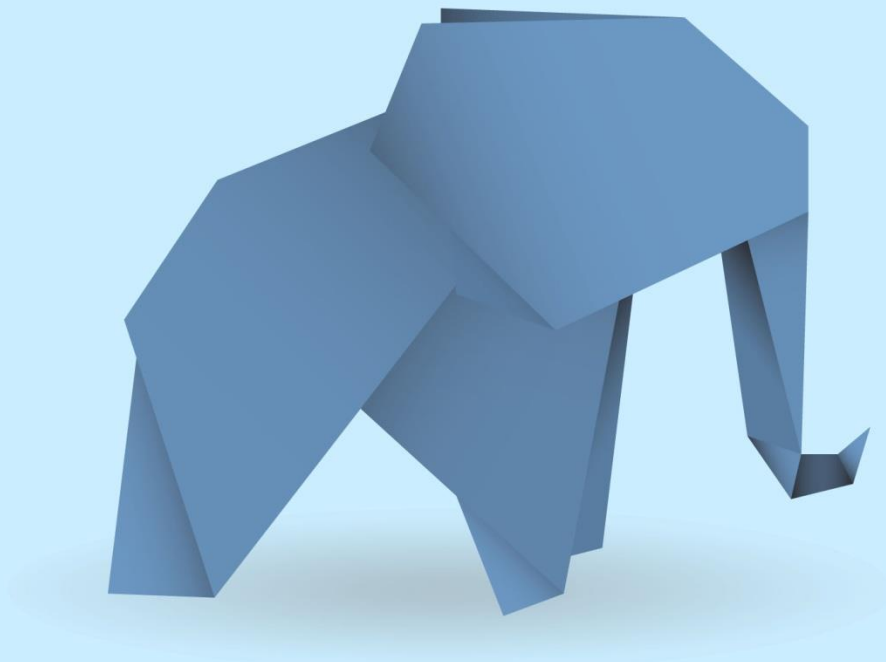


GRATIS

PANDUAN PRAKTIS PEMROGRAMAN PHP UNTUK PEMULA

PHP Fundamental



Sugono galih Aprianto

indosmartdigital.com



EBOOK

Panduan Praktis Pemrograman PHP Untuk Pemula

Disusun Oleh : Sugono Galih Aprianto

Founder indosmartdigital.com

Daftar Isi Ebook

Bab 1 Mari Mengenal PHP	1
Sejarah PHP	1
Alasan Memilih PHP	2
Bab 2 Instalasi PHP dan Software Pendukungnya	3
Menginstal XAMPP di Windows	3
Menjalankan Apache, MySQL dan Membuka Localhost.....	9
Bab 3 Struktur Dasar Skrip PHP.....	10
Memahami Aturan Penulisan Skrip PHP	10
Memberikan Komentar pada Skrip PHP	11
Memahami Variabel di PHP.....	12
Memberi Nilai pada Variabel.....	12
Bab 4 Operator Aritmatika dengan PHP	14
Jenis Operator Aritmatika dalam PHP	14
Tingkat Presedensi	15
Kombinasi Operator Aritmatik dan Assignment.....	16
Operator Pre/Post Increment dan Decrement.....	17
Bab 5 Memahami Method GET Dan POST Pada PHP	18
Kapan menggunakan GET?.....	18
Kapan menggunakan POST?	18
Bab 6 Statement Kontrol – IF	20
Operator Relasional.....	21
Operator Logika	22
Bab 7 Statement SWITCH.....	29
Bab 8 Statement FOR	31
Bab 9 Statement WHILE	35
Bab 10 Mengenal Array.....	38
Memahami Fungsi Array	38
Bab 11 Fungsi (Function)	40
Pengertian Fungsi (Function) Dalam PHP	40
Built in Function dalam PHP.....	42

Bab 12 MySQL Dan PHPMyAdmin.....	48
Perbedaan SQL dan MySQL	48
Keunggulan MySQL	49
Mengelola Database melalui PhpMyAdmin.....	50
 Bab 13 Studi Kasus (Latihan).....	 55

BAB 1

MARI MENGENAL PHP

Anda mungkin sudah pernah mendengar tentang istilah PHP. Namun apakah Anda tahu apa kepanjangan dari PHP itu sebenarnya?

PHP adalah singkatan dari PHP Hypertext Preprocessor, merupakan bahasa pemrograman berbasis web yang dijalankan pada sisi server.

Sejarah PHP

Menurut Wikipedia, Pada awalnya PHP merupakan kependekan dari Personal Home Page (Situs personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama Form Interpreted (FI), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dari web.

Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI. Dengan perilsan kode sumber ini menjadi sumber terbuka, maka banyak pemrogram yang tertarik untuk ikut mengembangkan PHP.

Pada November 1997, dirilis PHP/FI 2.0. Pada rilis ini, interpreter PHP sudah diimplementasikan dalam program C. Dalam rilis ini disertakan juga modul-modul ekstensi yang meningkatkan kemampuan PHP/FI secara signifikan.

Pada tahun 1997, sebuah perusahaan bernama Zend menulis ulang interpreter PHP menjadi lebih bersih, lebih baik, dan lebih cepat. Kemudian pada Juni 1998, perusahaan tersebut merilis interpreter baru untuk PHP dan meresmikan rilis tersebut sebagai PHP 3.0 dan singkatan PHP diubah menjadi akronim berulang PHP: Hypertext Preprocessor.

Pada pertengahan tahun 1999, Zend merilis interpreter PHP baru dan rilis tersebut dikenal dengan PHP 4.0. PHP 4.0 adalah versi PHP yang paling banyak dipakai pada awal abad ke-21. Versi ini banyak dipakai disebabkan kemampuannya untuk membangun aplikasi web kompleks tetapi tetap memiliki kecepatan dan stabilitas yang tinggi.

Pada Juni 2004, Zend merilis PHP 5.0. Dalam versi ini, inti dari interpreter PHP mengalami perubahan besar.

Versi ini juga memasukkan model pemrograman berorientasi objek ke dalam PHP untuk menjawab perkembangan bahasa pemrograman ke arah paradigma berorientasi objek.

Versi terbaru dari bahasa pemrograman PHP adalah versi 7.1.0

Alasan memilih PHP

Berikut ini adalah beberapa alasan yang membuat bahasa Pemrograman PHP yang termasuk dalam kategori bahasa yang banyak digunakan oleh programmer di dunia :

- Free alias gratis, PHP bersifat Open Source, sehingga kita bebas menggunakan PHP tanpa harus membayar.
- Lintas Platform, PHP dapat dijalankan pada sistem operasi apapun, seperti Linux, Windows, OpenBSD, FreeBSD, Mac OS, Solaris dan lain-lain.
- Didukung berbagai Web Server Handal seperti Apache, IIS (Windows), Xitami, Lighttpd, OmniSecure, WebWeaver, dan lain-lain.
- Mampu koneksi dengan berbagai database, support untuk aplikasi database seperti MySQL, Ms. SQL Server, ORACLE, PostgreSQL, Interbase, SQLite, dBase, dan lainnya.
- Memiliki tingkat akses paling cepat dibanding pemrograman web lainnya.

Untuk bekerja dengan PHP dibutuhkan tiga buah software yaitu :

- Apache (Server untuk menjalankan PHP)
- PHP (Program pengeksekusi script PHP)
- MySQL (untuk bekerja dengan database)
- Untuk Windows, Anda bisa menggunakan aplikasi paket instalasi seperti XAMPP ataupun Wampserver. Sedangkan di Linux Anda dapat melakukan instalasi Apache, PHP, dan MySQL secara manual atau menggunakan LAMPP.

BAB 2

INSTALASI PHP DAN SOFTWARE PENDUKUNGNYA

Untuk dapat menjalankan file PHP di komputer lokal berbasis Windows, Kita bisa memanfaatkan XAMPP atau WAMP Server. Dan pada tutorial ini Saya akan membahas tentang XAMPP. Mungkin Anda bertanya-tanya apa fungsi dari XAMPP dan bagaimana cara kerjanya? XAMPP merupakan satu paket instalasi yang di dalamnya terdapat aplikasi-aplikasi pendukung untuk menjalankan file PHP. Selain tersedia pada platform Windows, XAMPP juga tersedia untuk Linux dan OS x.

Pada versi baru yang saya download saat ini (4/10/2016) versi XAMPP telah dilengkapi dengan beberapa update aplikasi yang Kita butuhkan sebagai berikut :

- Updated PHP versi 5.5.38 / 5.6.24 / 7.0.9
- Updated Apache versi 2.4.23
- Updated MariaDB versi 10.1.16

Sebagai informasi penting, XAMPP saat ini tidak lagi menyertakan MySQL sebagai default instalasi database. MySQL digantikan oleh MariaDB. Untuk menjaga kompatibilitas, pihak pembuat tidak banyak mengganti environment MySQL. MariaDB sendiri merupakan versi pengembangan terbuka dan mandiri dari MySQL.

Menginstall XAMPP di Windows

Pertama Anda harus kunjungi website Apache Friend dan download XAMPP disana. Ukuran file XAMPP sekitar 112MB.

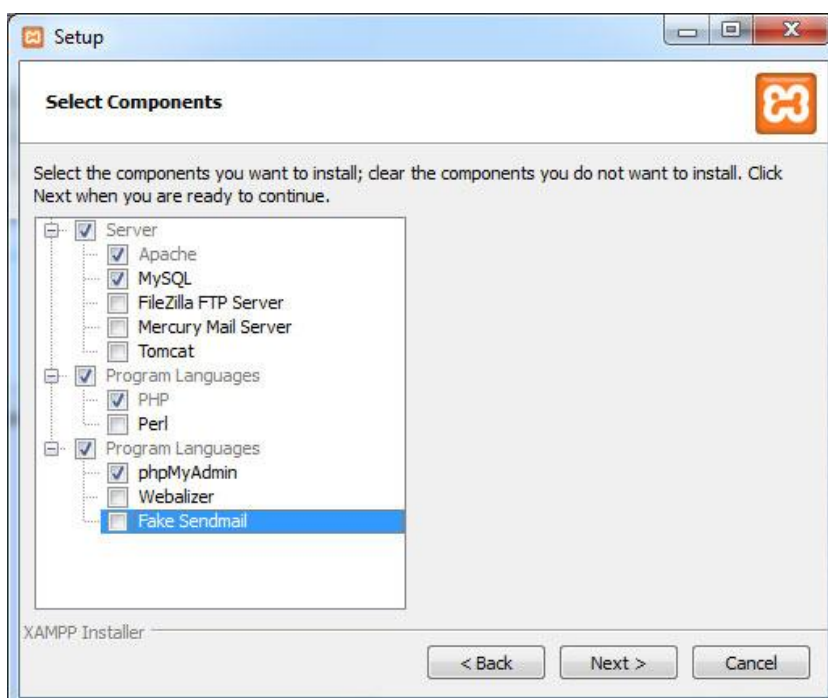


Setelah selesai di download, selanjutnya tinggal jalankan file tersebut dengan melakukan klik dua kali pada file tersebut.

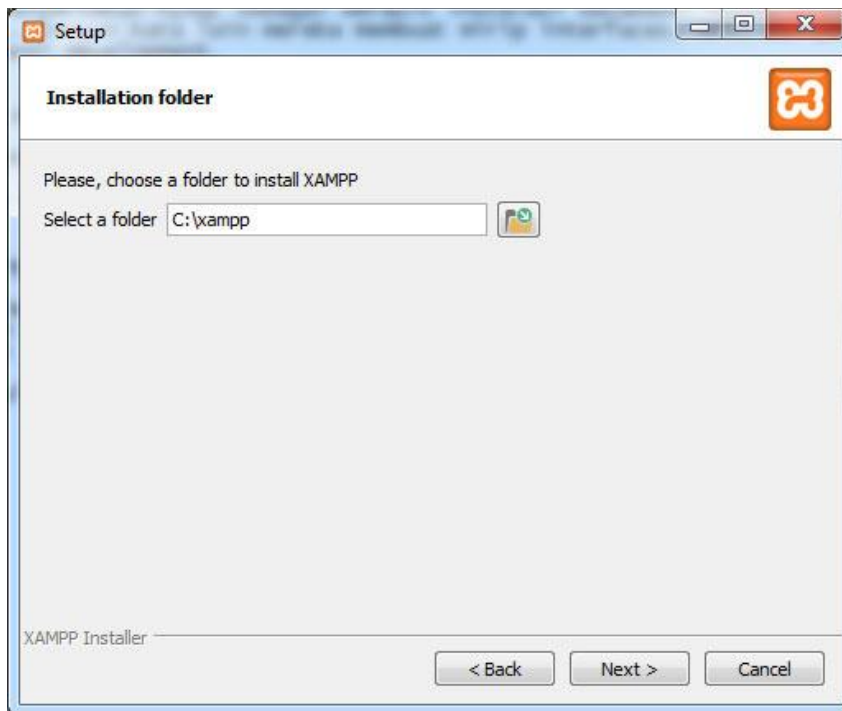
Selama proses instalasi, Anda mungkin akan mendapatkan beberapa peringatan seperti Windows menanyakan apakah Anda yakin ingin menginstall, dan lain sebagainya. Klik saja **"Yes"** dan **OK** untuk melanjutkan.



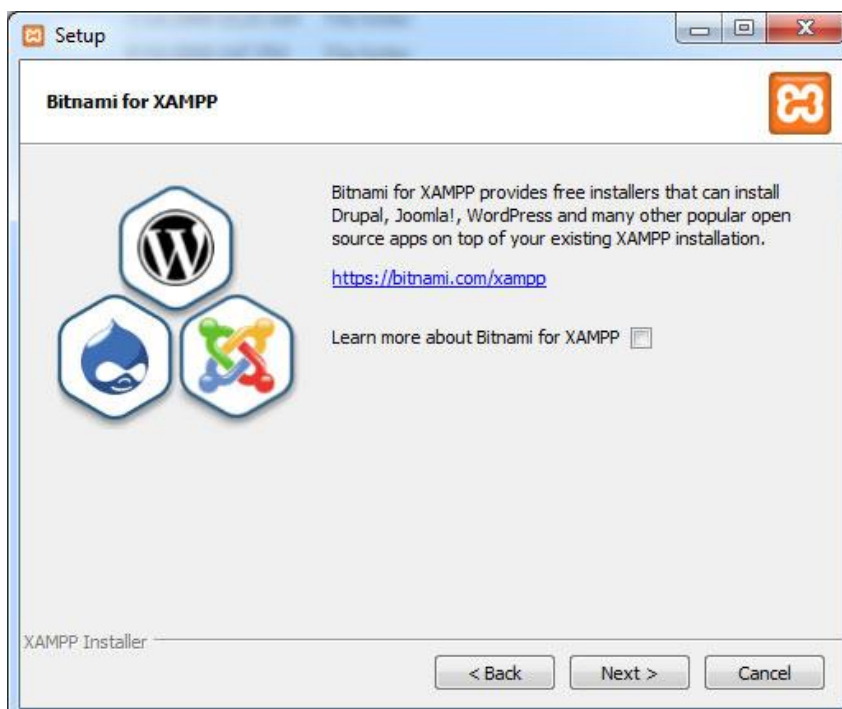
Berikutnya akan muncul tampilan XAMPP Setup Wizard yang akan memandu proses instalasi. Klik **Next**.



Pada jendela berikutnya Anda akan tampil jendela pilihan komponen apa saja yang ingin Kita install. Apache dan PHP secara otomatis akan terinstall sehingga tanda ceklis pada keduanya tidak bisa dihilangkan, sedangkan untuk yang lainnya masih bisa disesuaikan dengan kebutuhan, pastikan MySQL dan PHPMyAdmin Kita ceklis karena ini akan sangat berguna untuk Kita nantinya. Klik **Next**

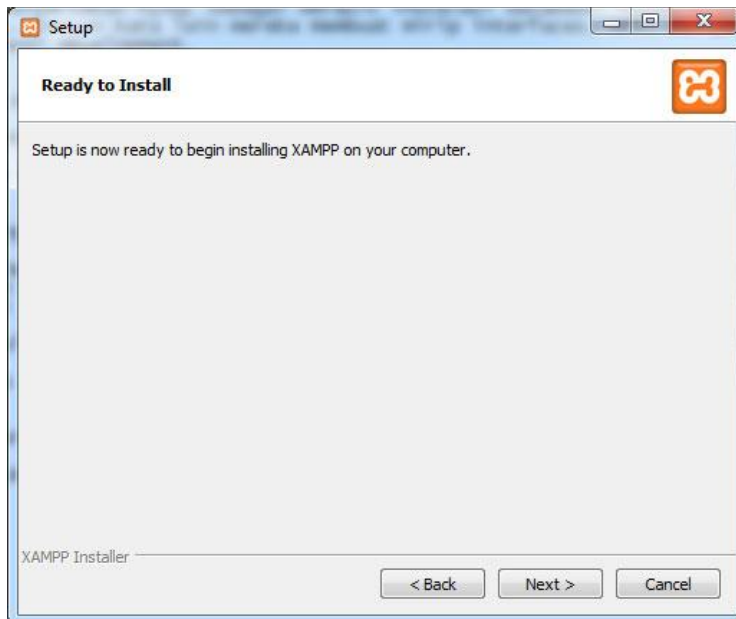


Pilih folder lokasi yang Anda inginkan untuk menginstall XAMPP pada komputer Anda, pada contoh ini kita coba untuk membuatnya di folder **C:\xampp**.



