

Bab 1

Sekilas Tentang Teknologi Internet

Sekarang siapa sih yang tidak kenal dengan Facebook, Friendster, Kaskus, Detikcom atau website yang lainnya? Ya situs-situs terkenal tersebut semuanya sudah menjadi bagian yang tidak lepas dari keseharian kehidupan kita. Situs-situs tersebut berada di alam maya (virtual) yang bernama internet.

Internet atau disebut dengan *world wide web* (www) merupakan kumpulan jaringan komputer yang saling berkomunikasi dengan aturan komunikasi tertentu. Suatu situs web (website) di internet umumnya berupa sebuah komputer/server yang dilengkapi dengan perangkat lunak yang bernama *webserver*, sehingga komputer ini dapat dapat diakses dari seluruh dunia. Informasi dari *webserver* tersebut diakses melalui protokol standar yang bernama

http (hyper text transfer protocol).

Format dokumen standar di internet saat ini adalah HTML (Hypertext Markup Language). HTML menentukan bagaimana informasi ditampilkan sebagai halaman web pada browser

Pemrograman Internet

Pemrograman di Internet tidak sama seperti pemrograman pada komputer lokal (misal pada Visual Basic). Pemrograman di Internet berkaitan dengan format dokument yang digunakan di Internet. Secara umum ada dua macam pemrograman di internet, yaitu : *Client Side Programming* dan *Server Side Programming*.

Client Side Programming

Pada pemrograman jenis ini program didownload oleh pengunjung dan dijalankan pada komputer pengunjung. Pemrograman jenis ini memerlukan dukungan browser pengunjung. Selain itu ukuran program harus diperhatikan karena jika program terlalu besar, maka waktu yang diperlukan untuk membuka halaman web menjadi lebih lama. Kerugian lainnya adalah pengunjung dapat dengan mudah melihat dan menyalin *source code* program. Contoh program client side adalah : HTML, CSS, JavaScript dan VBScript.

Server Side Programming

Sesuai dengan namanya, jenis pemrograman ini

program dijalankan di dalam webserver. Hasil pengolahannya dikirimkan pada browser pengunjung sebagai halaman HTML. Ukuran program relatif dapat lebih besar karena pengunjung tidak perlu mendownload programnya, tetapi hanya hasil pengolahannya. Keuntungan lainnya yakni *Source Code* tidak dapat dilihat. Contoh pemrograman Server Side diantaranya : Perl, PHP, ASP, dan JSP.

Mengenal PHP

PHP dikenal sebagai sebuah bahasa scripting yang menyatu dengan tag-tag HTML yang dieksekusi di server dan digunakan untuk membuat halaman web yang dinamis seperti guestbook, statistik pengunjung, polling, email, upload foto, blog dan masih banyak lagi.

Versi pertama PHP dibuat oleh Rasmus Lerdorf pada tahun 1994 dan sampai saat ini pengembangan versinya telah mencapai PHP versi 5.0. Berdasarkan hasil survei di website resmi PHP (www.php.net/usage.php), pada Mei 2002, di seluruh dunia PHP telah dipakai oleh lebih dari 9 juta domain dan pada Juli 2007 terjadi peningkatan signifikan menjadi lebih dari 21 juta domain. Jumlah ini akan selalu meningkat dengan cepat yang menunjukkan PHP telah semakin populer digunakan di dunia. PHP versi pertama diberi nama *Personal Homepage* dan entah kapan namanya berubah menjadi *PHP Hypertext Preprocessor*.

Beberapa situs terkenal seperti Detik.Com, Kompas.Com, Friendster.Com dan Facebook.Com menggu-

nakan komponen bahasa PHP dalam pembuatan situsnya.

Mengapa PHP

- PHP bersifat *open source* dan gratis. Efek dari program open source ini sangat terasa sekarang yakni teknologi internet semakin canggih, terjangkau alias murah, mudah dipelajari dan aman sehingga dapat menggerakkan ekonomi dunia.
- *Life Cycle* yang sangat singkat , sehingga PHP selalu *up to date* mengikuti perkembangan teknologi internet.
- *Cross Platform*, yakni PHP dapat dipakai di hampir semua webserver yang ada di pasaran (terutama Apache dan Microsoft IIS) dan dijalankan pada berbagai sistem operasi (Linux, Windows, FreeBSD).
- PHP mendukung koneksi ke banyak database baik yang gratis maupun komersil, seperti mySQL, PostgreSQL, Oracle, Microsoft SQL Server, Interbase, dan banyak lagi.
- Kemudahan dalam mendapatkan dokumentasi. Di Internet, Kita tidak akan sulit untuk mencari baik itu referensi, resource kode-kode PHP yang sudah jadi dan juga mengajukan pertanyaan pada grup-grup diskusi yang di dalamnya banyak sekali para master PHP.
- Banyaknya aplikasi berbasis web yang dibuat dengan PHP dan kita tinggal pakai, modifikasi dan kembangkan. Misalnya:

- Aplikasi untuk Blog : Wordpress, Texpattern
- Aplikasi untuk Content Management Service (CMS) : Joomla, Drupal, Mambo
- Aplikasi untuk Shopping Cart : ZendCart, VirtueMart
- Dan masih banyak lagi.

Webdatabase

Webdatabase adalah database yang dapat diakses melalui internet dengan menggunakan pemrograman Web. Webdatabase merupakan suatu keharusan untuk menampung informasi yang semakin besar dan kompleks. MySQL merupakan salah satu **webdatabse** yang sering digunakan dalam pemrograman PHP.

Instalasi dan Konfigurasi

Sebelum perang dimulai, beberapa peralatan yang harus dipersiapkan diantaranya :

1. Webserver: bisa dipilih apakah : Apache atau Microsoft IIS
2. Source Program PHP
3. Database MySQL

Kita tidak perlu bersusah payah untuk instal dan konfigurasi secara manual ketiga komponen di atas karena sekarang sudah tersedia aplikasi integrated yang menggabungkan ketiga unsur di atas yakni PHP, Webserver Apache dan Database MySQL menjadi satu set aplikasi

yang secara otomatis bisa dideploy. Nama programnya adalah XAMPP. Bisa didownload di <http://www.apache-friends.org>. Terserah kita mau pilih versi yang mana apakah versi Windows ataupun Linux. Adapun dalam buku ini kita pilih XAMPP versi Windows. Paket XAMPP ini terdiri dari

- Webserver : Apache HTTPD + Openssl
- Database MySQL
- Engine PHP 5
- phpMyAdmin : Aplikasi berbasis web untuk administrasi database
- Filezilla FTP Server
- Mercury mail transport : Aplikasi untuk server email / SMTP server

deutsch · english · français · italiano · português (brasil) · 日本語 · 繁體中文 · 简体中文

[Home](#) [XAMPP](#) [News](#) [Team](#) [Projects](#) [Docs](#) [Extras](#) [Links](#) [Support](#)

Database/SQL Tool
For DB2, SQL Server, Derby, Mimer Informix, Oracle and more
www.dbvis.com

Ads by Google

XAMPP for Windows

We've released the new version of XAMPP 1.7.3,
including:

- Apache 2.2.14 (IPv6 enabled) + OpenSSL 0.9.8l
- MySQL 5.1.41 + PBXT engine
- PHP 5.3.1
- phpMyAdmin 3.2.4
- Perl 5.10.1
- FileZilla FTP Server 0.9.33
- Mercury Mail Transport System 4.72

Please visit the [win32 compatibility list](#).

For all "newcomers": You only need the XAMPP basic package, "Add-On" packages are optional.

Download older versions of XAMPP (even the "old" WAMPP)
directly from [SourceForge](#).

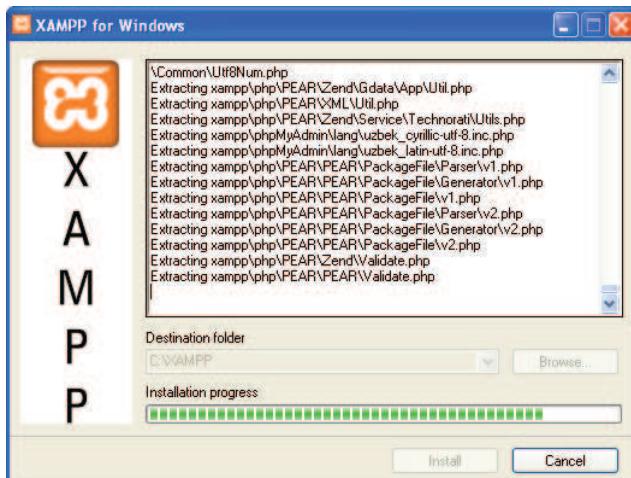
Vista Note:
Because of missing or insufficient write permissions in
"C:\Program Files", we recommend to use alternate folder
for XAMPP ("C:\xampp" or "C:\meinverzeichnis\xampp").

see also
[FAQ - Vista problems](#)



Instalasi dan Konfigurasi

1. Pastikan anda telah mendownload aplikasi XAMPP for Windows di www.apachefriends.com. Agar proses instalasinya simple, pilih XAMPP for Windows yang basic package executable (EXE).
2. Double klik file paket instalasi tersebut dan pilihlah folder default instalasi misal ke drive C:\. Kemudian klik tombol **Install**. Hasilnya program akan melakukan proses instalasi dan membuat folder C:\XAMPP secara otomatis



Gb. 1.1 Proses Instalasi XAMPP ke drive C:XAMPP

```
#####
## XAMPP 1.7.3 - Setup
##
## Copyright 2009 Carsten Wiedmann (FreeBSD License)
##
## Authors: Carsten Wiedmann <carsten_sttgt@gmx.de>
##          Kay Uogelgesang <kvo@apachefriends.org>
#####

Should I add shortcuts to the startmenu/desktop? <y/n>: y
```

Gb.1.2 Pilih yes untuk menempatkan icon shortcut XAMPP
di desktop

```
#####
## XAMPP 1.7.3 - Setup
##
## Copyright 2009 Carsten Wiedmann (FreeBSD License)
##
## Authors: Carsten Wiedmann <carsten_sttgt@gmx.de>
##          Kay Uogelgesang <kvo@apachefriends.org>
#####

Current directory does not match configured directory.
I must relocate the XAMPP paths correctly.

Should I proceed? <y/x=exit setup>: y
```

Gb. 1.3 Pilih yes untuk mengoreksi relokasi path XAMPP

```
#####
## XAMPP 1.7.3 - Setup
##
## Copyright 2009 Carsten Wiedmann <FreeBSD License>
##
## Authors: Carsten Wiedmann <carsten_sttgt@gmx.de>
##          Kay Vogelgesang <kvo@apachefriends.org>
#####
Should I make a portable XAMPP without drive letters?
NOTE: - You should use drive letters, if you want use services.
      - With USB sticks you must not use drive letters.
```

Gb. 1.4 Pilih no karena kita tidak hendak mengintall XAPP
di USB flashdisk

```
Should I make a portable XAMPP without drive letters?
NOTE: - You should use drive letters, if you want use services.
      - With USB sticks you must not use drive letters.

Your choice? <y/n>: n
relocating XAMPP...
relocate XAMPP base package
relocate Apache
relocate FileZilla FTP Server
relocate Mercury
relocate MySQL
relocate OpenSSL
relocate Perl
relocate PHP
relocate phpMyAdmin
relocate Sendmail
relocate Webalizer
relocate XAMPP Demopage
relocating XAMPP successful.

XAMPP is ready to use.

Please /Datums\ to continue.
```

Gb. 1.5 XAMPP sudah siap digunakan.

```
#####
## XAMPP 1.7.3 - Setup
##
## Copyright 2009 Carsten Wiedmann <FreeBSD License>
##
## Authors: Carsten Wiedmann <carsten_sttgt@gmx.de>
##          Kay Vogelgesang <kvo@apachefriends.org>
#####
I have set the timezone in 'php.ini' and 'my.ini' to "Asia/Krasnoyarsk".
You should correct these values if my guess was wrong.
```

Gb. 1.6 XAMPP melakukan setting automatis untuk time-zone

Di akhir instalasi akan muncul jendela

■ Aplikasi Berbasis Web dengan PHP & MySQL *Learning By Doing and Make it Simple!*

```
#####
## XAMPP 1.7.3 - Setup
##
## Copyright 2009 Carsten Wiedmann <FreeBSD License>
## Authors: Carsten Wiedmann <carsten_sttgt@gmx.de>
##          Kay Vogelgesang <kvo@apachefriends.org>
#####

1. start XAMPP Control Panel
2. relocate XAMPP
   (current path: C:\XAMPP)
3. disable HTTPS (SSL)
4. disable Server Side Includes (SSI)
5. enable IPv4 only (current: IPv4/6 <auto>)
6. disable mod_perl
7. disable Apache::ASP

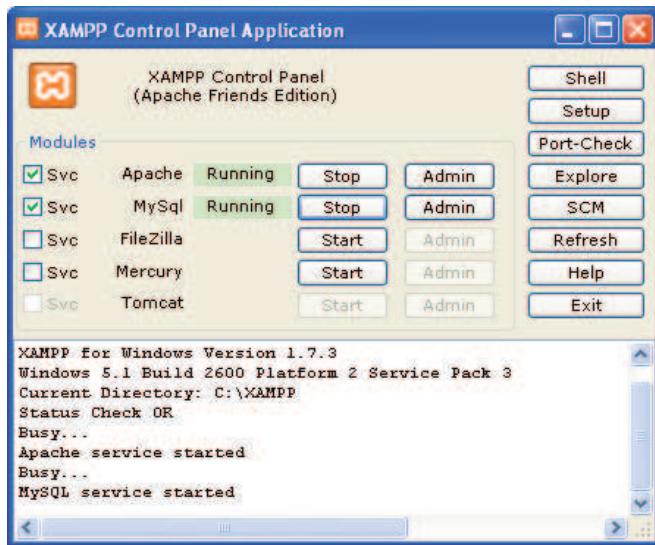
x Exit

Please choose <1-7/x>:
```

Gb.1.7 Proses instalasi sudah selesai. Pilih x untuk keluar dari proses instalasi.

Tahap berikutnya adalah melakukan konfigurasi XAMPP sehingga aplikasi wesberver Apache dan databse MySQL berjalan sebagai service.

1. Klik menu Start -> **All Program** -> **XAMPP for Windows** -> **XAMPP Control Panel** . hasilnya akan muncul jendela XAMPP Control Panel
2. Centang checkbox **Svc Apache** dan **Svc MySQL** sehingga mereka dapat berjalan sebagai service di windows. Kemudian klik **OK**
3. Kemudian tekan Tombol **Start** untuk menghidupkan masing-masing service tersebut sehingga kedua aplikasi tersebut sekarang telah berjalan sebagai server dan sudah siap digunakan



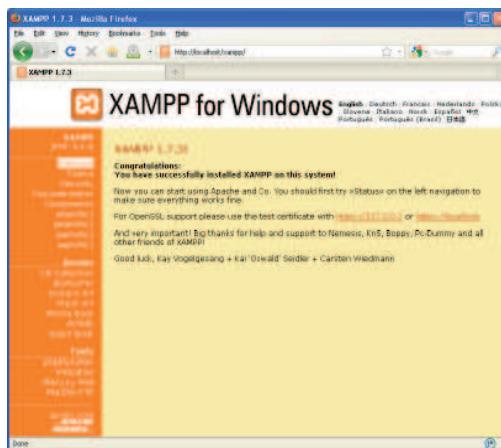
Gb.1.8 XAMPP control panel

Untuk melihat apakah webserver yang kita install benar-benar bekerja dengan baik, caranya adalah dengan mengeceknya melalui browser. Kita tinggal buka browser dan kemudian ketikkan **alamat** untuk webserver tersebut yakni `http://localhost` atau `http://127.0.0.1`. Hasilnya **akan** muncul halaman program XAMPP sebagai berikut.

■ Aplikasi Berbasis Web dengan PHP & MySQL *Learning By Doing and Make it Simple!*



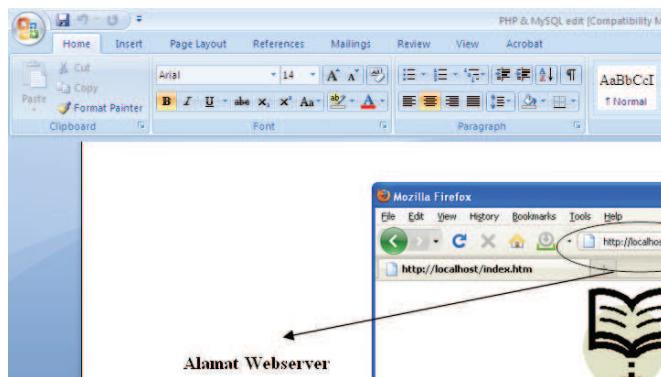
Gb. 1.9 Tampilan splash XAMPP



Gb. 1.10 Dashboard XAMPP

Adapun untuk menyimpan file-file web yang kita buat, yakni ditempatkan di folder **C:\XAMPP\htdocs**

Untuk mentest apakah file web yang kita buat bisa ditampilkan di webserver, terlebih dahulu kita harus membuat satu dokumen HTML sederhana misal dengan Microsoft Word dan disimpan (**Save as**) sebagai *Webpage (*.htm, *.html)*, misal dengan nama *index.htm*. Tempatkan dokumen tersebut di *home directory* yakni **C:\XAMPP\htdocs**. Setelah itu bukalah browser, dan ketikan di address barnya : <http://localhost/>



Gb.1.11 Webserver akan menampilkan halaman web dengan alamat <http://localhost/index.htm>

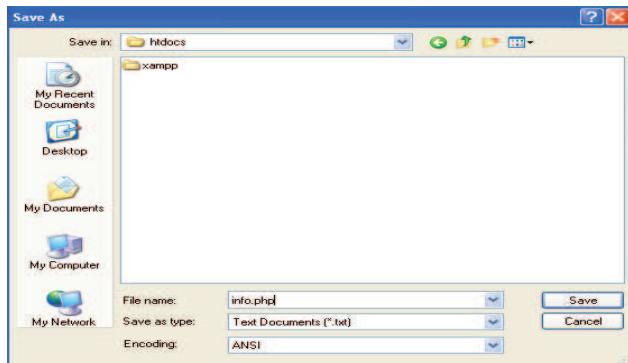
Standard baku untuk halaman pertama suatu alamat web nama file-nya diberi nama **index.htm** atau **index.html** atau **index.php**

Sekarang kita coba untuk membuat file PHP kita yang pertama. Ketikkan dengan menggunakan notepad text sebagai berikut:

```
<?php  
phpinfo();
```

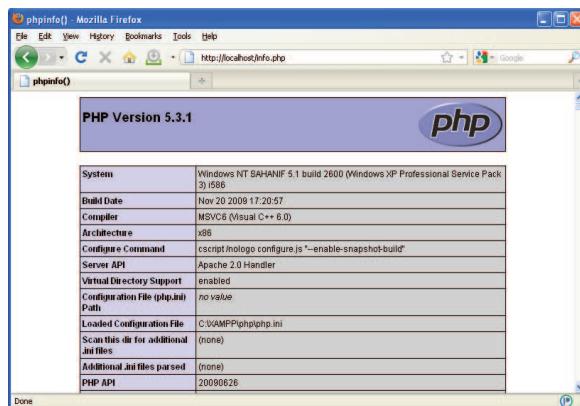
?>

Setelah itu simpan file tersebut dengan nama info.php dan simpan di folder **C:\XAMPP\htdocs**.



Gb. 1.12 Simpan sebagai info.php

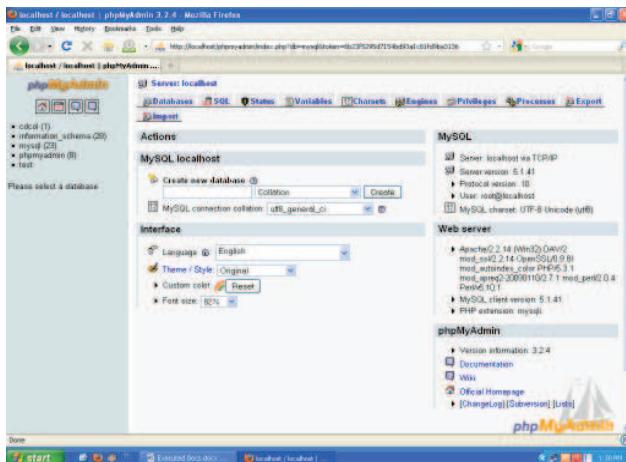
Bukalah browser, dan ketikan di address barnya : <http://localhost/info.php> . Jika jendela seperti di bawah ini muncul, berarti program PHP sudah terinstall dengan baik.



Gb.1.13 Tampilan file info.php menampilkan informasi versi PHP yang telah diinstall

Verifikasi Instalasi MySQL

XAMPP secara default sudah melakukan instalasi MySQL dengan sendirinya. Untuk menceknya adalah dengan menjalankan program phpMyAdmin apakah berjalan atau tidak yang beralamt di <http://xampp/phpmyadmin>



Gb.1.14 Tampilan aplikasi berbasis web phpMyAdmin untuk mengelola database MySQL

So What Next ?

Selamat, Anda sudah menginstal perlengkapan yang Anda butuhkan untuk belajar web programming dengan PHP dan MySQL. Ada baiknya untuk mempermudah proses pengetikan kode-kode latihan PHP, sebaiknya Anda menggunakan editor text alternatif selain Notepad yakni Notepad++ sehingga kode-kode Anda menjadi lebih rapi dan mudah ter-

■ Aplikasi Berbasis Web dengan PHP & MySQL *Learning By Doing and Make it Simple!*

baca. Notepad++ dapat didownload secara gratis di <http://notepad-plus.sourceforge.net/uk/site.htm>. Perjuangan belum berakhir, melainkan baru permulaan, tetap semangat.

Bab 2

Mengenal HTML

HTML (HyperText Markup Language) adalah “bahasa” yang digunakan pada untuk membuat halaman web di internet. HTML memungkinkan kita untuk menampilkan teks, gambar, link ke halaman lain, membuat form isian, menyisipkan video, suara, dan lain-lain.

Struktur Dokumen HTML

Dokumen HTML sebenarnya hanya berupa file teks biasa yang dilengkapi dengan tanda-tanda khusus (tag) yang menentukan bagaimana teks tersebut ditampilkan. Karena itu, dokumen HTML dapat dibuat menggunakan teks editor sederhana seperti *Notepad*, *Notepad++*, *EditPlus*, *vi*, dan sebagainya. Namun sekarang ini telah banyak visual editor untuk HTML yang bisa mempermudah seperti *Adobe*

Dreamweaver, Bluefish, dan lain-lain.

Tag HTML biasanya adalah tag-tag yang berpasangan dan ditandai dengan simbol < dan >, sedangkan ‘pasangan’ atau akhir perintah dari sebuah tag ditandai dengan tanda ‘/’, misalnya pasangan dari tag <tag> adalah </tag>.

