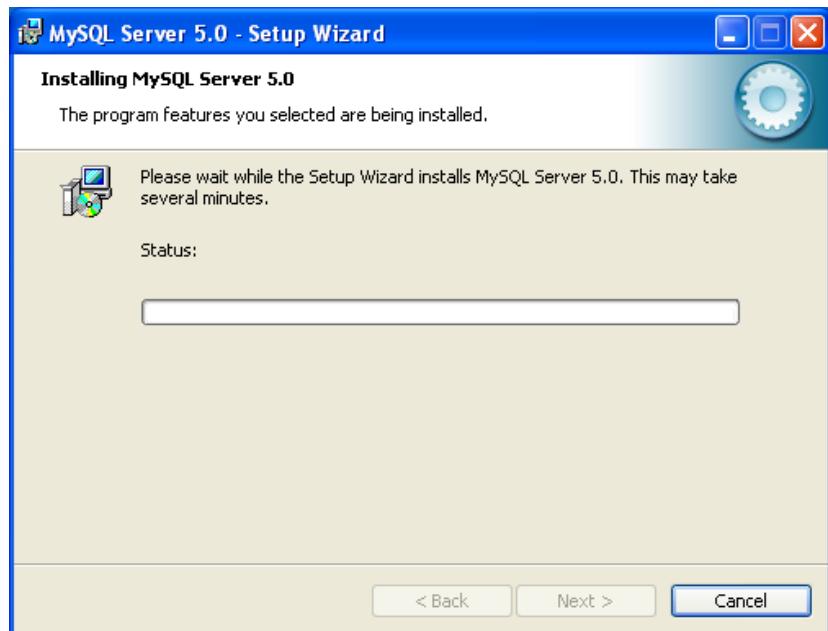
Gambar 2.4. Pilih tipe instalasi

6. Selanjutnya akan ditampilkan window informasi konfigurasi MySQL, yaitu tipe instalasi dan folder tujuan instalasi. Klik **Install** untuk memulai proses instalasi.



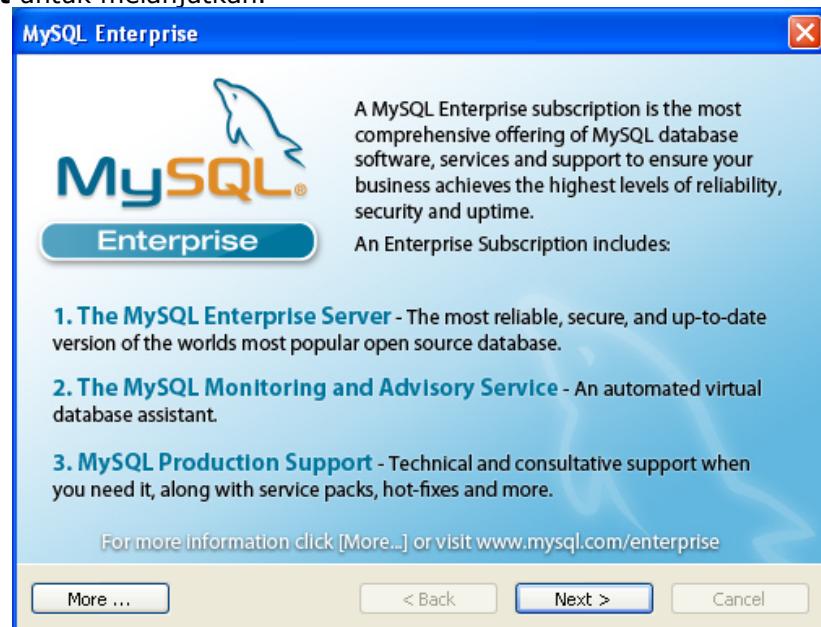
Gambar 2.5. Window Informasi Konfigurasi Instalasi

7. Proses instalasi dimulai.

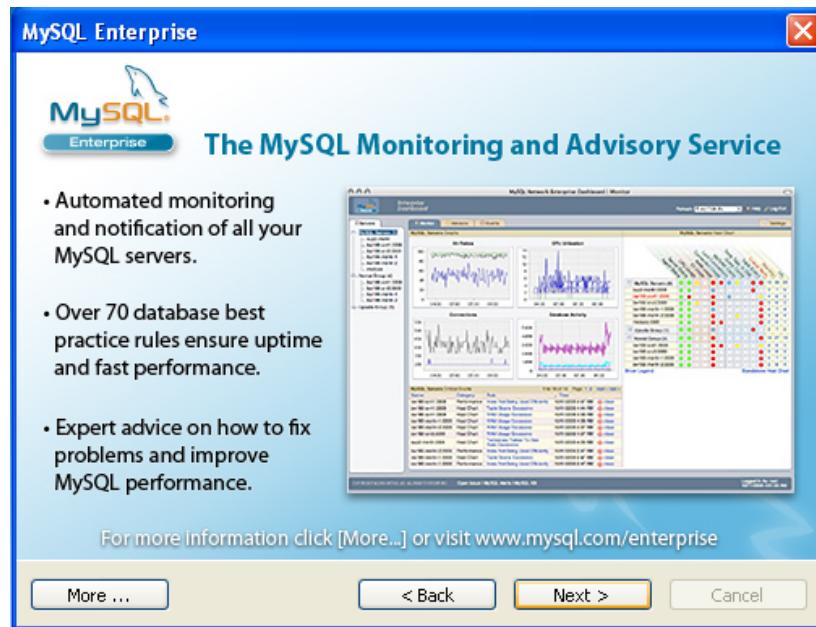


Gambar 2.6. Window Setup Type

8. Selanjutnya ditampilkan window informasi mengenai **MySQL Enterprise**. Klik **Next** untuk melanjutkan.

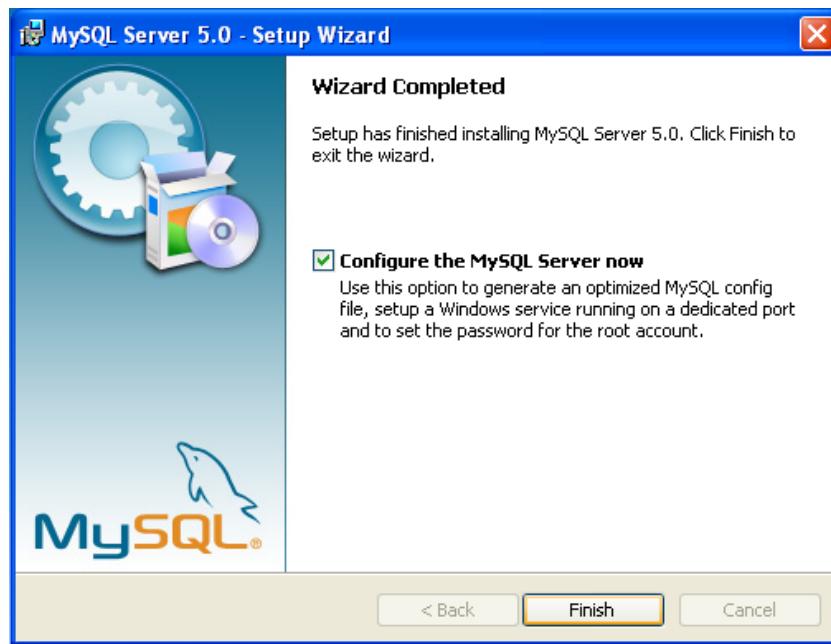


Gambar 2.7. Window MySQL Enterprise



Gambar 2.8. Window informasi MySQL Monitoring

9. Proses instalasi selesai dan akan ditampilkan seperti pada gambar di bawah ini. Jika kita ingin langsung mengkonfigurasikan server MySQL (password, service dll) maka pilihkan checkbox **Configure the MySQL Server now** dan tekan tombol **Finish**.



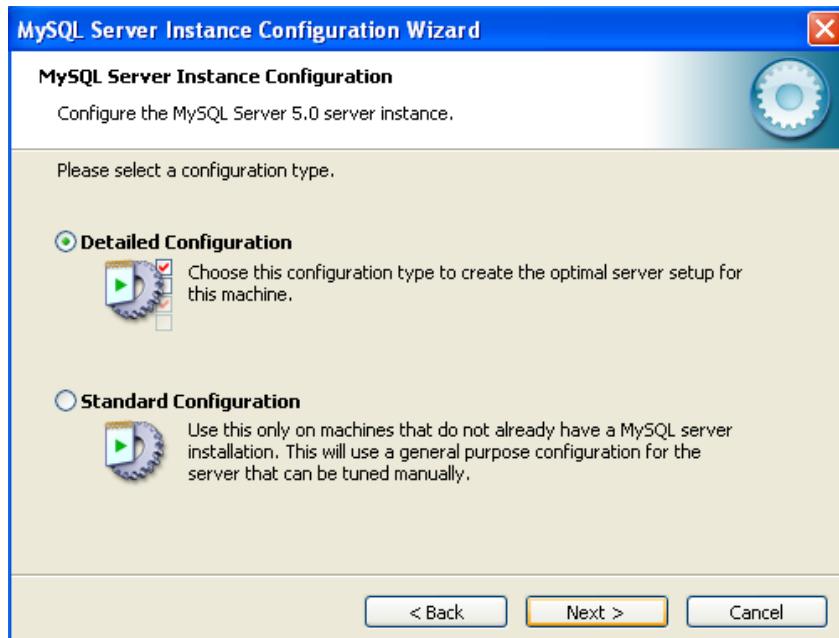
Gambar 2.9. Proses instalasi Selesai

10. Selanjutnya ditampilkan window **MySQL Server Instance Configuration Wizard**. Klik **Next** untuk melanjutkan.



Gambar 2.10. Window MySQL Server Instance Configuration Wizard

11. Selanjutnya terdapat pilihan tipe konfigurasi yang diinginkan, **Detailed Configuration** atau **Standard Configuration**. Pilih dan klik **Next** untuk melanjutkan.