Pada jendela berikutnya, Anda akan ditanya apakah Anda ingin menginstall Bitnami for XAMPP, ini merupakan tool gratis untuk memudahkan Anda menginstall WordPress, Drupal dan Joomla.

Pada contoh kali ini kita tidak menginstall Bitnami karena tidak Kita butuhkan, Hilangkan ceklis pada Learn more about Bitnami for XAMPP. Klik **Next** untuk melanjutkan proses instalasi.



Setelah semua pengaturan dipilih, XAMPP kini siap untuk di install ke komputer Anda. Klik **Next** untuk melanjutkan.



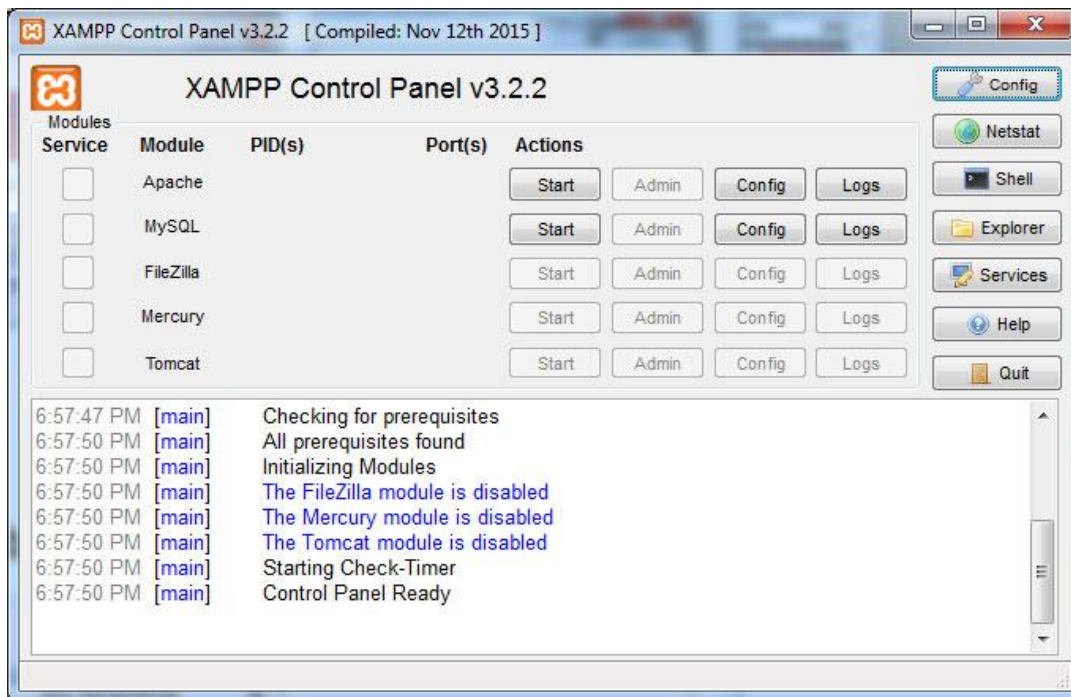
Proses Instalasi sedang berjalan, mohon Anda jangan kemana-mana ya karena proses ini tidak lama :-)



Setelah XAMPP terinstall, Anda akan ditanya apakah Anda ingin memulai XAMPP Control Panel, yang menyediakan sebuah antarmuka untuk menjalankan lingkungan localhost Anda. Anda bisa ceklis pada pilihan ini lalu klik **Finish**.

XAMPP Control Panel secara otomatis akan terbuka, namun apabila tadi Anda tidak menceklis pilihan, Anda bisa membukanya melalui folder XAMPP di komputer Anda, dan buka XAMPP Control Panel sebagai gantinya.

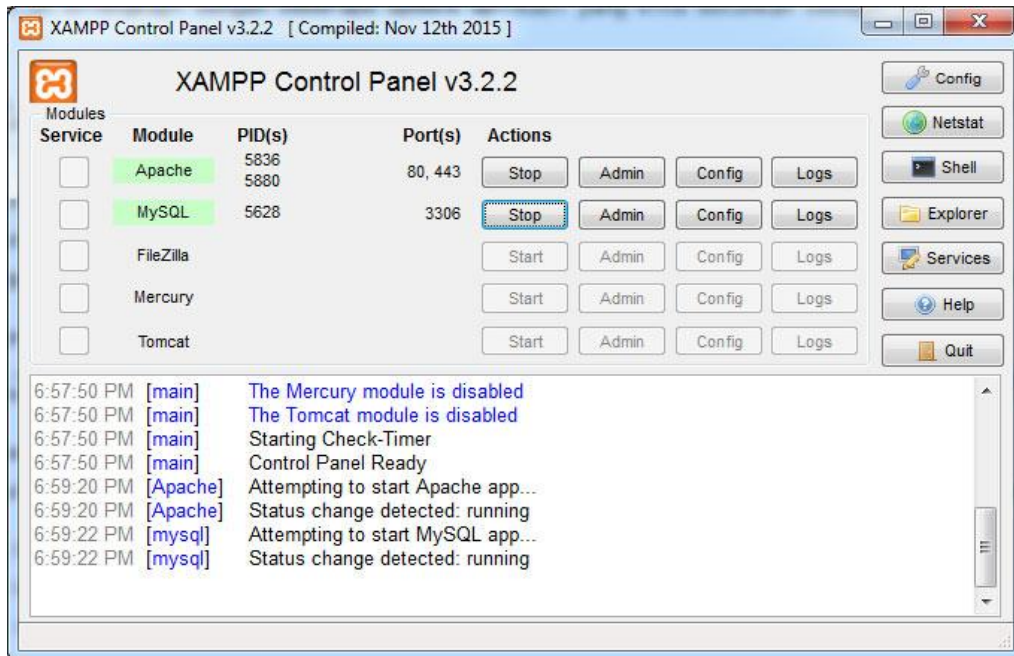
Maka tampilan XAMPP Control Panel akan terbuka seperti berikut.



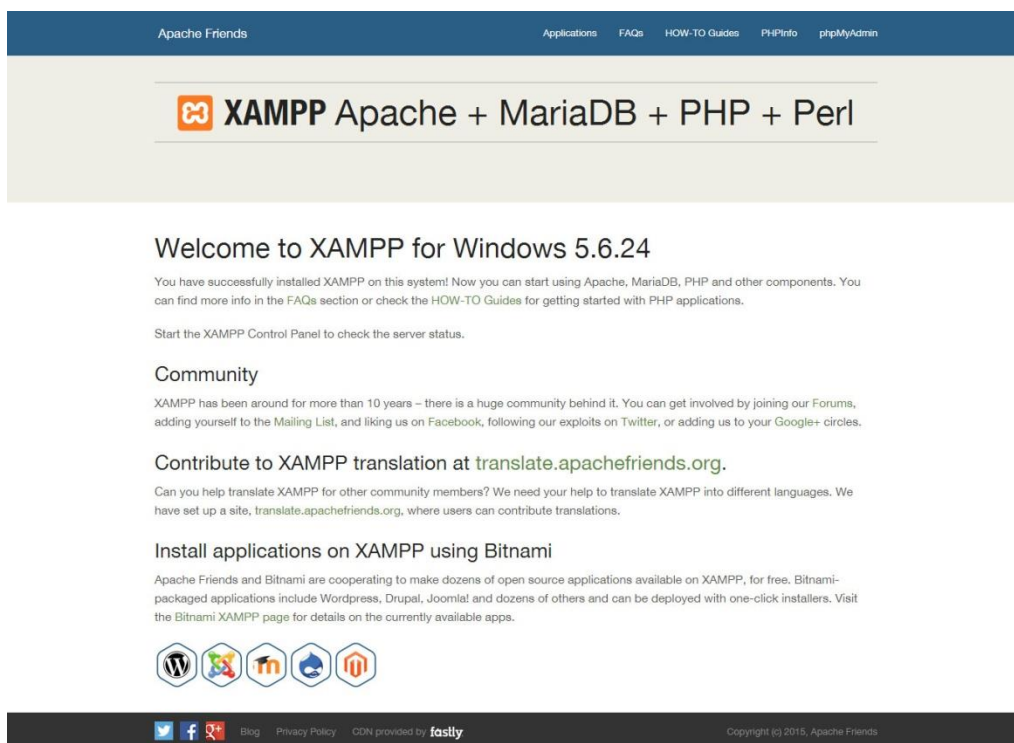
Bila tampilan jendela di layar desktop Anda sama seperti tampilan di layar komputer saya, Selamat Anda telah sukses melakukan instalasi XAMPP.

Menjalankan Apache, MySQL dan Membuka Localhost

Untuk menjalankan Apache Web Server dan MySQL, maka klik tombol Start pada keduanya, sehingga akan tampak tampilan seperti gambar berikut ini.



Untuk melakukan tes server lokal, Anda ketikkan alamat <http://localhost/> atau <http://127.0.0.1> pada address browser Anda untuk memastikan apakah XAMPP sudah terinstall. Jika berhasil maka akan terlihat tampilan utama seperti gambar berikut ini.



BAB 3

STRUKTUR DASAR SKRIP PHP

Untuk memberikan kemudahan bagi Anda dalam mempelajari bahasa pemrograman PHP, Saya akan mencoba memberikan tutorial secara step by step mengenai materi PHP mulai dari pemahaman dasar hingga implementasinya berupa studi kasus sederhana. Sebelum Kita mulai belajar, pastikan Anda sudah menyiapkan aplikasi-aplikasi yang dibutuhkan, Jalankan Service Apache dan MySQL Server Anda melalui XAMPP Control Panel, buka aplikasi Text Editor Anda seperti NotepadPlusPlus atau SublimeText.

Memahami Aturan Penulisan Skrip PHP

Aturan penulisan yang tidak boleh dilanggar adalah bahwa format penulisan skrip PHP harus dibuka oleh tag `<?php` dan ditutup dengan `?>`.

```
<?php
// Tulis Skrip PHP Disini
?>
```

Ekstensi file default untuk file PHP adalah **".php"**.

Sebuah file PHP biasanya berisi tag HTML, dan beberapa kode Skrip PHP. Skrip PHP memiliki kemampuan untuk mengintegrasikan format tampilan dengan database dalam membangun sebuah sistem. Untuk menyisipkan file HTML kedalam skrip PHP atau sebaliknya, Kita dapat menggunakan 2 cara, yaitu :

1. Embedded Script

```
<!DOCTYPE html>
<html>
<body>
<h1>Halaman PHP Pertamaku</h1>
<?php
echo "Halo, Saya sedang belajar bahasa pemrograman PHP";
?>
</body>
</html>
```

2. Non-Embedded Script

```
<?php
echo "<html>
<body>
    <h1>Halaman PHP Pertamaku</h1>
    <p>Halo, Saya sedang belajar bahasa pemrograman PHP</p>

</body>
</html>";

?>
```

Adapun cara yang akan Anda gunakan, silahkan pilih sesuai selera Anda, karena baik menggunakan cara Embedded Script maupun Non-Embedded Script hasilnya sama saja.

Memberikan Komentar pada Skrip PHP

Skrip komentar pada PHP digunakan untuk memberikan keterangan yang tidak terbaca pada saat dieksekusi oleh program. Hal ini sangatlah penting bagi seorang programmer untuk menandai atau memberikan keterangan pada masing-masing fungsi skrip.

Fungsi komentar pada skrip PHP digunakan untuk :

- Agar orang lain memahami apa yang Anda lakukan pada fungsi kode program tertentu.
- Mengingat pada diri Anda sendiri tentang bagian dari program yang sudah Anda kerjakan.

Ada dua cara memberikan komentar pada skrip PHP, yaitu :

- Komentar satu baris menggunakan tanda // atau #.
- Komentar lebih dari satu baris menggunakan tanda /* ... */.
- Berikut adalah contoh latihan menggunakan komentar pada skrip PHP

```
<?php
/*
Berikut adalah Program untuk menampilkan
Tanggal saat ini menggunakan
Fungsi date
*/

// Membuat variabel berisi tanggal hari ini
$tanggal = date("d-M-Y");

# Menampilkan tanggal di browser menggunakan fungsi echo
echo "Tanggal hari ini : $tanggal";

?>
```

Memahami Variabel di PHP

Variabel adalah merupakan suatu tempat / wadah untuk menampung data atau konstanta di memori yang mempunyai nilai atau data yang dapat berubah-ubah selama proses program.

Ketentuan menulis nama variabel yang benar di PHP

- Variabel selalu diawali dengan tanda dollar (\$), lalu diikuti nama variabel.
- Hanya ada tiga jenis karakter yang dapat digunakan untuk pemberian nama variabel, yaitu huruf, angka, dan garis bawah (_).
- Karakter pertama setelah tanda \$ harus berupa huruf atau garis bawah, tidak boleh angka atau yang lain.
- Jika nama variabel lebih dari satu kata, tidak boleh memisahkannya dengan spasi, alternatif lainnya dapat menggunakan garis bawah (_).
- Nama variabel bersifat case sensitif, artinya PHP membedakan huruf besar dan huruf kecil.

Berikut adalah contoh latihan untuk memudahkan Anda dalam memahami cara menulis variabel di PHP

```
<?php
$nama;
$nama_mahasiswa;
$alamat1;
$alamat1alamat2;
$_telpon;
echo "Jika teks ini tampil, <br>
maka semua pemberian nama variabel sudah benar";
?>
```

Memberi Nilai pada Variabel

```
<?php
// memberi nilai variabel
$nama_mahasiswa = "Novita Sari"; // String
$mata_kuliah = "Pemrograman Basis Data";
$jumsks = 3; // Integer
$nilai = 85.5; // Floating point

// Menampilkan nilai variabel
echo "Nama Mahasiswa : $nama_mahasiswa<br>
Mata Kuliah : $mata_kuliah<br>
Jumlah SKS : $jumsks<br>
Nilai : $nilai";
?>
```

Variabel dapat menyimpan berbagai jenis data, dan tipe data yang berbeda dapat melakukan hal-hal yang berbeda. PHP mendukung jenis data sebagai berikut:

String

String adalah sebuah tipe data yang terdiri dari kata, bisa berupa kata tunggal maupun kalimat. Penulisan string harus diapit oleh tanda petik, baik petik tunggal(' ') maupun petik ganda (" ").

Integer

Tipe data integer adalah tipe data yang berfungsi untuk menyimpan bilangan bulat, bukan desimal. Tipe data integer mempunyai range antara -2,147,483,648 sampai dengan 2,147,483,647.

Float

Tipe data floating point numbers biasa juga disebut dengan double, float atau real adalah tipe data yang berguna untuk menyimpan bilangan desimal.

Boolean

Tipe Data ini adalah tipe data yang paling sederhana. Hanya berupa true atau false. Penggunaan huruf besar atau kecil sama sekali tidak mempengaruhi dalam penulisan tipe data boolean.

Array

tipe data Array digunakan untuk menyimpan beberapa nilai dalam satu variabel tunggal.

Object

Objek adalah tipe data yang menyimpan data dan informasi tentang cara mengolah data tersebut.

NULL

Null adalah tipe data khusus yang hanya dapat memiliki satu nilai: NULL. Sebuah variabel tipe data NULL adalah variabel yang tidak memiliki nilai yang ditugaskan untuk itu. Jika variabel yang dibuat tanpa nilai, maka secara otomatis diberi nilai NULL.

Resource

Tipe Data resources ini di berfungsi untuk menyimpan resource, sumber atau alamat. Variabel tersebut hanya dapat dibuat oleh suatu fungsi khusus yang mengembalikan nilai berupa resource seperti penggunaan fungsi mysql_connect, mysql_query, fopen, opendir dan semacamnya.

BAB 4

OPERATOR ARITMATIKA DENGAN PHP

Dalam mempelajari bahasa pemrograman dasar, Kita selalu dihadapkan dengan istilah operator aritmatika. Operator Aritmatika adalah operator matematis yang terdiri dari operator penambahan, pengurangan, perkalian, pembagian, modulus, plus, dan minus.

Dan pada tutorial kali ini, saya akan memberikan contoh latihan operator aritmatika dengan menggunakan bahasa pemrograman PHP.