Dokumen HTML biasanya disimpan dengan *extension* .HTM atau .HTML. Untuk membuka dokumen HTML ini, diperlukan sebuah *Web Browser*. Contoh web browser yang banyak dipakai adalah Mozilla Firefox, Microsoft Internet Explorer, Opera, dan sebagainya. Berikut ini adalah sebuah contoh dokumen HTML sederhana:

```
<HTML>
<HEAD>
    <TITLE>Judul Yang Muncul Pada Title Bar Browser</TITLE>
</HEAD>
<BODY>
<H1>Hello World</H1>
Di dalam BODY ditempatkan teks, gambar, link atau isi dokumen Anda.
</BODY>
</HTML>
```

Kode 2.1 helo.html – Struktur dokumen HTML sederhana

Jika dokumen ini dibuka pada web browser akan tampak seperti pada gambar di bawah ini.



Gb.2-1 Tampilan File hello.html

Dokumen HTML dapat dibagi menjadi dua bagian, yaitu bagian kepala (**HEAD**) dan tubuh (**BODY**). Bagian kepala ditandai dengan tag **<HEAD></HEAD>**. Pada bagian ini biasanya diisi dengan informasi-informasi tambahan yang berguna bagi web browser ketika menampilkan halaman tersebut, atau informasi untuk *search engine*. Bagian tubuh diapit tag **<BODY></BODY>**. Isi dokumen dan tag-tag HTML lainnya diletakkan di bagian ini.

Salah satu elemen penting yang terletak pada bagian **head** adalah **TITLE**. Seperti tampak pada contoh di atas, teks yang diapit tag **<TITLE></TITLE>** ditampilkan pada *title bar* web browser. Teks ini adalah judul halaman web bersangkutan. Judul ini berguna untuk memberi gambaran umum pada pengunjung tentang isi halaman bersangkutan.

Heading

Tag yang pertama kali akan kita pelajari adalah tag **heading**, yang berfungsi untuk memformat Heading (Judul/sub judul) dalam sebuah dokumen. Ada 7 buah heading

yang dikenal dalam HTML yaitu <H1> sampai <H7>. Untuk mengetahui ukuran tiap heading, anda dapat mencoba baris-baris berikut :

```
<HTML>
<HEAD>
<TITLE>Latihan Heading</TITLE>
</HEAD>

<BODY>
<H1>Heading 1</H1>
<H2>Heading 2</H2>
<H3>Heading 3</H3>
<H6>Heading 6</H6>
<H7>Heading 7</H7>

</BODY>
</HTML>
```

Kode 2.2 heading.html – Tampilan Huruf untuk judul dan subjudul

Paragraf

Untuk memformat paragraf kita bisa menggunakan tag <P>. Dengan tag ini kita bisa menentukan *alignment* (posisi) dari dokumen HTML. Untuk mengatur posisi tulisan kita menggunakan atribut ALIGN. Atribut ALIGN diikuti dengan posisi yang diinginkan.

Selain tag <P> ada pula tag
 untuk pindah baris, <HR> untuk membuat garis, <BLOCKQUOTE> untuk membuat kutipan, <PRE> untuk menampilkan text sama dengan yang anda ketikkan dalam dokumen HTML, misal anda mengetikkan:

```
<HTML>
<HEAD>
  <TITLE>TEST</TITLE>
<HEAD>
<BODY>
```

```
Hallo  
    Hallo  
Hallo  
</BODY>  
</HTML>
```

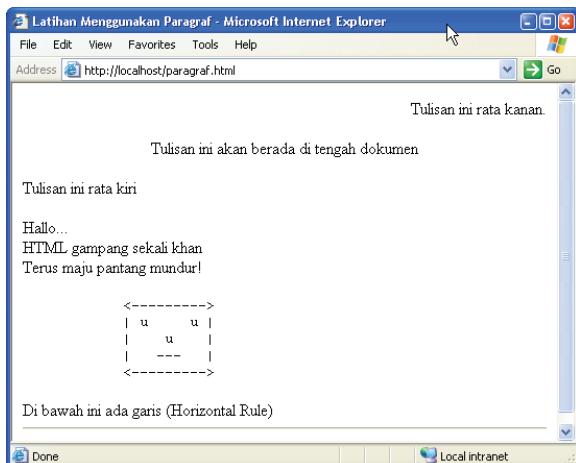
Maka yang akan ditampilkan oleh browser adalah

```
HalloHalloHallo
```

Hal itu terjadi karena HTML tidak mengenali tombol **Enter** sehingga tombol tersebut diabaikan. Untuk mengatasinya hal tersebut maka kita menggunakan tag **<PRE>** tersebut.

```
<HTML>  
<HEAD>  
    <TITLE>Latihan Menggunakan Paragraf</TITLE>  
</HEAD>  
  
<BODY>  
    <P ALIGN="right"> Tulisan ini rata kanan.  
    <P ALIGN="center"> Tulisan ini akan berada di tengah dokumen  
    <P ALIGN="left"> Tulisan ini rata kiri  
    <P align="left">  
    Hallo...<BR>  
    HTML gampang sekali khan<BR>  
    Terus maju pantang mundur!<BR>  
  
<PRE>  
    <----->  
    |   u       u   |  
    |           u     |  
    |           ---  |  
    <----->  
</PRE>  
  
Di bawah ini ada garis (Horizontal Rule) <HR>  
  
</BODY>  
</HTML>
```

Kode 2.3 paragraf.html – Latihan memakai paragraf



Gb.2-2 Tampilan file paragraf.html

Pengaturan Huruf (Font)

Untuk mengatur jenis, ukuran, warna suatu font, kita bisa menggunakan tag , tag ini memiliki beberapa atribut diantaranya FACE dan COLOR.

Atribut FACE digunakan untuk menentukan jenis font yang akan kita gunakan, semisal kita akan menggunakan jenis font Comic Sans MS, maka kita dapat menuliskan . Sedangkan COLOR berfungsi untuk menentukan warna font, bisa dengan cara menyebutkan nama warna seperti red, green dan blue atau menggunakan nilai RGB (Red Green Blue) dari suatu warna, misalnya FF0000 untuk red, 00FF00 untuk green dan 0000FF untuk blue. Ada juga atribut SIZE untuk menentukan ukuran font

■ Aplikasi Berbasis Web dengan PHP & MySQL *Learning By Doing and Make it Simple!*

```
<HTML>
<HEAD>
<TITLE>Latihan Font</TITLE>
</HEAD>

<BODY>
<B>Ini Font Biasa yang dicetak Tebal (BOLD) </B><BR>
<FONT FACE="Arial" COLOR="red" SIZE=4>
Ini Menggunakan Font Arial, warna merah dan ukurannya 4
</FONT><BR>
<FONT FACE="Verdana" COLOR="blue" SIZE=2>
Ini menggunakan Verdana, warna biru dan ukurannya 2
</FONT>

</BODY>
</HTML>
```

Kode 2.4 font.html – Contoh pemakaian jenis font



Gb.2-3 Tampilan file font.html

Warna Dokumen

Suatu dokumen web akan terlihat lebih menarik jika diberi warna-warna atau gambar sebagai backgroundnya. Untuk itu, pada HTML tersedia atribut-atribut untuk mengatur hal tersebut, atribut tersebut diletakkan pada tag <BODY> sehingga efeknya akan dapat dilihat di seluruh dokumen. Atribut yang digunakan dalam tag <BODY> yaitu

Atribut	Fungsi
BGCOLOR	Menentukan warna background suatu halaman web
B G A C K - GROUND	Menentukan gambar/images yang akan dipasang sebagai background halaman web
TEXT	Menentukan warna teks normal
LINK	Menentukan warna teks yang berfungsi sebagai link
ALINK	Menentukan warna teks link ketika diklik
VLINK	Menentukan warna teks link ketika selesai diklik

Tabel 2.1 Atribut Warna dokumen yang diletakkan di BODY

Contoh pemakaianya adalah sebagai berikut :

```
<BODY BGCOLOR="yellow" TEXT="black" LINK="blue"
      ALINK="red" VLINK="green">
```

Link

Dalam web, selalu ada yang dinamakan link, link berfungsi menghubungkan antara satu halaman dengan halaman lainnya. Untuk link ini HTML menyediakan tag **<A>** (*Anchor*). Tag **<A>** memiliki beberapa atribut mis : HREF untuk menghubungkan antara satu halam dengan halaman lainnya, dan NAME untuk memberikan nama pada suatu paragraf/tulisan dalam dokumen HTML, sehingga mempermudah navigasi dalam suatu halaman web.

```
<HTML>
<HEAD>
<TITLE>Latihan Link</TITLE>
</HEAD>

<BODY>
<A NAME="top">Bagian ini ditandai dengan top</A><BR>
<A HREF="heloo.html">Link ke Hello World (heloo.html)</
A><BR>
<A HREF="paragraf.html">Link ke Latihan Paragraf
(paragraf.html)</A><BR>
<A HREF="font.html">Link ke Latihan Font (font.html)</
```

```
A><BR>
<A HREF="#top">Link ke tempat yang ditandai</A>
</BODY>
</HTML>
```

Kode 2.5 link.html – Penggunaan Link

Menampilkan Gambar (Image)

Suatu situs web yang bagus, biasanya menampilkan gambar-gambar untuk menarik pengunjung masuk kedalamnya, bayangkan saja jika situs tersebut hanya menampilkan tulisan saja, tentu amat membosankan. HTML menyediakan fasilitas untuk menampilkan gambar di halaman web kita menggunakan tag .

Tag memiliki bermacam atribut seperti SRC yang digunakan untuk menentukan lokasi gambar yang akan ditampilkan, ALT sebagai teks pengganti apabila ada browser yang tidak mendukung grafik, BORDER berfungsi untuk memberikan suatu border di sekeliling gambar, Height dan weight berfungsi untuk menentukan tinggi atau lebar suatu gambar dalam halaman web kita (dalam ukuran pixel), HSPACE dan VSPACE berfungsi menentukan jarak spasi dengan objek disekitar gambar secara horisontal maupun vertikal.

```
<HTML>
<HEAD>
<TITLE>Latihan Penggunaan IMG</TITLE>
</HEAD>

<BODY>

<IMG SRC="apache.gif" ALIGN=LEFT HSPACE=20 VSPACE=1>
```

■ Aplikasi Berbasis Web dengan PHP & MySQL Learning By Doing and Make it Simple!

```
<A HREF="http://www.apache.org">Apache Web Server</A>  
Adalah Web Server open source dan gratis.<BR>  
Sangat tangguh dan paling banyak digunakan di  
Internet.<BR>  
</BODY>  
</HTML>
```

Kode 2.6 img.html – IMG digunakan untuk memasukkan gambar ke dokumen HTML



Gb.2-4 Tampilan File img.html

Menggunakan Tabel

Untuk membuat tampilan website yang bagus, kebanyakan para desainer web menggunakan tabel untuk mempermudah penggerjaan mereka, coba saja lihat situs www.detik.com dan www.kompas.com, dimana tabel digunakan untuk mempercantik situs tersebut.

Untuk membuat tabel, HTML menyediakan tag <TABLE>. Dalam tag <TABLE> terdapat banyak atribut, yaitu :

Atribut	Fungsi
BORDER	Menentukan ukuran border/garis tabel
WIDTH	Menentukan lebar tabel
HEIGHT	Menentukan tinggi tabel
BGCOLOR	Menentukan background tabel

BACKGROUND	Menentukan gambar yang digunakan untuk background tabel
------------	---

Tabel 2.2 Atribut yang dipakai dalam <TABLE>

Untuk membuat baris, setelah menuliskan tag <TABLE> maka digunakan tag <TR>. Sedangkan untuk membuat kolom digunakan tag <TD>. Sebagian besar atribut dalam tag <TABLE> dapat digunakan dalam tag <TR> maupun dalam tag <TD>.

Penggabungan baris/kolom

Untuk menggabungkan baris/kolom (*merge*) digunakan atribut COLSPAN dan ROWSPAN. COLSPAN digunakan untuk menggabungkan beberapa kolom menjadi 1 sedangkan ROWSPAN menggabungkan beberapa baris menjadi 1.

CELLPADDING dan CELLSPACING

CELLPADING digunakan untuk mengatur spasi antara border dengan tulisan, sedangkan CELLSPACING digunakan untuk mengatur spasi antar dua buah sel.

```
<HTML>
<HEAD>
    <TITLE>Contoh Penggunaan Tabel</TITLE>
</HEAD>

<BODY>
<TABLE BORDER="1" CELLSPACING="2" CELLPADDING="3"
WIDTH="100%">
<TR      BGCOLOR="yellow"      ALIGN="center"><TD>No</
TD><TD>Nama</TD><TD>Alamat</TD></TR>
<TR><TD>1</TD><TD>Faisal</TD><TD>Bandung</TD></TR>
<TR><TD>2</TD><TD>Ade</TD><TD>Jakarta</TD></TR>
<TR><TD>3</TD><TD>Herry</TD><TD>Cimahi</TD></TR>
```

■ Aplikasi Berbasis Web dengan PHP & MySQL Learning By Doing and Make it Simple!

```
<TR><TD>4</TD><TD COLSPAN="2" BGCOLOR="green">Nama &  
Alamat digabung</TD><TR>  
</TABLE>  
</BODY>  
</HTML>
```

Kode 2.7 tabel.html – Contoh pembuatan tabel dan manipulasinya

No	Nama	Alamat
1	Faisal	Bandung
2	Ade	Jakarta
3	Henry	Cimahi
4	Nama & Alamat digabung	

Gb.2-5 Tampilan file tabel.html

Komentar

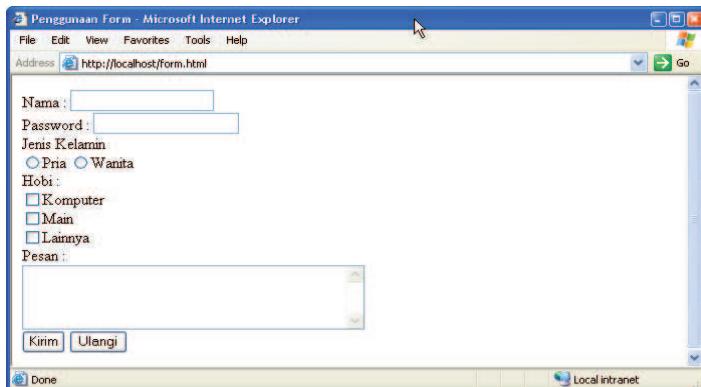
Dalam kode HTML, kita dapat menuliskan keterangan penulisan dimana keterangan biasanya membantu programmer web untuk memberikan petunjuk/komentar penulisan di baris-baris kode yang ditulis. Komentar ini tidak akan ditampilkan di halaman web. Komentar dapat dituliskan di dalam dokumen HTML dengan cara mengapitkan tag `<!--` dan ditutup dengan tag `-->`:

```
<!-- Ini adalah keterangan atau komentar penulisan -->
```

Membuat Form

Untuk melakukan interaksi dengan user, biasanya suatu situs menggunakan form untuk menerima masukan/input dari user dan memproses hasil inputan tersebut di

server. Misalkan saja :



Gb.2-6 Halaman Web dengan Form Interaktif

Dalam gambar di atas dapat dilihat untuk nama, kita menggunakan tipe textbox, password menggunakan tipe password, untuk jenis kelamin menggunakan radio button dan untuk hobi menggunakan checkbox. Untuk mengatur tipe form, menggunakan atribut TYPE, struktur form dapat dibuat sebagai berikut.

```
<FORM METHOD="..." ACTION="...">
<INPUT TYPE="..." NAME="..." VALUE="...">
<INPUT TYPE="Submit" VALUE="...">
</FORM>
```

Untuk membuat form seperti pada Gb. 2.x dalam HTML dapat ditulis :

```
<HTML>
<HEAD>
    <TITLE> Penggunaan Form </TITLE>
</HEAD>

<BODY>
<!-- Latihan membuat Form -->
```

■ Aplikasi Berbasis Web dengan PHP & MySQL Learning By Doing and Make it Simple!

```
<FORM METHOD=POST ACTION="hello.html">
Nama : <INPUT TYPE="Text" NAME="nama"><BR>
Password : <INPUT TYPE="Password" NAME="pass"><BR>
Jenis Kelamin <BR>
<INPUT TYPE="Radio" NAME="kelamin" VALUE="pria">Pria
<INPUT TYPE="Radio" NAME="kelamin" VALUE="wanita">Wanita<BR>
Hobi : <BR>
<INPUT TYPE="Checkbox" NAME="hobi1" VALUE="komputer">Komputer<BR>
<INPUT TYPE="Checkbox" NAME="hobi2" VALUE="main">Main</BR>
<INPUT TYPE="Checkbox" NAME="hobi3" VALUE="lain">Lainnya</BR>
Pesan : <BR>
<TEXTAREA name="pesan" ROWS="4" COLS="40"></TEXTAREA><BR>
<INPUT TYPE="Submit" VALUE="Kirim">
<INPUT TYPE="Reset" VALUE="Ulangi">

</BODY>
</HTML>
```

Kode 2.8 form.html – Contoh pemakaian form di halaman web

Atribut-atribut diatas dapat dijelaskan sebagai berikut:

Atribut	Fungsi
METHOD	Metode pengiriman data ke file tujuan. Ada 2 yaitu POST dan GET
ACTION	Aksi yang akan dilakukan jika user menekan tombol Submit
NAME	Memberikan nama tiap input
VALUE	Memberikan nilai suatu input
TYPE	Tipe form yang akan kita gunakan.

Tabel 2.3 Atribut Umum pada FORM

<INPUT>

Elemen <INPUT> digunakan untuk menampung masukan dari pengunjung. Bentuk masukan dapat diubah-ubah sesuai kebutuhan dengan atribut TYPE. Berikut ini adalah nilai-nilai

yang mungkin untuk atribut TYPE elemen <INPUT>:

TYPE	Keterangan
TEXT	Kotak teks biasa
PASSWORD	Kotak password (teks ditampilkan sebagai '**')
CHECKBOX	Checkbox (dapat memilih lebih dari satu pilihan)
RADIO	Radio (hanya dapat memilih salah satu pilihan)
IMAGE	Tombol berupa gambar
BUTTON	Tombol
SUBMIT	Tombol untuk mengirimkan isi form
RESET	Tombol untuk mengembalikan isian ke harga default

Tabel 2.4 Atribut TYPE elemen <INPUT>

Elemen <INPUT> umumnya memiliki atribut *NAME* dan *VALUE*. Setelah form dikirimkan pada web server, NAME biasanya akan dijadikan nama variabel yang menampung nilai VALUE.

<SELECT> dan <OPTION>

Elemen <SELECT> memungkinkan pengguna memilih salah satu pilihan yang disediakan. Pilihan ditampilkan menggunakan elemen <OPTION>.

<TEXTAREA>

Elemen <TEXTAREA> digunakan untuk menerima masukan yang terdiri dari banyak baris, seperti komentar pada buku tamu. Tinggi dan lebar text area dapat ditentukan dari atribut *ROWS* dan *COLS*.

So What Next ?

Di bab ini kita telah berlatih dasar-dasar HTML se-

bagai bekal untuk ke tahap selanjutnya. Halaman website HTML yang ada di bab ini sifatnya statis karena HTML ini lebih berperan sebagai pembentuk tampilan dasar halaman web. Adapun dengan bekal dasar HTML ini diharapkan kita lebih familiar dan tidak kaget dengan tag-tag yang membentuk suatu halaman web. Tag-tag HTML ini akan selalu berdampingan dengan kode-kode PHP dalam **membentuk** program berbasis web yang dinamis dan interaktif.

Bab 3

Bermain dengan PHP

Pada bab sebelumnya kita telah belajar mengenai tag-tag HTML yang digunakan untuk membuat halaman web yang statis. Nah pada bab ini kita akan belajar PHP untuk ditancapkan (*embed*) pada dokumen HTML sehingga menghasilkan halaman web yang dinamis. Untuk lebih jelasnya lihat contoh berikut :

```
<HTML>
<HEAD>
    <TITLE>Contoh Skrip PHP </TITLE>
</HEAD>

<BODY>
<?php
// menampilkan teks
echo "Hello world, ini adalah kode php pertama saya";
?>

</BODY>
</HTML>
```

Kode 1: hello.php – Script PHP sederhana yang ditancapkan di dokumen HTML

Pada contoh di atas, kode PHP `echo` berfungsi untuk menampilkan tulisan "Hello world, ini adalah kode php pertama saya" pada halaman web. Yang disebut kode PHP adalah kode yang berada diantara tag pembuka `<?php` dan tag penutup `?>`.

Komentar

Komentar sangat berguna untuk memperjelas isi kode PHP Anda dan memudahkan saat *debugging*. PHP mengetahui beberapa cara penulisan komentar, yaitu:

- C style. Komentar diawali dengan tanda pembuka `/*` dan penutup `*/`, umumnya dipakai untuk banyak baris.
- C++ style. Komentar diawali dengan tanda `//`. Hanya untuk 1 baris.
- Unix Shell style. Komentar diawali dengan tanda `#`. Juga hanya untuk 1 baris.

Statement Penulisan

Seperti pada Turbo Pascal dan C/C++, setiap statement dalam PHP harus selalu diakhiri dengan tanda ; (titik koma). Seperti terlihat pada contoh di bawah :

```
echo ("Hello world, PHP...siapa takut");
```

Cara penulisan PHP bersifat semi case sensitif. Case-sensitive itu apa sih? Gampangnya, sifat case-sensitive akan membedakan huruf besar dan huruf kecil. Untuk syntax, operator dan fungsi, umumnya PHP tidak case-sensitive, artinya statement `echo` sama dengan `ECHO`

atau Echo. Sedangkan untuk variabel, PHP bersifat case-sensitif. Artinya, variabel \$nama berbeda dengan variabel \$Nama. Dengan demikian cara penulisan PHP bersifat semi Case Sensitif.

Meng-escape Karakter

Misalnya Anda ingin menulis teks di bawah ini di browser dengan menggunakan PHP:

```
Welcome to "PHP" the best place to explore the 'new world'
```

Mungkin Anda berpikiran cara menulisnya adalah begini:

```
<?php  
    echo " Welcome to "PHP" the best place to explore the 'new  
world' ";  
?>
```

Ternyata cara penulisan di atas akan menimbulkan error, karena terdapat tanda double-quote dan apostrof di dalam string yang dicetak. Untuk mengatasinya, tanda double-qoute dan apostrof harus di-escape terlebih dahulu sebelum di echo. Escape karakter dilakukan dengan menulis tanda backslash (\) sebelum karakter tersebut. Jadi, cara yang benar adalah

```
<?php  
    echo " Welcome to \"PHP\" the best place to  
explore the \'new world\' ";  
?>
```

Contoh lain adalah jika Anda ingin menulis kode HTML dengan statement echo:

```
<?php
```

```
echo "<TABLE BORDER=\"1\" ALIGN=\"CENTER\">
    <TR><TD><FONT FACE=\"arial\" SIZE=\"2\">Test..
test</FONT></TD></TR>
</TABLE>";
?>
```

Variable dan Konstanta

Variable

Setiap bahasa pemrograman pastilah mempunyai suatu variabel. Variabel berfungsi untuk menyimpan beberapa bit data, sehingga untuk memanggil suatu data kita tinggal memanggil variabel tersebut. Dalam PHP, setiap variabel selalu dimulai dengan tanda '\$'. Tidak peduli data tersebut apakah integer, real maupun string, PHP akan secara otomatis mengkonversi data menurut tipenya.