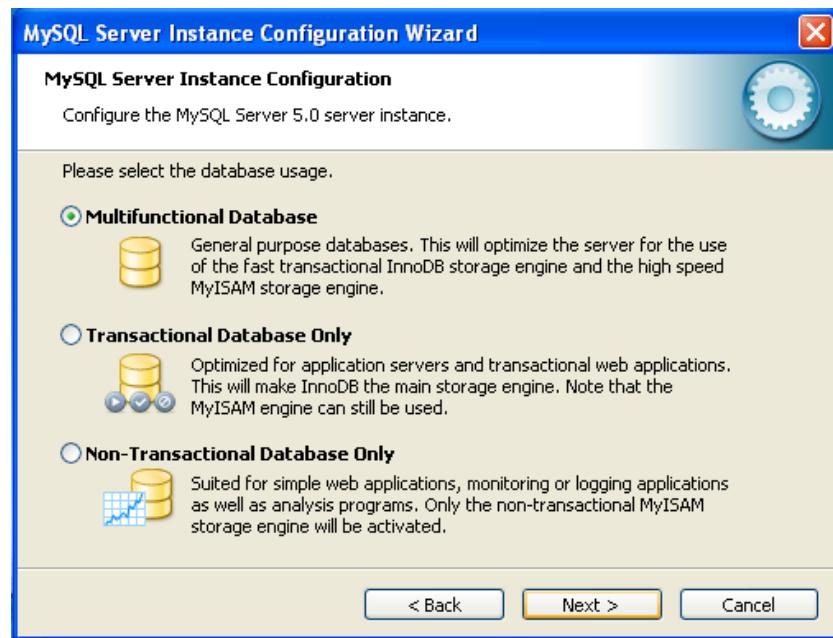
Gambar 2.11. Window Pilihan tipe konfigurasi

12. Selanjutnya terdapat pilihan tipe server yang diinginkan, **Developer, Server**, atau **Dedicated MySQL Server**. Pilih salah satu dan klik **Next** untuk melanjutkan.



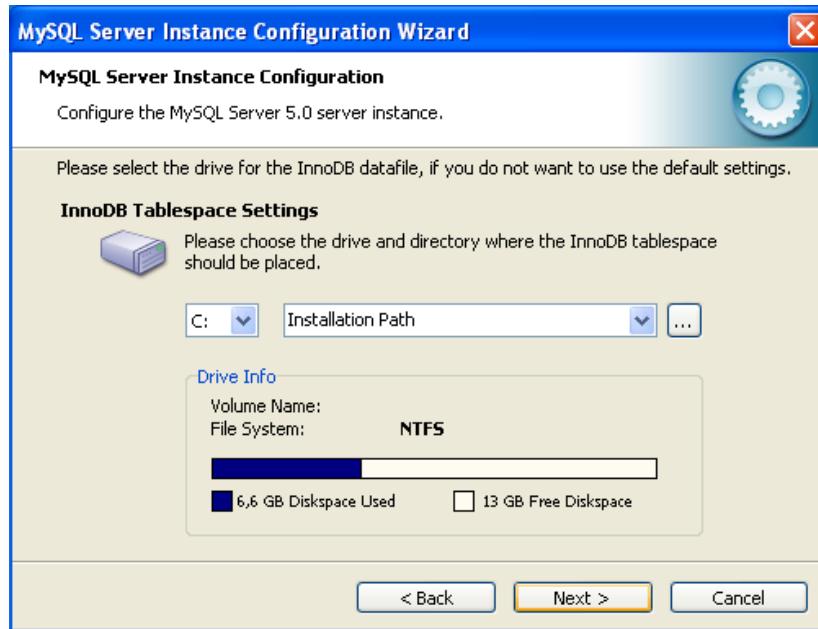
Gambar 2.12. Pilihan tipe server MySQL

13. Selanjutnya terdapat pilihan penggunaan database MySQL, untuk **Multifunctional, Transactional Only** atau **Non-Transactional Only**. Pilih salah satu dan klik **Next** untuk melanjutkan.



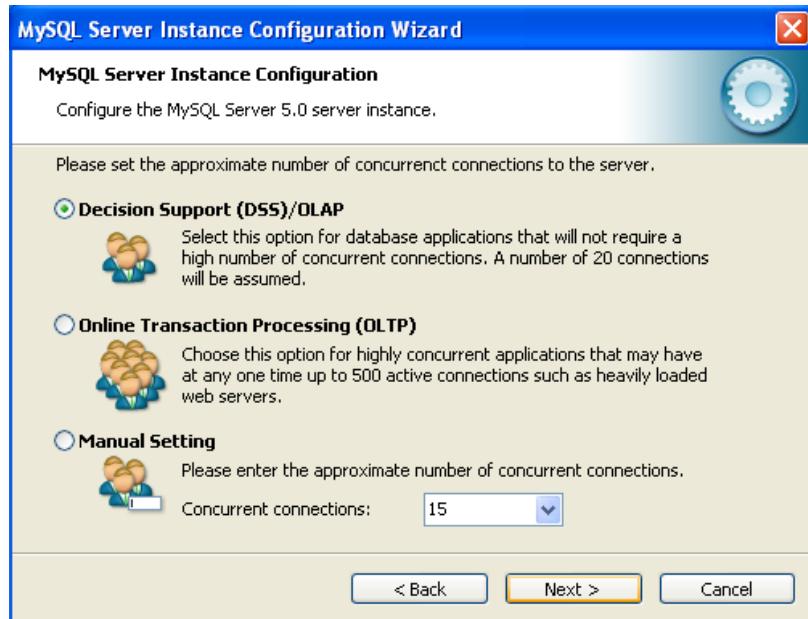
Gambar 2.13. Window Pilihan penggunaan Database.

14. Selanjutnya terdapat setting-an **InnoDB Tablespace Settings** dimana diminta memilih tempat untuk *tablespace* InnoDB. Klik **Next** untuk melanjutkan.



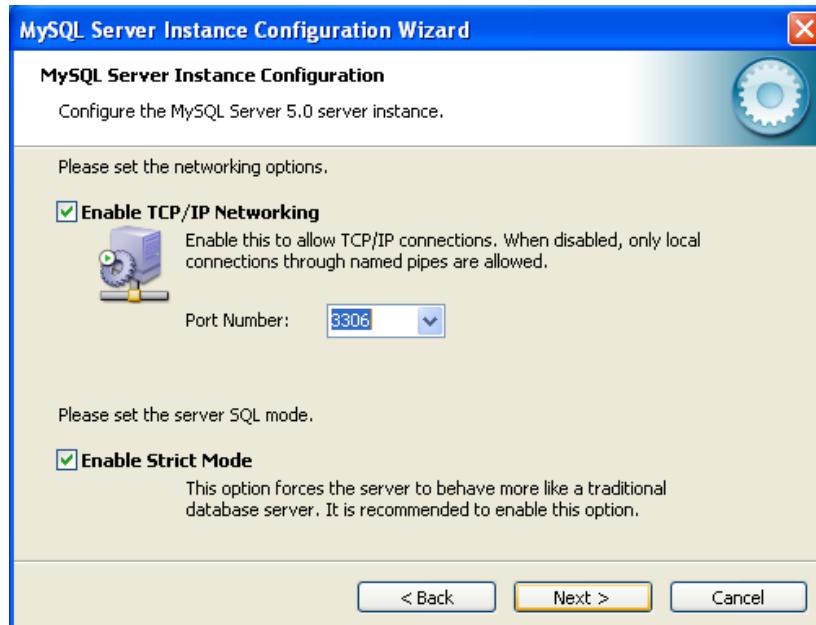
Gambar 2.14. Window InnoDB Tablespace Settings.

15. Selanjutnya terdapat pilihan perkiraan seberapa besar koneksi user ke server. Pilih salah satu dan klik **Next** untuk melanjutkan.



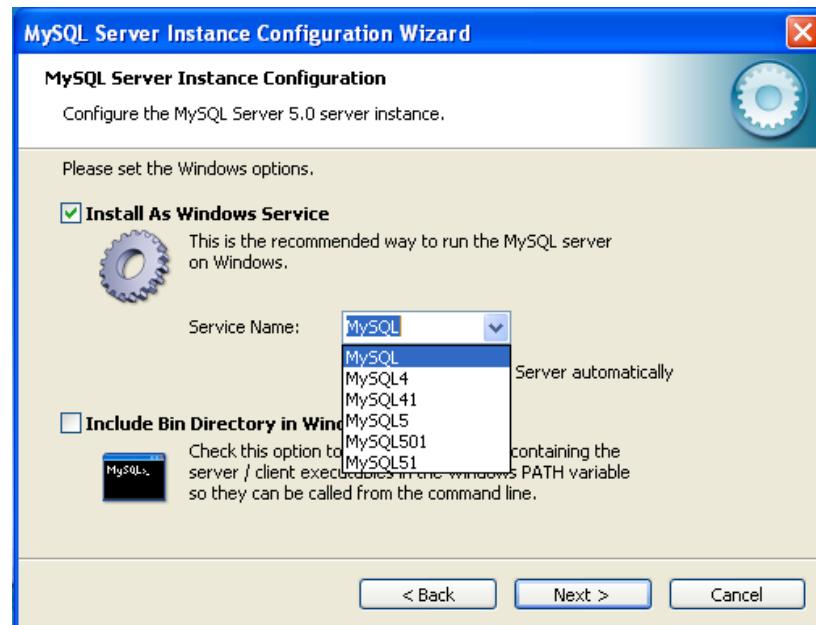
Gambar 2.15. Pilihan Perkiraan Seberapa Besar Koneksi User ke Server

16. Selanjutnya terdapat window untuk memilih nomor PORT yang digunakan untuk MySQL. **Next** untuk melanjutkan.



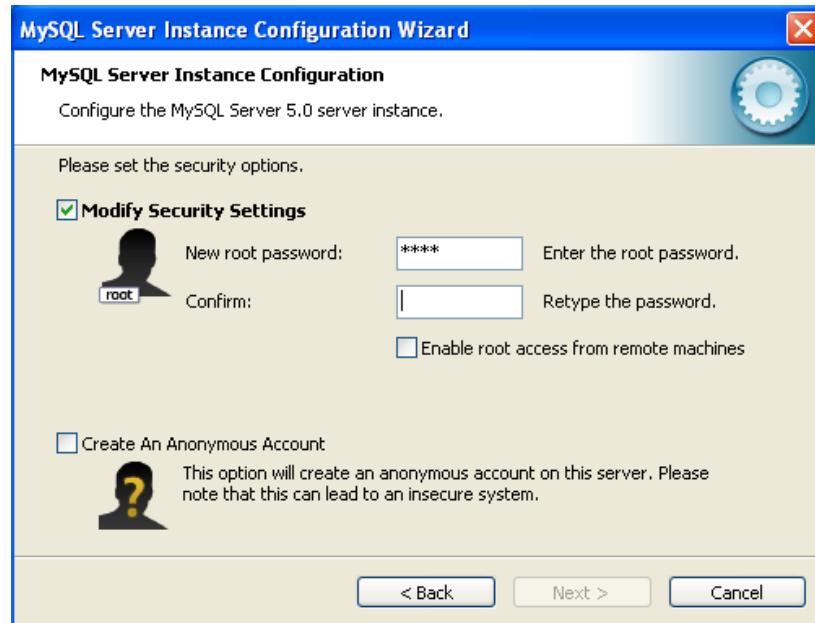
Gambar 2.16. Window pilihan port MySQL.

17. Selanjutnya terdapat pilihan nama service MySQL yang akan digunakan oleh Windows. Pilih salah satu dan klik **Next** untuk melanjutkan.