Jenis Operator Aritmatika dalam PHP

$\$a + \b	Penambahan	Total dari \$a dan \$b
$\$a - \b	Pengurangan	Selisih dari \$a dan \$b
$\$a * \b	Perkalian	Hasil kali dari \$a dan \$b
$\$a / \b	Pembagian	Hasil bagi dari \$a dan \$b
$\$a \% \b	Mod / Sisa hasil bagi	Sisa dari pembagian \$a/\$b

Contoh Penggunaan Operator Aritmatika di dalam PHP

Nama file : **aritmatika.php**

```
<?php
$a = 35;
$b = 10;
$hasil1 = $a + $b; // Penjumlahan
$hasil2 = $a - $b; // Pengurangan
$hasil3 = $a * $b; // Perkalian
$hasil4 = $a / $b; // Pembagian
$hasil5 = $a % $b; // Modulus

echo "Nilai a : $a<br>
Nilai b : $b <br>
Hasil penjumlahan $a + $b = $hasil1<br>
Hasil pengurangan $a - $b = $hasil2<br>
Hasil perkalian $a x $b = $hasil3<br>
```

```
Hasil pembagian $a / $b = $hasil4<br>
Sisa dari pembagian $a / $b = $hasil5";
?>
```

Contoh Penggunaan Operator Aritmatika dengan Form

Nama file : **form_jumlah.php**

```
<form method="POST" action="hasil.php">
    Nilai a : <input type="text" name="a"><br><br>
    Nilai b : <input type="text" name="b"><br><br>
    <input type="submit" value="Jumlahkan">
</form>
```

Nama file : **hasil.php**

```
<?php
// Ambil variabel dari form
$a = $_POST['a'];
$b = $_POST['b'];
$hasil = $a + $b;

echo "Nilai a : $a<br>
Nilai b : $b<br>
Hasil Penjumlahan $a + $b = $hasil";
?>
```

Tingkat Presedensi

Dalam menggunakan operator aritmatik, kita harus berhati-hati terutama ketika menggunakan lebih dari satu operator yang berbeda dalam satu statement perhitungan.

Sebagai contoh skrip berikut ini:

```
<?php
$a = 3 + 4 * 5 - 6;
echo $a;
?>
```

Apabila skrip di atas dijalankan, maka hasil yang muncul bukan 29, tapi 17. Mengapa demikian? itu dikarenakan operasi aritmatik yang dikerjakan terlebih dahulu adalah perkalian (*). Mengapa demikian? Karena perkalian memiliki tingkat presedensi yang lebih tinggi daripada + dan -. Setelah perkalian(*) dikerjakan, baru dikerjakan operasi penjumlahan(+) atau pengurangan(-).

Operasi penjumlahan(+) dan pengurangan(-) memiliki tingkat presedensi yang sama. Oleh karenanya, maka yang dikerjakan lebih dahulu adalah yang terletak di bagian yang lebih kiri, yaitu penjumlahan (+).

Bagaimana dengan operator pembagian (/)? Operator ini memiliki tingkat presedensi yang sama dengan perkalian (*). Keduanya memiliki tingkat presedensi yang lebih tinggi daripada penjumlahan (+) dan pengurangan (-). Sedangkan operator modulus (%) levelnya juga sama dengan perkalian (*) dan pembagian (/).

Jika kita ingin yang mengerjakan penjumlahan terlebih dulu, caranya adalah dengan memberikan tanda kurung seperti contoh di bawah ini.

```
<?php
$a = (3 + 4) * 5 - 6;
echo $a;
?>
```

Kombinasi Operator Aritmatik dan Assignment

Selain bentuk operator aritmatik yang dibahas sebelumnya, ada juga operator yang merupakan kombinasi antara operator aritmatik dengan assignment.

Dalam pemrograman seringkali dijumpai proses yang melibatkan proses increment (kenaikan nilai). Misalkan kita menginginkan proses increment dengan tingkat kenaikan

1, maka perintah yang dituliskan dapat berupa

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

Maksud dari perintah di atas adalah, nilai variabel *\$counter* yang baru diperoleh dari nilai *\$counter* yang lama ditambah 1.

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

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

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

Operator	Contoh	Operasi yang ekuivalen
+=	<i>\$x</i> += 2;	<i>\$x</i> = <i>\$x</i> + 2;
-=	<i>\$x</i> -= 4;	<i>\$x</i> = <i>\$x</i> - 4;

<code>*=</code>	<code>\$x *= 3;</code>	<code>\$x = \$x * 3;</code>
<code>/=</code>	<code>\$x /= 2;</code>	<code>\$x = \$x / 2;</code>
<code>%=</code>	<code>\$x %= 5;</code>	<code>\$x = \$x % 5;</code>
<code>.=</code>	<code>\$var_str.="hello";</code>	<code>\$var_str = \$var_str . "hello";</code>

Operator Pre/Post Increment dan Decrement

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

Berikut ini adalah operator yang termasuk jenis ini :

`$x++;`

ekuivalen dengan `$x += 1;` atau `$x = $x + 1;`

`$x--;`

ekuivalen dengan `$x -= 1;` atau `$x = $x - 1;`

Contoh :

```
<?php
$x = 4;
$x++;
echo "Nilai x yang baru : ". $x;
$x = 4;
$x--;
echo "Nilai x yang baru : ". $x;
?>
```

BAB 5

MEMAHAMI METHOD GET DAN POST PADA PHP

Form pada PHP mempunyai dua method pengiriman data, yaitu GET dan POST. Dengan menggunakan method GET, maka nilai variabel yang dikirimkan ke server melalui url pada address bar browser. Sedangkan method POST akan mengirimkan nilai variabel ke server secara terpisah, sehingga nilai variabelnya tidak terlihat.

Kapan menggunakan GET?

Informasi yang dikirim dari form dengan metode GET dapat terlihat oleh semua orang (semua nama variabel dan nilai-nilai variabel ditampilkan di URL). GET juga memiliki batasan pada jumlah informasi untuk mengirim. pembatasan itu sekitar 2000 karakter.

Namun, karena variabel ditampilkan di URL, ini memungkinkan untuk sebagai penunjuk halaman. Dan hal ini dapat berguna dalam beberapa kasus tertentu. GET dapat digunakan untuk mengirim data yang tidak bersifat sensitif.

Kapan menggunakan POST?

Informasi yang dikirim dari form dengan metode POST tidak dapat terlihat oleh orang lain (semua nama / nilai tersembunyi dalam HTTP Request) dan tidak memiliki batasan pada jumlah informasi untuk mengirim.

Selain itu POST mendukung fungsi seperti dukungan untuk multi-part binary input ketika mengupload file ke server. Namun, karena variabel tidak ditampilkan di URL, adalah tidak mungkin untuk penunjuk halaman. Pengembang lebih memilih POST untuk mengirim data form.

Contoh skrip dengan form method GET

Nama file : **formmethod_get.php**

```
<html>
<body>
<form method="get" action="kirimmethode_get.php">
Nama: <input type="text" name="nama"><br>
E-mail: <input type="text" name="email"><br>
<input type="submit" value="Kirim">
</form>
</body>
</html>
```

Nama file : **kirimmethod_get.php**

```
<?php
echo "Selamat Datang, $_GET[nama]<br>
Alamat E-mail Anda adalah : $_GET[email]";
?>
```

Contoh skrip dengan form method POST

Nama file : **formmethod_post.php**

```
<html>
<body>
<form method="post" action="kirimmethod_post.php">
Nama: <input type="text" name="nama"><br>
E-mail: <input type="text" name="email"><br>
<input type="submit" value="Kirim">
</form>
</body>
</html>
```

Nama file : **kirimmethod_post.php**

```
<?php
echo "Selamat Datang, $_POST[nama]<br>
Alamat E-mail Anda adalah : $_POST[email]";
?>
```

Berikut beberapa pertimbangan kapan sebaiknya Anda menggunakan method GET dan POST, diantaranya :

- Apabila data yang dikirim ke server bersifat rahasia, Anda dapat menggunakan method POST, misalnya untuk mengirimkan data username dan password.
- Apabila data yang dikirim ke server berupa data umum dan biasanya untuk memperjelas data yang dimasukkan di form, Anda dapat menggunakan method GET, misalnya untuk form pencarian data, polling dan lainnya.
- Jika data yang dikirimkan cukup banyak, gunakanlah method POST
- Jika data yang dikirimkan berupa file, gunakan method POST

BAB 6

STATEMENT CONTROL – IF

Saat ini kita masuk ke bab yang sangat penting dalam belajar programming yaitu statement kontrol. Mengapa pembahasan ini sangat penting ? Karena dengan statement ini kita bisa mengontrol aliran program / skrip.

Pada contoh pembahasan sebelumnya, saya hanya memberikan contoh skrip yang terdiri dari satu aliran saja, yaitu top-down. Artinya setiap perintah dijalankan satu persatu mulai dari atas sampai dengan paling bawah, dan selesai. Seperti contoh skrip berikut :

```
<?php
$panjang = 20;
$lebar = 25;
$luas = $panjang * $lebar;
echo "Luas persegi panjang = $luas satuan luas";
?>
```

Skrip di atas menggunakan sebuah aliran saja. Pertama jalankan perintah paling atas (\$panjang=20), lalu jalankan perintah kedua (\$lebar=25), kemudian jalankan perintah untuk menghitung luas persegi panjang, dan terakhir tampilkan hasilnya dan selesai.

Namun, tidak semua skrip/program hanya terdiri dari sebuah aliran saja. Seringkali kita membuat skrip yang di dalamnya terdapat perintah yang harus diulang beberapa kali, ataupun perintah yang hanya dijalankan pada kondisi atau berdasarkan syarat tertentu. Dalam kondisi ini kita dapat menggunakan statement kontrol.

Dalam programming, terdapat 2 statement kontrol yaitu : statement kontrol kondisional (bersyarat) dan statement kontrol perulangan (looping).

Statement kontrol kondisional adalah statement kontrol yang digunakan untuk mengatur kapan suatu perintah akan dijalankan.

Sedangkan statement kontrol perulangan digunakan untuk mengatur perintah yang dijalankan secara berulang-ulang.

Dalam PHP, terdapat dua buah statement kontrol yang termasuk statement kontrol kondisional, yaitu **IF** dan **SWITCH**. Sedangkan yang termasuk statement kontrol perulangan adalah : **FOR**, **WHILE**, **DO WHILE** dan **FOREACH**.

Pada bab ini kita hanya difokuskan pada statement kontrol IF.

Adapun sintaks atau penulisan IF adalah sebagai berikut :

```
if (kondisi)
{
    statement;
}
```

Bentuk IF yang lain :

```
if (kondisi)
{
    statement 1;
}
else
{
    statement 2;
}
```

Dalam aturan sintaks di atas, bagian 'statement' akan dijalankan jika 'kondisi' terpenuhi. Jika tidak, maka bagian 'statement' tidak akan dijalankan.

Operator Relasional

Untuk menyatakan suatu kondisi, kita dapat menggunakan operator pembandingan atau relasional. Hasil yang didapat dari penggunaan operator operasional adalah nilai BENAR atau SALAH. Berikut adalah beberapa operator relasional yang dapat digunakan :

Simbol	Makna
<	Lebih kecil dari
>	Lebih besar dari
>=	Lebih besar atau sama dengan
<=	Lebih kecil atau sama dengan

==	Sama dengan
!=	Tidak sama dengan

Contoh penggunaan operator relasional pada statement IF

```
<?php
$bil = 15;
if ($bil > 10){
    echo "Bilangan lebih besar dari 10";
}
?>
```

Pada skrip di atas, nilai dari variabel \$bil adalah 15. Selanjutnya terdapat statement IF yang di dalamnya terdapat syarat : Jika nilai \$bil lebih besar dari 10, maka browser akan menampilkan teks "Bilangan lebih besar dari 10". Pada contoh ini kondisi bernilai BENAR, yaitu 15>10. Dikarenakan kondisi bernilai BENAR, maka browser akan menampilkan teks Bilangan lebih besar dari 10.

Namun, coba Anda perhatikan skrip berikut :

```
<?php
$bil = 15;
if ($bil == 10)
{
    echo "Bilangan sama dengan 10";
}
?>
```

Pada skrip di atas, kondisi bernilai SALAH, dikarenakan nilai bil 15 tidak sama dengan 10. Dengan demikian perintah echo "Bilangan sama dengan 10" tidak akan dijalankan.

Operator Logika

Dalam pemrograman PHP, terdapat beberapa operator logika yang bisa kita gunakan, yaitu :

Operator	Makna
&&	AND

	OR
!	NOT

Berikut adalah tabel kebenaran dari masing-masing operator

Tabel hasil operasi logika **AND**

Operand1	Operand2	Hasil
BENAR	BENAR	BENAR
BENAR	SALAH	SALAH
SALAH	BENAR	SALAH
SALAH	SALAH	SALAH

Tabel hasil operasi logika **OR**

Operand1	Operand2	Hasil
BENAR	BENAR	BENAR
BENAR	SALAH	BENAR
SALAH	BENAR	BENAR
SALAH	SALAH	SALAH

Tabel hasil operasi logika **NOT**

Operand1	Hasil
BENAR	SALAH
SALAH	BENAR

Berikut adalah contoh penggunaan operator logika

```
<?php
$kdpegawai = "003";
$kdbagian = "A";
$gajipokok = 1300000;
if (($kdpegawai == "003") || ($kdbagian == "A"))
{
    $tunjstruktural = 0.05 * $gajipokok;
    echo "Tunjangan struktural Anda : $tunjstruktural";
}
else {
    $tunjstruktural = 0;
    echo "Anda tidak mendapat tunjangan struktural";
}
?>
```

Latihan fungsi IF :

```
<?php
$nilai = 70;
if ($nilai > 50)
{
    echo "Nilai lebih besar dari 50";
}
else if($nilai < 50)
{
    echo "Nilai lebih kecil dari 50";
}
else
{
    echo "Nilai sama dengan 50";
}
?>
```

Script di atas menghasilkan "Nilai lebih besar dari 50" dikarenakan kondisi "\$nilai > 50" bernilai BENAR. Ketika sudah menemukan kondisi yang bernilai benar, maka kondisi yang terletak di bawahnya diabaikan.

Anda boleh melakukan uji coba dengan melakukan perubahan berikut :

```
<?php
$nilai = 50;
if ($nilai > 50)
{
    echo "Nilai lebih besar dari 50";
}
else if($nilai < 50)
{
    echo "Nilai lebih kecil dari 50";
}
else
{
    echo "Nilai sama dengan 50";
}
?>
```

Pada skrip di atas, kondisi "\$nilai > 50" bernilai SALAH. Karena kondisi bernilai SALAH, maka akan dicek kondisi berikutnya "\$nilai < 50". Kondisi inipun juga SALAH. Dengan demikian program akan mengambil alternatif lainnya yaitu "Nilai sama dengan 50".

Skrip untuk mencari bilangan terbesar dari 3 buah bilangan : 6,9 dan 10

Bagaimana ide untuk mencari bilangan terbesar dari ketiga bilangan tersebut? Idennya adalah dengan membandingkan setiap bilangan.

Berikut adalah langkah-langkahnya :

1. Cari terlebih dahulu bilangan terbesar di antara bilangan pertama dan kedua (misalkan dinamakan 'max')
Untuk mencari 'max' caranya adalah dengan membandingkan bilangan pertama dengan kedua.
Jika bilangan pertama > bilangan kedua, maka nilai 'max' adalah bilangan pertama. Jika tidak, maka 'max' adalah bilangan kedua.
2. Setelah Kita tahu bilangan terbesar di antara bilangan pertama dan kedua ('max'), langkah selanjutnya adalah mencari bilangan terbesar diantara 'max' dengan bilangan ketiga. Nilai terbesar dari ketiga bilangan adalah nilai yang terbesar diantara bilangan ketiga dengan 'max'.

Nama File : **bilangan_terbesar.php**

```
<?php
$bil1 = 6;
$bil2 = 9;
$bil3 = 10;
if ($bil1 > $bil2)
{
    $max = $bil1;
}
else
```

```

{
    $max = $bil2;
}
if ($max > $bil3)
{
    $maxSemua = $max;
}
else
{
    $maxSemua = $bil3;
}
echo "Nilai terbesar dari ketiga bilangan adalah ".$maxSemua;
?>

```

Skrip untuk mengilustrasikan proses login.