Umumnya, variabel berasal dari 3 sumber. Pertama, variabel secara explisit diisi (assign) dalam skrip kita. Kedua, variabel berasal dari halaman HTML (biasanya dari form input). Ketiga, variabel bisa berasal dari lingkungan PHP, yaitu variabel yang pre-defined dalam PHP.

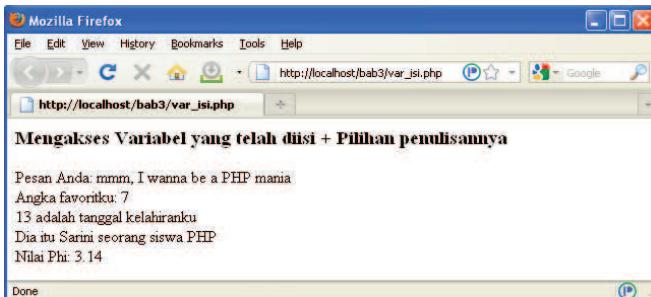
a. Assignment Variable

PHP membuat bekerja dengan variabel menjadi sangat mudah. Dalam skrip PHP, Anda tidak perlu mendefinisikan variabel terlebih dahulu sebelum mengisi atau memakainya. Apapun yang ingin Anda isikan ke sebuah variabel – mau string, integer, float, array, dsb – tinggal isi saja, tanpa harus mendefinisikam tipe variabelnya.

Contoh-contoh assignment (pengisian) variabel:

```
<?php  
//file var_isi.php  
$pesan = "mmm, I wanna be a PHP mania"; //mengisi  
string  
$angka = 7; //mengisi integer  
$Angka = "13"; //mengisi string  
$phi = 3.14; //mengisi bilangan floating point (real)  
$nama = "Sarini";  
  
echo "<h3>Mengakses Variabel yang telah diisi + Pili-  
han penulisannya</h3>";  
echo "Pesanan Anda: $pesan <br>"; //penulisan variable  
biasa (di dalam "")  
echo "Angka favoritku: $angka<br>"; //penulisan vari-  
able biasa (di dalam "")  
echo $Angka ." adalah tanggal kelahiranku <br>"; //  
penulisan variable di depan  
echo "Dia itu ".$nama." seorang siswa PHP <br>"; //  
penulisan variable di tengah  
echo "Nilai Phi: ", $phi; //penulisan variable di be-  
lakang  
?>
```

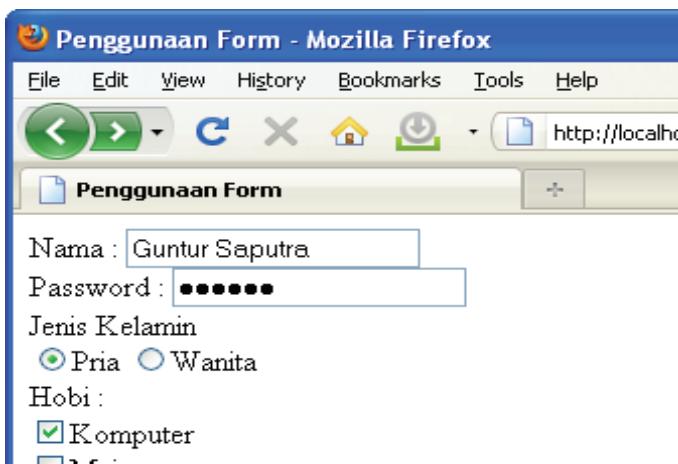
Kode 2: var_isi.php – Contoh variable yang telah diisikan nilainya



Gb.3-1 Tampilan halaman var_isi.php

b. Mengakses Variabel yang berasal dari form input HTML

Sebelumnya (lihat hal. 15) kita telah membuat sebuah file bernama `form.html` yang merupakan form biodata. Form tersebut belum bisa berfungsi karena apabila ditekan tombol **Kirim**, form tersebut akan melakukan **action** ke file yang bernama `proses.php`. Nah kita akan berusaha mengakses variable yang ada pada form tersebut di halaman `proses.php`. Nama variable dari setiap input berasal dari atribut NAME. Mari kita lihat gambarannya :



Gb.3-2 Atribut NAME pada elemen form menentukan nama variable yang akan diakses

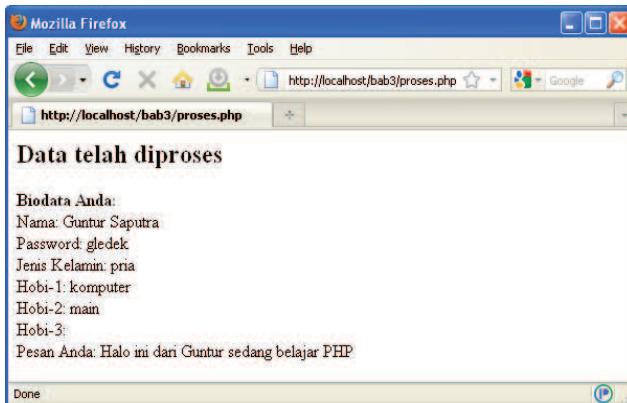
Pada halaman `proses.php` kita dapat mengakses nilai dari variable-variable yang telah dikirimkan (disubmit atau di-POST). Berikut adalah kode dari halaman `proses.php`:

```
<?php  
  
echo "<H2>Data telah diproses</H2>";  
echo "<B>Biodata Anda:</B> <BR>";  
echo "Nama: $_POST[nama] <BR>";  
echo "Password: $_POST[pass] <BR>";  
echo "Jenis Kelamin: $_POST[kelamin] <BR>";  
echo "Hobi-1: $_POST[hobi1] <BR>";  
echo "Hobi-2: $_POST[hobi2] <BR>";  
echo "Hobi-3: $_POST[hobi3] <BR>";  
echo "Pesan Anda: $_POST[pesan]";  
?>
```

Kode 3: proses.php – Halaman yang digunakan untuk memproses form yang di-POST

Ingat bahwa untuk mengakses variable yang disubmit melalui form, maka kita harus menambahkan syntax **\$_POST[nama_variable_yang_dipost_via_form]**. \$_POST ini dinamakan *predefine variable* yang khusus digunakan untuk memproses variable yang disubmit oleh form.

Apabila tombol **Kirim** di klik, maka form akan melakukan action ke halam `proses.php`, hasilnya akan seperti di bawah ini:



Gb.3-3 Tampilan halaman proses.php dari hasil action form biodata

Dari program diatas, dapat terlihat bahwa atribut NAME yang terdapat pada form secara otomatis diubah oleh PHP ke dalam variabel-variabel, sehingga memudahkan kita untuk mengidentifikasi data yang dikirim oleh form. Penggunaan variable ini lebih terasa gunanya jika data yang disubmit itu akan disimpan ke dalam database.

c. Predefined Variabel

PHP memiliki variable-variable yang telah didefinisikan sebelumnya, variable ini nilainya bergantung pada beberapa faktor, seperti jenis server, setting server, dan lain-lain. Predefine variable dapat dilihat dengan memakai fungsi `phpinfo()`. Bisa dilihat pada halaman `info.php` yang telah kita buat sebelumnya.

Berikut ini adalah beberapa predefine variable yang dibuat pada Webserver Apache, jadi tidak ada jaminan apakah variable-varibale ini bisa dikenali pada jenis web-server lainnya.

- `REMOTE_ADDR`: IP address dari komputer yang dipakeai untuk mengakses
- `HTTP_USER_AGENT`: Keterangan jenis browser yang dipakai untuk mengakses, misal: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
- `SERVER_NAME`: Nama dari server host dimana script dieexecute. Misal: localhost.
- `SERVER_SOFTWARE`: Menunjukkan webserver yang digunakan, contoh: APACHE/2.2.4.

Apache Environment

Variable	Value
MIBDIRS	C:\XAMPP\php/extras/mibs
MYSQL_HOME	C:\XAMPP\mysql\bin
OPENSSL_CONF	C:\XAMPP\apache\bin\openssl.cnf
PHP_PEAR_SYSCONF_DIR	C:\XAMPP\php
PHPRC	C:\XAMPP\php
TMP	C:\XAMPP\tmp
HTTP_HOST	localhost
HTTP_USER_AGENT	Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.2.3) Gecko/20100401 Firefox/3.6.3
HTTP_ACCEPT	text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
HTTP_ACCEPT_LANGUAGE	en-us,en;q=0.5
HTTP_ACCEPT_ENCODING	gzip,deflate
HTTP_ACCEPT_CHARSET	ISO-8859-1,utf-8;q=0.7,*;q=0.7
HTTP_KEEP_ALIVE	115
HTTP_CONNECTION	keep-alive
PATH	C:\Program Files\PC Connectivity Solution;C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem;C:\Program Files\Microsoft SQL Server\9.0\Tools\bin\;C:\Program Files\Samsung\Samsung PC Studio\3;C:\Program Files\QuickTime\QTSystem\;C:\Program Files\Nmap
SystemRoot	C:\WINDOWS

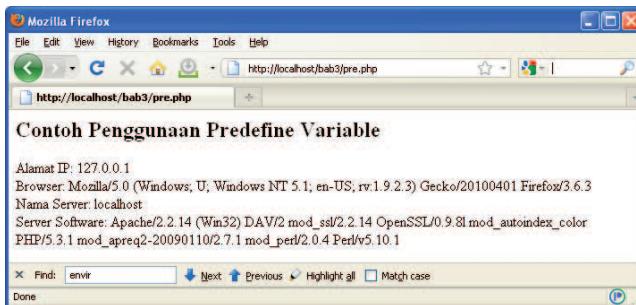
Gb.3-4 Daftar Variable-variable predefined dapat dilihat di info.php

Berikut ini adalah contoh penggunaan predefined variable.

```
<?php  
//file pre.php  
echo "<h2>Contoh Penggunaan Predefine Variable</h2>";  
echo "Alamat IP: $_SERVER[REMOTE_ADDR]";  
echo "<br>Browser: ", $_SERVER[HTTP_USER_AGENT];  
echo "<br>Nama Server: ", $_SERVER[SERVER_NAME];  
echo "<br>Server Software: ", $_SERVER[SERVER_SOFTWARE];  
?>?>
```

Kode 4: pre.php – Contoh penggunaan Predefine Variable khusus untuk server.

Pengunaannya didahului oleh syntax **\$_SERVER[predefined_variable]**



Gb.3-5 Tampilan halaman dari penggunaan predefined variable

Konstanta

Konstanta adalah nilai yang tak pernah berubah. Misalnya, nama Anda. Jika Anda ingin mencetak nama Anda di seluruh halaman web, daripada mencetak string nama Anda lebih baik mendefinisikan konstanta misalnya MYNAME. Tidak seperti variable, konstanta tidak didahului tanda \$. Konstanta dapat berisi string atau angka. Untuk membuat sebuah konstanta dalam PHP, kita bisa menggunakan fungsi *built-in* PHP yaitu define () .

Misal :

```
define ("MYNAME", "Faisal Brotosewiryo");
define ("PERUSAHAAN", "PT. Angin Ribut");
define ("ALAMAT", "Ciledug Raya Jakarta ");
```

Dalam contoh di atas, kita membuat 2 buah konstanta yaitu PERUSAHAAN yang berisi identitas perusahaan dan konstanta ALAMAT yang berisikan alamat tempat perusahaan tersebut berada. Untuk memanggil konstanta tersebut, kita bisa menuliskan sebagai berikut :

```
<?
//konst.php
define ("MYNAME"," Faisal Brotosuwiryo");
define ("PERUSAHAAN", "PT. Angin Ribut");
define ("ALAMAT", "Jl. Ciledug Raya Jakarta");

echo "Nama saya " . MYNAME . "<BR>";
echo "Saya kerja di " . PERUSAHAAN . "<BR>";
echo "Yang beralamat di " . ALAMAT . "<BR>";
?>
```

Kode 5: konst.php – Contoh penerapan konstanta di PHP

Jika kita jalankan program tersebut, maka secara otomatis akan kita dapatkan hasil :

Nama saya Faisal Brotosuwiryo
Saya kerja di PT. Angin Ribut
Yang beralamat di Jl. Ciledug Raya Jakarta

Operator

Operator digunakan untuk menghasilkan sebuah nilai dengan melakukan operasi pada satu atau lebih nilai lainnya. Nilai yang dioperasikan (digunakan dalam operasi) disebut operand. PHP mengenal beberapa jenis operator, yang kebanyakan hampir sama dengan operator C/C++. Berikut ini akan dijabarkan satu persatu operator yang umum dipakai.

a. Operator Aritmatika

Hampir disetiap bahasa pemrograman selalu mengenal operator aritmatika. Operator aritmatika yang dikenal di PHP terdapat pada tabel berikut.

Operator	Penggunaan	Contoh	Keterangan
+	Penjumlahan	$9 + 2$	Hasil = 9
-	Pengurangan	$9 - 2$	Hasil = 7

*	Perkalian	$9 * 2$	Hasil = 18
/	Pembagian	$9 / 2$	Hasil = 4.5
%	Modulus	$9 \% 2$	Operator modulus akan menghasilkan sisa pembagian. Hasil operasi di samping adalah 1

Table 3.x Operator aritmatika

Catatan : PHP mentolerir adanya white space antara operator dan operand. Jadi, statement $9+2$ sama saja dengan $9 + 2$.

b. Operator String

Hanya ada satu operator string, yaitu operator *concatenation* (.). Operator ini digunakan untuk menyambung dua buah ekspresi string.

```
<?php
//concat.php
$a="Hallo ";
$b= $a . "Selamat Belajar PHP";
echo "$b <br>";
//$b berisikan "Hallo Selamat belajar PHP"

$input=<INPUT TYPE=\"text\" NAME=\"nama\">>;
$tombol=<INPUT TYPE=\"submit\" VALUE=\"Kirim\">>;
echo "<FORM ACTION=\"proses.php\" METHOD=\"post\">>". 
$input.$tombol."</FORM>";
?>
```

Kode 5: conc.php – Contoh pemakaian operator string concatenation

c. Operator Perbandingan

PHP memiliki operator-operator perbandingan yang sangat berguna untuk menyusun ekspresi kondisional. Ekspresi yang menggunakan operator perbandingan sel-

Jika menghasilkan nilai TRUE atau FALSE. Misalkan \$alpha = 9 dan \$beta = 5.

Operator	Arti	Ekspresi	Hasil Evaluasi Nilai
==	sama dengan	\$alpha == \$beta	False
!=	tidak sama dengan	\$alpha != \$beta	True
>	lebih besar daripada	\$alpha > \$beta	True
<	lebih kecil daripada	\$alpha < \$beta	False
>=	lebih besar atau sama dengan	\$alpha >= \$beta	True
<=	lebih kecil atau sama dengan	\$alpha <= \$beta	False

Tabel 3.x Operator perbandingan.

Operator perbandingan lainnya adalah operator trinary “?:” yang memiliki bentuk :

```
(ekspresi) ? (jika benar) : (jika salah);
```

Contoh :

```
<?php  
$umur=5;  
$status=($umur<=5) ? ("termasuk balita") : ("bukan  
balita lagi");  
echo "Jadi $umur tahun adalah $status";  
?>
```

Kode 6: trinary.php – pemakaian operator trinary

Skrip di atas akan menghasilkan :

Jadi 5 tahun adalah termasuk balita

d. Operator Logika

Operator logika dipakai untuk menggabungkan beberapa kondisi, sehingga dapat dievaluasi sebagai sebuah

■ Aplikasi Berbasis Web dengan PHP & MySQL *Learning By Doing and Make it Simple!*
ekspresi.

Contoh	Nama Operator	Hasil evaluasi akan TRUE jika
(kondisi1) and (kondisi2)	And	Kedua kondisi bernilai TRUE
(kondisi1) && (kondisi2)	And	Kedua kondisi bernilai TRUE
(kondisi1) or (kondisi2)	Or	Salah satu atau kedua kondisi bernilai TRUE
(kondisi1) (kondisi2)	Or	Salah satu atau kedua kondisi bernilai TRUE
(kondisi1) xor (kondisi2)	Xor	Salah satu bernilai TRUE, tidak dua-duanya
!(Kondisi1)	Not	Kondisi1 bernilai FALSE

Tabel 3.x Operator Logika

e. Operator Penugasan

Operator penugasan (assignment) digunakan sebagai jalan pintas dalam melakukan suatu operasi diantara dua operan. Biasanya operan pertama adalah variable dan hasilnya disimpan pada variable yang sama. Sebagai contoh misal \$a=8 dan \$b=2. Penerapannya dapat dilihat sebagai berikut:

Contoh	Penggunaan	Equivalen dengan	Hasil
<code>+=</code>	<code>\$a += \$b</code>	<code>\$a = \$a + \$b</code>	<code>\$a = 10</code>
<code>-=</code>	<code>\$a -= \$b</code>	<code>\$a = \$a - \$b</code>	<code>\$a = 8</code>
<code>*=</code>	<code>\$a *= \$b</code>	<code>\$a = \$a * \$b</code>	<code>\$a = 16</code>
<code>/=</code>	<code>\$a /= \$b</code>	<code>\$a = \$a / \$b</code>	<code>\$a = 4</code>
<code>%=</code>	<code>\$a %= \$b</code>	<code>\$a = \$a % \$b</code>	<code>\$a = 0</code>
<code>.=</code>	<code>\$a .= \$b</code>	<code>\$a = \$a . \$b</code>	<code>\$a = 8 2</code>
<code>++</code>	<code>\$a++</code>	<code>\$a = \$a + 1</code>	<code>\$a = 9</code>
<code>--</code>	<code>\$a--</code>	<code>\$a = \$a - 1</code>	<code>\$a = 7</code>

Tabel 3.x Operator Penugasan

f. Operator Increment / Decrement

PHP mendukung operator increment dan decrement seperti yang dipakai pada bahasa C/C++.

Operator	Keterangan	Efek
<code>++\$a</code>	Pre-increment	Menambah isi variable \$a dengan 1, lalu mengembalikan isi \$a
<code>\$a++</code>	Post-Incre-ment	Mengembalikan isi \$a dulu, baru menambah isi variable \$a dengan 1
<code>--\$a</code>	Pre-decrement	Mengurangi isi variable \$a dengan 1, lalu mengembalikan isi \$a
<code>\$a--</code>	Post-decre-ment	Mengembalikan isi \$a dulu, baru mengurangi isi variable \$a dengan 1

Tabel 3.x Operator increment/decrement

Contoh :

```
//Contoh post-increment  
$a = 2;  
echo $a++; //hasilnya masih : 2  
echo $a;   //hasilnya sudah : 3  
  
//Contoh pre-increment  
$a = 2;  
echo ++$a; //hasilnya sudah : 3  
echo $a;   //hasilnya tetap : 3
```

Studi Kasus

Berikutnya kita akan menerapkan beberapa aspek fundamental yang telah dipelajari di atas dengan membuat halaman web sebuah toko komputer online sederhana. Pada halaman tersebut kita akan membuat form pemesanan barang yang kemudian hasil submitnya akan diproses lebih lanjut. Berikut adalah bentuk tampilan halamannya:

The screenshot shows a Mozilla Firefox browser window with the title bar 'PALASARI Computer Shop - Mozilla Firefox'. The address bar displays the URL 'http://localhost/bab3/shop.h...'. The main content area contains a form titled 'Form Pemesanan Barang'. The form has two columns: 'Item' and 'Banyaknya'. It lists three items: Monitor (2), Speaker (1), and Keyboard (2). Below the table is a blue 'Submit Order' button. At the bottom left of the form area, there is a 'Done' link.

Gb.3-6 Tampilan Form Pemesanan Barang

Sedangkan listing source codenya adalah sebagai berikut :

```
<HTML>
<HEAD>
<TITLE>PALASARI Computer Shop</TITLE>
</HEAD>

<BODY>
<h2>Form Pemesanan Barang</h2>

<FORM ACTION="order_proses.php" METHOD=post>
<tabel border=0>
<tr bgcolor="#cccccc">
<td width=150>Item</td> <td width=15>Banyaknya</td>
</tr>

<tr>
<td>Monitor</td>
<td align=center><input type="text" name="jum_monitor" size=3></td>
</tr>

<tr>
<td>Speaker</td>
<td align=center><input type="text" name="jum_speaker" size=3></td>
</tr>

<tr>
<td>Keyboard</td> <td align=center> <input
```

■ Aplikasi Berbasis Web dengan PHP & MySQL *Learning By Doing and Make it Simple!*

```

        type="text" name="jum_keyboard" size=3></td>
    </tr>

    <tr>
        <td colspan=2 align=center><input type=submit value="Submit Order"></td>
    </tr>

</table>
</FORM>

</BODY>
</HTML>
```

Kode 3-7: shop.html – Kode untuk form pemesanan barang

Apabila tombol **Submit Order** diklik, maka form akan diproses ke halaman `order_proses.php` yang tampilannya adalah sebagai berikut:



Gb.7 Halaman setelah form diproses

Source code halaman `order_proses.php` adalah sebagai berikut:

```
<?php
DEFINE ("COMPANY","<h2>PHP Computer Shop</h2>");
DEFINE ("RP_MONITOR",650000); //harga 1 monitor
DEFINE ("RP_SPEAKER",60000); // harga 1 speaker
DEFINE ("RP_KEYBOARD",25000); // harga 1 keyboard

//mengambil nilai variable yang dikirimkan oleh form
//ke dalam variable yang lebih singkat
$monitor=$_POST[jum_monitor];
$speaker=$_POST[jum Speaker];
$keyboar=$_POST[jum_keyboard];

$waktu=date("H:m:s d-M-Y"); // date adalah fungsi untuk waktu

$jum_total = $monitor + $speaker + $keyboar;
$rp_total=($monitor*RP_MONITOR)+($speaker*RP_SPEAKER)+($keyboar*RP_KEYBOARD);
$pajak=0.1 * $rp_total;
$rp_bayar = $rp_total + $pajak;

echo COMPANY;
echo "<h3>Hasil Pemesanan</h3>";
echo "<p>Pemesanan diproses pada $waktu</p>";
echo "Barang yang Anda pesan adalah sebagai berikut:<br>";
echo "$monitor buah Monitor<br>";
echo "$speaker buah Speaker<br>";
echo "$keyboar buah Keyboard<br>";

echo "Jumlah total barang yang dipesan: $jum_total<br>";
echo "Harga sebelum pajak: Rp.$rp_total<br>";
echo "Harga setelah pajak 10%: Rp.$rp_bayar<br>";

?>
```

Kode 8: `order_proses.php` – Halaman yang digunakan untuk memproses form pemesanan

Pada halaman `order_proses.php` di atas, kita dapat mengakses 3 variable yang dikirimkan oleh form pemesanan dengan menggunakan `$_POST[nama_variable_yg_dipost]`.