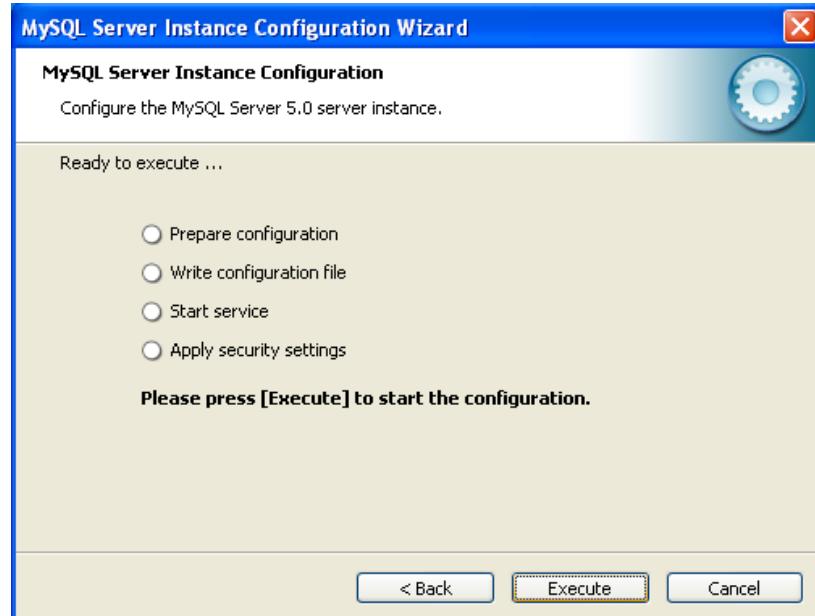
Gambar 2.17. Window pilihan Nama Service MySQL.

18. Selanjutnya diminta memodifikasi security. Pilih password untuk root (user tertinggi di MySQL) dan klik **Next** untuk melanjutkan.



Gambar 2.18. Window Security Setting.

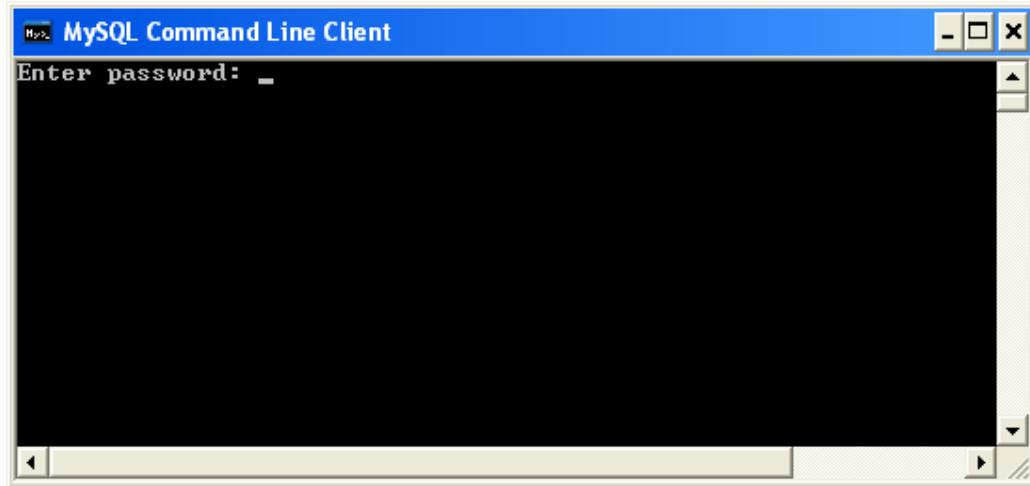
19. Proses konfigurasi selesai dan klik **Execute** untuk menyimpan konfigurasi dan menjalankan servis MySQL.



Gambar 2.19. Proses konfigurasi server MySQL selesai.

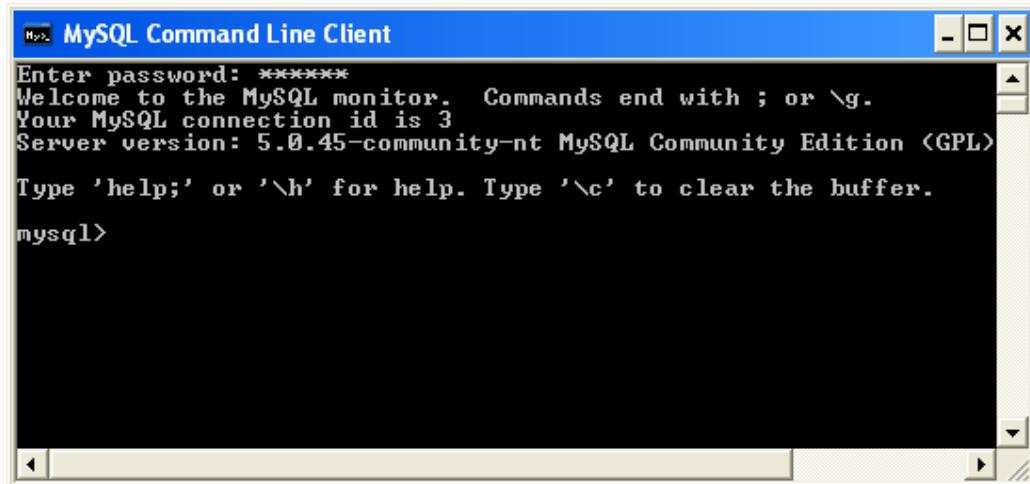
Koneksi ke Server MySQL dengan MySQL Client

MySQL menyediakan tools untuk melakukan koneksi ke server MySQL, yaitu MySQL Command-Line Client. Tools tersebut dapat diakses dari menu **Start > All Programs > MySQL > MySQL Server 5> MySQL Command Line Client**. Tampilannya kurang lebih tampak pada gambar berikut ini:



Gambar 12.13. MySQL Command Line Client

Untuk melakukan koneksi ke server MySQL, Anda cukup mengetikkan password koneksi MySQL. Password ini didefinisikan pada saat proses instalasi. Jika passwordnya benar, maka akan ditampilkan window sbb :



Gambar 2.20. Koneksi ke Server MySQL dengan User root

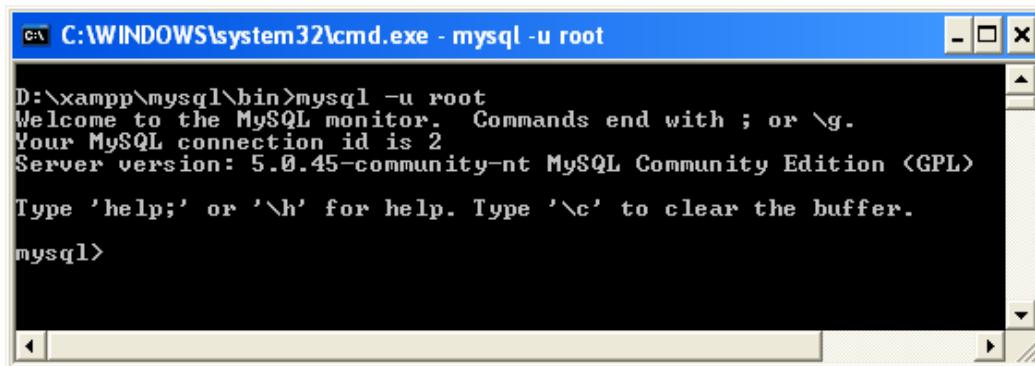
Setelah koneksi ke server MySQL berhasil dilakukan, maka akan ditampilkan prompt **mysql>** seperti pada gambar 12.14. Query atau perintah-perintah MySQL dapat dituliskan pada prompt MySQL ini. Akhiri setiap query dengan titik-koma (;). Selanjutnya untuk keluar dari server MySQL dapat dilakukan dengan mengetikkan perintah **quit** atau **\q** pada prompt **mysql>**.

Berbagai MySQL Client untuk Administrasi Server MySQL

Berikut ini beberapa *tools* yang biasa digunakan dalam mempermudah administrasi server MySQL. *Tools* berikut ini hanya digunakan untuk mempermudah administrasi MySQL, jadi tidak harus digunakan.

1. MySQL Command Line Client

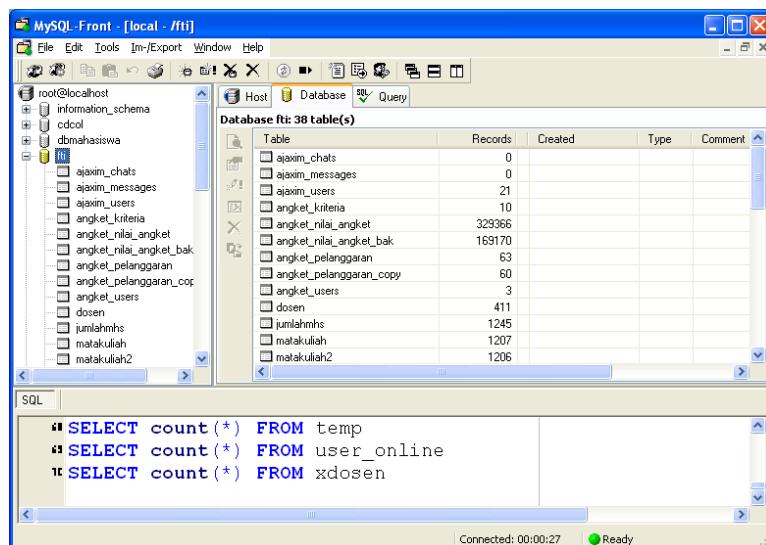
MySQL Command Line Client merupakan tools *default* MySQL yang sudah disertakan dalam file instalasi MySQL. Aplikasi ini dapat digunakan untuk melakukan koneksi ke MySQL melalui *text-based mode*.