Nama File : **form_login.php**

```

<form method="POST" action="login.php">
    Username : <input type="text" name="username"><br>
    Password : <input type="password" name="pass"><br>
    <input type="submit" name="submit" value="Login">
</form>

```

Nama File : **login.php**

```

<?php
$username = $_POST['username'];
$password = $_POST['pass'];

if((( $username=='galih') && ($password=='adivalucu')) || (( $username=='novi')
&& ($password=='cantik')) )
{
    echo "Login Sukses";
}
else
{
    echo "Login Gagal";
}
?>

```

Skrip untuk mengkonversi nilai angka ke huruf

Di salah satu Universitas menerapkan kebijakan konversi nilai angka ke huruf sebagai berikut :

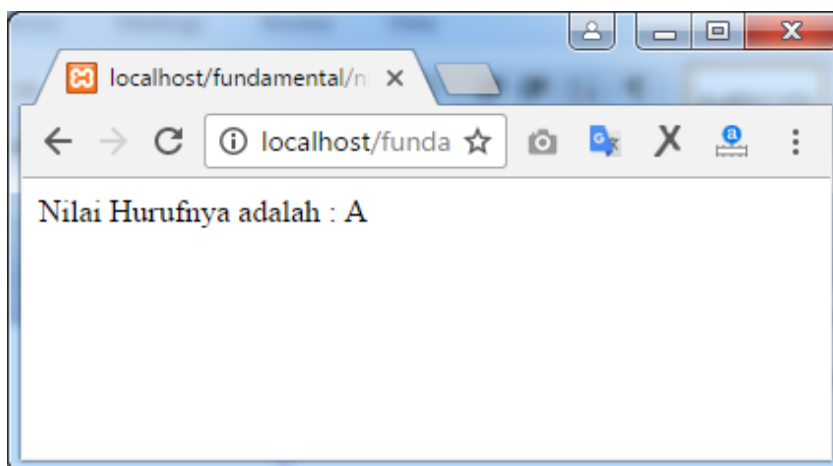
Nilai Angka	Nilai Huruf
80 s.d 100	A
70 s.d 79	B
60 s.d 69	C
50 s.d 59	D
0 s.d 49	E

Nama File : **nilai1.php**

```
<?php
$nilaiAngka = 85;
if (($nilaiAngka<0) || ($nilaiAngka>100))
{
    echo "Nilai angka yang diberikan salah";
}
else
{
    if (($nilaiAngka>=80)&&($nilaiAngka<=100))
    {
        $nilaiHuruf = "A";
    }
    else if (($nilaiAngka>=70)&&($nilaiAngka<=79))
    {
        $nilaiHuruf = "B";
    }
    else if (($nilaiAngka>=60)&&($nilaiAngka<=69))
    {
        $nilaiHuruf = "C";
    }
    else if (($nilaiAngka>=50)&&($nilaiAngka<=59))
    {
        $nilaiHuruf = "D";
    }
    else if (($nilaiAngka>=0)&&($nilaiAngka<=49))
    {
        $nilaiHuruf = "E";
    }

    echo "Nilai Hurufnya adalah : $nilaiHuruf";
}
?>
```

Dan hasil di browser adalah sebagai berikut :



Skrip di atas akan sama dengan skrip berikut ini :

Nama File : **nilai2.php**

```
<?php
$nilaiAngka = 85;
if (($nilaiAngka<0) || ($nilaiAngka>100))
{
    echo "Nilai angka yang diberikan salah";
}
else
{
    if ($nilaiAngka>=80)
    {
        $nilaiHuruf = "A";
    }
    else if($nilaiAngka>=70)
    {
        $nilaiHuruf = "B";
    }
    else if($nilaiAngka>=60)
    {
        $nilaiHuruf = "C";
    }
    else if($nilaiAngka>=50)
    {
        $nilaiHuruf = "D";
    }
    else if($nilaiAngka>=0)
    {
        $nilaiHuruf = "E";
    }

    echo "Nilai Hurufnya adalah : $nilaiHuruf";
}
?>
```


BAB 7

STATEMENT SWITCH

SWITCH adalah termasuk statement yang dapat dipergunakan untuk menyatakan suatu pernyataan kondisional atau bersyarat seperti halnya IF.

Adapun sintaks secara umum statement SWITCH dalam PHP adalah sebagai berikut :

```
switch (var)
{
case value1 : statement A;
    .
    .
    break;
case value2 : statement B;
    .
    .
    break;
case value3 : statement C;
    .
    .
    break;
.
.
default : statement D;
    .
    .
}
```

SWITCH tidak bisa digunakan untuk kondisional yang di dalamnya terdapat proses perbandingan seperti 'lebih besar dari', 'lebih kecil atau sama dengan', 'lebih besar atau sama dengan' dan 'tidak sama dengan'. Melihat hal ini memang penggunaan statement IF untuk menyatakan kondisional lebih fleksibel daripada SWITCH.

Berikut adalah kelebihan SWITCH dibanding dengan IF

Di dalam SWITCH, persyaratan hanya diuji sekali saja. Diantara sekian pilihan nilai, secara otomatis PHP hanya akan menjalankan blok yang nilainya cocok saja. Proses pencarian nilai yang cocok ini tidak dimulai dari pilihan nilai paling atas, namun akan langsung tertuju pada nilai yang cocok tersebut.

Sedangkan pada IF syarat diuji beberapa kali. Proses pengujian dimulai dari syarat IF paling atas, jika tidak terpenuhi maka proses akan berlanjut ke nilai berikutnya.

Akibatnya, apabila skrip Anda sangat kompleks atau banyak menggunakan pernyataan kondisional, penggunaan SWITCH akan menghasilkan proses yang lebih cepat dari IF.

Latihan fungsi SWITCH :

Skrip untuk menampilkan nama bulan sekarang ke dalam bahasa Indonesia

Nama File : **bulan_indonesia.php**

```
<?php
$angkaBln = date("n");

switch($angkaBln)
{
    case 1 : $namaBln = "Januari";
        break;
    case 2 : $namaBln = "Februari";
        break;
    case 3 : $namaBln = "Maret";
        break;
    case 4 : $namaBln = "April";
        break;
    case 5 : $namaBln = "Mei";
        break;
    case 6 : $namaBln = "Juni";
        break;
    case 7 : $namaBln = "Juli";
        break;
    case 8 : $namaBln = "Agustus";
        break;
    case 9 : $namaBln = "September";
        break;
    case 10 : $namaBln = "Oktober";
        break;
    case 11 : $namaBln = "November";
        break;
    case 12 : $namaBln = "Desember";
        break;
}

echo "Nama bulan sekarang adalah : $namaBln";
?>
```

BAB 8

STATEMENT FOR

Statement ini merupakan salah satu statement yang dapat digunakan untuk menyatakan proses yang diulang-ulang atau looping.

Coba Anda perhatikan pernyataan berikut :

Hari ini saya minum kopi 10 kali.

Hari ini saya terus minum kopi sampai saya kembang.

Kedua pernyataan di atas sama-sama mengandung pengertian sebuah proses yang diulang-ulang yaitu 'minum kopi'. Namun proses 'minum kopi' dari kedua pernyataan tersebut berbeda cara berhentinya. Pernyataan pertama menunjukkan bahwa 'minum kopi' akan diulang-ulang sampai 10 kali. Setelah sampai 10 kali, maka 'minum kopi' akan berhenti. Sedangkan pernyataan kedua, dalam hal ini jumlah berapa kali 'minum kopi' tidak diketahui, namun hanyalah syarat kapan perulangan ini akan berhenti yaitu ketika 'sudah kembang'.

Untuk menyatakan looping yang telah diketahui jumlah perulangannya, kita dapat menggunakan statement FOR atau WHILE.

Sedangkan bila perulangannya tidak diketahui jumlahnya atau hanya diketahui kondisi kapan harus berakhir, maka kita dapat menggunakan WHILE.

Pada bab ini Kita akan membahas terlebih dahulu statement FOR.

Sintaks :

```
for (variabel = nilaiawal; variabel operatorlogika nilaiakhir; increment)
{
    Proses yang diulang;
}
```

Contoh skrip sederhana :

```
<?php
for($i=1; $i<10; $i++)
{
    echo "data ke-$i <br>";
}
?>
```

Penjelasan Skrip :

1. Nilai awal \$i adalah 1 (\$i=1), maka tampilkan "data ke-1".

2. Kemudian nilai i ditambah dengan 1, lihat pola perulangannya adalah $i++$, artinya pola tersebut adalah $i=i+1$, setelah ditambah 1, maka sekarang menjadi $i=2$.
3. Selanjutnya dicek syarat perulangannya, apakah nilai i sekarang yang bernilai 2 lebih kecil dari 10 (syarat perulangannya $i<10$), jika memenuhi syarat (TRUE), maka tampilkan "data ke-2".
4. Sekarang nilai i adalah 2, tambahkan lagi sesuai dengan pola perulangannya, maka nilai i berikutnya adalah 3. Cek lagi apakah 3 lebih kecil dari 10, jika iya tampilkan "data ke-3".
5. Begitu seterusnya nilai i ditambahkan 1 sampai bernilai 9 akan ditampilkan terus sampai i bernilai 10 dan tidak akan ditampilkan, karena sudah tidak memenuhi syarat perulangan $10<10 = \text{FALSE}$.

Skrip perulangan FOR dalam bentuk Form.

Nama file : **form_for.php**

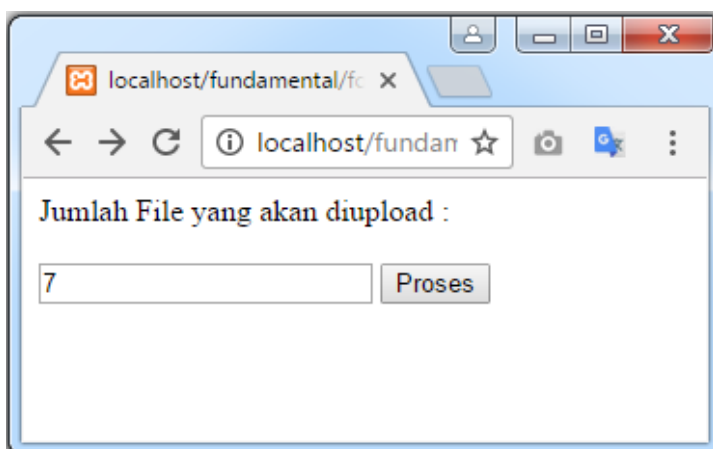
```
<form method="POST" action="hasil_for.php">
    Jumlah File yang akan diupload : <br><br>
    <input type="text" name="jumlah">
    <input type="submit" value="Proses">
</form>
```

Nama file : **hasil_for.php**

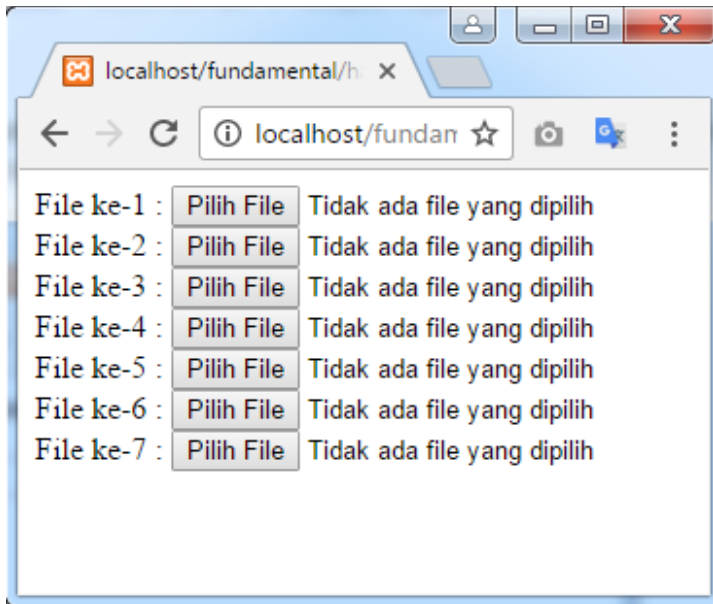
```
<?php
$jumlah = $_POST['jumlah'];

for ($i=1; $i<=$jumlah; $i++)
{
    echo "File ke- $i$  : <input type='file' name='nama $i$ '> <br>";
}
?>
```

Pertama, jalankan skrip form_for.php, kemudian isikan berapa jumlah file yang akan diupload, misalnya 7, lalu klik tombol **Proses**.



Maka akan tampil 7 komponen input file seperti berikut :



Skrip untuk menampilkan bilangan bulat 5 s.d 100 yang berkelipatan 10.

Konsep untuk menampilkan bilangan bulat tersebut bisa menggunakan looping. Looping dimulai dari bilangan 5, 6, 7 dst sampai dengan 100. Untuk setiap bilangan akan dicek apakah kelipatan 10 atau tidak. Jika bilangan tersebut merupakan kelipatan 10, maka akan ditampilkan. Jika tidak, maka akan diabaikan.

Nama File : **latihan_for.php**

```
<?php
for($bil=5; $bil <=100; $bil++)
{
    if ($bil % 10 == 0)
    {
        echo "$bil <br>";
    }
}
?>
```

Skrip untuk menjumlahkan bilangan bulat 2 s.d 50

Yang dimaksud disini adalah mencari hasil dari 2+3+4+5+6+..+50

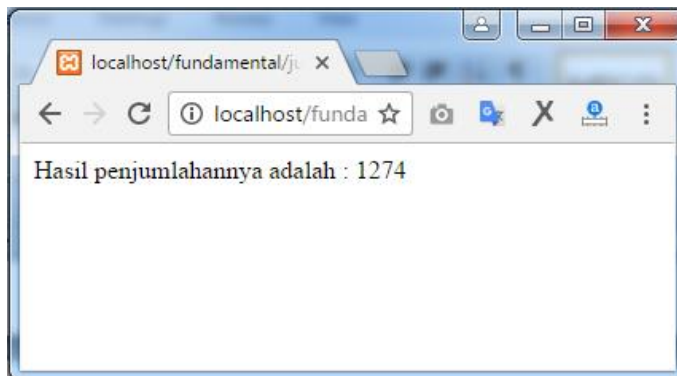
Konsepnya adalah kita ambil variabel sembarang terlebih dahulu, misalkan namanya 'jumlah'. Pertama-tama variabel 'jumlah' ini bernilai 0. Selanjutnya kita jumlahkan dengan bilangan pertama yaitu 2, dan hasilnya kita simpan ke dalam 'jumlah' yang baru. Sehingga nilai 'jumlah' sekarang adalah 0 + 2 = 2, selanjutnya kita jumlahkan lagi 'jumlah' dengan bilangan kedua yaitu 3. Dan simpan kembali ke dalam 'jumlah' lagi. Sehingga hasil 'jumlah' yang baru adalah 2 + 3 = 5. Berikutnya 'jumlah' dijumlahkan kembali dengan bilangan ketiga yaitu 4, dan diperoleh 'jumlah' adalah 5 + 4 = 9. proses ini berlanjut sampai dengan 50.

Nama File : **jumlah_for.php**

```
<?php
$jumlah = 0;
for ($bil=2; $bil<=50;$bil++)
{
    $jumlah = $jumlah + $bil;
}

echo "Hasil penjumlahannya adalah : $jumlah";
?>
```

Hasil latihan di browser adalah sebagai berikut :



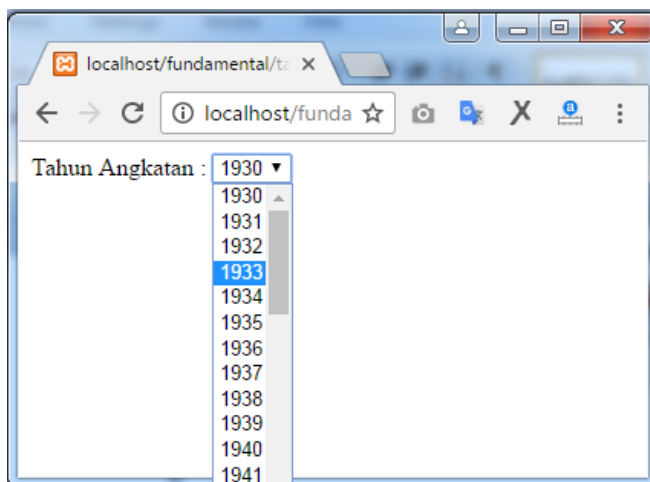
Skrip untuk menghasilkan list combo box berisi tahun mulai dari tahun 1930 s.d 2017

Nama File : **tahun_for.php**

```
<?php
echo "Tahun Angkatan : <select name='tahun'>";

for ($tahun=1930;$tahun<=2017;$tahun++)
{
    echo "<option value='$tahun'>$tahun</option>";
}
echo "</select>";
?>
```

Hasil latihan di browser adalah sebagai berikut :



BAB 9

STATEMENT WHILE

While adalah perintah pada **PHP** dan bahasa pemrograman lainnya untuk membuat sebuah perulangan yang tidak di ketahui berapa kali perulangan tersebut di lakukan.