Dimana `nama_variable_yg_dipost` diberikan oleh attribut `NAME` pada elemen form INPUT antara lain `jum_monitor`, `jum_keyboard` dan `jum_speaker`. Agar lebih singkat maka kita masukkan nilai yang dipost tersebut ke variable baru yakni `$monitor`, `$speaker`, dan `$keyboar`,

```
$monitor=$_POST[jum_monitor];  
$speaker=$_POST[jum_speaker];  
$keyboar=$_POST[jum_keyboard];
```

`DEFINE` digunakan untuk mendefinsikan konstanta (nilai yang sifatnya tetap) baik itu untuk string maupun integer. Konstanta bertipe string diterapkan pada `COMPANY`, sedangkan konstanta bertipe integer diterapkan pada `RP_MONITOR`, `RP_SPEAKER` dan `RP_KEYBOARD`, yang merupakan harga dari masing-masing produk.

Variable `$jum_total` digunakan untuk mendefinisikan total banyaknya produk yang dipesan,

```
$jum_total = $monitor + $speaker + $keyboar
```

Sedangkan variable `$rp_total` digunakan untuk mendefinisikan total harga berdasarkan banyaknya barang yang dipesan, yakni harga dasar dikali banyaknya yang dipesan kemudian ditambahkan untuk masing-masing produk.

```
$rp_total=($monitor*RP_MONITOR)+($speaker*RP_SPEAKER)+($keyboar*RP_KEYBOARD)
```

Pengenaan pajak 10% dikenakan pada total harga, rumusnya adalah `$pajak=0.1*$rp_total`. Sedangkan harga setelah pajak ditentukan oleh rumus `$rp_bayar=$rp_total+$pajak`.

`Date()` merupakan salah satu fungsi yang dimiliki PHP dari sekian banyak fungsi-fungsi bawaan yang lain. Fungs `date()` digunakan untuk menyatakan waktu. Di dalamnya dapat disimpan format string yang menunjukkan tanggal maupun waktu. Di abwah ini format string yang sering dipakai untuk menyatakan waktu.

- d - tanggal hari dalam setiap bulan, misal “01” to “31”
- D - nama hari yang dinyatakan dengan 3 huruf pertama; misal “Fri”
- F - nama bulan dalam format lengkap. misal “January”
- h - jam dalam format 12, misal “01” to “12”
- H - jam, dalam format 24, misal “00” to “23”
- i - menit; i.e. “00” to “59”
- j - tanggal hari dalam setiap bulan tanpa embel-embel nol, misal “1” to “31”
- m - nomor bulan dalam setahun; misal “01” to “12”
- M – nama bulan dalam fromat 3 karakter; misal “Jan”
- n – nomor bulan tanpa nol di depan, misal “1” to “12”
- s - detik; misal. “00” to “59”
- Y – tahun berformat 4 digits; misal “1999”

- y – year berformat 2 digits, misal “99”

Jadi jika sekarang kita menuliskan fungsi
date ("H:m:s d-M-Y")

Hasil keluarannya adalah jam dan tanggal hari ini,
misal 11:12:25 04-Dec-2009

Bab 4

Struktur Kontrol Dalam PHP

Struktur kontrol (statement) merupakan elemen yang sangat penting dalam tiap bahasa pemrograman, karena dengan struktur kontrol kita dapat mengendalikan jalannya eksekusi program. Berikut ini dijabarkan beberapa elemen penting struktur kontrol di PHP yang sering digunakan dalam dunia nyata.

Strukur Kondisional

a. Statement IF

Berfungsi untuk melakukan eksekusi suatu statement secara bersyarat. Dengan statement kondisional `if`, sebuah atau sekumpulan statement hanya akan dieksekusi ketika sebuah ekspresi yang dievaluasi bernilai TRUE. Sintaks :

```
if (kondisi)
{
```

```
Statement  
...  
}
```

Sebagai contoh kita bisa menerapkannya pada form pemesanan sebelumnya. Jika setelah disumbit jumlah total pemesanannya nol (mungkin karena tidak sengaja kepencet), maka kita akan memberikan pernyataan bahwa user belum melakukan data pemesanan. Scriptnya akan seperti ini:

```
If ($jum_total==0)  
echo "Order belum bisa diproses. Anda belum memasukkan data pemesanan! <br>";
```

Ingin bahwa operator perbandingan (==) berbeda dengan operator penugasan (=). Kondisi \$jum_total==0 akan TRUE jika jika \$jum_total bernilai nol. Jika \$jum_total tidak sama dengan nol, kondisi akan FALSE. Ketika kondisi TRUE, statement echo akan dieksekusi.

Untuk statement yang lebih dari satu baris, maka penulisannya harus diberikan tanda kurung kurawal {}.

Contoh :

```
If ($jum_total==0) {  
    echo "<font color=red>";  
    echo "Order belum bisa diproses. Anda belum memasukkan data pemesanan! <br>";  
    echo "</font>";  
}
```

b. Statement ELSE

Jika ekspresi yang diuji bernilai FALSE, Anda dapat

mengeksekusi sebuah atau sekumpulan statement yang lain dengan menggunakan kata kunci `else`. Sintaks:

```
if(kondisi){  
    Statement1;  
}else{  
    Statement2;  
}
```

Contoh kita akan menerapkannya pada form pemesanan. Kodanya adalah sebagai berikut.

```
If ($jum_total==0){  
    echo "Order belum bisa diproses. Anda belum me-  
masukkan data pemesanan! <br>";  
} else {  
    echo "Barang yang Anda pesan adalah sebagai  
berikut:<br>";  
    echo "$jum_monitor buah Monitor<br>";  
    echo "$jum_speaker buah Speaker<br>";  
    echo "$jum_keyboard buah Keyboard<br>";  
}
```

Kita bisa membuat proses logika yang lebih “canggih” dengan menggunakan statement if bersarang. Statementnya adalah jika jumlah salah satu produk lebih dari 0 maka jumlahnya akan ditampilkan, namun jika sama dengan nol maka tidak akan ditampilkan. Kodanya adalah sebagai berikut:

```
If ($jum_total==0){  
    echo "Order belum bisa diproses. Anda belum me-  
masukkan data pemesanan! <br>";  
} else {  
    echo "Barang yang Anda pesan adalah sebagai  
berikut:<br>";  
    if ($jum_monitor>0)  
        echo "$jum_monitor buah Monitor <br>";
```

```
if ($jumlah_speaker>0)
    echo "$jumlah_speaker buah Speaker <br>";
if ($jumlah_keyboard>0)
    echo "$jumlah_keyboard buah Keyboard <br>";
}
```

c. Statement ELSEIF

Kita bisa membuat sebuah rangkaian dari banyak pilihan dengan menggunakan statement elseif. Statement elseif merupakan kombinasi dari pernyataan else dan if. Dengan memberikan serangkaian kondisi, program bisa memeriksanya sampai ditemukan salah satu yang bernilai TRUE.

Misal kita akan memberikan diskon untuk pembelian yang besar. Skema diskonya adalah sebagai berikut :

- Untuk jumlah pembelian monitor kurang dari 10 – tidak didiskon.
- Pembelian 10-49 monitor – diskon 5%.
- Pembelian 50-99 monitor– diskon 10%
- Pembelian >= 100 monitor– diskon 15%

Kita dapat membuat program untuk memeriksa diskon yang diberikan dengan menggunakan statement elseif. Berikut adalah kodennya:

```
If ($jumlah_monitor < 10)
    $diskon = 0;
elseif ($jumlah_monitor >= 10 && $jumlah_monitor <= 49 )
    $diskon = 0.05;
elseif ($jumlah_monitor >= 50 && $jumlah_monitor <= 99 )
    $diskon = 0.1;
elseif ($jumlah_monitor > 100 )
    $diskon = 0.15;
```

d. Statement Switch...Case

Switch ... case biasanya digunakan untuk eksekusi statement kondisional yang memiliki banyak kemungkinan jawaban. Bentuk umum **switch** adalah:

```
switch ($variabel) {  
    case nilai_1:  
        statement_1;  
        break;  
    case nilai_1:  
        statement_2;  
        break;  
    default:  
        statement_3;  
        break;  
}
```

statement_1 dieksekusi jika \$variabel bernilai sama dengan nilai_1. statement_2 dieksekusi jika \$variabel bernilai sama dengan nilai_2. statement_3 dieksekusi jika \$variabel **tidak** bernilai sama dengan nilai_1 maupun nilai_2.

Mari kita aplikasikan ke dunia nyata. Misal kita ingin mengetahui dari mana konsumen mendapatkan informasi tentang toko kita. Tambahkan kode berikut ini pada file shop.html setelah baris keyboard.

```
<tr>  
    <td>Mengetahui Toko PHP dari</td>  
    <td><select name="tahu">  
        <option value = "a">Langsung datang  
        <option value = "b">Iklan  
        <option value = "c">Buku telepon  
        <option value = "d">Word of mouth  
    </select></td>  
</tr>
```

Halaman formnya akan menjadi seperti di bawah:

PALASARI Computer Shop - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost/bab4/shop.php

PALASARI Computer Shop

Form Pemesanan Barang

Item	Banyaknya
Monitor	<input type="text"/>
Speaker	<input type="text"/>
Keyboard	<input type="text"/>

Mengetahui Toko

PALASARI dan Langsung datang

Submit Order

Done

Gb.4-1 Tampilan halaman form pemesanan setelah penambahan kode

Penambahan elemen form <select> di atas telah menambah variable baru \$tahu yang bisa bernilai “a”, “b”, “c” atau “d”. Kita bisa menangani variable baru ini dengan serangkaian statement if dan elseif seperti di bawah:

```
if($tahu == "a")
    echo "<P>Langsung datang.";
elseif($tahu == "b")
    echo "<P>Tahu dari iklan.";
elseif($tahu == "c")
    echo "<P>Tahu dari buku telepon.";
elseif($tahu == "d")
    echo "<P>Tahu dari word of mouth.";
```

Cara lain adalah dengan menuliskannya dengan statement switch

```
switch($tahu)
{
case "a" :
```

```
echo "<P>Pelanggan langsung datang.";
break;
case "b" :
echo "<P>Pelanggan tahu dari iklan.";
break;
case "c" :
echo "<P>Pelanggan tahu dari buku telepon.";
break;
case "d" :
echo "<P>Pelanggan tahu dari word of mouth.";
break;
default :
echo "<P>Kami bingung pelanggan tahu dari mana.";
break;
}
```

e. Statement exit

Statement exit berguna untuk mengakhiri pengeksekusian skrip di server. Dengan dijalankannya perintah ini, segala kode baik PHP maupun HTML tidak dikirim ke browser.

Contohnya

```
If ($jumlah_total==0) {
    echo "Order belum bisa diproses. Anda belum memasukkan data pemesanan! <br>";
    exit;
} else {
...
}
```

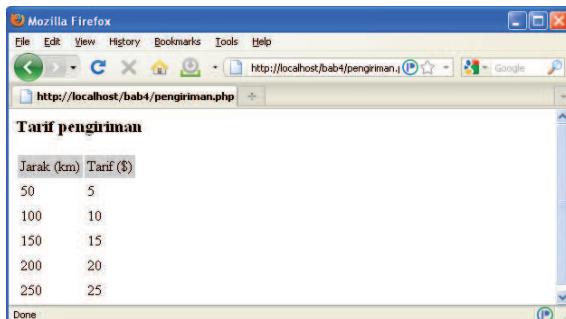
Jika anda telah familiar dengan statement-statement di atas, barangkali anda bertanya, “Manakah yang terbaik untuk digunakan?”. Jawabannya adalah Anda harus coba menggunakan statement kondisional mana yang paling mudah dibaca dan paling cocok pada situasi yang anda

miliki. Anda akan merasakannya dengan pengalaman yang tiada henti.

Pengulangan

Salah satu keunggulan yang dimiliki oleh komputer adalah kemampuannya untuk melakukan pekerjaan yang berulang secara otomatis. Jika terdapat sesuatu yang ingin kita lakukan dengan cara yang sama berkali-kali, anda dapat menggunakan suatu loop untuk mengulangi beberapa bagian dari program anda. Loop berarti kita mengeksekusi beberapa baris program dalam waktu tertentu, atau sampai suatu kondisi telah terpenuhi.

Misal kita ingin menampilkan halaman tarif pengiriman kepada konsumen dimana biayanya akan ditambahkan pada harga pemesanan. Tampilan halamannya adalah seperti di bawah:



Gb.4-2 Halaman tarif pengiriman.html

Kode HTML halaman di atas adalah sebagai berikut:

```
<html>
<body>
<h2>Tarif Pengiriman</h2>
<tabel border = 0 cellpadding = 3>
```

```
<tr>
<td bgcolor = "#CCCCCC" align = center>Jarak (km)</td>
<td bgcolor = "#CCCCCC" align = center>Tarif ($)</td>
</tr>
<tr>
<td align = right>50</td>
<td align = right>5</td>
</tr>
<tr>
<td align = right>100</td>
<td align = right>10</td>
</tr>
<tr>
<td align = right>150</td>
<td align = right>15</td>
</tr>
<tr>
<td align = right>200</td>
<td align = right>20</td>
</tr>
<tr>
<td align = right>250</td>
<td align = right>25</td>
</tr>
</table>
</body>
</html>
```

Kode 4.1: pengiriman.html – halaman untuk tarif pengiriman

Daripada bersusah payah menuliskan keterangan tarif yang berulang dan teratur secara manual, lebih baik kita gunakan PHP. Statement perulangan (loop) di PHP yang sering dipakai diantaranya: while, do..while, dan for. Mari kita bantai satu-satu.

a. While

Perintah ini digunakan untuk mengulangi sebuah perintah sampai jumlah tertentu. While adalah jenis

pengulangan yang paling mudah. Standar penulisannya adalah:

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

Pengulangan `while` akan memeriksa apakah `kondisi` bernilai TRUE atau FALSE. Jika TRUE, satu atau sekumpulan statement di dalam kurung kurawal akan dieksekusi. Setelah semua statement dalam kurung kurawal di eksekusi, `kondisi` akan dievaluasi lagi. Jika hasil evaluasi masih TRUE, semua statement dalam kurung kurawal akan di eksekusi lagi. Begitulah seterusnya, hingga evaluasi terhadap `kondisi` bernilai FALSE. Contoh skrip untuk menampilkan bilangan 0 sampai 5:

```
$nomor=0;  
while ($nomor <=5) {  
    echo "$nomor <br>";  
    $nomor++;  
}
```

Hasil keluarannya akan sebagai berikut:

```
0  
1  
2  
3  
4  
5
```

Jika pertama kali variabel `$nomor` diisi dengan 6, maka pengulangan `while` tidak akan terjadi. Hal tersebut

disebabkan ekspresi (`$a <= 6`) bernilai FALSE.

Sekarang mari kita aplikasikan penggunaan while ini untuk menggenerate tabel tarif pengiriman seperti pada halaman pengiriman.html.

```
<body>
<h2>Tarif Pengiriman</h2>
<tabel border = 0 cellpadding = 3>
<tr>
<td bgcolor = "#CCCCCC">Jarak (km)</td>
<td bgcolor = "#CCCCCC">Tarif ($)</td>
</tr>

<?php
$jarak = 50;
while ($jarak <= 250 ) {
    echo "<tr><td>$jarak</td><td>". $jarak / 10 ."/<
td></tr>";
    $jarak += 50;
}
?>

</tabel>
</body>
</html>
```

Kode 4.2: pengiriman.php – Penggunaan while untuk menggenerate tabel tarif

b. For

For digunakan untuk melaksanakan statemen perulangan yang jumlahnya telah diketahui Standar penulisannya adalah:

```
for (expr1; expr2; expr3)
{
    statement;
}
```

Statement pengulangan for membutuhkan 3 ekspresi di dalam tanda kurung, yang masing-masing dipisahkan oleh karakter “;”. Ekspresi pertama (`expr1`) di-

gunakan untuk menginisialisasi pengulangan, dan hanya dieksekusi sekali pada awal pengulangan. Ekspresi ke-2 (*expr2*) adalah ekspresi boolean, yang dievaluasi pada setiap awal pengulangan.

Jika hasil evaluasi *expr2* ternyata TRUE, pengulangan dilanjutkan dan statement dalam kurung kurawal dieksekusi. Jika *expr2* bernilai FALSE, pengulangan berhenti. Pada setiap akhir pengulangan, ekspresi ke-3 (*expr3*) dieksekusi. Ekspresi ke-3 ini biasanya dipakai untuk *increment* atau *decrement* nilai variabel. Contoh pengulangan *for* untuk menggenerate tarif pengiriman :

```
<?php
for($jarak = 50; $jarak <= 250; $jarak += 50) {
    echo "<tr> <td> $jarak </td><td>". $jarak / 10 ."
</td></tr>";
}
?>
```

Ternyata pengulangan versi while dan versi for keduaanya identik. Loop for agak lebih padat, bisa menghemat beberapa baris. Kedua tipe loop tersebut adalah equivalent – tidak ada yang lebih baik ataupun lebih jelek. Pada situasi tertentu Anda bisa memilih mana yang lebih intuitif dan enak digunakan.

c. Do...While

Standar penulisannya adalah:

```
do
{
    statement;
    ...
}
```

```
while () ;
```

Pengulangan do...while sebenarnya hampir sama dengan while. Perbedaan utamanya terletak pada pemeriksaan ekspresi. Pengulangan do...while akan memeriksa ekspresi setelah semua statement dalam kurung kurawal dieksekusi. Jika hasil pemeriksaan ekspresi bernilai TRUE, pengulangan diteruskan. Jika hasilnya FALSE, pengulangan berhenti. Jadi, dengan pengulangan do...while, statement dalam kurung kurawal akan dieksekusi paling tidak satu kali. Berikut ini adalah contoh penggeneratan tabel tarif pengiriman dengan menggunakan loop do..while:

```
<?php  
$jarak=50;  
do{  
    echo "<tr><td>$jarak</td> <td>". $jarak / 10 ."</td></tr>";  
    $jarak += 50;  
} while ($jarak<=250)  
?>
```

Require() dan Include()

Statement require() dan include() akan menyertakan dan mengevaluasi sebuah file yang diinginkan ke dalam file dimana statement tersebut berada. Perbedaan antara kedua statement tersebut terletak pada cara menangani kesalahan yang terjadi.

Jika terjadi kesalahan, misalnya karena file yang ingin di-include tidak ada, statement include() hanya akan meghasilkan peringatan (warning) dan eksekusi skrip

diteruskan, sedangkan statement `require()` akan menghasilkan *fatal error* dan eksekusi skrip dihentikan. Sintaksnya:

```
require ("nama_file");
include ("nama_file");
```

Sebagai contoh kita akan menampilkan halaman tarif pengiriman di dalam form pemesanan. Daripada menulis kodennya lagi di halaman form pemesanan, alangkah baiknya kita tinggal memakai fungsi `include` untuk menyertakan halaman tersebut. Coba sertakan script :

```
include "pengiriman.php";
```

Di bagian bawah file `shop.html` (setelah `</FORM>`), kemudian simpan (save as) filenya sebagai `shop.php`. Hasil tampilannya akan nampak seperti di bawah :

The screenshot shows a Mozilla Firefox window with the title "PALASARI Computer Shop - Mozilla Firefox". The address bar shows the URL "http://localhost/bab4/shop.php". The main content area displays a form titled "Form Pemesanan Barang". The form has two columns: "Item" and "Banyaknya". Under "Item", there are three options: "Monitor", "Speaker", and "Keyboard". Each item has a corresponding input field for quantity. Below the items is a dropdown menu with options "Mengetahui Toko PALASARI dari" and "Langsung datang". At the bottom of the form is a "Submit Order" button. Below the form is a section titled "Tarif pengiriman" containing a table:

Jarak (km)	Tarif (\$)
50	5
100	10
150	15
200	20
250	25

Gb.4-3 Halaman pemesanan barang yang menyertakan (include) halaman tarif pengiriman

Studi Kasus

Setelah kita memperlajari beberapa statement di atas, maka kita sekarang akan mengaplikasikannya menjadi program yang lebih nyata dan bagus. Berikut ini adalah kode untuk halaman `shop.php`:

```
<html>
<head>
    <title>PALASARI Computer Shop</title>
</head>

<body>
<h2>Form Pemesanan Barang</h2>
```

```
<form action="order_proses.php" method="post">

<tabel border=0>
<tr bgcolor="#cccccc"><td width=150>Item</td>    <td width=15>Banyaknya</td></tr>

<tr><td>Monitor</td><td align=center><input type="text" name="jum_monitor" size=3></td></tr>

<tr><td>Speaker</td><td align=center><input type="text" name="jum_speaker" size=3></td></tr>

<tr><td>Keyboard</td><td align=center><input type="text" name="jum_keyboard" size=3></td></tr>

<tr>
    <td>Mengetahui Toko PALASARI dari</td>
    <td><select name="tahu">
        <option value = "a">Langsung datang
        <option value = "b">Iklan
        <option value = "c">Buku telepon
        <option value = "d">Word of mouth
    </select></td>
</tr>
<tr><td colspan=2 align=center><input type=submit value="Submit Order"></td></tr>

</tabel>

</form>

<?php include "pengiriman.php" ?>

</body>
</html>
```

Kode 4.3 : shop.php – Kode baru untuk halaman form pemesanan dengan penambahan include

Berikut ini adalah halaman `order_proses.php` yang telah ditambahi alur struktur kondisi `if else` yang

lebih canggih:

```
<?php
```

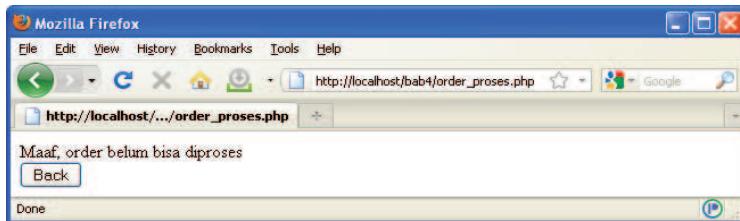
```
DEFINE("COMPANY","<h2>PALASARI Computer Shop</h2>");  
DEFINE("RP_MONITOR",650000);  
DEFINE("RP_SPEAKER",60000);  
DEFINE("RP_KEYBOARD",25000);  
  
//mengambil nilai variable yang dikirimkan oleh form  
//ke dalam variable yang lebih singkat  
$monitor=$_POST[jum_monitor];  
$speaker=$_POST[jum Speaker];  
$keyboard=$_POST[jum Keyboard];  
  
$waktu=date("H:m:s d-M-Y");  
  
$jum_total= $monitor+ $speaker + $keyboard;  
$rp_total = ($monitor*RP_MONITOR)+($speaker*RP_  
SPEAKER)+($keyboard*RP_KEYBOARD);  
$pajak = 0.1 * $rp_total;  
$rp_bayar = $rp_total + $pajak;  
  
if ($jum_total==0) {  
    echo "Maaf, order belum bisa diproses<br>";  
    echo "<input type=button value=Back onclick=history.  
go(-1)>";  
    exit;  
} else {  
    echo COMPANY;  
    echo "<h3>Hasil Pemesanan</h3>";  
    echo "<p>Pemesanan diproses pada $waktu</p>";  
    echo "<hr>"; //<hr> adalah horizontal rule atau  
garis horizontal  
    echo "Barang yang Anda pesan adalah sebagai  
berikut:<br>";  
    if ($monitor>0) {  
        echo "$monitor buah Monitor<br>";  
    }  
    if ($speaker>0) {  
        echo "$speaker buah Speaker<br>";  
    }  
    if ($keyboard>0) {  
        echo "$keyboard buah Keyboard<br>";  
    }  
  
echo "Jumlah yang dipesan: $jum_total<br>";
```

■ Aplikasi Berbasis Web dengan PHP & MySQL *Learning By Doing and Make it Simple!*

```
echo "Harga sebelum pajak: Rp.$rp_total<br>";
echo "Harga setelah pajak 10%: Rp.$rp_bayar<br>";
echo "<hr>";
echo "Referensi: ";
switch( $_POST[tahu] ) //untuk mengetahui pelanggan
tahu dari mana
{
    case "a" :
        echo "Pelanggan langsung datang.";
        break;
    case "b" :
        echo "Pelanggan tahu dari iklan.";
        break;
    case "c" :
        echo "Pelanggan tahu dari buku telepon.";
        break;
    case "d" :
        echo "Pelanggan tahu dari word of mouth.";
        break;
    default :
        echo "Kami bingung pelanggan tahu dari mana.";
        break;
}
?>
```

Kode 4.4: order_proses.php – Sekarang menjadi semakin lengkap

Di bawah ini adalah tampilan halaman order_proses.php dengan berbagai kondisi :



Gb.4-4 Proses order dimana jika konsumen tidak memasukkan jumlah barang yang dipesan



Gb.4-5 Proses order dimana konsumen telah memasukkan jumlah barang yang dipesan

Bab 5

Penanganan File

Penanganan berkas (file) merupakan salah satu aspek yang tidak pernah lepas dalam programming. Dengannya kita bisa menyimpan data masukan yang disubmit melalui form pemesanan ke dalam suatu file. Untuk selanjutnya kita dapat mengakses maupun memanipulasi data yang ada di dalam file tersebut.