Gambar 2.21. Tampilan MySQL command line client

2. MySQL-Front

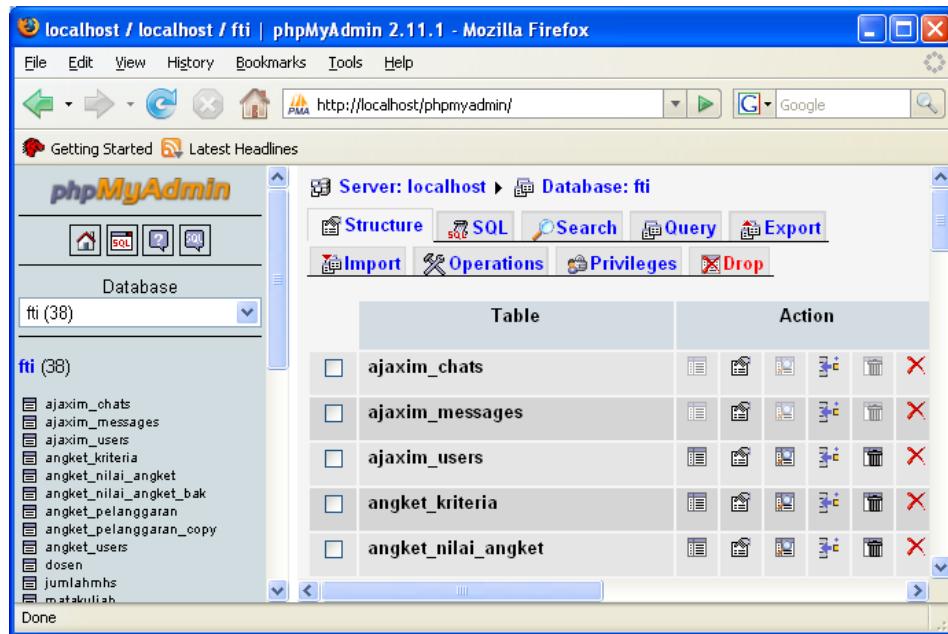
MySQL-Front merupakan front-end MySQL berbasis Windows yang cukup banyak digunakan. MySQL-Front memiliki user interface yang cukup mudah digunakan, bahkan oleh user pemula. Pada awalnya MySQL-Front merupakan software yang free, namun mulai versi 3.0 ke atas, software ini menjadi software yang bersifat shareware dengan masa percobaan selama 30 hari. Jika Anda ingin mencoba software ini, cobalah MySQL-Front versi 2.5 karena selain masih bebas untuk didownload, versi 2.5 cukup stabil dan sudah teruji. Situs resmi MySQL-Front beralamat di <http://www.mysqlfront.de>



Gambar 2.22. Tampilan MySQL Front

3. PHPMyAdmin

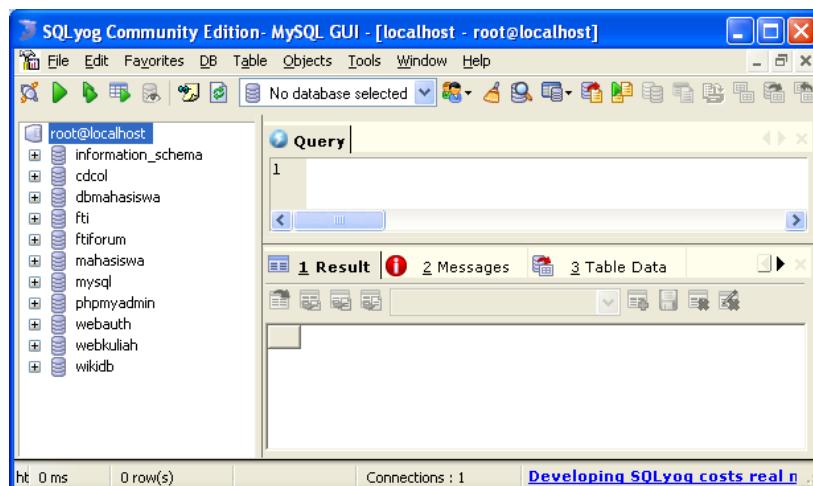
PHPMyAdmin merupakan front-end MySQL berbasis web. PHPMyAdmin dibuat dengan menggunakan PHP. Saat ini, PHPMyAdmin banyak digunakan dalam hampir semua penyedia hosting yang ada di internet. PHPMyAdmin mendukung berbagai fitur administrasi MySQL termasuk manipulasi database, tabel, index dan juga dapat mengeksport data ke dalam berbagai format data. PHPMyAdmin juga tersedia dalam 50 bahasa lebih, termasuk bahasa Indonesia. PHPMyAdmin dapat didownload secara gratis di <http://www.phpmyadmin.net>



Gambar 2.23. Tampilan halaman PHPMyAdmin

4. SQLYog

SQLYog merupakan salah satu front-end MySQL yang cukup populer saat ini. Dengan dukungan fitur yang cukup banyak dan lengkap, SQL Yogi tersedia versi commercial dan community (free). SQLYog dapat didownload di situsnya <http://www.webyog.com>



Gambar 2.24. Tampilan layar SQLYog

5. MySQL Administrator dan MySQL Query Browser

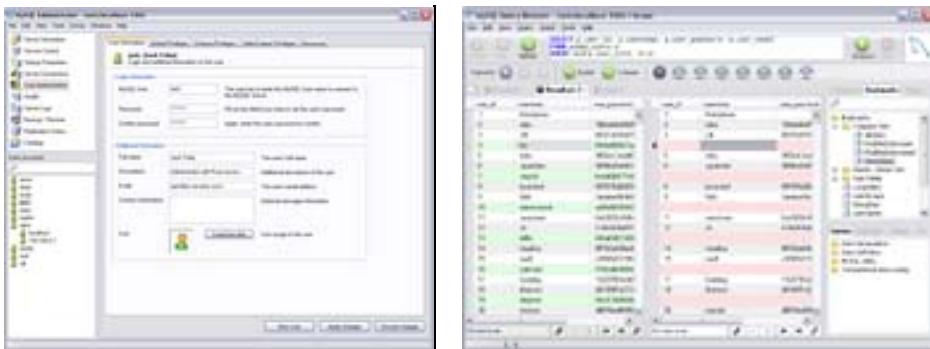
MySQL Administrator dan MySQL Query Browser merupakan tools administrasi database MySQL yang tersedia di situs resmi MySQL (<http://www.mysql.com>). Keduanya dapat didownload di alamat <http://www.mysql.com/products/tools/>.

Beberapa fitur **MySQL Administrator**, antara lain:

- Administrasi user.
- Halaman monitoring server.
- Optimatisasi MySQL
- Informasi umum keadaan server
- Status *replication*.
- *Cross-platform*.

Beberapa fitur **MySQL Query Browser**, antara lain:

- Tampilan dan menu yang mudah (*user-friendly*).
- Mendukung beberapa window hasil (*result preview*) sekaligus.
- Kemudahan dalam menulis query dengan *visual tools*.
- Manipulasi database.
- Membuat dan manipulasi tabel.
- SQL *statements debugging*.



Gambar 2.25. Tampilan layar MySQL Administrator dan MySQL Control Center

Bab 12

Pengenalan Structured Query Language

- ❖ Apa Itu SQL ?
- ❖ Membuat, Menampilkan, Membuka dan Menghapus Database
- ❖ Membuat, Mengubah dan Menghapus Table
- ❖ Menambah Record dengan INSERT
- ❖ Mengedit Record dengan UPDATE
- ❖ Menghapus Record dengan DELETE
- ❖ Menampilkan Record dengan SELECT

Apa Itu SQL ?

SQL merupakan singkatan dari *Structured Query Language*. SQL atau juga sering disebut sebagai query merupakan suatu bahasa (*language*) yang digunakan untuk mengakses database. SQL dikenalkan pertama kali dalam IBM pada tahun 1970 dan sebuah standar ISO dan ANSI ditetapkan untuk SQL. Standar ini tidak tergantung pada mesin yang digunakan (IBM, Microsoft atau Oracle). Hampir semua software database mengenal atau mengerti SQL. Jadi, perintah SQL pada semua software database hampir sama.

Terdapat 2 (dua) jenis perintah SQL, yaitu :

1. DDL atau Data Definition Language

DDL merupakan perintah SQL yang berhubungan dengan pendefinisian suatu struktur database, dalam hal ini *database* dan *table*. Beberapa perintah dasar yang termasuk DDL ini antara lain :

- CREATE
- ALTER
- RENAME
- DROP

2. DML atau Data Manipulation Language

DML merupakan perintah SQL yang berhubungan dengan manipulasi atau pengolahan data atau *record* dalam *table*. Perintah SQL yang termasuk dalam DML antara lain :

- SELECT
- INSERT
- UPDATE
- DELETE

Membuat, Menampilkan, Membuka dan Menghapus Database

Membuat Database

Sintaks umum SQL untuk membuat suatu database adalah sebagai berikut :

```
CREATE DATABASE [IF NOT EXISTS] nama_database;
```

Bentuk perintah di atas akan membuat sebuah database baru dengan nama *nama_database*. Aturan penamaan sebuah database sama seperti aturan penamaan sebuah variabel, dimana secara umum nama database boleh terdiri dari huruf, angka dan *under-score* (_). Jika database yang akan dibuat sudah ada, maka akan muncul pesan error. Namun jika ingin otomatis menghapus database yang lama jika sudah ada, aktifkan option IF NOT EXISTS. Setiap kita membuat database baru, maka sebenarnya MySQL akan membuat suatu folder (direktori) sesuai dengan nama databasenya yang ditempatkan secara *default* di C:\mysql\data. Di dalam folder tersebut nantinya akan terdapat file-file yang berhubungan dengan tabel dalam database.

Berikut ini contoh perintah untuk membuat database baru dengan nama "**mahasiswa**" :

```
| CREATE DATABASE mahasiswa;
```

Jika query di atas berhasil dieksekusi dan database berhasil dibuat, maka akan ditampilkan pesan sebagai berikut :

```
| Query OK, 1 row affected (0.02 sec)
```

Membuat Database

Untuk melihat database yang baru saja dibuat atau yang sudah ada, dapat menggunakan perintah sebagai berikut :

```
| SHOW DATABASES;
```

Hasil dari perintah di atas akan menampilkan semua database yang sudah ada di MySQL. Berikut ini contoh hasil dari query di atas :

```
+-----+
| Database      |
+-----+
| mahasiswa    |
| mysql         |
| test          |
+-----+
3 rows in set (0.02 sec)
```

Membuka Database

Sebelum melakukan manipulasi tabel dan record yang berada di dalamnya, kita harus membuka atau mengaktifkan databasenya terlebih dahulu. Untuk membuka database "**mahasiswa**", berikut ini querinya :

```
| USE mahasiswa;
```

Jika perintah atau query di atas berhasil, maka akan ditampilkan pesan sebagai berikut :

```
| Database changed
```

Menghapus Database

Untuk menghapus suatu database, sintaks umumnya adalah sbb :

```
DROP DATABASE [IF EXISTS] nama_database;
```

Bentuk perintah di atas akan menghapus database dengan nama `nama_database`. Jika databasenya ada maka database dan juga seluruh tabel di dalamnya akan dihapus. Jadi berhati-hatilah dengan perintah ini! Jika nama database yang akan dihapus tidak ditemukan, maka akan ditampilkan pesan error. Aktifkan option `IF EXISTS` untuk memastikan bahwa suatu database benar-benar ada.

Berikut ini contoh perintah untuk menghapus database dengan nama "**mahasiswa**" :

```
DROP DATABASE mahasiswa;
```

Membuat, Mengubah dan Menghapus Table

Membuat Table

Bentuk umum SQL untuk membuat suatu *table* secara sederhana sebagai berikut :

```
CREATE TABLE nama_tabel (
    field1 tipe.panjang,
    field2 tipe.panjang,
    ...
    fieldn tipe.panjang,
    PRIMARY KEY (field_key)
);
```

Bentuk umum di atas merupakan bentuk umum pembuatan tabel yang sudah disederhanakan. Penamaan tabel dan field memiliki aturan yang sama dengan penamaan database.