Sering kali pada saat menuliskan kode program kita membutuhkan **perintah perulangan while** ini untuk membuat perulangan yang tidak perlu kita ketahui berapa kali perulangan harus di lakukan. Misalnya seperti pada saat kita ingin menampilkan data dari database.

Berikut adalah penulisan dan struktur cara penggunaan perulangan WHILE di PHP.

```
while (kondisi)
{
    statement
}
```

Dapat di lihat pada **contoh penulisan while** di atas setelah menuliskan while kemudian di dalam tanda kurung berisi kondisi nilai yang harus terpenuhi agar **while** dijalankan. Kemudian di dalam kurung kurawal berisi statement yang di jalankan dalam bentuk perulangan apabila kondisi terpenuhi. statement di sini maksudnya adalah perintah syntax yang akan di jalankan dalam bentuk **perulangan** apabila kondisi terpenuhi.

Skrip latihan statement WHILE

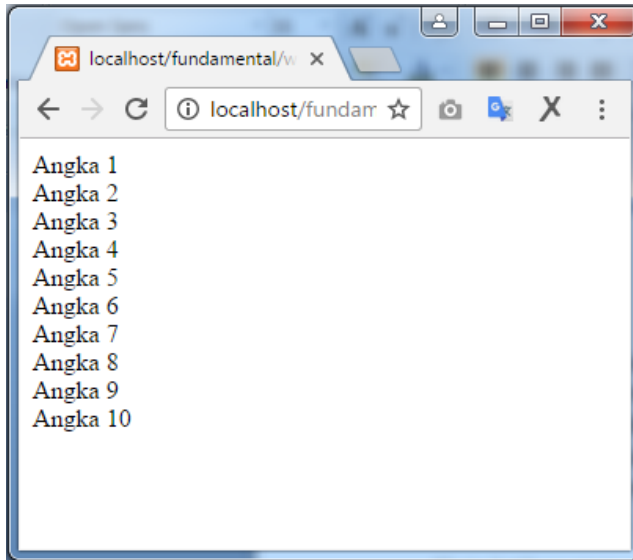
```
<?php
$x = 1;

while($x <= 10) {
    echo "Angka $x <br>";
    $x++;
}
?>
```

Nilai awal dari \$x adalah 1. Selanjutnya terdapat WHILE dengan kondisi \$x <= 10. Apabila dicek kondisinya, maka dalam hal ini kondisi terpenuhi dan bernilai BENAR, yaitu 1 <= 10, dan akan menampilkan "Angka 1". Berikutnya nilai \$x akan bertambah menjadi 2 (\$x++).

Perulangan akan terus dilanjutkan dan akan berhenti sampai kondisi \$x <=10 tidak terpenuhi.

Hasil setelah dijalankan browser adalah sebagai berikut :



Contoh-contoh skrip menggunakan WHILE

Skrip untuk menampilkan bilangan bulat 5 s.d 100 yang berkelipatan 10

```
<?php
$bil = 5;
while($bil <= 100)
{
    if($bil % 10==0)
    {
        echo "$bil <br>";
    }
    $bil++;
}
?>
```

Skrip untuk menjumlahkan bilangan bulat 2 s.d 50

```
<?php
$jumlah = 0;
$bil=2;

while($bil <= 50)
{
    $jumlah = $jumlah + $bil;
    $bil++;
}

echo "Hasilnya adalah : $jumlah";
?>
```

Skrip untuk mencari banyaknya bilangan bulat mulai dari 3 s.d 127 yang merupakan kelipatan 6.

```
<?php
$hitung = 0;
$bil = 3;
while($bil<=127)
```



```

{
    if($bil % 6 == 0)
    {
        $hitung = $hitung + 1;
    }
    $bil++;
}
echo "Banyaknya bilangan bulat adalah : $hitung";
?>

```

Skrip untuk membuat tabel dengan n baris dan m kolom secara otomatis

```

<?php
$jumBaris = 10;
$jumKolom = 6;

echo "<table border='1'>";

$baris = 1;
while($baris <= $jumBaris)
{
    echo "<tr>";
    $kolom = 1;
    while($kolom <= $jumKolom)
    {
        echo "<td>...</td>";
        $kolom++;
    }
    echo "</tr>";
    $baris++;
}
echo "</table>";
?>

```

Skrip untuk menghasilkan listi combo box berisi tahun mulai dari 1930 s.d 2017 secara otomatis.

```

<?php
echo "Pilih Tahun : <select name='tahun'>";

$tahun = 1930;

while($tahun <= 2017)
{
    echo "<option value='$tahun'>$tahun</option>";
    $tahun++;
}

echo "</select>";
?>

```

BAB 10

MENGENAL ARRAY

Memahami Fungsi Array

Array adalah suatu tipe data variabel yang mampu menyimpan banyak data atau nilai. Perhatikan skrip berikut :

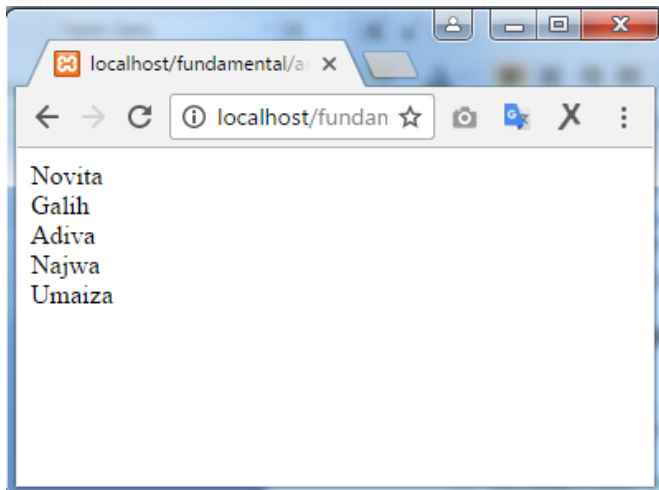
```
$nama = "Novita";  
$nama = "Galih";  
$nama = "Adiva";  
$nama = "Najwa";  
$nama = "Umaiza";
```

Menurut Anda, ketika data variabel nama ditampilkan, nama siapa yang akan tampil? Tentu saja hanya satu nama saja, yaitu variabel terakhir yang berisi data "Umaiza", karena satu variabel hanya dapat menyimpan satu data. Bagaimana caranya agar semua data dalam variabel tersebut bisa ditampilkan semua? Disinilah fungsinya Array. Perhatikan contoh skrip berikut :

Nama File : **array.php**

```
<?php  
// Memberi nama array serta isi datanya  
$nama = array("Novita", "Galih", "Adiva", "Najwa", "Umaiza");  
  
// Menampilkan semua isi data array  
echo "$nama[0] <br>";  
echo "$nama[1] <br>";  
echo "$nama[2] <br>";  
echo "$nama[3] <br>";  
echo "$nama[4] ";  
?>
```

Jalankan skrip array.php, maka browser akan menampilkan kelima nama yang ada di dalam data array.



Bagaimana caranya jika data array-nya banyak, dimana Anda harus menuliskan satu persatu array beserta indeks untuk menampilkan datanya, tentu ini sangat tidak efisien. Disinilah peran perulangan (looping) dibutuhkan untuk mengatasi masalah tersebut. Berikut adalah contoh skrip penyelesaiannya :

Nama File : **foreach.php**

```
<?php
// Memberi nama array serta isi datanya
$nama = array("Novita", "Galih", "Adiva", "Najwa", "Umaiza");

// Menampilkan semua isi data array
foreach($nama as $stampilnama)
{
    echo "$stampilnama <br>";
}
?>
```

Kali ini, kita akan menerapkan array pada komponen form yaitu checkbox yang menyediakan beberapa pilihan untuk memilih data.

Buat contoh skrip implementasi array pada komponen form seperti berikut :

Nama File : **form_hobi.php**

```
<form method="POST" action="hasil_hobi.php">
    Pilih Hobi Anda : <br>
    <input name="hobi[]" type="checkbox" value="Membaca">Membaca
    <input name="hobi[]" type="checkbox" value="Olahraga">Olahraga
    <input name="hobi[]" type="checkbox" value="Fotografi">Fotografi
    <input name="hobi[]" type="checkbox" value="Traveling">Traveling
    <input name="hobi[]" type="checkbox" value="Programming">Programming
    <input name="hobi[]" type="checkbox" value="Otomotif">Otomotif
    <input name="hobi[]" type="checkbox" value="Memancing">Memancing
    <input name="hobi[]" type="checkbox" value="Musik">Musik
    <input name="hobi[]" type="checkbox" value="Menyanyi">Menyanyi<br><br>
    <input type="submit" value="Proses">
</form>
```

Nama File : **hasil_hobi.php**

```
<?php
echo "Hobi yang kamu pilih : <br>";
$hobi = $_POST['hobi'];

// Hitung jumlah hobi yang dicek atau dipilih
$jumlah_hobi = count($hobi);

for($i=0; $i<$jumlah_hobi; $i++)
{
    echo "$hobi[$i] <br>";
}
?>
```

BAB 11

FUNGSI (FUNCTION)

Pengertian Fungsi (Function) dalam PHP

Dalam merancang kode program, seringkali kita membuat kode yang memiliki tugas yang sama secara berulang-ulang, seperti membaca tabel dari database, menampilkan penjumlahan, dan lain-lain. Tugas yang sama ini akan lebih efektif jika dipisahkan dari program utama, dan dirancang menjadi sebuah fungsi.

Fungsi (atau Function) dalam bahasa pemrograman adalah kode program yang dirancang untuk menyelesaikan sebuah tugas tertentu, dan merupakan bagian dari program utama. Kita dapat membuat fungsi sendiri, atau menggunakan fungsi yang dibuat oleh programmer lain.

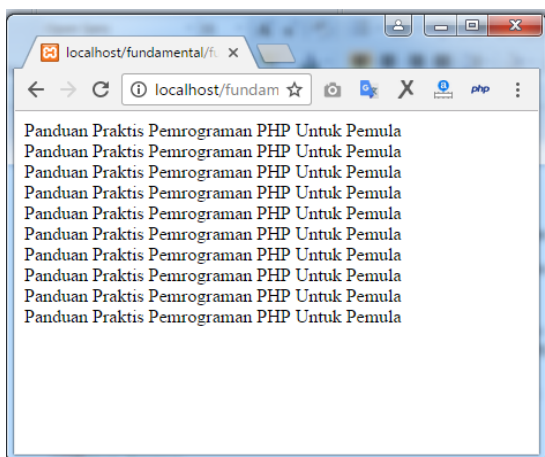
Untuk memudahkan Anda dalam memahami fungsi di PHP, ketikkan skrip berikut :

Nama File : **fungsi_perulangan.php**

```
<?php
// Membuat Function
function ulang($angka, $tulisan)
{
    for($i=0; $i<$angka; $i++)
    {
        echo "$tulisan <br>";
    }
}

// Menggunakan atau memanggil function
// menampilkan tulisan sebanyak 10 kali
ulang(10, "Panduan Praktis Pemrograman PHP Untuk Pemula");
?>
```

Jalankan skrip fungsi_perulangan.php, maka tulisan 'Panduan Praktis Pemrograman PHP Untuk Pemula' sebanyak 10 kali di browser.



Kerjakan latihan-latihan berikut untuk memudahkan Anda dalam memahami fungsi (function).

Skrip membuat fungsi penjumlahan dua bilangan

Nama File : **fungsi_jumlah.php**

```
<?php
function jumlah($a,$b)
{
    $c = $a + $b;
    return $c;
}

echo "Hasil Penjumlahan : ". jumlah(5,12);
?>
```

Skrip membuat fungsi penjumlahan (3+5) ditambah (7+6)

Nama File : **fungsi_jumlah2.php**

```
<?php
function jumlah($a,$b)
{
    $c =$a + $b;
    return $c;
}

$hasil = jumlah(jumlah(3,5),jumlah(7,6));
echo "Hasil Penjumlahan : " . $hasil;
?>
```

Skrip membuat fungsi dropdown nama hari

Nama File : **fungsi_combohari.php**

```
<?php
function combohari($namahari)
{
    $hari =
array("Minggu", "Senin", "Selasa", "Rabu", "Kamis", "Jumat", "Sabtu");

    echo "<select name='namahari'>";
    for ($i=0; $i<=6; $i++){
        echo "<option value='$hari[$i]'>$hari[$i]</option>";
    }
    echo "</select>";
}
?>
```

Nama File : **hari_lahir.php**

```
<?php
// Memanggil file yang berisi function combohari
include "fungsi_combohari.php";

echo "Hari Lahir Anda : ";

// Memanggil fungsi combohari
combohari("harilahir");
?>
```

Built in Function dalam PHP

Sebuah function dapat dibuat sendiri seperti halnya di atas, namun ada juga function yang disediakan oleh PHP untuk dapat juga kita gunakan secara langsung. Function yang sudah disediakan oleh PHP disebut dengan built in functions.

Berikut ini adalah beberapa built in functions berdasarkan kategori penggunaannya.

Mathematics Built in Functions

abs()

Untuk menghitung nilai mutlak atau absolute.

Contoh :

```
<?php
$bil = -10;
echo abs($bil); // menghasilkan 10
?>
```

ceil()

Untuk membulatkan ke atas suatu bilangan real

Contoh :

```
<?php
$bil = 19.1;
echo ceil($bil); // menghasilkan 20
?>
```

floor()

Untuk membulatkan ke bawah suatu bilangan real

Contoh :

```
<?php
$bil = 19.5;
echo floor($bil); // menghasilkan 19
?>
```

max()

Mencari nilai terbesar dari suatu data bertipe array

Contoh :

```
<?php
$data = array(19, 23, 11, 45);
$max = max($data);
echo "Nilai max = $max"; // menghasilkan 45;
?>
```

min()

Mencari nilai minimum dari suatu tabel bertipe array

mt_rand(x,y)

Untuk menghasilkan bilangan bulat random antara x s.d y

Contoh :

```
<?php
$random = mt_rand(4,10); // menghasilkan bilangan random antara 4 s.d 10
echo $random;
?>
```

pow(x,y)

Digunakan untuk mencari hasil x pangkat y

Contoh :

```
<?php
$hasil = pow(4,-5); // menghitung 4 pangkat -5
echo $hasil;
?>
```

round()

Membulatkan bilangan real ke bawah jika desimal di belakang koma kurang dari 0.5, dan membulatkan ke atas jika di belakang koma lebih dari atau sama dengan 0.5

Contoh :

```
<?php
echo round(10.23) . "<br>"; // menghasilkan 10
echo round(10.6) . "<br>"; // menghasilkan 11
echo round(-10.2); // menghasilkan -10
?>
```

Array Built in Functions

Berikut ini beberapa function yang dapat digunakan untuk mengolah data berbentuk array.

array_multisort()

Digunakan untuk mensorting beberapa array terkait sekaligus. Dapat juga digunakan untuk mensorting dalam array saja.

Sintaks dari penggunaan array_multisort() adalah

`array_multisort(x, metode, a, b, c, ...);`

x = adalah array yang digunakan sebagai acuan dalam sorting

metode = adalah metode yang digunakan sorting (ascending atau descending)

a, b, c, ... = adalah array lain yang ikut disorting.

Contoh :

Skrip berikut akan mengurutkan data mahasiswa (NIM dan NAMA) berdasarkan NIM dari 2 buah array yang bersesuaian secara ascending.

```
<?php
$nim = array("A0197008", "A0197009", "A0197005", "A0197002", "A0197003");
$nama = array("Andi", "Budi", "Wati", "Tono", "Yudi");

array_multisort($nim, SORT_ASC, $nama);

for ($i=0; $i<=count($nim)-1; $i++) {
    echo $nim[$i]. " ". $nama[$i]. "<br>";
}
?>
```

array_rand()

Digunakan untuk mengambil salah satu elemen dari array secara random.

Contoh :

```
<?php
$bil = array(3,1,4,5,2,6);
$ambilAcak = array_rand($bil);
echo $ambilAcak;
?>
```

array_reverse()

Digunakan untuk membalik urutan data dalam array

Contoh :

```
<?php
$data = array("mangga", "jambu", "jeruk", "apel");
$balik = array_reverse($data);

foreach($balik as $index => $buah)
{
    echo "$buah <br>"; // Menampilkan apel, jeruk, jambu, mangga
}
?>
```

array_search()

Digunakan untuk mencari nomor urut elemen tertentu dalam array (dimulai dari 0)

Contoh :

Skrip berikut ini akan mencari nomor urut elemen dari data 'jambu' dalam array buah

```
<?php
$data = array("mangga", "jambu", "jeruk", "apel");
$no = array_search("jambu", $data);

echo $no; // Akan menampilkan 1 (nomor urut elemen dalam array)
?>
```

count()

Digunakan untuk menghitung jumlah elemen yang ada dalam suatu array.

String Built in Functions

explode()

Digunakan untuk memecah suatu string ke dalam array berdasarkan karakter tertentu.