Konsep Penanganan File

File merupakan tempat untuk menyimpan data secara permanen dalam suatu media penyimpanan misal hardisk, disket dan yang lainnya. Secara umum format file terbagi menjadi dua yaitu file teks dan binary.

Konsep dasar penanganan file secara umum terdiri atas tiga tahap :

- Membuka file

- Memproses/memanipulasi file
- Menutup file

Membuka dan Menutup File

Untuk membuka dan mengakses file digunakan fungsi `fopen()`. Fungsi ini dapat mengakses file baik itu secara langsung (dari sistem file) maupun melalui alamat HTTP atau FTP di internet. Sintaksnya:

```
fopen (nama_file, mode_akses)
```

Dalam hal ini :

- Parameter `nama_file` adalah string yang menyatakan nama file yg hendak dibuka
- Parameter `mode_akses` adalah metode pengaksesan apakah file tersebut hanya untuk dibaca, ditulis atau ditambah.

Pilihan mode akses yang dapat digunakan adalah sebagai berikut.

Mode Akses	Keterangan
a	Membuka file untuk diubah . Data akan ditambahkan di akhir file yang diakses, jika file tidak ditemukan, maka PHP secara otomatis akan membuat file tersebut
a+	Membuka file untuk diubah dan dibaca . Data akan ditambahkan di akhir file yang akses, jika file tidak ditemukan, maka PHP secara otomatis akan membuat file tersebut.
r	Membuka file untuk dibaca saja
r+	Membuka file untuk dibaca dan ditulis . Data akan ditulis di awal file

w	Membuka file untuk ditulis saja. Data dalam file yang lama akan hilang dan digantikan dengan data baru yang dimasukkan. Jika file tidak ditemukan, PHP secara otomatis akan membuat file tersebut
w+	Membuka file untuk ditulis dan dibaca . Data dalam file yang lama akan hilang dan digantikan dengan data baru yang dimasukkan. Jika file tidak ditemukan, PHP secara otomatis akan membuat file tersebut

Tabel 1: Mode Pengaksesan file

Perlu juga diketahui, fopen dapat digunakan untuk membuka file yang berada pada suatu server. Biasanya pembukaannya diawali dengan tulisan `http://` atau `ftp://`. Secara lebih khusus apabila penulisannya diawali dengan HTTP, file hanya dibuka dengan mode baca (r), sedangkan kalau diawali dengan FTP, operasi pembacaan atau penulisan diperkenankan.

Contoh pemanggilan fopen:

```
$pegangan=fopen("http://karedok.com/index.htm", "r");
```

atau

```
$pegangan=fopen("count.dat","w");
```

Untuk menutup file/berkas digunakan fungsi `fclose()`. Sintaksnya adalah:
`fclose(pegangan)`

Parameter pegangan menyatakan pegangan file yang diperoleh saat memanggil `fopen`. Nilai balik fungsi

ini dapat berupa:

- TRUE apabila berkas berhasil ditutup
- FALSE apabila berkas gagal ditutup

Memasukkan data ke file (fputs)

Fungsi `fputs` digunakan untuk memasukkan/merekam data ke file. Bentuk pemanggilannya:

```
fputs(pegangan, data)
```

Dalam hal ini parameter pegangan menyatakan pegangan file yang diperoleh ketika memanggil `fopen`. Dengan kata lain, parameter menetukan berkas yang akan ditulisi dan parameter `data` (bertipe string) menyatakan data yang akan direkam.

Fungsi ini memberikan nilai balik berupa:

- TRUE kalau data berhasil direkam
- FALSE kalau data gagal direkam

Setelah data direkam, penunjuk file akan menunjuk ke posisi sesudah karakter terakhir yang direkam.

Membaca data dari file (fgets)

Fungsi `fgets` **adalahkebalikan** dari `fputs`. Kegunaannya adalah untuk membaca data yang terdapat pada berkas. Sintaksnya:

```
fgets(pegangan, panjang)
```

Parameter `panjang` menyatakan jumlah karakter yang akan dibaca.

☞ Jumlah maksimal karakter yang akan dibaca adalah

panjang-1 karakter. Namun apabila sebelum sebanyak karakter tersebut telah ditemukan adanya *newline* atau tanda baris baru, jumlah karakter yang dihasilkan akan kurang dari panjang-1.

- ☞ String yang menjadi nilai balik fungsi ini dengan sendirinya akan diakhiri dengan *newline*.
- ☞ PHP juga menyediakan fungsi bernama fgets yang **kengunaannya** serupa dengan fgets, tetapi akan membuang semua tag HTML dan PHP. Secara lengkap sintaxnya adalah sebagai berikut:

```
fgets(pegangan,      panjang[, tag-tag_di-
perkenankan])
```

Parameter ketiga menyatakan string dari tag-tag yang tidak ikut dibuang.

Fungsi feof

Fungsi feof berguna untuk memeriksa apakah penunjuk berkas sedang menunjuk akhir berkas atau tidak. Sintaksnya:

```
feof(pegawai)
```

Fungsi ini menghasilkan nilai TRUE kalau penunjuk berkas sedang menunjuk akhir berkas. Untuk keadaan lainnya, nilai baliknya berupa FALSE.

Studi Kasus 1: Menyimpan data form ke dalam File

Setelah Anda mengenal sejumlah fungsi yang terkait dengan file, berikutnya Anda akan mendapatkan gambaran tentang penggunaan sejumlah fungsi tersebut dengan

mengimplementasikannya pada form pemesanan yang telah dibuat. Modifikasilah form pemesanan sehingga tampilannya seperti di bawah ini:

The screenshot shows a Mozilla Firefox browser window titled "PALASARI Computer Shop - Mozilla Firefox". The address bar displays "http://localhost/bab5/shop.php". The main content area shows a modified shopping cart form. It includes fields for Name and Address, a table for item quantities, a dropdown for information source, and a "Submit Order" button.

Item	Banyaknya
Monitor	2
Speaker	4
Keyboard	3

Mengetahui Toko PHP
dari

Submit Order

Gb.5-1 Form pemesanan dengan tambahan input data nama dan alamat

Berikut ini adalah kode yang harus ditambahkan. Tempatkan kode tersebut di atas tabel yang terdahulu, tetapi tetap harus di dalam tag `<form></form>`.

```
<tabel border=0>
```

```
<tr><td>Nama</td><td><input type="text" name="nama"></td></tr>
<tr><td align="top">Alamat</td><td><textarea name="alamat"></textarea></td></tr>
</tabel>
```

Setelah tombol **Submit Order** diklik, maka form akan melakukan action ke file `order_proses.php`. Berikut adalah skrip `order_proses.php` yang baru. Di dalam

skrip yang baru ini dilakukan beberapa penambahan dan modifikasi sehingga data yang dikirimkan form dapat tersimpan dalam sebuah file.

```
<?php
```

```
DEFINE ("COMPANY","<h2>PALASARI Computer Shop</h2>");  
DEFINE ("RP_MONITOR",650000);  
DEFINE ("RP_SPEAKER",60000);  
DEFINE ("RP_KEYBOARD",25000);  
  
//mengambil nilai variable yang dikirimkan oleh form  
//ke dalam variable yang lebih singkat  
$monitor=$_POST[jum_monitor];  
$speaker=$_POST[jum_speaker];  
$keyboard=$_POST[jum_keyboard];  
  
$waktu=date("H:m:s d-M-Y");  
  
$jum_total=$monitor+ $speaker + $keyboard;  
$rp_total=($monitor*RP_MONITOR)+($speaker*RP_  
SPEAKER)+($keyboard*RP_KEYBOARD);  
$pajak=0.1 * $rp_total;  
$rp_bayar=$rp_total + $pajak;  
  
if ( (empty($_POST[nama]))OR (empty($_POST[alamat]))  
OR ($jum_total==0) ) {  
    echo "Maaf, order belum bisa diproses<br>";  
    echo "Nama, Alamat dan Jumlah pesanan harus  
diisi<br>";  
    echo "<input type=button value=Back onclick=history.  
go(-1)>";  
    exit;  
} else {  
    echo COMPANY;  
    echo "<h3>Hasil Pemesanan</h3>";  
    echo "<p>Pemesanan diproses pada $waktu</p>";  
    echo "<hr>"; //<hr> adalah horizontal rule atau  
garis horizontal  
    echo "Barang yang Anda pesan adalah sebagai  
berikut:<br>";  
    if ($monitor>0){  
        echo "$monitor buah Monitor<br>";  
    }else {  
        $monitor=0;  
    }
```

```
if ($speaker>0) {
    echo "$speaker buah Speaker<br>";
} else {
    $speaker=0;
}

if ($keyboard>0) {
    echo "$keyboard buah Keyboard<br>";
} else {
    $keyboard=0;
}

echo "Jumlah yang dipesan: $jum_total<br>";
echo "Harga sebelum pajak: Rp.$rp_total<br>";

echo "Harga setelah pajak adalah Rp.$rp_bayar";
echo "<hr>";
echo "Referensi: ";
switch( $_POST[tahu] ) //untuk mengetahui pelanggan
tahu dari mana
{
    case "a" :
        echo "Pelanggan langsung datang.";
        break;
    case "b" :
        echo "Pelanggan tahu dari iklan.";
        break;
    case "c" :
        echo "Pelanggan tahu dari buku telepon.";
        break;
    case "d" :
        echo "Pelanggan tahu dari word of mouth.";
        break;
    default :
        echo "Kami bingung pelanggan tahu dari mana.";
        break;
}

//menyimpan data ke file
$pegangan=fopen("order.txt","a+");
fputs($pegangan, "$_POST[nama]\t");
fputs($pegangan, "$_POST[alamat]\t");
fputs($pegangan, "$monitor Monitor\t");
fputs($pegangan, "$speaker Speaker\t");
fputs($pegangan, "$keyboard Keyboard\t");
fputs($pegangan, "Rp.$rp_bayar\t");
```

■ Aplikasi Berbasis Web dengan PHP & MySQL *Learning By Doing and Make it Simple!*

```
fputs($pegangan, "$waktu\n");
fclose($pegangan);
echo "<h4>Halo, $_POST[nama]. Data order Anda sudah
disimpan</h4>";
echo "Terima Kasih<br><br>";
echo "<a href=\"shop.php\">Form Pemesanan</a>";
}
?>
```

Kode 1:`order_proses.php` – Kode yang dicetak tebal adalah kode tambahan/modifikasi



Gb.5-2 `order_proses.php` dimana sekarang data form disimpan di dalam file `order.txt`

Berikut ini adalah isi dari file `order.txt` dimana data dari form pemesanan disimpan (**dilihat dari homesite**):

1	Sarini arifah	Tarogong	11 Garut	1	Monitor	2	Speaker	1	Keyboard	Rp.874500	10:12:12	09-Dec-2003
2	Aditya Sumardi	Gunugn Baru	10 Bandung	3	Monitor	1	Speaker	2	Keyboard	Rp.2266000	10:12:35	09-Dec-2003
3	Ade Gartinach	Kebayoran Lama	10 Jaksel	1	Monitor	0	Speaker	3	Keyboard	Rp.797500	10:12:03	09-Dec-2003

Pada skrip `order_proses.php` di atas, pernyataan:

```
if (empty($_POST[nama])) OR (empty($_POST[alamat]))  
OR ($jum_total==0) {  
    echo "Maaf, order belum bisa diproses<br>";  
    echo "Nama, Alamat dan Jumlah pesanan harus  
diisi<br>";  
    echo "<input type=button value=Back onclick=history.  
go(-1)>";  
} else {...  
...  
}
```

Digunakan untuk memeriksa keadaan variable `$nama`, `$alamat` dan `$jum_total`. Fungsi `empty()` digunakan untuk memeriksa apakah isi variable dalam keadaan kosong atau tidak. Pernyataan OR akan TRUE jika ada salah satu pernyataan bernilai TRUE. Jadi Jika :

```
empty($_POST[nama])= TRUE  
OR  
empty($_POST[alamat])= TRUE                   TRUE : Order tidak  
akan diproses  
OR  
$jum_total==0 =TRUE  
  
empty($_POST[nama])= TRUE  
OR  
empty($_POST[alamat])= TRUE                   TRUE : Order tidak  
akan diproses  
OR  
$jum_total==0 =TRUE  
  
empty($_POST[nama])= FALSE  
OR  
empty($_POST[alamat])=FALSE                  FALSE : Order akan  
diproses dan data akan disimpan
```

OR

```
$jum_total==0 =FALSE
```

Sedangkan pernyataan :

```
if ($monitor>0) {  
    echo "$monitor buah Monitor<br>";  
} else {  
    $monitor=0;  
}
```

Digunakan untuk memeriksa apakah variable `$monitor` yang dimasukkan lebih dari nol. Jika lebih dari nol maka keterangannya akan dicetak di browser. Jika tidak diisi atau `$monitor` yang dimasukkan adalah nol, maka keterangannya tidak akan dicetak namun variablenya akan diset nilainya 0.

Pernyataan :

```
$pegangan =fopen("order.txt","a+")
```

Digunakan untuk membuka file `order.txt` dengan mode “append” (artinya file akan diubah, jikalau berkas tidak ada maka akan diciptakan. Kalau berkas sudah ada, penambahan – dilakukan oleh `fputs` – akan dilakukan di belakang).

Berikutnya sederetan `fputs` dipakai untuk merekam data ke dalam file. Pada masing-masing `fputs`, “\t” dimaksudkan agar setiap data akan diakhiri/dipisahkan oleh tab horizontal. Sedangkan “\n” digunakan untuk men- gakhiri data dengan newline (baris baru).

Pernyataan :

```
fputs($pegangan, " $_POST[nama]\t");
```

■ Aplikasi Berbasis Web dengan PHP & MySQL Learning By Doing and Make it Simple!

```
fputs($pegangan, " $_POST[alamat]\t");
fputs($pegangan, " $monitor Monitor\t");
fputs($pegangan, " $speaker Speaker\t");
fputs($pegangan, " $keyboard Keyboard\t");
fputs($pegangan, " Rp.$rp_bayar\t");
fputs($pegangan, " $waktu\n");
```

Akan equivalen dengan pernyataan seperti dibawah:

```
fputs($pegangan,"$_POST[nama]\t      $_POST[alamat]\t
$monitor monitor\t $speaker speaker\t $keyboard
keyboard\t Rp.$rp_bayar\t $waktu\n")
```

Lalu berkas ditutup dengan memakai pernyataan:
fclose(\$pegangan);

Adapun pernyataan :

```
echo "<h4>Halo, $_POST[nama]. Data order Anda sudah
disimpan</h4>";
echo "Terima Kasih<br><br>";
echo "<a href=\"shop.php\">Form Pemesanan</a>";
```

Digunakan untuk memberikan pesan kepada para pengguna bahwa data order sudah berhasil diproses dan disimpan.

Studi kasus 2: Membaca data order

Sekarang setelah pelanggan melakukan pemesanan dan datanya disimpan dalam file order.txt, maka sekarang giliran staf dari toko komputer untuk melihat data pemesanan yang masuk. Berikut adalah skrip yang digunakan untuk membaca data pemesanan dari file order.txt.

```
<html>
<head><title>PALASARI Computer Shop - Data Order
```

```
Pelanggan</title></head>
<body>
<h2>Data Order Pelanggan</h2>
<?php
@$pegangan = fopen("order.txt", "r");
if (!$pegangan) {
    echo "<p><strong>Belum ada Order</strong></p>";
exit;
}

while (!feof($pegangan)) {
    $order= fgets($pegangan, 255);
    echo "$order<br>";
}
fclose($pegangan);
?>
</body>
</html>
```

Kode 2: lihat_order.php – Kode untuk melihat data order

Pernyataan :

```
@$pegangan = fopen("order.txt", "r");
```

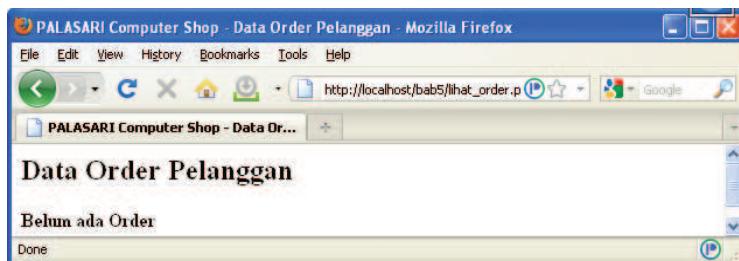
Digunakan untuk membuka file `order.txt` dengan mode “read” atau baca. Penyataan di atas digunakan tanda “@” yang berfungsi untuk menghandle error. Apabila tidak menggunakan “@” maka error yang berkaitan dengan fungsi `fopen` akan terlihat. Errornya misal disebabkan oleh file `order.txt` tidak ada. Jika tidak menggunakan “@” errornya akan terlihat seperti di bawah:

■ Aplikasi Berbasis Web dengan PHP & MySQL Learning By Doing and Make it Simple!



Gb .5-3 fopen tanpa menggunakan "@"

Jika menggunakan "@", maka error akan dihandle dan halaman akan ditampilkan seperti dibawah:



Gb.5-4 fopen dengan menggunakan "@"

Pernyataan

```
if (!feof($pegangan))
```

Digunakan untuk memeriksa apakah !\$pegangan adalah TRUE , jika TRUE berarti datanya tidak ada atau belum ada order.

Sedangkan pernyataan:

```
while (!feof($pegangan))
```

Digunakan untuk melakukan pengulangan pema-

caan dan pemrosesan terhadap data di file selama akhir dari isi file tidak ditemui.

Berikut adalah hasil halaman `lihat_order.php` yang memperlihatkan hasil pembacaan data di `order.txt`:



Gb.5-5 Tampilan halaman `lihat_order.php`

Berbagai Fungsi Untuk File

a. Memeriksa keberadaan file

Untuk memeriksa suatu file ada atau tidak, kita bisa menggunakan fungsi `file_exists`. Sintaksnya:

```
file_exists(namafile)
```

Contoh skripnya :

```
If (file_exists(order.txt)) {  
    echo "Ada order yang siap diproses";  
} else {  
    echo "sampai sekarang belum ada order yang datang";  
}
```

b. Memperoleh Ukuran File

Ukuran file dapat diperoleh dapat diperoleh dengan

meamanggil fungsi `filesize`. Sintaksnya:

```
file_size(namafile)
```

Nilai balik fungsi ini bertipe integer, yang menyatakan ukuran berkas dalam satuan byte.

c. Mengetahui tanggal Pengaksesan Terakhir pada File

Untuk mengetahui waktu terakhir suatu file diakses, kita bisa memanggil fungsi bernama `fileatime`. Sintaksnya:

```
fileatime(namafile)
```

Nilai balik fungsi ini berupa jumlah detik semenjak tanggal 1 Januari 1970 atau FALSE kalau terjadi kegagalan. Contoh skripnya:

```
<?
$file="order.txt";
$waktu_akses=fileatime($file);
echo ("terakhir kali diakses : ", date("d F Y, h:m",
$waktu_akses));
?>
```

Keluarannya di browser akan sebagai berikut:
terakhir kali diakses : 09 December 2009, 11:12

Studi Kasus 3 : Counter Pengunjung

Kita dapat membuat counter pengunjung untuk mengetahui jumlah terakhir orang yang mengakses halaman web kita dengan mencatatnya di sebuah file. Setiap kali halaman web diakses, isi dari file tersebut dinaikkan sebesar satu. Berikut adalah conoth skrip lengkapnya:

```
<?
```

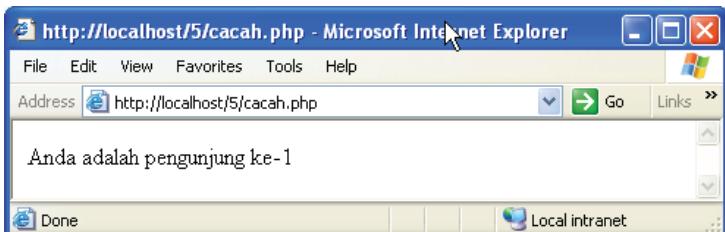
```
//Periksa file
if (file_exists("counter.txt")){
    $pegangan=fopen("counter.txt", "r");
    $cacah=fgets($pegangan,255);
    $cacah++;
    fclose($pegangan);
} else
    //simpan pencacah
    $cacah=1;
    $pegangan=fopen("counter.txt","w");
    fputs($pegangan,$cacah);
    fclose($pegangan);

//tulis ke halaman web
echo "Anda adalah pengunjung ke-$cacah<br>";
?>
```

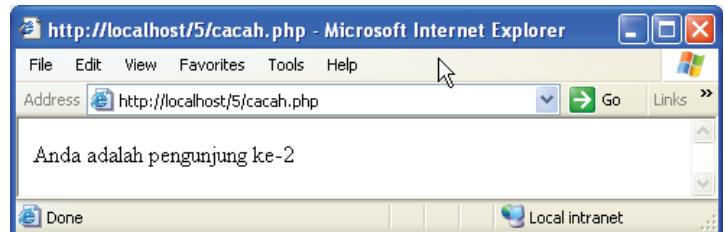
Kode 3 : cacah.php – Skrip untuk menghitung jumlah pengunjung halaman web

Pada skrip di atas, file yang digunakan untuk menyimpan data kunjungan adalah `counter.txt`. Mula-mula file diperiksa (melalui fungsi `file_exists`). Jika ada, maka berkas dibuka dengan mode “hanya baca” (parameter `r` pada fungsi `fopen`). Kemudian isinya dibaca melalui fungsi `fgets` dan disimpan ke variable `$cacah`. Lalu isi variable ini dinaikkan sebesar satu dan file segera ditutup (melalui `fclose`). Kalau file tidak ada, variable `$cacah` diisi dengan 1.