MySQL menyediakan berbagai tipe data dengan spesifikasi dan panjang masing-masing. Tipe data untuk field dalam MySQL diantaranya ditampilkan pada tabel berikut ini :

JENIS TIPE	TIPE	KETERANGAN
NUMERIK	TINYINT	-128 s/d 127 SIGNED 0 s/d 255 UNSIGNED
	SMALLINT	-32768 s/d 32767 SIGNED 0 s/d 65535 UNSIGNED.
	MEDIUMINT	-8388608 s/d 8388607 SIGNED 0 s/d 16777215 UNSIGNED
	INT	-2147483648 s/d 2147483647 SIGNED 0 s/d 4294967295 UNSIGNED.
	BIGINT	-9223372036854775808 s/d 9223372036854775807 SIGNED 0 s/d 18446744073709551615 UNSIGNED.
	FLOAT	Bilangan pecahan presisi tunggal
	DOUBLE	Bilangan pecahan presisi ganda
	DECIMAL	Bilangan dengan desimal
DATE/TIME	DATE	Tanggal dengan format YYYY-MM-DD
	DATETIME	Tanggal dan waktu dengan format : YYYY-MM-DD HH:MM:SS
	TIMESTAMP	Tanggal dan waktu dengan format : YYYYMMDDHHMMSS
	TIME	Waktu dengan format HH:MM:SS
	YEAR	Tahun dengan format YYYY
STRING	CHAR	0 – 255 karakter
	VARCHAR	0 – 255 karakter
	TINYTEXT	String dengan panjang maksimum 255 karakter
	TEXT	String dengan panjang maksimum 65535 karakter
	BLOB	String dengan panjang maksimum 65535

		karakter
	MEDIUMTEXT	String dengan panjang maksimum 16777215 karakter
	MEDIUMBLOB	String dengan panjang maksimum 16777215 karakter
	LONGTEXT	String dengan panjang maksimum 4294967295 karakter
	LONGBLOB	String dengan panjang maksimum 4294967295 karakter
<hr/>		
KHUSUS	ENUM	Tipe data dengan isi tertentu
	SET	Tipe data dengan isi tertentu

Sebagai contoh, kita akan membuat tabel baru dengan struktur sebagai berikut :

Nama tabel : **mhs**

No	Nama Field	Tipe	Panjang
1	nim	Varchar	10
2	nama	Varchar	30
3	tglahir	Date	-
4	alamat	Text	-

Untuk membuat tabel tersebut di atas, query atau perintah SQL-nya adalah sebagai berikut :

```
CREATE TABLE mhs (
    nim varchar(10) NOT NULL,
    nama varchar(30) NOT NULL,
    tglahir date,
    alamat text,
    PRIMARY KEY(nim)
);
```

Jika query untuk membuat tabel di atas berhasil dijalankan, maka akan ditampilkan pesan sebagai berikut :

```
Query OK, 0 rows affected (0.16 sec)
```

Pada perintah di atas, beberapa hal yang perlu diperhatikan :

- **CREATE TABLE** merupakan perintah dasar dari pembuatan table.
- **mhs** merupakan nama tabel yang akan dibuat.
- **Nim, nama, tglahir** dan **alamat** merupakan nama field
- **Varchar, date** dan **text** merupakan tipe data dari field
- **NOT NULL** merupakan option untuk menyatakan bahwa suatu field tidak boleh kosong.
- **PRIMARY KEY** merupakan perintah untuk menentukan field mana yang akan dijadikan primary key pada tabel.
- **10** dan **30** di belakang tipe data merupakan panjang maksimal dari suatu field
- Untuk tipe data **date** dan **text** (dan beberapa tipe data lainnya) panjang karakter maksimalnya tidak perlu ditentukan.
- Jangan lupa akhiri perintah dengan titik-koma (;

Selanjutnya untuk melihat tabel mhs sudah benar-benar sudah ada atau belum, ketikkan perintah berikut ini :

```
SHOW TABLES;
```

Perintah di atas akan menampilkan seluruh tabel yang sudah ada dalam suatu database. Contoh hasil dari perintah di atas adalah sebagai berikut :

```
+-----+
| Tables_in_mahasiswa |
+-----+
| mhs                 |
+-----+
1 rows in set (0.01 sec)
```

Untuk melihat struktur tabel “**mhs**” secara lebih detail, cobalah perintah atau query sebagai berikut :

```
DESC mhs;
```

DESC merupakan singkatan dari **DESCRIBE** (dalam query bisa ditulis lengkap atau hanya 4 karakter pertama) dan **mhs** adalah nama tabel yang akan dilihat strukturnya. Dari perintah di atas, akan ditampilkan struktur tabel **mhs** sebagai berikut :

```
+-----+-----+-----+-----+-----+-----+
| Field      | Type       | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| nim        | varchar(10) |      | PRI |          |       |
| nama       | varchar(30)  |      |     |          |       |
| tgllahir   | date        | YES  |     | NULL    |       |
| alamat     | text         | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

Dari struktur tabel mhs yang ditampilkan di atas, dapat diketahui bahwa :

- Terdapat 4 (empat) field dengan tipe masing-masing.
- Primary Key dari tabel mhs adalah nim. Lihat kolom **Key** pada field nim.
- Untuk field nim dan nama defaultnya tidak boleh kosong. Lihatlah kolom **Null** dan **Default** pada field nim dan nama
- Untuk field tgllahir dan alamat defaultnya boleh kosong. Lihatlah kolom **Null** dan **Default** pada field tgllahir dan alamat.

Mengubah Struktur Table dengan ALTER

Untuk mengubah struktur suatu tabel, bentuk umum perintah SQL-nya sebagai berikut :

```
ALTER TABLE nama_tabel alter_options;
```

dimana :

- **ALTER TABLE** merupakan perintah dasar untuk mengubah tabel.
- **nama_tabel** merupakan nama tabel yang akan diubah strukturnya.
- **alter_options** merupakan pilihan perubahan tabel. Option yang bisa digunakan, beberapa di antaranya sebagai berikut :
 - » **ADD definisi_field_baru**
Option ini digunakan untuk menambahkan field baru dengan “**definisi_field_baru**” (nama field, tipe dan option lain).
 - » **ADD INDEX nama_index**

Option ini digunakan untuk menambahkan index dengan nama "**nama_index**" pada tabel.

- » ADD PRIMARY KEY (field_kunci)
Option untuk menambahkan primary key pada tabel
- » CHANGE field_yang_diubah definisi_field_baru
Option untuk mengubah field_yang_diubah menjadi definisi_field_baru
- » MODIFY definisi_field
Option untuk mengubah suatu field menjadi definisi_field
- » DROP nama_field
Option untuk menghapus field nama_field
- » RENAME TO nama_tabel_baru
Option untuk mengganti nama tabel

Beberapa contoh variasi perintah ALTER untuk mengubah struktur suatu tabel antara lain :

1. Menambahkan field "**agama**" ke tabel **mhs**

```
ALTER TABLE mhs ADD agama varchar(15) NOT NULL;
```

2. Menambahkan primary key pada suatu tabel

```
ALTER TABLE mhs ADD PRIMARY KEY(nim);
```

3. Mengubah **panjang field agama** menjadi 10 karakter dalam tabel **mhs**

```
ALTER TABLE mhs CHANGE agama agama varchar(10);
```

4. Mengubah **tipe field agama** menjadi char(2) dalam tabel **mhs**

```
ALTER TABLE mhs MODIFY agama char(2) NOT NULL;
```

5. Menghapus field **agama** dari tabel **mhs**

```
ALTER TABLE mhs DROP agama;
```

Mengubah Nama Tabel

Untuk mengubah nama suatu tabel, dapat menggunakan perintah SQL sbb :

```
RENAME TABLE mhs TO mahasiswa;  
ALTER TABLE mhs RENAME TO mahasiswa;
```

Perintah di atas akan mengubah tabel **mhs** menjadi **mahasiswa**.

Menghapus Tabel

Untuk menghapus sebuah tabel, bentuk umum dari perintah SQL adalah sebagai berikut :

```
DROP TABLE nama_tabel;
```

Contohnya kita akan menghapus tabel dengan nama "**mahasiswa**" maka perintah SQL-nya adalah :

```
DROP TABLE mahasiswa;
```

Menambah Record dengan INSERT

Bentuk umum perintah SQL untuk menambahkan *record* atau data ke dalam suatu tabel adalah sebagai berikut :

```
INSERT INTO nama_tabel VALUES ('nilai1','nilai2',...);
```

atau dapat dengan bentuk sebagai berikut :

```
INSERT INTO nama_tabel(field1,field2,...)
VALUES ('nilai1','nilai2',...);
```

atau dapat juga dengan bentuk sebagai berikut :

```
INSERT INTO nama_tabel
SET field1='nilai1', field2='nilai2',...;
```

Sebagai contoh, kita akan menambahkan sebuah record ke dalam tabel **mhs** yang telah kita buat sebelumnya. Berikut ini perintah SQL untuk menambahkan sebuah record ke dalam tabel **mhs** :

```
INSERT INTO mhs VALUES ('0411500121','Achmad Solichin',
'1982-06-05','Jakarta Selatan');
```

Jika perintah SQL di atas berhasil dieksekusi maka akan ditampilkan pesan sebagai berikut :

```
Query OK, 1 row affected (0.00 sec)
```

Setelah perintah SQL di atas berhasil dieksekusi, maka record atau data dalam tabel mhs akan bertambah. Jalankan perintah berikut ini untuk melihat isi tabel mhs !