Contoh :

```
<?php
$date = "14-02-2017";

$specah = explode("-", $date); // Memecah string berdasarkan karakter
echo "Tanggal : $specah[0] <br>"; // Menampilkan 14
echo "Bulan : $specah[1] <br>"; // Menampilkan 02
echo "Tahun $specah[2]"; // Menampilkan 2017
?>
```

md5()

Digunakan untuk mengenkripsi suatu data string. Biasanya function ini digunakan untuk keperluan enkripsi password login ke suatu aplikasi. Hasilnya adalah suatu string yang sulit untuk dibaca dengan panjang 32 karakter.

Contoh :

```
<?php
$passAsli = "indosmart";
$passEnkripsi = md5($passAsli);
echo $passEnkripsi; // Menampilkan hasil enkripsi :
d7159819c13361ce23c5374bc2f9c7f1
?>
```

number_format()

Memformat angka dengan mengelompokkan berdasarkan ribuan.

Contoh :

Berikut ini contoh script untuk menyajikan angka Rp. 1002372 menjadi bentuk Rp. 1.002.372,-

```
<?php
$harga = 1002372;
$formatted = number_format($harga, 0, "", ".");
echo "Harganya Rp. ".$formatted.",-";
?>
```

Berikut ini contoh script untuk menyajikan bilangan real 123456789.111111 menjadi bentuk 123.456.789,111 (tanda pemisah desimal menggunakan koma, jumlah digit desimal sebanyak 3 dan menggunakan tanda titik untuk pemisah ribuan).

```
<?php
$bil = 123456789.111111;
$formatted = number_format($bil, 3, ",", ".");
echo "Bilangannya : ".$formatted;
?>
```

str_replace()

Digunakan untuk mereplace suatu substring dengan substring lain dalam string.

Contoh :

```
<?php
$kalimat = "ular melingker lingker di atas pager";
$replace = str_replace("er", "ar", $kalimat);

echo $replace; // menampilkan ular melingkar lingkar di atas pagar
?>
```

strip_tags()

Digunakan untuk menghilangkan tag HTML dalam suatu string. Biasanya untuk keamanan dalam form guna mencegah disisipkannya tag-tag HTML yang dapat merusak halaman website.

Contoh :

```
<?php
$string = "<font face='Verdana' size='2'>Hello World</font>";
$hilangkanTag = strip_tags($string);

echo $hilangkanTag; // akan menghasilkan Hello World
?>
```

strlen()

Digunakan untuk menghitung jumlah karakter dari suatu string.

Contoh :

```
<?php
$kata = "Hello World";
$jumKarakter = strlen($kata);

echo $jumKarakter; // akan menghasilkan 11 (spasi dihitung)
?>
```

strtolower()

Digunakan untuk mengubah semua huruf penyusun string menjadi huruf kecil.

Contoh :

```
<?php
$kata = "Adiva Yasna Umaiza";
$kecil = strtolower($kata);

echo $kecil; // akan menghasilkan adiva yasna umaiza
?>
```

strtoupper()

Merupakan kebalikan dari strtolower();

substr_count ()

Menghitung jumlah substring dalam suatu string

```
<?php
$lagu = "topi saya bundar, bundar topi saya, kalau tidak bundar, bukan
topi saya";
$hitungTopi = substr_count($lagu, "topi");

echo $hitungTopi; // akan menghasilkan 3
?>
```

substr ()

Mengambil sejumlah karakter substring dari suatu string.

Contoh :

Skrip berikut ini akan mengambil tahun angkatan mahasiswa dari NIM 'A0197001' dimana tahun angkatan terletak pada digit ke 4 dan ke 5 dari NIM (angkatan 97).

```
<?php
$nim = "A0197001";
$angkatan = substr($nim, 3, 2);

echo $angkatan; // menampilkan 97
?>
```

Keterangan :

Parameter pertama dari substr() menunjukkan string asling. Parameter kedua menunjukkan posisi awal substring yang akan diambil (dihitung mulai dari 0 dari karakter paling kiri string), dan parameter ketiga menunjukkan jumlah karakter substring yang akan diambil.

Sehingga dari contoh di atas, karena posisi awal tahun angkatan itu terletak pada karakter ke-3 dari string NIM, dan tahun angkatan ini terdiri dari 2 karakter maka perintahnya adalah

substr(\$nim, 3, 2)

BAB 12

MYSQL DAN PHPMYADMIN

Perbedaan SQL dan MySQL

Tahukah Anda apa perbedaan antara SQL dan MySQL? Ada sebagian orang yang menganggap dua istilah ini adalah sama, padahal apabila Kita rajin membaca referensi yang ada maka sangat jelas perbedaannya. Seperti dikutip dari Wikipedia, SQL (Structured Query Language) adalah sebuah bahasa yang digunakan untuk mengakses data dalam basis data relasional. Bahasa ini merupakan bahasa standar yang digunakan dalam manajemen basis data relasional. Saat ini hampir semua server basis data yang ada mendukung bahasa ini untuk melakukan manajemen datanya.

Secara umum, SQL terdiri dari dua bahasa, yaitu Data Definition Language (DDL) dan Data Manipulation Language (DML).

Data Definition Language (DDL)

DDL digunakan untuk mendefinisikan, mengubah, serta menghapus basis data dan objek-objek yang diperlukan dalam basis data, misalnya tabel, view, user, dan sebagainya. Secara umum, DDL yang digunakan adalah CREATE untuk membuat objek baru, USE untuk menggunakan objek, ALTER untuk mengubah objek yang sudah ada, dan DROP untuk menghapus objek. DDL biasanya digunakan oleh administrator basis data dalam pembuatan sebuah aplikasi basis data.

Data Manipulation Language (DML)

DML digunakan untuk memanipulasi data yang ada dalam suatu tabel. Perintah yang umum dilakukan adalah:

- **SELECT** untuk menampilkan data
- **INSERT** untuk menambahkan data baru
- **UPDATE** untuk mengubah data yang sudah ada
- **DELETE** untuk menghapus data

Sedangkan MySQL adalah sebuah perangkat lunak sistem manajemen basis data atau DBMS (Database Management System) yang multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL dapat Anda dapatkan secara gratis di bawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus di mana penggunaannya tidak cocok dengan penggunaan GPL. Selain MySQL, Anda juga mungkin pernah mendengar perangkat lunak DBMS lainnya seperti : Microsoft SQL Server, Sybase, Interbase, XBase, Firebird,

PostgreSQL, Microsoft Access, dBase III, Paradox, FoxPro, Visual FoxPro, Visual dBase, Lotus Smart, db2, MongoDB dan lain-lain.

Jadi Bisa disimpulkan secara sederhana bahwa SQL adalah struktur bahasa untuk mengakses database, sedangkan MySQL adalah salah satu aplikasi database atau software untuk manajemen database.

Keunggulan MySQL

Berikut adalah beberapa keunggulan dari MySQL, antara lain :

Portabilitas

MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.

Open Source

MySQL didistribusikan sebagai perangkat lunak sumber terbuka, dibawah lisensi GPL sehingga dapat digunakan secara gratis.

Multi User

MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.

Performance Tuning

MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.

Ragam tipe data

MySQL memiliki ragam tipe data yang sangat kaya, seperti signed / unsigned integer, float, double, char, text, date, timestamp, dan lain-lain.

Perintah dan Fungsi

MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah Select dan Where dalam perintah (query).

Keamanan

MySQL memiliki beberapa lapisan keamanan seperti level subnetmask, nama host, dan izin akses user dengan sistem perizinan yang mendetail serta sandi terenkripsi.

Skalabilitas dan Pembatasan

MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (records) lebih dari 50 juta dan 60 ribu tabel serta 5 miliar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

Konektivitas

MySQL dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, Unix socket (UNIX), atau Named Pipes (NT).

Lokalisasi

MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya.

Interface

MySQL memiliki antar muka (interface) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (Application Programming Interface).

Clients dan Tools

MySQL dilengkapi dengan berbagai peralatan (tool) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk online.

Struktur tabel

MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan basis data lainnya semacam PostgreSQL ataupun Oracle.

Sejak diakuisisinya MySQL oleh Oracle pada September 2010, Monty Program sebagai penulis awal kode sumber MySQL memisahkan diri dari pengembangan dan membuat versi yang lebih mandiri yakni MariaDB. MariaDB merupakan versi pengembangan terbuka dan mandiri dari MySQL.

Mengelola Database melalui PhpMyAdmin



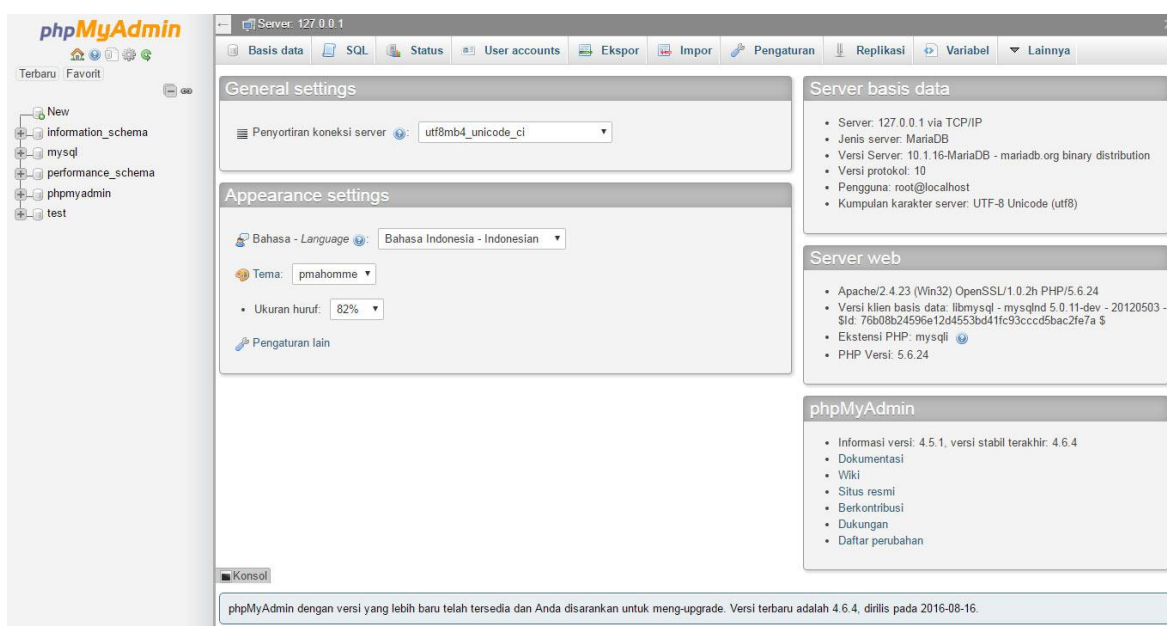
PHP dapat bekerja dengan berbagai macam software database, seperti MS. Access, SQL Server, Oracle dan lainnya. Namun, pada kesempatan kali ini Saya akan coba

membahas software database bawaan XAMPP yaitu MariaDB. Seperti dijelaskan pada artikel sebelumnya, bahwa XAMPP saat ini tidak lagi menyertakan MySQL sebagai default instalasi database. MySQL digantikan oleh MariaDB. Untuk menjaga kompatibilitas, pihak pembuat tidak banyak mengganti environment MySQL. Jadi Anda tidak perlu khawatir dengan banyaknya perubahan, karena semua connector, library dan aplikasi yang bekerja pada MySQL, dapat bekerja pada MariaDB.

Pada dasarnya, mengelola basis data dengan MariaDB/MySQL harus dilakukan dengan cara mengetikkan baris-baris perintah yang sesuai (command line) untuk setiap maksud tertentu. Jika seseorang ingin membuat basis data (database), ketikkan baris perintah yang sesuai untuk membuat basis data. Jika seseorang menghapus tabel, ketikkan baris perintah yang sesuai untuk menghapus tabel. Hal tersebut tentu saja sangat menyulitkan karena seseorang harus hafal dan mengetikkan perintahnya satu per satu.

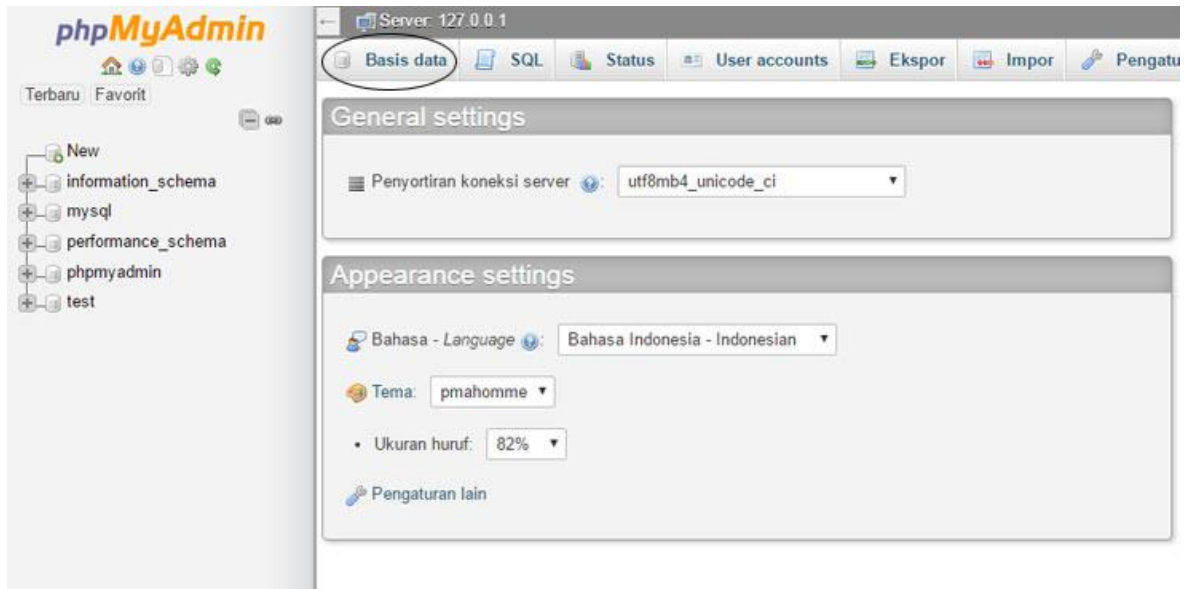
Saat ini banyak sekali perangkat lunak yang dapat dimanfaatkan untuk mengelola basis data dalam MySQL, salah satunya adalah phpMyAdmin. Dengan phpMyAdmin, seseorang dapat membuat database, membuat tabel, mengisi data, dan lain-lain dengan mudah, tanpa harus menghafal baris perintahnya.

Untuk membuka phpMyadmin, buka browser favorit Anda, lalu ketikkan alamat **<http://localhost/phpmyadmin>**, maka akan muncul halaman phpMyAdmin seperti berikut.