Pada tahap berikutnya, isi file tersebut dibuka dengan mode tulis(akibatnya isinya akan terhapus). Lalu isi varibale `$cacah` direkam dan kemudian file ditutup.



Gb.5-6 Awal pencacahan



Gb.5-7 Cacahan naik sebesar 1 setelah browser reload

Pilihan Terbaik Dalam Menyimpan Data

Sampai sejauh ini setidaknya kita telah memahami konsep dasar pemrosesan data dalam suatu file. Namun ada beberapa masalah yang perlu diketahui diantaranya:

1. Jika filenya bertambah besar, maka waktu pengaksesannya pun akan semakin lama dan berat
2. Proses pencarian suatu groups data tertentu akan semakin sulit jika data yang ditampung sangat banyak
3. Jika pengakses file begitu banyak, maka data hasil proses submiting kemungkinan akan saling tumpang tindih. Meskipun filenya dikunci, tetap saja akan ada kemungkinan terjadi bottleneck

4. Sulit sekali untuk dapat mengubah ataupun menghapus satu records data

Bagaimana sih untuk menghandle berabgai keterbatasan ini? Mau tahu...? Ya kuncinya adalah penggunaan database. Dengan database (kita akan gunakan MySQL) maka problem di atas akan dapat segera diatasi dengan baik. Karena:

- Database memberikan kecepatan dalam mengakses data
- Database memiliki intruksi untuk memilih berbagai kriteria data yang diinginkan
- Tidak akan terjadi data yang saling tumpang tindih dan tidak akan terjadi bottleneck
- Proses updating data begitu mudah dilakukan

Kita akan membahas database pada bab yang akan datang. Be patient OK!

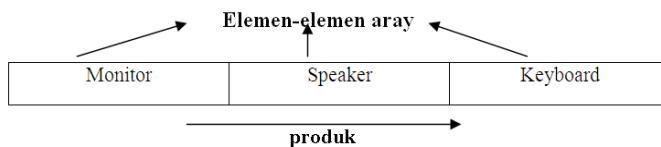
Bab 6

Array

Pada kali ini akan diperkenalkan salah satu aspek yang cukup penting dalam pemrograman PHP yakni **Array**. **Variabale** yang biasa kita gunakan sebelumnya adalah variable skalar yang hanya bisa menyimpan satu nilai saja. Sedangkan array adalah suatu variable yang dapat menyimpan sekumpulan atau serentetan nilai.

Gambaran Array

Sebagai contoh, kita bisa menyimpan sejumlah produk di dalam sebuah variable array \$produk.



Gb.1 Gambaran Array

Dalam terminologi array, array produk di atas mempunyai 3 buah elemen. Setiap elemen berisi 3 buah nilai. Elemen pertama berisi string “Monitor”, elemen kedua berisi string “Speaker”, dan elemen ketiga berisi string “Keyboard”.

Menciptakan Array

Suatu array dapat diciptakan dengan menggunakan konstruksi array. Sebagai contoh, array yang berisi 3 buah produk di atas dapat dibentuk dengan menggunakan pernyataan sebagai berikut:

```
$produk=array("Monitor","Speaker","Keyboard");
```

Alternatif lain, kita bisa menggunakan sederetan pernyataan berikut:

```
$produk[0] = "Monitor";  
$produk[1] = "Speaker";  
$produk[2] = "Keyboard";
```

Mengambil Isi Array

Untuk mengambil isi array, kita bisa menggunakan notasi:

```
$variable[indeks]
```

Contoh:

```
echo $produk[0];
```

Akan menampilkan isi elemen pertama pada array \$produk. Berikut adalah skrip lengkapnya:

```
<?
$produk=array("Monitor","Speaker","Keyboard");
echo "Elemen berindeks 0: $produk[0]<br>";
echo "Elemen berindeks 1: $produk[1]<br>";
echo "Elemen berindeks 2: $produk[2]<br>";
?>
```

Cara penulisan di atas adalah cara manual, untuk lebih canggihnya bisa digunakan loop `for`. Berikut adalah skripnya:

```
<?
$produk=array("Monitor","Speaker","Keyboard");
for ($i=0; $i<3; $i++){
echo "Elemen berindeks $i: $produk[$i]<br>";
}
?>
```

Kedua skrip diatas hasilnya adalah sebagai berikut:

```
Elemen berindeks 0: Monitor
Elemen berindeks 1: Speaker
Elemen berindeks 2: Keyboard
```

Array Asosiatif

PHP juga mendukung array yang bersifat asosiatif. Dalam suatu array asosiatif, kita dapat mengasosiasikan setiap indeks/key yang kita inginkan dengan nilainya. Skrip berikut adalah contoh inisialisasi array asosiatif dengan nama produk sebagai indeks/key dan harga sebagai nilainya.

```
$harga = array("Monitor"=>650000, "Speaker"=>60000,
"Keyboard"=>25000);
```

Bagian sebelum tanda => disebut key, sedangkan bagian setelah => disebut value.

Variable array asosiatif di atas equivalen dengan:

```
$harga["Monitor"] =650000;  
$harga["Speaker"] =60000;  
$harga["Keyboard"] =25000;
```

Berikut adalah contoh skripnya:

```
<?  
$harga = array("Monitor"=>650000, "Speaker"=>60000,  
"Keyboard"=>25000);  
echo "Harga Monitor: $harga[Monitor]<br>";  
echo "Harga Speaker: $harga[Speaker]<br>";  
echo "Harga Keyboard: $harga[Keyboard]<br>";  
?>
```

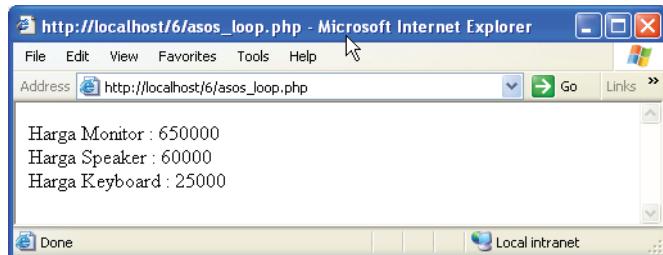
Kode 1: asos.php – Mengakses nilai array asosiatif dengan cara manual

Cara canggih untuk mengakses key dan value dalam array asosiatif, dapat dipakai fungsi list() dan each(). Berikut adalah skripnya:

```
<?  
$harga = array("Monitor"=>650000, "Speaker"=>60000,  
"Keyboard"=>25000);  
  
while (list($key,$value) = each($harga)) {  
    echo "Harga $key : $value <br>";  
    //key = nama produk dan value = Harga produk  
}  
?>
```

Kode 2: asos_loop.php – Mengakses nilai array asosiatif dengan loop

Fungsi list() digunakan untuk memisahkan sebuah elemen array menjadi sejumlah nilai key dan valuenya, sedangkan fungsi each() digunakan untuk mengambil key dan value dari element array. Karena kita memanggil each() dalam loop while, maka ia terus menerus mengembalikan key dan value setiap elemen array dan berhenti ketika posisi akhir array tercapai.



Gb.6-2 Tampilan halaman asos_loop.php

Array Mutidimensi

Array tidak harus selalu sesederhana dengan hanya mempunyai key dan value doang. Setiap lokasi dalam array dapat menampung array yang lainnya. Dengan cara inilah kita dapat membuat array 2 dimensi. Anda dapat membayangkan sebuah array 2 dimensi seperti sebuah matrix yang memiliki baris dan kolom.

Jika kita ingin menyimpan lebih banyak data tentang produk tersebut, maka kita dapat menggunakan array 2 dimensi. Pada gambar berikut **mempelihatkan** aray 2 dimensi dimana setiap baris menggambarkan produk dan masing-masing kolom menggambarkan atribut produk.

Atribut Produk		
Kode	Deskripsi	Harga
MON	Monitor	650000
SPK	Speaker	60000
KEY	Keyboard	25000

Skrip PHPnya untuk gambaran array di atas adalah sebagai berikut:

```
$produk = array( array( "MON", "Monitor", 650000 ),
                  array( "SPK", "Speaker", 60000 ),
                  array( "KEY", "Keyboard", 25000 )
);
```

Untuk menampilkan array di atas skrip manualnya adalah sebagai berikut:

```
echo      " | ".$produk[0][0]." | ".$produk[0]
[1].".|$produk[0][2]." |<BR>";
echo      " | ".$produk[1][0]." | ".$produk[1]
[1].".|$produk[1][2]." |<BR>";
echo      " | ".$produk[2][0]." | ".$produk[2]
[1].".|$produk[2][2]." |<BR>;
```

Cara alternatif lain yang lebih canggih adalah dengan menggunakan loop for dua kali. Satu kali untuk kolom dan satu kali untuk baris. Berikut skripnya:

```
for ( $baris = 0; $baris < 3; $baris++ ) {
    for ( $kolom = 0; $kolom < 3; $kolom++ ) {
        echo " | ".$produk[$baris][$kolom];
    }
    echo " |<BR>";
}
```

Dua cara di atas menghasilkan tampilan sebagai berikut:

	M	O	N		M	o	n	i	t	o	r		6	5	0	0	0	0	
	S	P	K		S	p	e	a	k	e	r		6	0	0	0	0	0	
KEY Keyboard 25000																			

Selain menggunakan nomor indeks, untuk mengakses isi array kita bisa langsung mencantumkan nama kolom/atributnya. Untuk membuatnya ([seperti pada Gb.x](#)) dapat digunakan array asosiatif. Kodennya adalah sebagai berikut:

```
$produk = array( array( "Kode" => "MON",
                        "Deskripsi" => "Monitor",
                        "Harga" => 650000
                    ),
                    array( "Kode" => "SPK",
                        "Deskripsi" => "Speaker",
                        "Harga" => 60000
                    ),
                    array( "Kode" => "KEY",
                        "Deskripsi" => "Keyboard",
                        "Harga" => 25000
                    )
    );
```

Untuk dapat mengaksesnya dapat digunakan loop **for** berikut:

```
for ( $baris = 0; $baris < 3; $baris++ ) {
    echo    "|". $produk[$baris]["Kode"] .    "|".
    $produk[$baris] ["Deskripsi"] .
    "|". $produk[$baris] ["Harga"] . "<BR>";
}
```

Berikutnya Anda mungkin bertanya, bagaimana sih caranya untuk mengakses array yang tidak simetris (sama)

baris dan kolomnya? Misal kita mempunyai data merk sebagai berikut :

- Monitor : LG, Sony, Samsung
- Speaker : Simbada, Altec
- Keyboard : Ace, BenQ, Logitech, Creative

Inisialisasi array untuk merk di atas adalah sebagai berikut:

```
$merk = array (
"Monitor" => array("LG", "Sony", "Samsung"),
"Speaker" => array("Simbada", "Altec"),
"Keyboard" => array("Ace", "BenQ", "Logitech", "Creative")
)
```

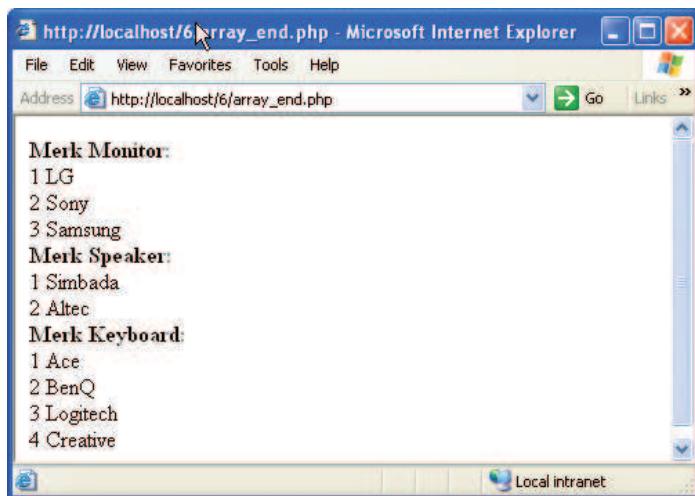
Untuk mengakses isi array di atas secara manual:

```
echo $merk["Monitor"][0]."<br>"; //akan menghasilkan
LG
echo $merk["Speaker"][1]."<br>"; //akan menghasilkan
Altec
echo $merk["Keyboard"][3]."<br>";//akan menghasilkan
Creative
```

Untuk menampilkan isi array di atas dengan canggih dapat digunakan fungsi `list()` dan `each()` :

```
while (list($indeks1,$nilai1) = each($merk)){
    echo "Merk $indeks1 : <br>:";
    $nomor =1;
    while(list($indeks2, $nilai2) = each($nilai1)){
        echo "$nomor $nilai2 <br>";
        $nomor++;
    }
}
```

Hasil tampilannya adalah :



Gb.6-3 : Hasil pengaksesan array untuk merk

Mengurutkan Array

PHP menyediakan sejumlah fungsi yang terkait dengan pengurutan array. Diantaranya:

Fungsi	Keterangan
sort	Mengurutkan data pada array dengan urutan naik (ascending)
asort	Mengurutkan data aray dengan pola naik (ascending) dengan tetap mempertahankan indeksnya
arsort	Mengurutkan data pada array secara menurun (descending) dengan tetap mempertahankan indeksnya
ksort	Mengurutkan array berdasarkan indeksnya dengan urutan turun (descending)
krsort	Mengurutkan array berdasarkan indeksnya dengan urutan naik (ascending)
shuffle	Mengacak array

Berikut adalah contoh skripnya:

```
<?
$harga=array("Speaker"=>60000, "Monitor"=>650000,"Ke
yboard"=>25000,"CPU"=>900000);

//urutan data asli
echo "Urutan Data Asli:<br>";
while (list($index,$nilai) = each($harga)) {
    echo "[{$index}] -> $nilai <br>";
}

//pengurutan dgn sort
echo "<br>Pengurutan dgn sort:<br>";
$harga_urut=$harga;
sort($harga_urut);
while (list($index,$nilai) = each($harga_urut)) {
    echo "[{$index}] -> $nilai <br>";
}

//pengurutan dgn asort
echo "<br>Pengurutan dgn asort:<br>";
$harga_urut=$harga;
asort($harga_urut);
while (list($index,$nilai) = each($harga_urut)) {
    echo "[{$index}] -> $nilai <br>";
}

//pengurutan dgn ksort
echo "<br>Pengurutan dgn ksort:<br>";
$harga_urut=$harga;
ksort($harga_urut);
while (list($index,$nilai) = each($harga_urut)) {
    echo "[{$index}] -> $nilai <br>";
}

//pengurutan dgn krsort
echo "<br>Pengurutan dgn krsort:<br>";
$harga_urut=$harga;
krsort($harga_urut);
while (list($index,$nilai) = each($harga_urut)) {
    echo "[{$index}] -> $nilai <br>";
}

//pengurutan dgn shuffle
echo "<br>Pengurutan dgn shuffle:<br>";
$harga_urut=$harga;
```

■ Aplikasi Berbasis Web dengan PHP & MySQL *Learning By Doing and Make it Simple!*

```
shuffle($harga_urut);
while (list($index,$nilai) = each($harga_urut)) {
    echo "[ $index ] -> $nilai <br>";
}
?>
```

Kode 3: urut.php – Contoh penerapan fungsi-fungsi pengurutan array

Urutan Data Asli:
[Speaker] -> 60000
[Monitor] -> 650000
[Keyboard] -> 25000
[CPU] -> 900000

Pengurutan dgnsort:
[0] -> 25000
[1] -> 60000
[2] -> 650000
[3] -> 900000

Pengurutan dgnsort:
[Keyboard] -> 25000
[Speaker] -> 60000
[Monitor] -> 650000
[CPU] -> 900000

Pengurutan dgnsort:
[CPU] -> 900000
[Keyboard] -> 25000
[Monitor] -> 650000
[Speaker] -> 60000

Pengurutan dgnsort:
[Speaker] -> 60000
[Monitor] -> 650000
[Keyboard] -> 25000
[CPU] -> 900000

Pengurutan dgnsort:
[0] -> 650000
[1] -> 25000
[2] -> 900000
[3] -> 60000

Gb.6-4 Tampil hasil pengurutan array

Studi Kasus 1 : Menampilkan Gambar Produk Secara Acak

Misal kita ingin menampilkan daftar gambar produk, namun supaya pengunjung tidak bosan kita ingin gambarnya berganti secara acak apabila browser direload. Dengan demikian pengunjung pun tidak akan bosan melihat website kita. Pengunjung akan melihat 3 produk yang berbeda

setiap kali dia berkunjung. Berikut adalah skrip lengkapnya:

```
<?php
$gambar=array("monitor.jpg","cpu.jpg","cd.
jpg","hardisk.jpg","keyboard.jpg", "printer.
jpg","scanner.jpg");
shuffle($gambar);
?>

<html>
<head><title>PALASARI Computer Shop</title></head>

<body>
<center>
<h1>PALASARI Hot Product!</h1>
<tabel width = 100%><tr>
<?php
for ( $i = 0; $i < 3; $i++ ) {
    echo "<td align = center><img src=\"$gambar[$i].\"></
td>";
}
?>
</tr></tabel>
</center>
</body>
</html>
```

Kode 4: gambar.php – Kode untuk menampilkan gambar secara acak



Gb.6-5 Tampilan halaman gambar.php



Gb.6-6 Tampilan halaman setelah direload, menampilkan gambar yang berbeda

Studi Kasus 2: Menampilkan Array dari File

Pada bab sebelumnya, kita telah menyimpan order pelanggan dalam sebuah file. Setiap baris data di dalam file order.txt merepresentasikan data setiap kali order disubmit. Berikut adalah salah satu contoh data yang masuk:

```
Sarini arifah Tarogong 11 Garut 1 Monitor 2  
Speaker 1 Keyboard Rp.874500 10:12:12 09-  
Dec-2003
```

Untuk memproses order ini, kita dapat menampilkannya kembali dalam bentuk array. Berikut ini adalah skrip sederhananya:

```
<?php  
$order = file("order.txt");  
$jumlah_order = count($order);  
if ($jumlah_order == 0) {  
    echo "<p><strong>Belum ada order yang datang!</strong></p>";  
}  
for ($i=0; $i<$jumlah_order; $i++) {  
    echo $order[$i]."<br>";  
}
```

?>

Kode 5: orderan.php – Menampilkan data order.txt ke dalam browser



Gb.6-7 Hasil pengaksesan data order.txt dengan manipulasi array. Bentuknya masih sederhana

Kita bisa menampilkan data order diatas dengan cantik yakni apabila kita dapat menklasifikasi dalam attribut-atribut kolom, misal nama, alamat, Monitor, Speaker, Keyboard, Total, dan waktu. Dengan demikian pembacaannya akan menjadi lebih mudah. Berikut adalah skrip modifikasinya:

```
<html>
<head><title>PALASARI Computer Shop - Order Pelanggan</title></head>

<body>
<h1>PHP Computer Shop</h1>
<h2>Order Pelanggan</h2>
<?php
    //Baca seluruh isi file. Setiap baris order dijadikan elemen array
    $order= file("order.txt");
    // Hitung jumlah order dalam array
    $jumlah_order = count($order);
    if ($jumlah_order == 0)
    {
        echo "<p><strong>No Belum ada order yang datang!</strong></p>";
    }
    echo "<table border=1 cellspacing=0 cellpadding=3>\n";
    echo "<tr bgcolor=\"#CCCCFF\">
        <th>Nama</th>
        <th>Alamat</th>
```

```
<th>Monitor</th>
<th>Speaker</th>
<th>Keyboard</th>
<th>Total</th>
<th>Waktu</th>
<tr>";
for ($i=0; $i<$jumlah_order; $i++)
{
    //Explode memisahkan data yg mengandung spasi
    //horizontal (tab) di setiap baris
    $data = explode( "\t", $order[$i] );
    // Hanya mengambil bilangannya saja
    $data[2] = intval( $data[2] ); // untuk monitor
    $data[3] = intval( $data[3] ); //untuk speaker
    $data[4] = intval( $data[4] ); //untuk keyboard

    // keluaran setiap order
    echo "<tr><td>$data[0]</td> //nama
          <td align = right>$data[1]</td> //alamat
          <td align = right>$data[2]</td> //
monitor
          <td align = right>$data[3]</td> //
speaker
          <td align = right>$data[4]</td> //
keyboard
          <td>$data[5]</td> //Rp. Total
          <td>$data[6]</td> //Waktu

        </tr>";
}
echo "</table>";

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

Kode 6: order_lihat.php – Mengakses data dalam order.txt dengan tampilan yang lebih menarik

Nama	Alamat	Monitor	Speaker	Keyboard	Total	Waktu
Sarini arifah	Tarogong 11 Garut	1	2	1	Rp.874500	10:12:12 09-Dec-2003
Guntur Saputra	Jl. Babakan Ciparay 100 Bandung	2	1	2	Rp.1551000	03:04:36 22-Apr-2010

Gb.6-8 Tampilan data order.txt sekarang menjadi lebih rapi

Fungsi `explode` berguna untuk menciptakan array berdasarkan suatu string. Bentuk pemanggilannya
`explode(string pemisah, string hasil)`

Hasilnya berupa array dengan masing-masing elemen berupa substring dari parameter kedua. Dalam hal ini parameter `pemisah` digunakan sebagai basis pemecahan string menjadi substring. Sebelumnya ketika menyimpan data order, kita menggunakan karakter tab (`\t`) sebagai pembatas di setiap baris data. Kemudian kita menggunakan `explode("\t","order[$i]")` dimana efeknya akan meledakkan(`explode`) string yang dipisahkan oleh karakter tab menjadi elemen array yang terpisah.

Sebagai contoh, string:

“Sarini arifah\t Tarogong 11 Garut\t 1 Monitor\t 2 Speaker\t 1 Keyboard\t Rp.874500\t 10:12:12 09-Dec-2003”

akan diexplode menjadi bagian-bagian string-string yang terpisah, yakni :

“Sarini arifah”, “Tarogong 11 Garut”, “1 Monitor”, “2 Speaker”, “1 Keyboard”, “Rp.874500” dan “10:12:12 09-Dec-2003”

Fungsi `intval()` digunakan untuk mengekstrak bilangan integer dari string yang. Sebagai contoh string : “1 Monitor” setelah di-`intval()` akan menghasilkan **1**.

Pada bab ini setidaknya kita mempunyai gambaran dalam penggunaan array dan aplikasinya. Bab berikutnya kita akan membahas fungsi-fungsi yang berkaitan dengan pemrosesan string, diantaranya fungsi yang berkaitan dengan pencarian, penggantian, pemisahan dan penggabungan string. Juga akan dibahas fungsi-fungsi ekspresi regular yang dapat melakukan hampir semua operasi validasi pada string.

Bab 7

Manipulasi String dan Ekspresi Reguler

Pada bab ini kita akan membahas bagaimana penggunaan fungsi-fungsi string yang ada di PHP untuk memformat dan memanipulasi teks. Kita pun akan membahas penggunaan fungsi-fungsi string atau fungsi-fungsi ekspresi regular untuk mencari (dan mengganti) kata, frase atau pola lain yang ada di dalam string.

Fungsi-fungsi ini sangatlah berguna dalam berbagai keadaan. Anda pasti akan sering memfilter data input yang user masukkan sehingga data tersebut benar-benar valid dan siap dimasukkan ke dalam database.

Studi Kasus 1: Form Pengiriman Email

Kali ini kita akan membuat aplikasi berupa form pengiriman email yang berfungsi sebagai sarana feed-

back pelanggan untuk toko komputer PHP. Di form ini akan dilakukan pemeriksaan dari data yang dimasukan oleh pelanggan sehingga data tersebut bisa benar-benar valid. Setelah semuanya valid, maka data akan dikirimkan ke alamat email yang dituju, misal marketing@palasari.com. Ok mari kita mulai penulisan skripnya:

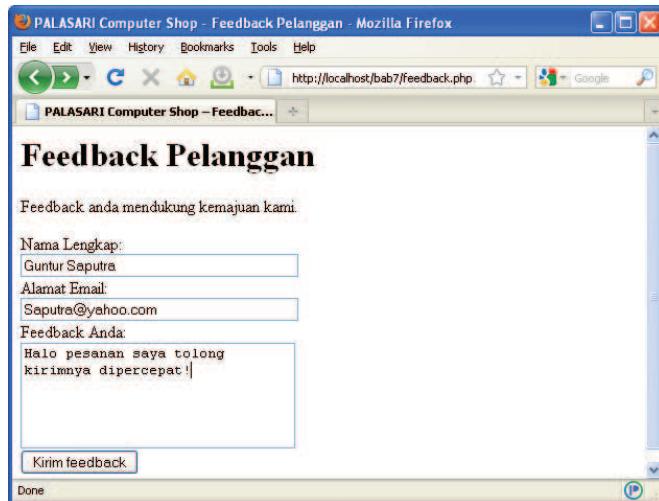
```
<html>
<head>
    <title>PALASARI Computer Shop - Feedback Pelang-
gan</title>
</head>
<body>