```
SELECT * FROM mhs;
```

Dan berikut ini hasil dari perintah SQL di atas :

nim	nama	tglahir	alamat
0411500121	Achmad Solichin	1982-06-05	Jakarta Selatan

1 row in set (0.19 sec)

Latihan

Tambahkan 10 data (record) baru ke tabel **mhs** sehingga isi tabel mhs menjadi sebagai berikut !

nim	nama	tglahir	alamat
0411500121	Achmad Solichin	1982-06-05	Jakarta Selatan
0411500123	Chotimatul M	1983-03-12	Jakarta Selatan
0422500111	Bajuri	1983-03-25	Tangerang
0444500011	Oneng	1980-05-22	Jakarta Utara
0433500115	Unyil	1980-08-29	Tangerang

0411500116 Ujang	1984-10-06 Jakarta Barat
0422500316 Jebleh	1984-10-06 Cengkareng
0433500333 Dono	1984-10-06 Jakarta Selatan
0422500433 Dini	1986-12-10 Jakarta Selatan
0411500331 Dana	1986-07-11 Jakarta Selatan
0444500315 Dani	1985-01-01 Jakarta Barat

Mengedit Record dengan UPDATE

Proses update bisa sewaktu-waktu dilakukan jika terdapat data atau record dalam suatu tabel yang perlu diperbaiki. Proses update ini tidak menambahkan data (record) baru, tetapi memperbaiki data yang lama. Perubahan yang terjadi dalam proses update bersifat permanen, artinya setelah perintah dijalankan tidak dapat di-cancel (*undo*).

Bentuk umum perintah SQL untuk mengedit suatu *record* atau data dari suatu tabel adalah sebagai berikut :

```
UPDATE nama_tabel SET field1='nilaibaru'
[WHERE kondisi];
```

Pada perintah untuk update di atas :

- UPDATE merupakan perintah dasar untuk mengubah *record* tabel.
- *nama_tabel* merupakan nama tabel yang akan diubah *recordnya*.
- Perintah SET diikuti dengan *field-field* yang akan diubah yang mana diikuti juga dengan perubahan isi dari masing-masing *field*. Untuk mengubah nilai dari beberapa *field* sekaligus, gunakan koma (,) untuk memisahkan masing-masing *field*.
- Perintah WHERE diikuti oleh kondisi tertentu yang menentukan record mana yang akan diedit (diubah). Perintah WHERE ini boleh ada boleh juga tidak. Jika WHERE tidak ditambahkan pada perintah update maka semua *record* dalam tabel bersangkutan akan berubah.

Perhatikan beberapa contoh perintah UPDATE tabel **mhs** berikut ini !

1. Mengubah alamat menjadi "Tangerang" untuk mahasiswa yang mempunyai nim 0411500121

```
UPDATE mhs SET alamat='Tangerang' WHERE
nim='0411500121';
```

Dan jika query di atas berhasil dieksekusi maka akan ditampilkan hasil sebagai berikut :

```
Query OK, 1 row affected (0.27 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

2. Mengubah tanggal lahir menjadi "12 Maret 1983" dan alamat menjadi "Bandung" untuk mahasiswa yang mempunyai nim 0422500316

```
UPDATE mhs SET tgllahir='1982-06-05', alamat='Jakarta
Selatan' WHERE nim='0422500316';
```

Menghapus Record dengan DELETE

Proses delete dilakukan jika terdapat data atau record dalam suatu tabel yang perlu dihapus atau dihilangkan. Perubahan yang terjadi dalam proses *delete* bersifat permanen, artinya setelah perintah dijalankan tidak dapat di-*cancel* (*undo*). Jadi berhati-hatilah dengan perintah *delete* !

Bentuk umum perintah SQL untuk menghapus suatu *record* atau data dari tabel adalah sebagai berikut :

```
DELETE FROM nama_tabel [WHERE kondisi];
```

Pada perintah untuk *delete* di atas :

- `DELETE FROM` merupakan perintah dasar untuk menghapus suatu *record* dari tabel.
- `nama_tabel` merupakan nama tabel yang akan dihapus *recordnya*.
- Perintah `WHERE` diikuti oleh kondisi tertentu yang menentukan record mana yang akan dihapus (didelete). Perintah `WHERE` ini boleh ada boleh juga tidak. Namun demikian, jika `WHERE` tidak ditambahkan pada perintah `delete` maka semua *record* dalam tabel bersangkutan akan **terhapus**. Jadi jangan lupa menambahkan WHERE jika kita tidak bermaksud mengosongkan tabel

Perhatikan beberapa contoh perintah `DELETE` dari tabel **mhs** berikut ini !

1. Menghapus data mahasiswa yang mempunyai nim 0411500331

```
DELETE FROM mhs WHERE nim='0411500331';
```

Dan jika query di atas berhasil dieksekusi maka akan ditampilkan hasil sebagai berikut :

```
Query OK, 1 row affected (0.11 sec)
```

2. Menghapus semua mahasiswa yang beralamat di "Bandung"

```
DELETE FROM mhs WHERE alamat='Bandung';
```

Menampilkan Record dengan SELECT

Perintah `SELECT` digunakan untuk menampilkan sesuatu. Sesuatu di sini bisa berupa sejumlah data dari tabel dan bisa juga berupa suatu ekspresi. Dengan `SELECT` kita bisa mengatur tampilan atau keluaran sesuai tampilan yang diinginkan.

Bentuk dasar perintah `SELECT` data dari tabel adalah sebagai berikut :

```
SELECT [field | *] FROM nama_tabel [WHERE kondisi];
```

Perhatikan beberapa contoh perintah `SELECT` dari tabel **mhs** berikut ini !

1. Menampilkan seluruh data atau record (*) dari tabel **mhs**

```
SELECT * FROM mhs;
```

Dan jika query di atas berhasil dieksekusi maka akan ditampilkan hasil sebagai berikut :

nim	nama	tglahir	alamat
0411500121	Achmad Solichin	1982-06-05	Jakarta Selatan
0411500123	Chotimatul M	1983-03-12	Jakarta Selatan
0422500111	Bajuri	1983-03-25	Tangerang
0444500011	Oneng	1980-05-22	Jakarta Utara
0433500115	Unyil	1980-08-29	Tangerang
0411500116	Ujang	1984-10-06	Jakarta Barat
0422500316	Jebleh	1982-06-05	Jakarta Selatan
0433500333	Dono	1984-10-06	Jakarta Selatan
0422500433	Dini	1986-12-10	Jakarta Selatan
0444500315	Dani	1985-01-01	Jakarta Barat

10 rows in set (0.25 sec)

2. Menampilkan field **nim** dan **nama** dari seluruh mahasiswa dalam tabel **mhs**

```
SELECT nim, nama FROM mhs;
```

Jika query di atas berhasil dieksekusi maka akan ditampilkan hasil sebagai berikut :

nim	nama
0411500121	Achmad Solichin
0411500123	Chotimatul M
0422500111	Bajuri
0444500011	Oneng
0433500115	Unyil
0411500116	Ujang
0422500316	Jebleh
0433500333	Dono
0422500433	Dini
0444500315	Dani

10 rows in set (0.11 sec)

3. Menampilkan data mahasiswa yang mempunyai nim **0411500123**

```
SELECT * FROM mhs WHERE nim = '0411500123';
```

Hasil query di atas adalah sbb :

nim	nama	tglahir	alamat
0411500123	Chotimatul M	1983-03-12	Jakarta Selatan

1 row in set (0.06 sec)

4. Menampilkan data semua mahasiswa yang beralamat di luar **Jakarta Selatan**

```
SELECT * FROM mhs WHERE alamat != 'Jakarta Selatan';
```

Hasil query di atas adalah sbb :

nim	nama	tglahir	alamat
0422500111	Bajuri	1983-03-25	Tangerang
0444500011	Oneng	1980-05-22	Jakarta Utara
0433500115	Unyil	1980-08-29	Tangerang
0411500116	Ujang	1984-10-06	Jakarta Barat
0444500315	Dani	1985-01-01	Jakarta Barat

5 rows in set (0.02 sec)

Berikut ini operator **perbandingan** yang dapat digunakan untuk membandingkan dua buah nilai dalam MySQL :

- **Operator =**, akan bernilai TRUE jika nilai yang dibandingkan sama.
- **Operator !=** atau **<>**, akan bernilai TRUE jika nilai yang dibandingkan TIDAK SAMA (berbeda).
- **Operator >**, akan bernilai TRUE jika nilai yang pertama lebih besar dari nilai kedua.
- **Operator >=**, akan bernilai TRUE jika nilai yang pertama lebih besar atau sama dengan nilai kedua.
- **Operator <**, akan bernilai TRUE jika nilai yang pertama lebih kecil dari nilai kedua.
- **Operator <=**, akan bernilai TRUE jika nilai yang pertama lebih kecil atau sama dengan nilai kedua.

5. Menampilkan data semua mahasiswa yang beralamat di **Jakarta Selatan** dan lahir pada tahun **1982**.

```
SELECT * FROM mhs WHERE alamat = 'Jakarta Selatan' &&
YEAR(tglahir) = '1982';
```

Hasil query di atas adalah sbb :

nim	nama	tglahir	alamat
0411500121	Achmad Solichin	1982-06-05	Jakarta Selatan
0422500316	Jebleh	1982-06-05	Jakarta Selatan

2 rows in set (0.00 sec)

Berikut ini operator **penghubung** yang dapat digunakan untuk menghubungkan antara dua kondisi dalam MySQL :

- **Operator &&** atau **AND**, akan menghubungkan dua kondisi dimana akan bernilai TRUE jika kedua kondisi bernilai TRUE.
- **Operator ||** atau **OR**, akan menghubungkan dua kondisi dimana akan bernilai TRUE jika salah satu atau kedua kondisi bernilai TRUE.
- **Operator !**, akan me-reverse nilai suatu kondisi logika.

Keterangan

Fungsi **YEAR** pada query di atas akan menghasilkan nilai TAHUN dari suatu tanggal. Selain fungsi YEAR, juga terdapat fungsi MONTH yang akan menghasilkan nama BULAN dari tanggal, fungsi DAY yang akan menghasilkan

hari dari suatu tanggal, dan masih banyak fungsi lain yang berhubungan dengan tanggal.

- Menampilkan **nim**, **nama** dan **umur** dari semua mahasiswa.

```
SELECT nim, nama, YEAR(now())-YEAR(tglahir) AS umur
FROM mhs;
```

Hasil query di atas adalah sbb :

nim	nama	umur
0411500121	Achmad Solichin	24
0411500123	Chotimatul M	23
0422500111	Bajuri	23
0444500011	Oneng	26
0433500115	Unyil	26
0411500116	Ujang	22
0422500316	Jebleh	24
0433500333	Dono	22
0422500433	Dini	20
0444500315	Dani	21

10 rows in set (0.05 sec)

Keterangan

Pada query di atas terdapat fungsi **YEAR** yang akan mengambil tahun dari suatu tanggal. Selanjutnya fungsi **now()** akan me-return tanggal dan waktu sistem saat query dieksekusi. Proses perhitungan umur dialiaskan dengan nama '**umur**'. Untuk mengaliaskan gunakan perintah **AS** yang diikuti nama alias.