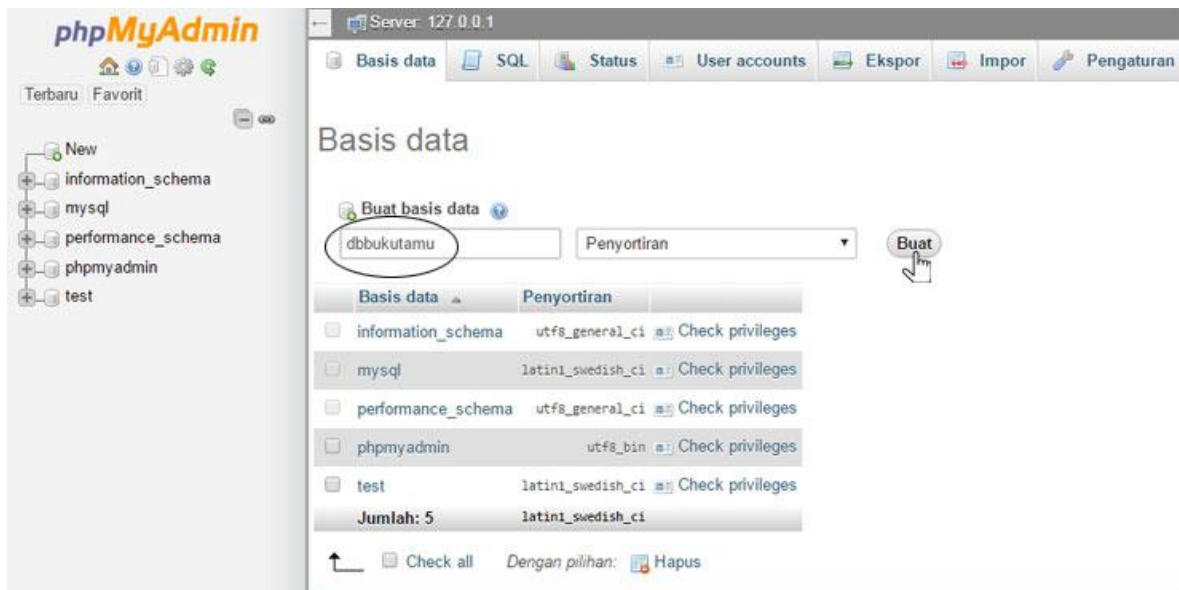


Membuat Database

Untuk membuat database, klik Basis data di halaman utama phpMyAdmin seperti gambar berikut :



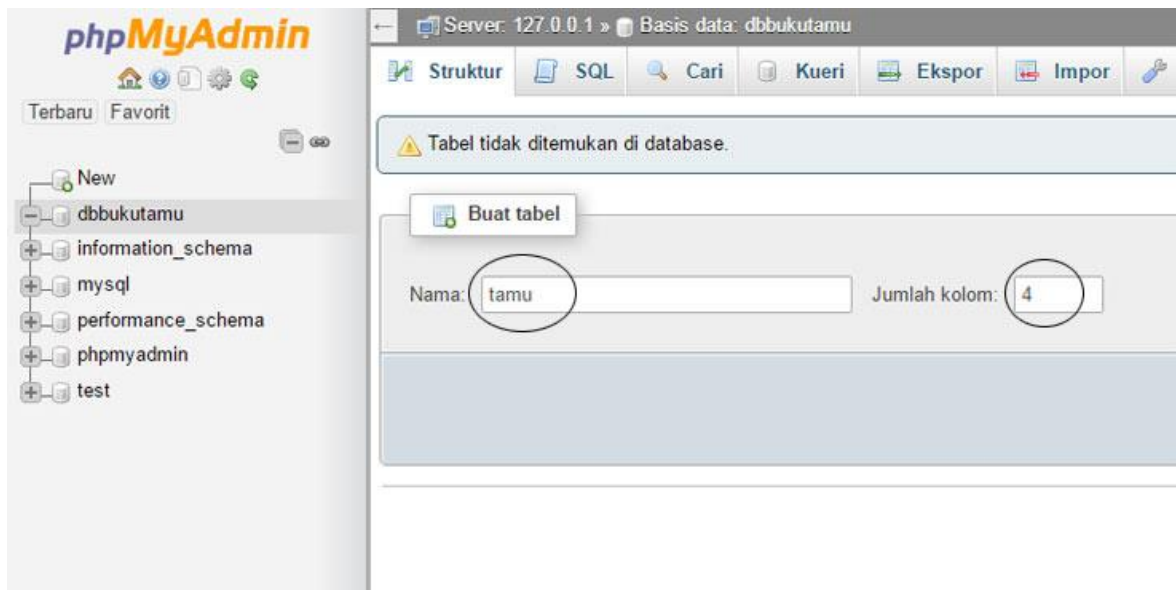
Silahkan Anda isi nama database sesuai keinginan Anda pada bagian Buat basis data, misalnya pada tutorial kali ini saya berikan contoh dbbukutamu, lalu klik tombol Buat



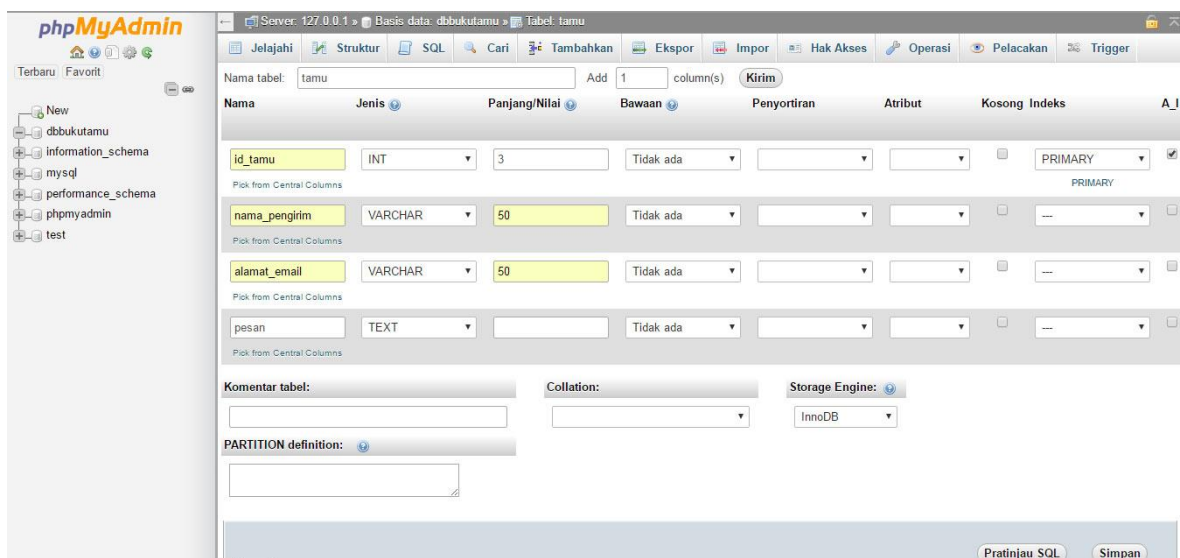
Maka akan terbentuk sebuah database baru dengan nama dbbukutamu.

Membuat Table dalam Database

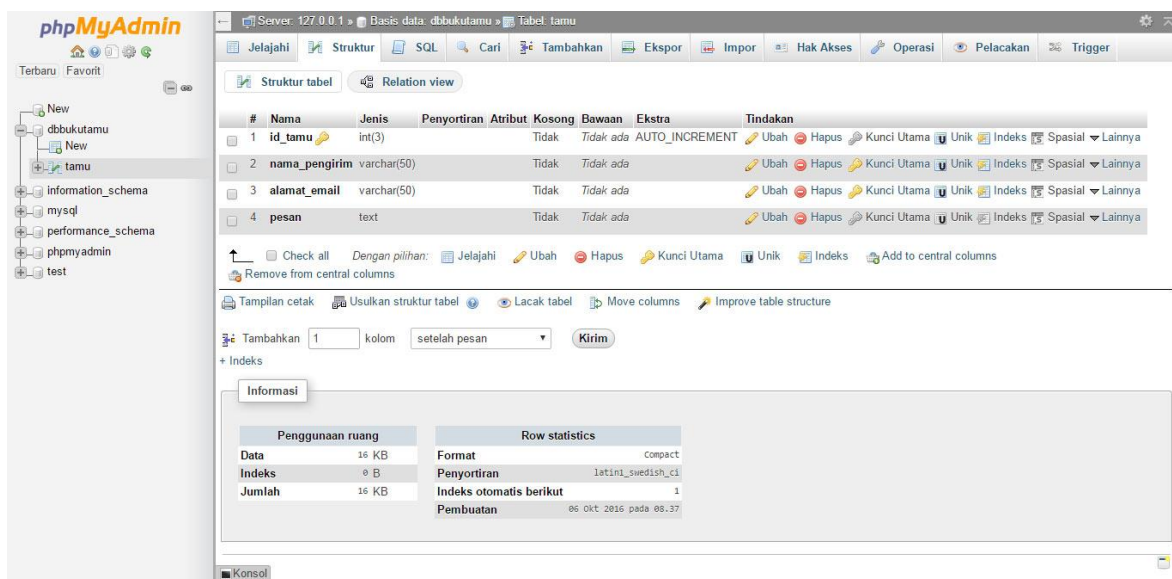
Untuk membuat tabel di dalam database dbbukutamu, silahkan klik nama databasenya, yaitu dbbukutamu, maka akan tampil halaman pembuatan tabel, sebagai contoh saya isikan Nama : tamu dan Jumlah kolom : 4



Silahkan isi Name masing-masing field, Type dan Length. Khusus field kunci, dalam hal ini id_tamu, silahkan pilih Index : PRIMARY dan ceklis pada bagian A_I (Auto Increment). Lalu klik tombol Simpan.



Jika berhasil disimpan, maka akan terbentuk sebuah table baru bernama tamu di database dbbukutamu.



TYPE DATA

Berikut adalah tipe-tipe data yang sering digunakan di dalam sebuah tabel :

- **INT**, data berupa angka.
- **VARCHAR**, data berupa karakter.
- **DATE**, data berupa tanggal.
- **TIME**, data berupa waktu.
- **TEXT**, data berupa karakter dalam jumlah yang banyak.
- **ENUM**, data yang sudah ditentukan isinya, misalnya Y (Yes) atau N (No) saja.

BAB 13

STUDI KASUS (LATIHAN)

Contoh Fungsi Siap Pakai (Konversi tanggal)

konversi_tanggal.php

```
<?php
// format penanggalan di database MySQL
$tanggal=date("Y-m-d");

// ambil atau pisahkan tanggal, bulan dan tahun
$ambiltahun = substr($tanggal,0,4);
$ambilbulan = substr($tanggal,5,2);
$ambiltanggal = substr($tanggal,8,2);

// ubah angka bulan menjadi nama bulan
if ($ambilbulan=="01") $namabulan="Januari";
elseif ($ambilbulan=="02") $namabulan="Februari";
elseif ($ambilbulan=="03") $namabulan="Maret";
elseif ($ambilbulan=="04") $namabulan="April";
elseif ($ambilbulan=="05") $namabulan="Mei";
elseif ($ambilbulan=="06") $namabulan="Juni";
elseif ($ambilbulan=="07") $namabulan="Juli";
elseif ($ambilbulan=="08") $namabulan="Agustus";
elseif ($ambilbulan=="09") $namabulan="September";
elseif ($ambilbulan=="10") $namabulan="Oktober";
elseif ($ambilbulan=="11") $namabulan="November";
elseif ($ambilbulan=="12") $namabulan="Desember";

echo "Format Tanggal Inggris: <b>$tanggal</b> <br><br>";

echo "Format Tanggal Indonesia: <b>$ambiltanggal-$namabulan-$ambiltahun</b>";
?>
```

Function str_replace

```
<?php
$stringawal = "SAYA SANGAT LAPAR SEKALI";
$stringakhir = str_replace("SA","GI",$stringawal);
echo "$stringawal > $stringakhir";
?>
```

Memformat Bilangan dengan Function number_format()

```
<?php
$number = 123456789.12345;

// menampilkan 123,456,789
$bil = number_format($number);
echo "$bil <br>";

// menampilkan 123,456,789.12
$bil = number_format($number,2);
echo "$bil <br>";
```

```
// menampilkan 123.456.789,12
$bil = number_format($number,2,"",".");
echo "$bil <br>";

// menampilkan 123#456#789-12
$bil = number_format($number,2,"-","#");
echo "$bil <br>";

// menampilkan Rp. 123.456.789,-
$bil = number_format($number,0,"",".");
echo "Rp $bil,-";

?>
```

Mengambil Substring dari Suatu String dengan Function substr()

```
<?php
$data = "31200801211455000100093408000002";
$kodeBagian = substr($data,0,2);
$tahun = substr($data,2,4);
$bulan = substr($data,6,2);
$tanggal = substr($data,8,2);
$jam = substr($data,10,2);
$menit = substr($data,12,2);
$keterangan = substr($data,14,4);
$kodeKary = substr($data,18,10);
$kodePresensi = substr($data,28,4);

echo "Kode Bagian : $kodeBagian <br>";
echo "Tahun Presensi : $tahun <br>";
echo "Bulan Presensi : $bulan <br>";
echo "Tanggal Presensi : $tanggal <br>";
echo "Jam Presensi : $jam <br>";
echo "Menit Presensi : $menit <br>";
echo "Keterangan Presensi : $keterangan <br>";
echo "Kode Karyawan : $kodeKary <br>";
echo "Kode Mesin : $kodePresensi <br>";

?>
```

Random Quotes dengan PHP

Buat database dan table dengan struktur di bawah ini :
nama table : tabelquotes

#	Name	Type	Collation	Attributes
<input type="checkbox"/> 1	<u>id</u>	int(11)		
<input type="checkbox"/> 2	quote	text	latin1_swedish_ci	

Isi data tabel

id	quote
1	Hari ini harus lebih baik dari kemarin
2	Jangan tunda sampai besok apa yang bisa kalian kerjakan hari ini
3	Ingin investasi dunia akhirat? Sharinglah ilmu yang Anda miliki dengan yang lain
4	Ilmu lebih baik dari harta. Apabila Anda memiliki harta, maka Andalah penjaganya. Namun jika Anda memiliki ilmu, maka ilmulah yang akan menjaga Anda.

```
<?php
    $koneksi = mysqli_connect("localhost", "root", "", "dbfundamental");

    $query = "SELECT * FROM tabelquotes ORDER BY RAND()";
    $hasil = mysqli_query($koneksi, $query);
    $data = mysqli_fetch_array($hasil);

    echo "Pesan saat ini : <strong>$data[quote]</strong>";
?>
```

Membuat Autentifikasi User di PHP

Buat tabel user

#	Name	Type
1	username	varchar(35)
2	password	varchar(100)

Membuat form register user (register.php)

```
<form method="post" action="submit.php">
    <table>
        <tr>
            <td>Masukkan Username </td>
            <td><input name="username" type="text"></td>
        </tr>
        <tr>
            <td>Masukkan Password</td>
            <td><input name="pass1" type="password"></td>
        </tr>
        <tr>
            <td>Ulangi Password</td>
            <td><input name="pass2" type="password"></td>
        </tr>
        <tr>
            <td>&nbsp;</td>
            <td><input type="submit" value="Submit"></td>
        </tr>
    </table>
</form>
```

submit.php

```
<?php
$username= $_POST['username'];
$password1 = $_POST['pass1'];
$password2 = $_POST['pass2'];

// cek kesamaan password1
if($password1 == $password2) {
    $koneksi = mysqli_connect("localhost", "root", "", "dbfundamental");

    // perlu dibuat sembarang pengacak
    $pengacak = "NDJS3289JSKS190JISJI";

    // mengenkripsi password dengan md5() dan pengacak
    $password1 = md5($pengacak. md5($password1) . $pengacak);

    // menyimpan username dan password terenkripsi ke database
```

```

$query = "INSERT INTO user VALUES('$username','$password1')";
$hasil = mysqli_query($koneksi,$query);

// menampilkan keterangan pendaftaran
if ($hasil){
    echo "Username sudah berhasil terdaftar";
}
else {
    echo "Username sudah terdaftar";
}
}
else {
    echo "Password yang dimasukkan tidak sama";
}
}
?>

```

login.php

```

<form method="post" action="loginsubmit.php">
<table>
<tr>
<td>Masukkan Username </td>
<td><input name="username" type="text"></td>
</tr>
<tr>
<td>Masukkan Password </td>
<td><input name="pass" type="password"></td>
</tr>
<tr>
<td>&nbsp;</td>
<td><input type="submit" name="submit" value="Submit"></td>
</tr>
</table>
</form>

```

loginsubmit.php

```

<?php
// menjalankan session
session_start();

$username = $_POST['username'];
$password = $_POST['pass'];

$koneksi = mysqli_connect("localhost","root","","dbfundamental");

// mencari password terenkripsi berdasarkan username
$query = "SELECT * FROM user WHERE username = '$username'";
$hasil = mysqli_query($koneksi,$query);
$data = mysqli_fetch_array($hasil);

$pengacak = "NDJS3289JSKS190JISJI";

/* cek kesesuaian password terenkripsi dari form login
dengan password terenkripsi dari database */
if (md5($pengacak.md5($password).$pengacak) == $data['password']){
    // jika sesuai, maka buat session untuk username
    $_SESSION['username'] = $username;

    // menampilkan menu ke halaman yang akan akses
    echo "<h2>Login Sukses</h2>";
    echo "<p><a href='#>Menu 1</a> | <a href='#>Menu 2</a></p>";
}
else {

```

```

        echo "<h2>Login Gagal</h2>";
    }
?>

```

Pemrosesan Form dengan Komponen Combo Box

buat table : **countries**

#	Name	Type
<input type="checkbox"/> 1	<u>id</u>	varchar(2)
<input type="checkbox"/> 2	<u>namanegara</u>	varchar(35)

Isikan data sebagai berikut :

	id	namanegara
<input type="checkbox"/> Edit Copy Delete	AU	Australia
<input type="checkbox"/> Edit Copy Delete	CA	Canada
<input type="checkbox"/> Edit Copy Delete	ID	Indonesia
<input type="checkbox"/> Edit Copy Delete	MY	Malaysia
<input type="checkbox"/> Edit Copy Delete	US	Amerika Serikat

form.php

```

<?php
    $konek=mysqli_connect("localhost","root","","dbfundamental");

    echo "
    <form method='post' action='submit.php'>
        Pilih salah satu negara :
        <select name='negara'>";
    $query = "SELECT * FROM countries";
    $hasil = mysqli_query($konek,$query);
    while($data=mysqli_fetch_array($hasil)){
        echo "<option value='$data[id] '$data[namanegara]</option>";
    }
    echo "</select>
    <input type='submit' value='Submit'>
    </form>";
?>