- Menampilkan semua mahasiswa **jurusan TI**

```
SELECT * FROM mhs WHERE SUBSTRING(nim,3,2)='11' ;
```

Hasil query di atas adalah sbb :

nim	nama	tglahir	alamat
0411500121	Achmad Solichin	1982-06-05	Jakarta Selatan
0411500123	Chotimatul M	1983-03-12	Jakarta Selatan
0411500116	Ujang	1984-10-06	Jakarta Barat

3 rows in set (0.19 sec)

Keterangan

Pada query di atas terdapat fungsi **SUBSTRING** yang berguna untuk memotong suatu string. Format fungsi SUBSTRING adalah sebagai berikut :
SUBSTRING(field, awal, panjang)

- Menampilkan semua data mahasiswa secara urut berdasarkan **nama** dengan perintah **ORDER BY**

```
SELECT * FROM mhs ORDER BY nama;
```

Hasil query di atas adalah sbb :

nim	nama	tglahir	alamat
0411500121	Achmad Solichin	1982-06-05	Jakarta Selatan
0422500111	Bajuri	1983-03-25	Tangerang
0411500123	Chotimatul M	1983-03-12	Jakarta Selatan
0444500315	Dani	1985-01-01	Jakarta Barat
0422500433	Dini	1986-12-10	Jakarta Selatan
0433500333	Dono	1984-10-06	Jakarta Selatan
0422500316	Jebleh	1982-06-05	Jakarta Selatan
0444500011	Oneng	1980-05-22	Jakarta Utara
0411500116	Ujang	1984-10-06	Jakarta Barat
0433500115	Unyil	1980-08-29	Tangerang

10 rows in set (0.01 sec)

9. Menampilkan semua data mahasiswa secara urut berdasarkan **nim** secara **DESCENDING**

```
SELECT * FROM mhs ORDER BY nim DESC;
```

Hasil query di atas adalah sbb :

nim	nama	tglahir	alamat
0444500315	Dani	1985-01-01	Jakarta Barat
0444500011	Oneng	1980-05-22	Jakarta Utara
0433500333	Dono	1984-10-06	Jakarta Selatan
0433500115	Unyil	1980-08-29	Tangerang
0422500433	Dini	1986-12-10	Jakarta Selatan
0422500316	Jebleh	1982-06-05	Jakarta Selatan
0422500111	Bajuri	1983-03-25	Tangerang
0411500123	Chotimatul M	1983-03-12	Jakarta Selatan
0411500121	Achmad Solichin	1982-06-05	Jakarta Selatan
0411500116	Ujang	1984-10-06	Jakarta Barat

10 rows in set (0.00 sec)

10. Menampilkan 5 record (data) pertama dari tabel **mhs** secara urut berdasarkan **nim** dengan **LIMIT**

```
SELECT * FROM mhs ORDER BY nim LIMIT 0,5;
```

Hasil query di atas adalah sbb :

nim	nama	tglahir	alamat
0411500116	Ujang	1984-10-06	Jakarta Barat
0411500121	Achmad Solichin	1982-06-05	Jakarta Selatan
0411500123	Chotimatul M	1983-03-12	Jakarta Selatan
0422500111	Bajuri	1983-03-25	Tangerang
0422500316	Jebleh	1982-06-05	Jakarta Selatan

5 rows in set (0.13 sec)

Keterangan

Pada query di atas bentuk **LIMIT** digunakan untuk membatasi hasil tampilan. LIMIT banyak digunakan untuk menampilkan data yang relatif banyak. Format fungsi LIMIT adalah sebagai berikut :

 LIMIT awal, jumlah_record

Bab 13

Aplikasi Berita dengan PHP dan MySQL

- ❖ Merancang Struktur Database
- ❖ Membuat File Koneksi Database
- ❖ Membuat Halaman Input Berita
- ❖ Menampilkan Berita Terbaru di Halaman Depan
- ❖ Menampilkan Berita Lengkap
- ❖ Membuat Halaman Arsip Berita
- ❖ Membuat Halaman Edit Berita
- ❖ Membuat Halaman Delete Berita
- ❖ CSS sebagai Pemanis Tampilan

Merancang Struktur Database

Untuk membuat aplikasi berita secara sederhana, langkah pertama adalah merancang tabel-tabel database yang diperlukan. Secara sederhana, struktur tabel dapat dibuat dengan spesifikasi sebagai berikut :

Tabel “**berita**”

Nama Field	Tipe	Panjang	Keterangan
id_berita	Int	5	Primary Key, Auto Increment
id_kategori	Int	3	Foreign Key
judul	Varchar	100	Judul berita
headline	Text	-	Berita singkat
isi	Text	-	Berita lengkap
pengirim	Varchar	15	Pengirim berita
tanggal	Datetime	-	Tanggal dan waktu kirim berita

Tabel “**kategori**”

Nama Field	Tipe	Panjang	Keterangan
id_kategori	Int	3	Primary Key, Auto Increment
nm_kategori	Varchar	30	Nama kategori
deskripsi	Varchar	200	Deskripsi kategori

Berikut ini perintah (SQL) untuk membuat tabel “**berita**”

```
CREATE TABLE berita (
    id_berita int(5) unsigned NOT NULL auto_increment,
    id_kategori int(3) unsigned NOT NULL default '0',
    judul varchar(100) NOT NULL default '',
    headline text NOT NULL,
    isi text NOT NULL,
    pengirim varchar(15) NOT NULL default '',
    tanggal datetime NOT NULL default '0000-00-00 00:00:00',
    PRIMARY KEY (id_berita)
) TYPE=MyISAM;
```

Berikut ini perintah (SQL) untuk membuat tabel “**kategori**”

```
CREATE TABLE kategori (
    id_kategori int(3) unsigned NOT NULL auto_increment,
    nm_kategori varchar(30) NOT NULL default '',
    deskripsi varchar(200) NOT NULL default '',
    PRIMARY KEY (id_kategori),
) TYPE=MyISAM;
```

Catatan : Isikan sejumlah data ke tabel **kategori** langsung di database

Membuat File Koneksi Database

Program 1

Nama File : *koneksi.php*
Deskripsi : Program koneksi ke database.

```
1 <?php
2 $host = "localhost";
3 $user = "root";
4 $pass = "";
5 $dbnm = "pw2";
6
7 $conn = mysql_connect ($host, $user, $pass);
8 if ($conn) {
9     $buka = mysql_select_db ($dbnm);
10    if (!$buka) {
11        die ("Database tidak dapat dibuka");
12    }
13 } else {
14     die ("Server MySQL tidak terhubung");
15 }
16
17 ?>
```

Membuat Halaman Input Berita

Program 2

Nama File : *input_berita.php*
Deskripsi : Program input berita.

```
1 <?php
2 include "koneksi.php";
3
4 //proses input berita
5 if (isset($_POST['Input'])) {
6     $judul = addslashes (strip_tags ($_POST['judul']));
7     $kategori = $_POST['kategori'];
8     $headline = addslashes (strip_tags ($_POST['headline']));
9     $isi_berita = addslashes (strip_tags ($_POST['isi']));
10    $pengirim = addslashes (strip_tags ($_POST['pengirim']));
11    //insert ke tabel
12    $query = "INSERT INTO berita
13 VALUES('','$kategori','$judul','$headline','$isi_berita','$pengirim',
now())";
14    $sql = mysql_query ($query);
15    if ($sql) {
16        echo "<h2><font color=blue>Berita telah berhasil
ditambahkan</font></h2>";
17    } else {
18        echo "<h2><font color=red>Berita gagal
ditambahkan</font></h2>";
19    }
20 }
21 ?>
22 <html>
23     <head><title>Input Berita</title>
24     <link rel="stylesheet" href="style.css">
25     </head>
26     <body>
27         <a href="index.php">Halaman Depan</a> |
28         <a href="arsip_berita.php">Arsip Berita</a> |
29         <a href="input_berita.php">Input Berita</a>
30         <br><br>
```

```

30      <FORM ACTION="" METHOD="POST" NAME="input">
31          <table cellpadding="0" cellspacing="0" border="0"
32              width="700">
33                  <tr>
34                      <td colspan="2"><h2>Input Berita</h2></td>
35                  </tr>
36                  <tr>
37                      <td width="200">Judul Berita</td>
38                      <td>: <input type="text" name="judul"
39                          size="30"></td>
40                  </tr>
41                  <tr>
42                      <td>Kategori</td>
43                      <td>:
44                      <select name="kategori">
45                          <?
46                              $query = "SELECT id_kategori, nm_kategori
47                          FROM kategori ORDER BY nm_kategori";
48                          $sql = mysql_query ($query);
49                          while ($hasil = mysql_fetch_array ($sql)) {
50                              echo "<option
51                                  value='".$hasil[id_kategori]."'>$hasil[nm_kategori]</option>";
52                          }
53                      ?>
54                      </select></td>
55                  </tr>
56                  <tr>
57                      <td>Headline Berita</td>
58                      <td>: <textarea name="headline" cols="50" rows="4">
59                          </textarea></td>
60                  </tr>
61                  <tr>
62                      <td>Isi Berita</td>
63                      <td>: <textarea name="isi" cols="50" rows="10">
64                          </textarea></td>
65                  </tr>
66                  <tr>
67                      <td>Pengirim</td>
68                      <td>: <input type="text" name="pengirim" size="20">
69                  </td>
70                  </tr>
71          </table>
72      </body>
73  </html>

```

Menampilkan Berita Terbaru di Halaman Depan

Program 3

Nama File : index.php

Deskripsi : Program menampilkan 5 berita terbaru di halaman depan.

```

1  <?php
2  include "koneksi.php";
3
4  ?>
5  <html>
6      <head><title>Index Berita</title>

```

```

7   <link rel="stylesheet" href="style.css">
8   </head>
9   <body>
10  <a href="index.php">Halaman Depan</a> | 
11  <a href="arsip_berita.php">Arsip Berita</a> | 
12  <a href="input_berita.php">Input Berita</a>
13  <br><br>
14  <h2>Halaman Depan ~ Lima Berita Terbaru</h2>
15  <?
16  $query = "SELECT A.id_berita, B.nm_kategori, A.judul,
A.headline, A.pengirim, A.tanggal
17  FROM berita A, kategori B WHERE
A.id_kategori=B.id_kategori
18  ORDER BY A.id_berita DESC LIMIT 0,5";
19  $sql = mysql_query ($query);
20  while ($hasil = mysql_fetch_array ($sql)) {
21      $id_berita = $hasil['id_berita'];
22      $kategori = stripslashes ($hasil['nm_kategori']);
23      $judul = stripslashes ($hasil['judul']);
24      $headline = nl2br(stripslashes
($hasil['headline']));
25      $pengirim = stripslashes ($hasil['pengirim']);
26      $tanggal = stripslashes ($hasil['tanggal']);
27      //
28      //tampilkan berita
29      echo "<font size=4><a
30      href='berita_lengkap.php?id=$id_berita'$judul</a></font><br>";
31      echo "<small>Berita dikirimkan oleh <b>$pengirim</b>
32      pada tanggal <b>$tanggal</b> dalam kategori
33      <b>$kategori</b></small>";
34      echo "<p>$headline</p>";
35      echo "<hr>";
36  }
37  ?>
38  </body>
39  </html>

```

Menampilkan Berita Lengkap

Program 4

Nama File : berita_lengkap.php
 Deskripsi : Program menampilkan berita secara lengkap.

```

1  <?php
2  include "koneksi.php";
3
4  if (isset($_GET['id'])) {
5      $id_berita = $_GET['id'];
6  } else {
7      die ("Error. No Id Selected! ");
8  }
9  ?>
10 <html>
11  <head><title>Berita Lengkap</title>
12  <link rel="stylesheet" href="style.css">
13  </head>
14  <body>
15  <a href="index.php">Halaman Depan</a> | 
16  <a href="arsip_berita.php">Arsip Berita</a> | 
17  <a href="input_berita.php">Input Berita</a>
18  <br><br>
19  <h2>Berita Lengkap</h2>
20  <?
21  $query = "SELECT A.id_berita, B.nm_kategori, A.judul,
A.isi, A.pengirim, A.tanggal
22  FROM berita A, kategori B WHERE

```

```

1 A.id_kategori=B.id_kategori && A.id_berita='$id_berita'";
23 $sql = mysql_query ($query);
24 $hasil = mysql_fetch_array ($sql);
25 $id_berita = $hasil['id_berita'];
26 $kategori = stripslashes ($hasil['nm_kategori']);
27 $judul = stripslashes ($hasil['judul']);
28 $isi = nl2br(stripslashes ($hasil['isi']));
29 $pengirim = stripslashes ($hasil['pengirim']);
30 $tanggal = stripslashes ($hasil['tanggal']);
31 //
32 //tampilkan berita
33 echo "<font size=5 color=blue>$judul</font><br>";
34 echo "<small>Berita dikirimkan oleh <b>$pengirim</b>
35 pada tanggal <b>$tanggal</b> dalam kategori
<b>$kategori</b></small>";
36 echo "<p>$isi</p>";
37 ?
38 </body>
39 </html>

```

Membuat Halaman Arsip Berita

Program 5

Nama File : arsip_berita.php
 Deskripsi : Program arsip keseluruhan berita.

```

1 <?php
2 include "koneksi.php";
3
4 ?>
5 <html>
6   <head><title>Arsip Berita</title>
7   <link rel="stylesheet" href="style.css">
8   <script language="javascript">
9     function tanya() {
10       if (confirm ("Apakah Anda yakin akan menghapus
11 berita ini ?")) {
12         return true;
13       } else {
14         return false;
15       }
16     }
17   </script>
18   </head>
19   <body>
20     <a href="index.php">Halaman Depan</a> |
21     <a href="arsip_berita.php">Arsip Berita</a> |
22     <a href="input_berita.php">Input Berita</a>
23     <br><br>
24     <h2>Arsip Berita</h2>
25     <ol>
26     <?
27       $query = "SELECT A.id_berita, B.nm_kategori, A.judul,
A.pengirim, A.tanggal
28           FROM berita A, kategori B WHERE
A.id_kategori=B.id_kategori
29           ORDER BY A.id_berita DESC";
30       $sql = mysql_query ($query);
31       while ($hasil = mysql_fetch_array ($sql)) {
32         $id_berita = $hasil['id_berita'];
33         $kategori = stripslashes ($hasil['nm_kategori']);
34         $judul = stripslashes ($hasil['judul']);
35         $pengirim = stripslashes ($hasil['pengirim']);
36         $tanggal = stripslashes ($hasil['tanggal']);
37         //

```

```
38         //tampilkan arsip berita
39         echo "<li><a
40             href='berita_lengkap.php?id=$id_berita'>$judul</a><br>";
41             echo "<small>Berita dikirimkan oleh <b>$pengirim</b>
42                 pada tanggal <b>$tanggal</b> dalam kategori
43                 <b>$kategori</b><br>";
44             echo "<b>Action : </b><a
45                 href='edit_berita.php?id=$id_berita'>Edit</a> | ";
46             echo "<a href='delete_berita.php?id=$id_berita'
47                 onClick='return tanya()">'Delete</a>";
48             echo "</small></li><br><br>";
49         ?
50     </ol>
51 </body>
52 </html>
```

Membuat Halaman Edit Berita

Program 6

Nama File : edit_berita.php
Deskripsi : Program edit berita.

```
1 <?php
2 include "koneksi.php";
3
4 if (isset($_GET['id'])) {
5     $id_berita = $_GET['id'];
6 } else {
7     die ("Error. No Id Selected! ");
8 }
9
10 $query = "SELECT id_berita, id_kategori, judul, headline, isi,
11 pengirim, tanggal
12         FROM berita WHERE id_berita='$id_berita'";
13 $sql = mysql_query ($query);
14 $hasil = mysql_fetch_array ($sql);
15 $id_berita = $hasil['id_berita'];
16 $id_kategori = stripslashes ($hasil['id_kategori']);
17 $judul = stripslashes ($hasil['judul']);
18 $headline = stripslashes ($hasil['headline']);
19 $isi = stripslashes ($hasil['isi']);
20 $pengirim = stripslashes ($hasil['pengirim']);
21 $tanggal = stripslashes ($hasil['tanggal']);
22
23 //proses edit berita
24 if (isset($_POST['Edit'])) {
25     $id_berita = $_POST['hidberita'];
26     $judul = addslashes (strip_tags ($_POST['judul']));
27     $kategori = $_POST['kategori'];
28     $headline = addslashes (strip_tags ($_POST['headline']));
29     $isi_berita = addslashes (strip_tags ($_POST['isi']));
30     $pengirim = addslashes (strip_tags ($_POST['pengirim']));
31     //update berita
32     $query = "UPDATE berita SET
33         id_kategori='$kategori', judul='$judul', headline='$headline',
34         isi='$isi_berita', pengirim='$pengirim' WHERE
35         id_berita='$id_berita'";
36     $sql = mysql_query ($query);
37     if ($sql) {
38         echo "<h2><font color=blue>Berita telah berhasil
diedit</font></h2>";
39     } else {
40         echo "<h2><font color=red>Berita gagal
diedit</font></h2>";
41     }
42 }
```

```

39     }
40   }
41 ?>
42 <html>
43   <head><title>Edit Berita</title>
44   <link rel="stylesheet" href="style.css">
45   </head>
46   <body>
47     <a href="index.php">Halaman Depan</a> |
48     <a href="arsip_berita.php">Arsip Berita</a> |
49     <a href="input_berita.php">Input Berita</a>
50     <br><br>
51     <FORM ACTION="" METHOD="POST" NAME="input">
52       <table cellpadding="0" cellspacing="0" border="0"
width="700">
53         <tr>
54           <td colspan="2"><h2>Input Berita</h2></td>
55         </tr>
56         <tr>
57           <td width="200">Judul Berita</td>
58           <td>: <input type="text" name="judul"
size="30" value=<? echo $judul ?>></td>
59         </tr>
60         <tr>
61           <td>Kategori</td>
62           <td>:
63             <select name="kategori">
64               <?
65                 $query = "SELECT id_kategori,
nm_kategori FROM kategori ORDER BY nm_kategori";
66                 $sql = mysql_query ($query);
67                 while ($hasil = mysql_fetch_array
($sql)) {
68                   $selected = ($hasil['id_kategori']==
$id_kategori) ? "selected" : "";
69                   echo "<option
value='<?=$hasil[id_kategori]>'$selected><?=$hasil[nm_kategori]</option>";
70                 }
71               ?>
72             </select></td>
73           </tr>
74           <tr>
75             <td>Headline Berita</td>
76             <td>: <textarea name="headline" cols="50"
rows="4"><?=$headline?></textarea></td>
77           </tr>
78           <tr>
79             <td>Isi Berita</td>
80             <td>: <textarea name="isi" cols="50"
rows="10"><?=$isi?></textarea></td>
81           </tr>
82           <tr>
83             <td>Pengirim</td>
84             <td>: <input type="text" name="pengirim"
size="20" value=<?=$pengirim?>></td>
85           </tr>
86           <tr>
87             <td>&nbsp;</td>
88             <td>&nbsp;&nbsp;
89             <input type="hidden" name="hidberita"
value=<?=$id_berita?>>
90             <input type="submit" name="Edit" value="Edit
Berita">&nbsp;
91             <input type="reset" name="reset"
value="Cancel"></td>
92           </tr>
93         </table>
94       </FORM>

```

```
95 </body>
96 </html>
```

Membuat Halaman Delete Berita

Program 7

Nama File : delete_berita.php
Deskripsi : Program untuk menghapus berita.

```
1 <?php
2 include "koneksi.php";
3
4 if (isset($_GET['id'])) {
5     $id_berita = $_GET['id'];
6 } else {
7     die ("Error. No Id Selected! ");
8 }
9 ?>
10 <html>
11     <head><title>Delete Berita</title>
12     <link rel="stylesheet" href="style.css">
13     </head>
14     <body>
15         <a href="index.php">Halaman Depan</a> |
16         <a href="arsip_berita.php">Arsip Berita</a> |
17         <a href="input_berita.php">Input Berita</a>
18         <br><br>
19         <?
20         //proses delete berita
21         if (!empty($id_berita) && $id_berita != "") {
22
23             $query = "DELETE FROM berita WHERE
24 id_berita='$id_berita'";
25             $sql = mysql_query ($query);
26             if ($sql) {
27                 echo "<h2><font color=blue>Berita telah berhasil
dihapus</font></h2>";
28             } else {
29                 echo "<h2><font color=red>Berita gagal
dihapus</font></h2>";
30             }
31             echo "Klik <a href='arsip_berita.php'>di sini</a>
untuk kembali ke halaman arsip berita";
32             } else {
33                 die ("Access Denied");
34             }
35         ?>
36     </body>
37 </html>
```

CSS sebagai Pemanis Tampilan

Program 8

Nama File : style.css
Deskripsi : File CSS sebagai pemanis tampilan.

```
1 body {
2     font-family:verdana;
3     font-size:11px;
4 }
5 a {
6     color:#0000ff;
```

```
7   text-decoration:underline;
8 }
9 a:hover {
10   color:#ffffff;
11   background-color:#8AC452;
12   text-decoration:none;
13 }
14 input, textarea, select, option {
15   font-family:verdana;
16 }
```

Tentang Penulis



Achmad Solichin. Adalah Lulusan Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur, Jakarta (S1, 2005). Saat ini sedang menempuh pendidikan S2 di Magister Teknologi Informasi Universitas Indonesia (2008). Kegiatan sehari-hari adalah sebagai Dosen di Universitas Budi Luhur (<http://www.bl.ac.id>), sekaligus sebagai Kepala Laboratorium Komputer Universitas Budi Luhur (<http://labkom.bl.ac.id>). Kegiatan lain aktif sebagai programmer, web developer, system analyst dan memberikan pelatihan di berbagai bidang komputer serta membuat tutorial-tutorial praktis di bidang komputer. Penulis juga terlibat dalam pengembangan E-Learning di Universitas Budi Luhur. Penulis memiliki situs utama di <http://achmatim.net> yang berisi berbagai tutorial praktis seputar pemrograman, web design, database dan materi kuliah. Penulis dapat dihubungi melalui email di achmatim@gmail.com atau YM achmatim.