```

Membuat Checkbox Dinamis dan Pemrosesannya

Buat table :

struktur table mk

#	Name	Type
<input type="checkbox"/> 1	<u>kodeMK</u>	varchar(10)
<input type="checkbox"/> 2	<u>namaMK</u>	varchar(35)

Isi table mk:

+ Options				kodeMK	namaMK
<input type="checkbox"/>				M001	Algoritma
<input type="checkbox"/>				M002	Pengantar Basis Data
<input type="checkbox"/>				M003	Sistem Pakar

struktur table ambilmk

#	Name	Type	Collation	Attrib
<input type="checkbox"/> 1	nim	varchar(10)	latin1_swedish_ci	
<input type="checkbox"/> 2	kodeMK	varchar(10)	latin1_swedish_ci	
<input type="checkbox"/> 3	nilai	float		

mk.php

```
<?php
$konek = mysqli_connect("localhost","root","","dbfundamental");
?>

<h1>Form Pengambilan Matakuliah</h1>
<form method="post" action="ambil.php">
Masukkan NIM <input type="text" name="nim"><br>
Daftar Matakuliah <br>
<?php
$query = "SELECT * FROM mk";
$hasil = mysqli_query($konek,$query);
$no = 1;
while($data = mysqli_fetch_array($hasil)){
    echo "<input type='checkbox' value='".$data[kodeMK]'"
name='mk$no'>$data[namaMK] <br>";
    $no++;
}
?>
<input type="hidden" name="jumMK" value="<?php echo $no-1; ?>">
<input type="submit" value="Submit">
</form>
```

ambil.php

```
<?php
$konek = mysqli_connect("localhost","root","","dbfundamental");

$nim = $_POST['nim'];
$jumMK = $_POST['jumMK'];

for($i = 1; $i < $jumMK; $i++)
{
    $mk = $_POST['mk'.$i];
    if(!empty($mk))
    {
        $query = "INSERT INTO ambilmk VALUES ('$nim','$mk',0)";
        mysqli_query($konek,$query);
    }
}
echo "Terimakasih sudah memilih matakuliah";
?>
```

Latihan INSERT, UPDATE, DELETE, SELECT

form_tamu.php

```
<html>
<title>Buku Tamu</title>
<body>
<h3>Form Buku Tamu</h3>
<form method="GET" action="input_tamu.php">
<table>
<tr><td>Nama</td><td>: <input type="text" name="nama"></td></tr>
<tr><td>E-mail</td><td>: <input type="text" name="email"></td></tr>
<tr><td>Pesan</td><td>: <textarea name="pesan" rows="5"
cols="30"></textarea></td></tr>
<tr><td></td><td><input type="submit" value="Kirim"></td></tr>
</table>
</form>
```

input_tamu.php

```
<?php
// koneksi ke database
$konek = mysqli_connect("localhost","root","","latihan");

// ambil variabel yang dikirim dari form
$nama = $_GET['nama'];
$email = $_GET['email'];
$pesan = $_GET['pesan'];

$input = "INSERT INTO tamu(nama,email,pesan)
VALUES('$nama','$email','$pesan')";
$hasil = mysqli_query($konek, $input);

// apabila query untuk menginput data benar
if ($hasil){
    // lakukan redirect
    header("location:tampil_tamu_admin.php");
}
else{
    echo "Input Data Tamu Gagal";
}
?>
```

tampil_tamu.php

```
<h3>Data Tamu</h3>

<?php
$konek = mysqli_connect("localhost","root","","latihan");

$tampil = "SELECT * FROM tamu ORDER BY id_tamu";
$hasil = mysqli_query($konek, $tampil);

echo "<table>
<tr>
<th>Nama</th>
<th>E-mail</th>
<th>Pesan</th>
</tr>";

while ($data=mysqli_fetch_array($hasil)){
    echo "<tr>
<td>$data[nama]</td>
<td>$data[email]</td>
```

```

        <td>$data[pesan]</td>
    </tr>";
}
echo "</table>";
?>

```

tampil_tamu_admin.php

```

<h3>Data Tamu</h3>
<p><a href="form_tamu.php">Input Tamu</a></p>

<?php
$konek = mysqli_connect("localhost","root","","latihan");

$stampil = "SELECT * FROM tamu ORDER BY id_tamu";
$hasil = mysqli_query($konek, $stampil);
$total = mysqli_num_rows($hasil);

echo "<table>
    <tr>
        <th>No</th>
        <th>Nama</th>
        <th>E-mail</th>
        <th>Pesan</th>
        <th>Aksi</th>
    </tr>";

// nomor awal untuk pengurutan
$no = 1;

// tampilkan data tamu
while ($data=mysqli_fetch_array($hasil)){
    echo "<tr>
        <td>$no</td>
        <td>$data[nama]</td>
        <td>$data[email]</td>
        <td>$data[pesan]</td>
        <td><a href='edit_tamu.php?id=$data[id_tamu]'">Edit</a> |
        <a href='hapus_tamu.php?id=$data[id_tamu]'">Hapus</a></td>
    </tr>";

    // tambahkan $no dengan 1
    $no++;
}

echo "</table>";

echo "<p>Jumlah Tamu: <b>$total</b> Orang</p>";
?>

```

edit_tamu.php

```

<h3>Edit Buku Tamu</h3>

<?php
$konek = mysqli_connect("localhost","root","","latihan");

// ambil id dari hasil klik link Edit
$id = $_GET['id'];

$edit = "SELECT * FROM tamu WHERE id_tamu='$id'";
$hasil = mysqli_query($konek, $edit);

```

```

$data = mysqli_fetch_array($hasil);

echo "<form method='GET' action='update_tamu.php'>
      <input type='hidden' name='id' VALUE='$id'>
      <table>
        <tr><td>Nama</td><td>: <input type='text' name='nama'
value='$data[nama]'></td></tr>
        <tr><td>E-mail</td><td>: <input type='text' name='email'
value='$data[email]'></td></tr>
        <tr><td>Pesan</td><td>: <textarea name='pesan' rows='5'
cols='30'>$data[pesan]</textarea></td></tr>
        <tr><td></td><td><input type='submit' value='Edit'></td></tr>
      </table>
    </form>";
?>

```

update_tamu.php

```

<?php
$konek = mysqli_connect("localhost","root","","latihan");

// ambil variabel yang dikirim dari form
$id    = $_GET['id'];
$nama  = $_GET['nama'];
$email = $_GET['email'];
$pesan = $_GET['pesan'];

$update = "UPDATE tamu SET nama='$nama', email='$email', pesan='$pesan'
WHERE id_tamu='$id'";
$hasil = mysqli_query($konek, $update);

// apabila query untuk mengupdate data benar
if ($hasil){
    // lakukan redirect
    header("location:tampil_tamu_admin.php");
}
else{
    echo "Update Data Tamu Gagal";
}
?>

```

hapus_tamu.php

```

<?php
$konek = mysqli_connect("localhost","root","","latihan");

// ambil id dari hasil klik link Hapus
$id    = $_GET['id'];

$hapus = "DELETE FROM tamu WHERE id_tamu='$id'";
$hasil = mysqli_query($konek, $hapus);

// apabila query untuk menghapus data benar
if ($hasil){
    // lakukan redirect
    header("location:tampil_tamu_admin.php");
}
else{
    echo "Hapus Data Tamu Gagal";
}
?>

```

Latihan Paging

paging.php

```
<?php
$konek = mysqli_connect("localhost","root","","latihan");

// Langkah 1. Tentukan batas,cek halaman & posisi data
$batas = 5;
$halaman = @$_GET['halaman'];
if(empty($halaman)){
    $posisi = 0;
    $halaman = 1;
}
else{
    $posisi = ($halaman-1) * $batas;
}

// Langkah 2. Sesuaikan query dengan posisi dan batas
$query = "SELECT * FROM anggota LIMIT $posisi,$batas";
$stampil = mysqli_query($konek, $query);

echo "<table>
      <tr><th>No</th><th>Nama</th><th>Alamat</th></tr>";

$no = $posisi+1;
while ($data=mysqli_fetch_array($stampil)){
    echo "<tr>
          <td>$no</td>
          <td>$data[nama]</td>
          <td>$data[alamat]</td>
        </tr>";
    $no++;
}
echo "</table>";

// Langkah 3: Hitung total data dan halaman serta link 1,2,3
$query2 = mysqli_query($konek, "select * from anggota");
$jmldata = mysqli_num_rows($query2);
$jmlhalaman = ceil($jmldata/$batas);

echo "<br> Halaman : ";

for($i=1;$i<=$jmlhalaman;$i++){
    if ($i != $halaman){
        echo " <a href='paging.php?halaman=$i'>$i</a> | ";
    }
    else{
        echo " <b>$i</b> | ";
    }
}
echo "<p>Total anggota : <b>$jmldata</b> orang</p>";
?>
```

Latihan Paging Ala Google

class_paging.php

```
<?php
class Paging{
    // Fungsi untuk mencek halaman dan posisi data
    function cariPosisi($batas){
        if(empty($_GET['halaman'])){
```

```

        $posisi = 0;
        $_GET['halaman'] = 1;
    }
    else{
        $posisi = ($_GET['halaman']-1) * $batas;
    }
    return $posisi;
}

// Fungsi untuk menghitung total halaman
function jumlahHalaman($jmldata, $batas){
    $jmlhalaman = ceil($jmldata/$batas);
    return $jmlhalaman;
}

// Fungsi untuk link halaman 1,2,3 ... Next, Prev, First, Last
function navHalaman($halaman_aktif, $jmlhalaman){
    $file = $_SERVER['PHP_SELF'];

    $link_halaman = "";

    // Link First dan Previous
    if ($halaman_aktif > 1){
        $prev = $halaman_aktif-1;
        $link_halaman .= " <a href='$file?halaman=1'>First</a> | 
                           <a href='$file?halaman=$prev'>Prev</a> | ";
    }
    else{
        $link_halaman .= " First | Prev | ";
    }

    // Link halaman 1,2,3, ...
    // Angka awal
    $angka = ($halaman_aktif > 3 ? " ... " : " "); // Ternary operator
    for ($i=$halaman_aktif-2; $i<$halaman_aktif; $i++){
        if ($i < 1) continue;
        $angka .= "<a href='$file?halaman=$i'>$i</a> | ";
    }

    // Angka tengah
    $angka .= " <b>$halaman_aktif</b> | ";
    for($i=$halaman_aktif+1; $i<($halaman_aktif+3); $i++){
        if($i > $jmlhalaman) break;
        $angka .= "<a href='$file?halaman=$i'>$i</a> | ";
    }

    // Angka akhir
    $angka .= ($halaman_aktif+2<$jmlhalaman ? " ... | <a 
href=\"'$file?halaman=$jmlhalaman'>$jmlhalaman</a> | " : " ");

    $link_halaman .= "$angka";

    // Link Next dan Last
    if ($halaman_aktif < $jmlhalaman){
        $next=$halaman_aktif+1;
        $link_halaman .= " <a href='$file?halaman=$next'>Next</a> | 
                           <a href='$file?halaman=$jmlhalaman'>Last</a> ";
    }
    else{
        $link_halaman .= " Next | Last ";
    }
    return $link_halaman;
}

```

```
}  
?>
```

use_paging.php

```
<?php  
$koneksi = mysqli_connect("localhost", "root", "", "cerdas");  
  
// Memanggil dan menginisiasi class  
include "class_paging.php";  
  
$p = new Paging;  
  
// Tentukan limit atau batas  
$batas = 5;  
  
// Cek halaman dan posisi data  
$posisi = $p->cariPosisi($batas);  
  
// Sesuaikan query dengan posisi dan batas  
$query = "SELECT * FROM anggota LIMIT $posisi,$batas";  
$stampil = mysqli_query($koneksi, $query);  
  
echo "<table>  
    <tr><th>No</th><th>Nama</th><th>Alamat</th></tr>";  
  
$no = $posisi+1;  
while ($data=mysqli_fetch_array($stampil)) {  
    echo "<tr>  
        <td>$no</td>  
        <td>$data[nama]</td>  
        <td>$data[alamat]</td>  
    </tr>";  
    $no++;  
}  
echo "</table><br>";  
  
// Dapatkan jumlah data keseluruhan  
$query2 = mysqli_query($koneksi, "select * from anggota");  
$jmlldata = mysqli_num_rows($query2);  
  
// Dapatkan jumlah halaman  
$jmlhalaman = $p->jumlahHalaman($jmlldata, $batas);  
  
// Cetak link navigasi halaman  
$linkHalaman = $p->navHalaman($_GET['halaman'], $jmlhalaman);  
echo $linkHalaman;  
?>
```

Teknik Upload File

Buat table dengan struktur seperti berikut :

Field	Type	Length	Index	Auto Increment
Id_upload	INT	3	PRIMARY	<input checked="" type="checkbox"/>
Nama_file	VARCHAR	100		
Deskripsi	TEXT			
Tgl_upload	DATE			

form_upload.php

```
<form enctype="multipart/form-data" method="POST"
action="hasil_upload.php">
File yang di upload : <input type="file" name="fupload"><br>
Deskripsi File : <br>
<textarea name="deskripsi" rows="8" cols="40"></textarea><br>
<input type="submit" value="Upload">
</form>
```

hasil_upload.php

```
<?php
// Baca lokasi file sementara dan nama file dari form (fupload)
$lokasi_file = $_FILES['fupload']['tmp_name'];
$nama_file   = $_FILES['fupload']['name'];

// Tentukan folder untuk menyimpan file
$folder = "files/$nama_file";

// tanggal sekarang
$ttl_upload = date("Ymd");

// Apabila file berhasil di upload
if (move_uploaded_file($lokasi_file,$folder)){
    echo "Nama File : <b>$nama_file</b> sukses di upload";

    // Masukkan informasi file ke database
    $koneksi = mysqli_connect("localhost","root","","latihan");

    $query = "INSERT INTO upload (nama_file, deskripsi, ttl_upload)
VALUES('$nama_file', '$_POST[deskripsi]', '$ttl_upload')";

    mysqli_query($koneksi, $query);
}
else{
    echo "File gagal di upload";
}
?>
```

TEKNIK THUMBNAIL

fungsi_thumbnail.php

```
<?php
function UploadImage($fupload_name){
    // Tentukan folder dan file gambar yang di upload
    $folder = "files/";

    $file_upload = $folder . $fupload_name;

    // Simpan gambar dalam ukuran aslinya
    move_uploaded_file($_FILES["fupload"]["tmp_name"], $file_upload);

    // Identitas file asli
    $gbr_asli = imagecreatefromjpeg($file_upload);
    $lebar    = imageSX($gbr_asli);
    $tinggi   = imageSY($gbr_asli);

    // Simpan dalam versi thumbnail (110 pixel)
    $thumb_lebar = 110;
    $thumb_tinggi = ($thumb_lebar/$lebar) * $tinggi;
```



```

// Proses perubahan dimensi ukuran
$gbr_thumb = imagecreatetruecolor($thumb_lebar,$thumb_tinggi);
imagecopyresampled($gbr_thumb, $gbr_asli, 0, 0, 0, 0, $thumb_lebar,
$thumb_tinggi, $lebar, $tinggi);

// Simpan gambar thumbnail
imagejpeg($gbr_thumb,$folder . "thumb_" . $fupload_name);

// Hapus gambar di memori komputer
imagedestroy($gbr_asli);
imagedestroy($gbr_thumb);
}
?>

```

hasil_thumbnail.php

```

<?php
include "fungsi_thumbnail.php";

$lokasi_file    = $_FILES['fupload']['tmp_name'];
$ tipe_file     = $_FILES['fupload']['type'];
$nama_file      = $_FILES['fupload']['name'];

// Membuat nama file yang unik
$acak          = rand(1,99);
$nama_file_unik = $acak.$nama_file;

if ($tipe_file!="image/jpeg" AND $tipe_file!="image/pjpeg"){
    echo "Upload Gagal, file yang di upload harus bertipe JPG";
}
else{
    // Gunakan fungsi thumbnail gambar untuk upload gambar
    UploadImage($nama_file_unik);

    // Masukkan informasi file ke database
    $koneksi = mysqli_connect("localhost","root","","latihan");
    $query = "INSERT INTO upload (nama file) VALUES('$nama_file_unik')";
    mysqli_query($koneksi, $query);

    // Tampilkan gambar asli di browser
    echo "<img src='http://localhost/files/$nama_file_unik'>";

    // Tampilkan gambar thumbnail di browser
    echo "<br><br>";
    echo "<img src='http://localhost/files/thumb_$nama_file_unik'>";
}
?>

```