<h1>Feedback Pelanggan</h1>

<p>Feedback anda mendukung kemajuan kami.</p>

<form method=post action="feedback_proses.php">
Nama Lengkap: <br><input type=text name="nama"
size=40><br>
Alamat Email: <br><input type=text name="email"
size=40><br>
Feedback Anda:<br><textarea name="feedback" rows=5
cols=30></textarea><br>
<input type=submit value="Kirim feedback">
</form>
</body>
```

Kode 15.: feedback.php – Form isian feedback pelanggan



Gb.7-1 Tampilan halaman feedback.php

Setelah disubmit, form di atas akan melakukan proses action ke halaman `feedback_proses.php`. Berikutnya adalah kodennya:

```
<?php
$tujuan = "marketing@palasari.com";
$subjek = "Feedback dari website PALASARI";
$isi_email = "Nama Pelanggan: ".$_POST[nama]."\n"
."Email Pelanggan: ".$_POST[email]."\n"
."Komentar Pelanggan: \n".$_POST[feedback]."\n";
$dari = "noreply@palasari.com";

//mengirim email dengan fungsi mail()
mail($tujuan, $subjek, $isi_email, $dari);
?>

<html>
<head>
<title>PHP Computer Shop - Feedback Telah Dikirim</title>
</head>
<body>
<h1>Terima Kasih</h1>
```

```
<p>Feedback Anda telah dikirim...</p>
</body>
</html>
```

Kode 7.2: feedback_proses.php – Kode yang digunakan untuk memproses action dari form

Kode 7.2 di atas akan melakukan pengiriman data via email dari form ke alamat tujuan. Fungsi yang digunakan untuk mengirim email ini adalah `mail()` yang bersifat boolean (return/nilai kembalinya TRUE atau FALSE).

Standard penulisannya:

```
bool mail(string ke, string subjek, string pesan,
string [header_tambahan]);
```

Tiga parameter pertama merepresentasikan alamat tujuan email, subjek email, isi pesannya. Parameter yang keempat dapat digunakan untuk mengirim header email tambahan. Disini kita dapat menggunakan parameter keempat untuk menambahkan alamat pengirim “From:”.

Kita pun bisa menambahkan field “Reply to:” dan “Cc:”. Jika kita ingin menambahkan lebih dari satu header tambahan, tinggal pisahkan saja dengan menggunakan string newline (\n). Contohnya bisa seperti ini:

```
$header_tambahan="From: noreply@palasari.com\n Reply-to: faisal@palasari.com";
```

Keseluruhan bab ini kita akan meningkatkan kemampuan skrip pengiriman email di atas dengan memakai penanganan string dan fungsi ekspresi reguler.

Catatan: Untuk menghilangkan pesan warning webserver di proses pengiriman email, maka kita dapat menambah-

kan karakter ‘@’ di depan function email tersebut sehingga menjadi

```
@mail($tujuan, $subjek, $isi_email, $dari, $header_tambahan);
```

Memformat String

PHP menyediakan sekumpulan fungsi-fungsi yang dapat digunakan untuk memformat ulang string dengan berbagai cara.

a. Menghapus Spasi Putih di String : trim()

Adakalanya diperlukan penghapusan karakter spasi putih (*whitespace*) yang berada di awal atau di belakang string. Meskipun kelihatannya sepele namun fungsi ini berguna apabila kita ingin menyimpan string ke dalam database ataupun ke file. Contoh penggunaannya :

```
$nama=trim($nama);
$email=trim($email);
$feedback=trim($feedback);
```

Yang termasuk spasi putih di antaranya:

- “ “ : spasi
- “\n” : baris baru
- “\t” : tab horizontal
- “\v” : tab vertikal

Contoh pemanggilan fungsi dan nilai baliknya(hasilnya) :

Pemanggilan fungsi	Nilai Balik
Trim("Guntur Saputra")	“Guntur Saputra”
Trim (“Guntur Saputra“)	“Guntur Saputra”
Trim (“Guntur Saputra “)	“Guntur Saputra”

b. Memformat string untuk penampilan

1. Newline to break : nl2br()

Fungsi nl2br() mengambil string sebagai parameter dan mengganti semua string newline("\n") dengan tag html
. Hal ini berguna untuk menampilkan string yangpanjang ke browser. Sebagai contoh kita menggunakan fungsi ini untuk memformat isi feedback pelanggan dan menampilkannya kembali di halaman feedback_proses.php. Berikut kodennya:

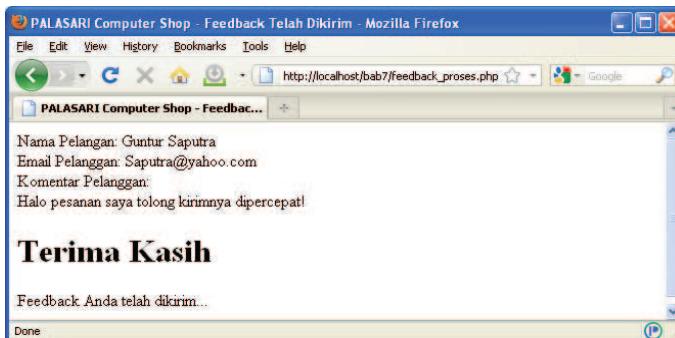
```
<?php
$tujuan = "marketing@palasari.com";
$subjek = "Feedback dari website PALASARI";
$isi_email = "Nama Pelangan: ".$_POST[nama]."\n"
."Email Pelanggan: ".$_POST[email]."\n"
."Komentar Pelanggan: \n".$_POST[feedback]."\n";
$dari = "noreply@palasari.com";

//mengirim email dengan fungsi mail()
@mail($tujuan, $subjek, $isi_email, $dari);

echo nl2br($isi_email);

?>

<html>
<head>
<title>PALASARI Computer Shop - Feedback Telah Dikirim</title>
</head>
<body>
<h1>Terima Kasih</h1>
<p>Feedback Anda telah dikirim...</p>
</body>
</html>
```



Gb.7-2 Tampilan hasil penambahan fungsi nl2br()

2. Mengubah huruf string: Huruf Kapital atau Huruf Kecil

Kita dapat pula memformat ulang huruf string sesuai dengan keinginan kita. Misal kita ingin memformat ulang string yang dikirimkan oleh variable \$subjek. Berikut ini contoh penggunaan beberapa fungsi string yang efeknya terlihat pada tabel sebagai berikut:

Fungsi	Deskripsi	Penggunaan	Nilai Balik
		\$subjek	feedback dari website toko PHP
Strtoupper()	Mengubah semua huruf string menjadi kapital		FEEDBACK DARI WEBSITE TOKO PHP
	Mengubah semua huruf string menjadi huruf kecil		feedback dari website toko php
Ucfirst()	Mengubah huruf pertama string menjadi huruf besar		Feedback dari toko PHP

Ucwords ()	M e n g u b a h huruf pertama dari setiap kata menjadi huruf kapital		Feedback Dari Website Toko PHP
------------	--	--	--------------------------------

Menggabungkan dan Memisahkan String : Explode() & Implode()

Sesekali pasti kita perlu mengidentifikasi suatu kandungan string dan membaginya menjadi beberapa bagian. Jika alamat email pelanggan berasal dari suatu perusahaan besar misal **microsoft.com**, maka email tersebut dikirim ke alamat email direktur@palasari.com, selain dari itu akan dikirimkan ke marketing@palasari.com.

Untuk mendapatkan nama domain dari alamat email pelanggan, kita bisa menggunakan fungsi `explode()`. Prototipe fungsi ini adalah:

```
array explode(string pemisah, string input);
```

Misal alamat email pelanggan adalah bill@microsoft.com, dengan `explode()` kita dapat memisahkan string email ini menjadi dua elemen array dengan "@" sebagai pemisah. Kodenya adalah sebagai berikut:

```
$array_email=explode("@",$email);
```

Hasilnya `$array_email[0]` berisi username bill, sedangkan `$array_email[1]` berisi nama domain microsoft.com. Sekarang kita dapat memeriksa nama domain untuk menentukan darimana asal pelanggan, kemudian mengirim feedbacknya ke tujuan email

yang ditentukan.

```
If ($array_email[1]=="microsoft.com")
    $ke="direktur@palasari.com";
else
    $ke="marketing@palasari.com";
```

Jika di alamat emailnya mengandung huruf kapital, maka skrip diatas tidak akan berjalan. Untuk menangginya, kita harus mengkonversi stringnya ke huruf kecil semua dengan fungsi `strtolower()`. Setelah itu baru diperiksa dengan fungsi if.

```
$array_email[1] = strtolower($array_email[1]);
```

Untuk menggabungkannya kembali, kita dapat menggunakan fungsi `implode()`.

```
$email=implode("@",$array_email);
```

Fungsi ini mengambil elemen array dari `$array_email` kemudian menggabungkannya dengan "@" sebagai penghubung.

Membandingkan string

a. `strcmp` dan `strcasecmp`

Untuk keperluan pembandingan dua **buah** string, PHP menyediakan fungsi `strcmp()` dan `strcasecmp()`. Prototypenya:

```
int strcmp(string string1, string string2)
int strcasecmp(string string1, string string2)
```

Fungsi `strcasecmp` berguna untuk membandingkan

dua buah string dengan perlakuan huruf kecil dan huruf kapital dianggap sama. Nilai balik (return) fungsi ini :

- <0 : Jika string1 lebih kecil dari string2
- =0 : Jika string1 sama dengan string2 (equivalen dengan FALSE)
- >0 : jika string1 lebih besar daripada string2 (equivalen dengan TRUE)

contoh :

```
$str1 = "Hello";
$str2 = "hello";
if (strcasecmp ($var1, $var2)==0) {
    echo "$str1 sama dengan $str2";
} else{
    echo "$str1 tidak sama dengan $str2";
}
```

b. Memeriksa panjang string dengan strlen()

Kita dapat memeriksa panjang karakter suatu string dengan fungsi `strlen()`. Sebagai contoh, `strlen("hallo")` menghasilkan nilai balik 5. Fungsi ini dapat diterapkan untuk memeriksa data yang dimasukkan oleh pelanggan.

Misal kita ingin memeriksa kevalidan data nama :

```
if (strlen($nama) <= 2){
    echo "Nama anda tidak valid";
    exit;
}
```

Mencari String di Dalam String

Untuk mencari string di dalam string, dapat digunakan fungsi `strstr()` ataupun `stristr()`. Prototipnya:

```
string strstr(string jerami, string jarum)
```

Dengan fungsi tersebut diibaratkan kita ingin mencari jarum di dalam jerami. Fungsi ini akan memberikan nilai balik TRUE jika jarum ditemukan dalam jerami, dan sebaliknya FALSE jika tidak temukan. Fungsi `strstr()` bersifat case sensitive sedangkan `stristr()` bersifat case insensitive.

Sebagai contoh, misal di form feedback kita ingin agar isi feedback yang mengandung kata tertentu dikirim ke alamat email tertentu juga. Berikut adalah kodennya:

```
$toaddress = "marketing@palasari.com"; // email default
```

```
// Ubah alamat email tujuan jika kriteria sebagai berikut ditemukan di dalam isi feedback
if (strstr($feedback, "kecewa"))
    $tujuan = "komplain@palasari.com";
else if (strstr($feedback, "pengiriman"))
    $tujuan = "kurir@palasari.com";
else if (strstr($feedback, "pelatihan"))
    $tujuan = "training@palasari.com";
```

Mengenal Ekspresi Reguler (Pencocokan Pola)

Ekspresi reguler merupakan cara dalam mencocokkan pola dari suatu rentetan teks. Pada aplikasi internet, ekspresi reguler digunakan khusus untuk mencek validasi dari input yang dimasukkan user di form. Kita dapat memeriksa validasi pola nomor telepon, nomor kartu kredit, alamat email dan lain sebagainya. Pencocokan yang telah kita lakukan sebelumnya adalah sebuah bentuk dari ekspresi reguler. Sebagai contoh, sebelumnya kita mencari penyataan ekspresi reguler seperti “kecewa” dan “pembayaran”.

Bila dibandingkan dengan perbandingan sama den-

gan (==), pencocokan ekspresi reguler dalam PHP mirip seperti pencocokan melalui `strstr()`, karena kita mencocokan string yang ada dalam suatu string. Sebagai contoh , string “kecewa” cocok dengan ekspresi reguler “kecewa”. Ia juga cocok dengan ekspresi reguler “e”, “ec” dan seterusnya. Kita dapat menggunakan karakter khusus untuk menunjukkan string yang dimaksud dengan tepat.

Dengan karakter khusus anda dapat mengatur apakah polanya harus terdapat pada awal atau akhir string, dimana bagian pola ini dapat diulangi. Berikut akan kita bahas bagaimana penempatan polanya satu persatu.

a. Set Karakter

Penggunaan set karakter memberikan ekspresi reguler lebih powerfull daripada pencocokan secara eksak. Pertama-tama kita akan menggunakan karakter . (titik). Sebagai contoh, ekspresi reguler

.an

akan cocok (true) dengan string “ban”, “san”, “dan”, “kan”, #an’ dan seterusnya. Jenis pencocokan ini banyak digunakan untuk pencocokan nama file di sistem operasi.

Kita dapat menggunakan karakter penghubung (-) untuk mencocokan range dari karakter. Contohnya bisa dilihat sebagai berikut:

```
[a-z] // mencocokan huruf kecil apapun  
[A-Z] //mencocokkan huruf besar apapun  
[a-zA-Z] // mencocokan huruf apapun baik besar maupun kecil  
[0-9] //mencocokan angka apapun  
[AaEeIiOoUu] //mencocokan huruf vokal apapun  
[0-9\.\-] // mencocokan angka apapun, titik atau tan-
```

da minus.

[^a-z] // mencocokan huruf apapun kecuali huruf kecil (hanya huruf besar saja)

Hati-hati, masing-masing dari kelas di atas digunakan untuk mencocokan **satu** karakter saja. Tanda *topi* ^ di dalam kurung menyatakan *bukan anggota*. Namun jika tanda ^ berada di luar kurung, ia mempunyai arti awalan.

Misal :

^[ab] // Mencocokan string yang berawalan "ab" seperti "abri", "abu", "abis", dst.

Tanda \$ digunakan untuk mencocokan akhiran. Misal:

[n]\$ //Mencocokan string yang berakhiran "n" spt "kawan", "dan" , "awan", dst.

So jika anda ingin mencocokan string seperti "a2" , "b2" atau "g7" bukan "ab2", "r2d2" atau "b52", anda harus menuliskan pola seperti ini :

^[a-z][0-9]\$

Di bawah ini adalah beberapa kelas karakter ekspresi regular bawaan PHP:

Kelas Karakter	Deskripsi
[:alnum:]	Alphanumeric characters
[:alpha:]	Alphabetic characters
[:lower:]	Lowercase letters
[:upper:]	Uppercase letters
[:digit:]	Decimal digits
[:blank:]	Tabs and spaces
[:space:]	Whitespace characters

b. Repetisi

Tanda kurung kurawal {} digunakan untuk mencocokan repetisi/pengulangan yang banyak. Misal:

```
^a{3}$ // mencocokan string "aaa"  
^a{2,4}$ // mencocokan string aa, aaa, atau aaaa  
^a{2} // mencocokan string yang dimulai dengan "aa" ,  
misal "aagym", "aauuu", dst.
```

Tanda + digunakan untuk pencocokan: *setidaknya satu atau lebih*. Misal:

```
[[:alnum:]]+ // mencocokan setidaknya satu karakter  
aphanumerik
```

Sedangkan tanda * digunakan untuk pencocokan: *setidaknya nol atau lebih*.

c. Subekspresi

Seringkali berguna untuk memisahkan ekspresi menjadi subekspresi sehingga anda dapat misal mengambarkan "*setidaknya satu dari string ini tepat diikuti satu atau banyak*". Anda dapat melakukan ini dengan menggunakan tanda kurung (). Sebagai contoh:

```
(sangat) *besar // mencocokan "besar", "sangat besar", "sangat sangat besar", dst.
```

d. Pencabangan

Anda dapat menyatakan suatu pilihan dalam sebuah ekspresi reguler dengan sebuah pipa vertikal |. Misal kita ingin mencocokan domain com, edu atau net, kita dapat menggunakan ekspresi:

```
(com) | (edu) | (net) //mencocokan string "com" atau  
"edu" atau "net"
```

e. Mencocokan karakter literal khusus

Jika Anda ingin mencocokkan salah satu karakter khusus seperti : ., {, atau \$, anda harus memberikan sebuah slash (\) di depannya. Jika anda ingin mencocokkan slash, maka tinggal tambahkan satu slash lagi yakni \\.

Menerapkan Ekspresi Reguler di Form Feedback

Kita akan mencoba menvalidasi pola alamat email dan isi feedback pelanggan yang diinput dgn menggunakan ekspresi reguler. Fungsi yang digunakan untuk mencocokkan pola ekspresi reguler adalah `ereg()` dan `eregi()`. Prototipenya:

```
int ereg(string pola, string sumber, array_pola  
[posisi_cocok]);
```

Fungsi ini mencari string dari *sumber* dengan berusaha untuk mencocokan berdasarkan string *pola*. Argument ketiga dari fungsi di atas adalah opsional, boleh ada boleh tidak. Fungsi `eregi()` sama seperti `ereg()`, namun `eregi()` bersifat tidak sensitive (tidak membedakan huruf besar/kecil).

Nilai kembalian dari kedua fungsi ini adalah integer positif (true) yakni jika pola berhasil di temukan. Namun jika pola tidak ditemukan, ia akan memberikan nilai kembalian nol (false).

Berikut adalah penerapannya di form feedback:

```
if      (!ereg("^[a-zA-Z0-9_]+@[a-zA-Z0-9\-\]+\.[a-zA-  
Z0-9\-\.\.]+\+$", $email)) {  
    echo "Alamat email Anda tidak valid!<br>";  
    echo "Klik tombol Back Untuk Kembali";  
    exit;
```

}

Pola ekspresi reguler:

`^ [a-zA-Z0-9_] + @ [a-zA-Z0-9\-\-] + \. [a-zA-Z0-9\-\.\-] + $`

digunakan untuk mencek validasi alamat email. Misal user-name@namahost.com

Subekspresi `^ [a-zA-Z0-9_]` + memiliki arti :

“awal dari string setidaknya satu huruf, nomer atau underscore _, atau kombinasi dari elemen-elen tersebut”

Simbol @ mencocokkan karakter literal @ yang ada di setiap email.

Subekspresi `[a-zA-Z0-9\-\-]` + mencocokkan bagian pertama dari nama host email termasuk karakter alphanumerik dan karakter penghubung. Sebagai catatan kita melakukan slash (\) untuk mengescape karakter penghubung (-) karena di dalam kurung persegi karakter ini merupakan karakter khusus.

Kombinasi \. Digunakan untuk mencocokan pola . (titik) di email.

Subekspresi `[a-zA-Z0-9\-\.\-]` + \$ mencocokkan nama domain email; termasuk huruf, angka, tanda penghubung dan titik, sampai akhir string.

Kode berikut adalah proses pencarian pola untuk isi dari feedback:

```
if (eregi("komplain | kecewa | marah | keluhan",  
$feedback))  
    $tujuan = "komplain@palasari.com";  
else if (eregi("pengiriman | kirim | terlambat | lam-  
bat", $feedback))  
    $tujuan = "kurir@palasari.com";  
else if (ereg("pelatihan | training | belajar", $feed-
```

```
back))
$tujuan = "training@palasari.com";

if (eregi("microsoft\.com", $email))
$tujuan = "direktur@palasari.com";
```

Skrip di bawah ini adalah skrip final `feedback_proses.php` dengan memakai ekspresi reguler:

```
<?php
//menghapus karakter putih dari form
$nama=trim($_POST[nama]);
$email=trim($_POST[email]);
$feedback=trim($_POST[feedback]);

//cek panjang nama
if (strlen($nama) <=2){
    echo "Nama anda tidak valid. Tekan tombol BACK untuk kembali!";
    exit;
}

//cek pola email yang dimasukkan
if (!ereg("^[a-zA-Z0-9_]+@[a-zA-Z0-9\-\]+\.[a-zA-Z0-9\-\.\.]+\$", $email)){
    echo "Alamat email Anda tidak valid!<br>";
    echo "Klik tombol BACK Untuk Kembali";
    exit;
}

$tujuan = "marketing@palasari.com"; // alamat email default
$subjek = "Feedback dari website toko PHP";
$isi_email = "Nama Pelanggan: ".$nama."\n"
."Email Pelanggan: ".$email."\n"
."Komentar Pelanggan: \n".$feedback."\n";
$dari = "webserver@palasari.com";

//cek isi dari feedback
if (eregi("komplain | kecewa | marah | keluhan",
$feedback))
$tujuan = "komplain@palasari.com";
else if (eregi("pengiriman | kirim | terlambat | lambat",
$feedback))
$tujuan = "kurir@palasari.com";
else if (ereg("pelatihan | training | belajar", $feed-
```

```
back))
$tujuan = "training@palasari.com";

if (eregi("microsoft\.com", $email))
$tujuan = "direktur@palasari.com";

// mengirim email dengan fungsi mail()
@mail($tujuan, $subjek, $isi_email, $dari);

// menampilkan data dari form
echo nl2br($isi_email);

?>

<html>
<head><title>PALASARI Computer Shop - Proses Feed-
back</title></head>
<body>
<h1>Terima Kasih</h1>
<p>Feedback Anda telah dikirim..</p>
</body>
</html>
```

Kode 3 feedback_proses.php – Halaman proses yang lebih canggih dengan memakai ekspresi reguler

Mana Yang lebih baik?

Secara umum, fungsi ekspresi reguler berjalan lebih efisien dibanding fungsi string dengan fungsi yang sama. Jika aplikasi yang anda buat cukup sederhana untuk menggunakan fungsi string, just do it! Banyak jalan menuju Roma. Isn't easy?

Bab 8

Bermain Database MySQL

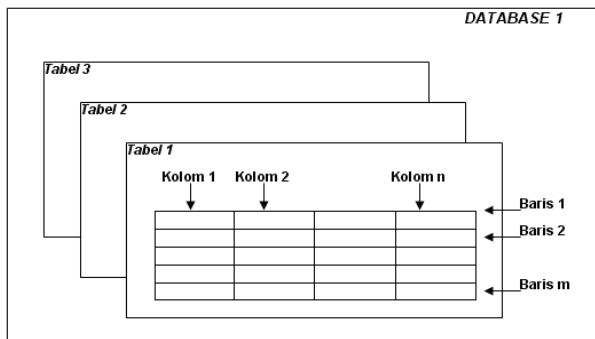
Pada bab terdahulu telah dibahas bagaimana cara menyimpan data pada suatu file. Namun kita merasakan adanya kelemahan jika datanya semakin membesar: bottleneck, lambat dan susah dimanage. Nah solusinya mau tidak mau kita harus menggunakan database agar kelemahan di atas dapat diatasi. Kali ini kita akan menggunakan database MySQL sebagai sarana untuk menyimpan data.



Gb.8-1 Logo Database MySQL

MySQL merupakan database server yang bersifat

multiuser dan multi-threaded. SQL adalah bahasa database standar yang memudahkan penyimpanan, pengubahan dan akses informasi. Pada MySQL dikenal istilah *database* dan *tabel*. Table adalah sebuah struktur data dua dimensi yang terdiri dari baris-baris record dan kolom.



Gb.1 Ilustrasi database dan tabel pada database MySQL

Mengapa Memilih MySQL ?

a. Free alias gratis

Bayangkan jika Anda menggunakan Oracle, sudah siap merogoh kocek antara \$10.000 - \$100.000 atau lebih ? Tidak ada yang meragukan kehebatan Oracle, Sybase, dan Informix, tapi biaya yang harus dikeluarkan bisa menjadi hambatan bagi rakyat kecil seperti kita.

Bandingkan dengan MySQL, gratis abiss eeuuy. Anda bisa menginstall dan memakainya tanpa harus membayar sepeserpun. MySQL merupakan salah satu database open source yang paling banyak digunakan di dunia

■ Aplikasi Berbasis Web dengan PHP & MySQL *Learning By Doing and Make it Simple!*

dan telah berhasil merevolusi teknologi internet dunia.

b. Cepat and Powerful

Untuk ukuran database kecil sampai menengah, MySQL diklaim sangat cepat. Untuk membaca 2.000.000 baris (dengan bantuan index), ternyata MySQL lebih cepat dari Oracle sekalipun.

c. Banyak dukungan

Sekarang sudah banyak bahasa pemrograman dan aplikasi yang mendukung MySQL sebagai solusi databas-enya. Mulai dari PHP, Delphi, ASP.Net, VB.Net, dll.

Memulai MySQL

Karena MySQL yang kita gunakan sudah bersatu dengan paket instalasi dari XAMPP, maka kita dapat melihat status service MySQL itu jalan atau tidak dengan menggunakan XAMPP control panel. Ada di menu **Start > Program > Apache Friends > XAMPP >XAMPP Control Panel**. Pastikan service MySQL-nya running seperti pada gambar 1.8 (bab 1).

Apabila kita melakukan penginstalan MySQL di Windows 2000 atau Windows XP, maka MySQL akan berjalan sebagai service dan kita tidak usah menjalankannya secara manual. Namun jika komputer anda adalah Windows 98, maka Anda harus menjalankan secara manual. Untuk menjalankan MySQL secara manual yakni melalui MS-DOS Prompt, ketikan perintah sebagai berikut:

```
C:\WINDOWS>cd \mysql\bin
```

```
C:\mysql\bin>mysql
```

a. Login ke MySQL

Dengan melakukan login ke MySQL , seluruh operasi pembuatan database, ekseskuasi query dan sebagainya, dapat dilakukan dengan mudah. Untuk login ke MySQL, gunakan MS-DOS Prompt dengan masuk ke menu **Start > Program > Accesories > Command Prompt**. Ketikkan command yang bercetak tebal sebagai berikut

```
Microsoft Windows XP [Version 5.1.2600]  
(C) Copyright 1985-2001 Microsoft Corp.
```

```
C:\Documents and Settings\Administrator>cd c:\xampp\  
mysql\bin  
C:\xampp\mysql\bin> mysql -u root
```

Program MySQL ada di folder **c:\xampp\mysql\bin**. Parameter –u root berarti login ke server menggunakan username root. Jika tidak menggunakan login root, kita tidak bisa melakukan operasi-operasi seperti pembuatan database. Jika login berhasil, pada layar akan tampak seperti berikut:

```
Welcome to the MySQL monitor. Commands end with ;  
or \g.  
Your MySQL connection id is 8  
Server version: 5.0.33 Source distribution  
  
Type 'help;' or '\h' for help. Type '\c' to clear the  
buffer.  
  
mysql>
```

Prompt **mysql>** diatas adalah prompt tempat Anda

■ Aplikasi Berbasis Web dengan PHP & MySQL *Learning By Doing and Make it Simple!*

bekerja.

Untuk logout dari MySQL, gunakan perintah `quit` atau `exit` atau `\q`.

```
mysql>quit  
bye
```

Server MySQL bisa di-shutdown dengan perintah:

```
C:\xampp\mysql\bin> mysqladmin -u root shutdown
```

b. Mempassword Login

Ketika login ke MySQL server pertama kali, tidak diperlukan password, seperti terlihat pada perintah:

```
C:\xampp\mysql\bin> mysql -u root
```

Bagaimana agar setiap kali login dengan user root, diperlukan password, agar tidak sembarang orang dapat login sebagai root ? (Catatan: root adalah user yang berkuasa penuh atas suatu server).

Pertama, lakukan login dengan perintah seperti di atas, tanpa password. Kemudian di prompt mysql, lakukan perintah-perintah berikut:

```
mysql> UPDATE mysql.user SET password=PASSWORD("hallo")  
where user="root" and host="localhost";  
Query OK, 1 row affected (0.00 sec)  
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql> flush privileges;  
Query OK, 0 rows affected (0.00 sec)
```

```
mysql>
```

Hasilnya nanti setiap kali login username `root` harus memasukkan password: `hallo`. Perintahnya :

```
C:\xampp\mysql\bin> mysql -u root -p  
Enter password:
```

Parameter `-p` berarti dalam proses login, kita harus memasukkan password. Jika tanpa `-p`, maka login kita ke MySQL akan ditolak. Berikut adalah contoh lengkap proses login dengan menggunakan password:

```
C:\xampp\mysql\bin> mysql -u root -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ;
or \g.
Your MySQL connection id is 7 to server version:
4.0.12-nt

Type 'help;' or '\h' for help. Type '\c' to clear the
buffer.

mysql>
```

Membuat database

Setelah login berhasil, kita dapat mulai pekerjaan kita di MySQL. Sebelum kita dapat membuat dan bekerja dengan tabel, perlu dibuat database terlebih dahulu. Perintah untuk membuat database baru adalah **CREATE DATABASE :**

```
mysql>CREATE DATABASE nama_database;
Query OK, 1 row affected (0.00 sec)
```

Ok, misal kita ingin membuat database `palasari`, yang nantinya akan disimpan tabel data order, feedback, dan lain-lain. Perintahnya adalah sebagai berikut:

```
mysql>CREATE DATABASE palasari;
Query OK, 1 row affected (0.00 sec)
```

Hasilnya database `palasari` akan terbentuk.

Melihat database: SHOW

Untuk melihat database yang sudah ada dalam server MySQL, dapat dilakukan dengan perintah SHOW DATABASES..

```
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| isal    |
| mysql   |
| palasari |
| test    |
+-----+
4 rows in set (0.00 sec)
```

Memilih database aktif: USE

Sebelum operasi yang berkaitan dengan tabel dilakukan, database yang aktif harus dipilih terlebih dahulu. Untuk memilih database digunakan perintah USE. Misal kita akan membuat tabel di database palasari, maka kita harus mengaktifkan terlebih dahulu database palasari-ya. Berikut perintahnya:

```
mysql> USE palasari;
Database changed
mysql>
```

Membuat Table: CREATE TABLE

Table dapat dibuat dengan perintah CREATE TABLE. Misal sekarang kita ingin membuat tabel untuk menyimpan data feedback dari pelanggan. Nama kolomnya antara lain: no, nama, email, isi. Berikut adalah

perintahnya di MySQL:

```
mysql> CREATE TABLE feedback(
-> no int,
-> nama varchar(40),
-> email varchar(30),
-> isi text
-> );
Query OK, 0 rows affected (0.02 sec)
```

Gambarkan nyata tabel `feedback` yang telah dibuat akan nampak sebagai berikut:

The diagram illustrates a MySQL table structure. It features a grid of 6 columns by 7 rows. The columns are labeled at the top: 'no', 'nama', 'email', and 'isi'. To the right of the grid, two curly braces provide labels: one brace groups the four columns from 'no' to 'isi' and is labeled 'Atribut Table'; another brace groups all seven rows and is labeled 'Record utk data'.

no	nama	email	isi			

Dalam satu baris, perintah pembuatan tabel terlihat sebagai berikut:

```
CREATE TABLE feedback (no int, nama varchar(40), email varchar(30), isi text );
```

Untuk memperjelas pembacaan, nama kolom ditulis dengan huruf tebal. Sedangkan tipe datanya ditulis dengan huruf miring.

MySQL mendukung beberapa tipe data, yang dapat dibagi menjadi: numerik, string (karakter) dan tanggal/waktu. Beberapa tipe data yang sering digunakan terlihat pada tabel berikut:

Tipe Data	Keterangan	Kategori
TINYINT	Signed: -128 s/d 127 Unsigned: 0 s/d 255	NUMERIK
INT	Signed: -2147483648 s/d 2147483647 Unsigned: 0 s/d 4294967295	
BIGINT	Signed: -2 ⁶³ s/d 2 ⁶³ -1 Unsigned: 0 s/d 2 ⁶³ -1	
DOUBLE	Bilangan pecahan/desimal dengan range ±1.7976931348623157E308	
VARCHAR	String dengan panjang bervariasi tergantung datanya. Rangennya: 1 s/d 255 karakter	STRING
TINYTEXT atau TINYBLOB	TEXT atau BLOB dengan panjang maksimum 255 karakter	
TEXT atau BLOB	TEXT atau BLOB dengan panjang maksimum 65535 karakter	
L O N G - TEXT	TEXT atau BLOB dengan panjang maksimum 2 ³² karakter	
DATE	Format "YY-MM-DD", "YYYY-MM-DD", atau YYMMDD"	TANGGAL/WAKTU
TIME	Format "HH:MM:SS", "HHMMSS", "HHMM" atau "HH"	
D A T E - TIME	Format "YYYY-MM-DD HH:MM:SS"	

Table 1. Tipe-tipe data pada MySQL

Memasukkan Data: INSERT

Data dapat dimasukkan ke dalam tabel dengan perintah **INSERT**. Bentuk perintahnya:

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

Berikut penerapan pemasukkan data pada tabel feedback:

```
INSERT INTO feedback VALUES(1, 'Guntur Saputra',  
'guntur@yahoo.com', 'Hallo barang di Palasari murah  
ya');
```

Atau jika kita ingin memasukkan data pada kolom yang ditentukan, perintahnya adalah

```
INSERT INTO nama_tabel (Kolom1, Kolom2, ...) VALUES  
(nilail, nilai2,...);
```

Contohnya:

```
INSERT INTO feedback (nama, email) VALUES('Sarini',  
'ririen@fan.fisika.net');
```

Perintah di atas hanya memasukkan data ke kolom nama dan email saja.

Perlu diingat, pemasukkan data yang bertipe string dan waktu harus memakai ` ` atau '' , sedangkan untuk tipe numerik tidak pelu.

Berikut ini contoh pemasukkan data yang terlihat di shell

MySQL:

```
mysql> INSERT INTO feedback VALUES (  
    -> 2,  
    -> 'Tina Talisa',  
    -> "tina@gmail.com",  
    -> 'Boleh gak COD ke kantor?');  
Query OK, 1 row affected (0.00 sec)
```

```
mysql>
```

Dalam pemasukan data string ke dalam tabel, ada beberapa karakter yang tidak bisa langsung dimasukan begitu saja, melainkan harus didahului (diescape) dengan

karakter backslash (''). Karakter tersebut antara lain:

- single quote (')
- double quote (")
- backslash (\)
- tanda persen (%)
- underscore (_)

Contoh: Memasukan string Faisal 'John' Rudi-janto N!

```
INSERT INTO (nama) VALUES ('Faisal \'John\'  
Rudijanto');
```

Menampilkan Data: SELECT

Data yang telah tersimpan dalam tabel database MySQL dapat ditampilkan dengan perintah SELECT. Bentuk perintahnya:

```
SELECT { * | kolom1 [, kolom2, ...] } FROM nama_tabel
```

Contoh:

```
SELECT * FROM feedback
```

Atau

```
SELECT nama, email FROM feedback
```

Berikut contoh tampilan perintah SELECT di shell MySQL:

```
mysql> SELECT * FROM feedback;
+----+-----+-----+-----+
| no | nama | email | isi |
+----+-----+-----+-----+
| 1 | Guntur Saputra | guntur@yahoo.com | Hallo barang di Palasari murah ya |
| 2 | Tina Talisa | tina@gmail.com | Boleh gak COD ke kantor? |
| 3 | Eva Wulandari | eva@live.com | Saya perlu cepet bukunya |
+----+-----+-----+-----+
3 rows in set <0.00 sec>
mysql>
```

```
mysql> SELECT nama,email FROM feedback;
+-----+-----+
| nama | email |
+-----+-----+
| Guntur Saputra | guntur@yahoo.com |
| Tina Talisa | tina@gmail.com |
| Eva Mulandari | eva@live.com |
+-----+-----+
3 rows in set (0.00 sec)
```

Untuk menspesifikasi hasil query, dapat dilakukan penyaringan dengan klausula WHERE dan dibantu oleh operator equality/inequality, operator logic serta keyword lainnya seperti BETWEEN, LIKE dan ORDER BY. Contoh:

```
mysql> SELECT * FROM feedback WHERE no BETWEEN 1 AND 2;
+-----+-----+-----+
| no | nama | email | isi |
+-----+-----+-----+
| 1 | Guntur Saputra | guntur@yahoo.com | Hallo barang di Palasari murah ya |
| 2 | Tina Talisa | tina@gmail.com | Boleh gak COD ke kantor? |
+-----+-----+-----+
2 rows in set (0.00 sec)
```

BETWEEN digunakan untuk menampilkan data range ANTARA apa dan apa

```
mysql> SELECT * FROM feedback WHERE email LIKE '%yahoo%';
+-----+-----+-----+
| no | nama | email | isi |
+-----+-----+-----+
| 1 | Guntur Saputra | guntur@yahoo.com | Hallo barang di Palasari murah ya |
+-----+-----+-----+
1 row in set (0.00 sec)
```

LIKE digunakan untuk menampilkan data yang MIRIP dengan kata yang dimaksud.

```
mysql> SELECT * FROM feedback ORDER BY nama ASC;
+----+----+----+----+
| no | nama | email | isi |
+----+----+----+----+
| 3 | Eva Wulandari | eva@live.com | Saya perlu cepet bukunya |
| 1 | Guntur Saputra | guntur@yahoo.com | Hallo barang di Palasari murah ya |
| 2 | Tina Talisa | tina@gmail.com | Boleh gak COD ke kantor? |
+----+----+----+----+
3 rows in set (0.00 sec)
```

ORDER BY digunakan untuk mengurutkan data berdasarkan apa. Contoh di atas dilakukan pengurutan berdasarkan abjad nama secara menaik (ASC = Ascending)

```
mysql> SELECT * FROM feedback ORDER BY no DESC;
+----+----+----+----+
| no | nama | email | isi |
+----+----+----+----+
| 3 | Eva Wulandari | eva@live.com | Saya perlu cepet bukunya |
| 2 | Tina Talisa | tina@gmail.com | Boleh gak COD ke kantor? |
| 1 | Guntur Saputra | guntur@yahoo.com | Hallo barang di Palasari murah ya |
+----+----+----+----+
3 rows in set (0.00 sec)
```

ORDER BY digunakan untuk mengurutkan data berdasarkan apa. Contoh di atas dilakukan pengurutan berdasarkan nomor secara menurun (DESC = Descending)

Mengubah Data: UPDATE

Untuk mengubah atau meng-update data dapat dilakukan dengan perintah UPDATE. Contoh:

```
mysql> UPDATE feedback SET nama='Guntur Saputra SH' WHERE nama='Guntur Saputra';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

Hasilnya nama **Guntur Saputra** akan diganti dengan **Guntur Saputra SH**. Perubahannya bisa dilihat pake SELECT:

```
mysql> SELECT * FROM feedback;
+----+----+----+----+
| no | nama | email | isi |
+----+----+----+----+
| 1 | Guntur Saputra SH | guntur@yahoo.com | Hallo barang di Palasari murah ya |
| 2 | Tina Talisa | tina@gmail.com | Boleh gak COD ke kantor? |
| 3 | Eva Wulandari | eva@live.com | Saya perlu cepet bukunya |
+----+----+----+----+
3 rows in set (0.00 sec)
```

Menghapus Record: DELETE

Untuk menghapus barisan record , digunakan perintah DELETE. Contoh:

```
mysql> DELETE FROM feedback WHERE nama='Eva Wulandari';
Query OK, 1 row affected (0.00 sec)
```

Hasil penghapusan bisa dilihat dengan perintah SELECT.

Berikut data record yang tersisa:

```
mysql> SELECT * FROM feedback;
+---+----+-----+-----+
| no | nama | email   | isi          |
+---+----+-----+-----+
| 1  | Guntur Saputra SH | guntur@yahoo.com | Hallo barang di Palasari murah ya |
| 2  | Tina Talisa    | tina@gmail.com  | Boleh gak COD ke kantor?             |
+---+----+-----+-----+
2 rows in set (0.00 sec)
```

Hasilnya record dengan nama **Eva Wulandari** telah dihapus dari tabel.

Mengubah Table : ALTER

ALTER digunakan untuk mengubah property tabel. Pengubahannya antara lain:

- Menambah Kolom, diikuti dengan perintah: ADD COLUMN nama_kolom tipe_data
- Menghapus Kolom, diikuti dengan perintah: DROP COLUMN nama_kolom
- Mengubah Kolom, diikuti dengan perintah: CHANGE kolom_lama kolom_baru tipe_datanya

☞ Berikut adalah contoh perintah untuk menambah kolom di tabel:

```
mysql> ALTER TABLE feedback ADD COLUMN waktu DATETIME;
Query OK, 2 rows affected (0.05 sec)
Records: 2  Duplicates: 0  Warnings: 0
```

Untuk melihat perubahan tabel dari kolom yang ditambahkan, bisa digunakan perintah SELECT:

```
mysql> SELECT * FROM feedback;
+---+-----+-----+-----+-----+
| no | nama | email | isi   | waktu |
+---+-----+-----+-----+-----+
| 1  | Guntur Saputra SH | guntur@yahoo.com | Hallo barang di Palasari murah ya | NULL  |
| 2  | Tina Falisa | tina@gmail.com | Boleh gak COD ke Kantor? | NULL  |
+---+-----+-----+-----+-----+
2 rows in set (0.02 sec)
```

Atau untuk melihat struktur tabel, kita bisa menggunakan perintah **DESC nama_tabel**.

```
mysql> DESC feedback;
+-----+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key  | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| no    | int(11) | YES  |      | NULL    |       |
| nama  | varchar(40) | YES  |      | NULL    |       |
| email | varchar(30) | YES  |      | NULL    |       |
| isi   | text    | YES  |      | NULL    |       |
| waktu | datetime | YES  |      | NULL    |       |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.01 sec)
```

☛ Jika kita ingin menghapus salah satu kolom di tabel, dapat digunakan perintah **DROP**:

```
ALTER TABLE feedback DROP COLUMN nama_kolom
```

☛ Berikut ini contoh perintah untuk mengubah nama kolom dan tipe datanya serta atributnya.

```
mysql> ALTER TABLE feedback CHANGE no id INT NOT NULL PRIMARY KEY AUTO_INCREMENT;
Query OK, 2 rows affected (0.05 sec)
Records: 2  Duplicates: 0  Warnings: 0
```

Perintah di atas mengubah nama kolom **no** menjadi **id** yang bertipe INTEGER. Kolom **id** ini bersifat NOT NULL, PRIMARY KEY serta AUTO_INCREMENT. Keterangan adalah sebagai berikut:

- Penambahan atribut NOT NULL menyatakan bahwa isi kolom **id** tidak mempunyai nilai NULL (kosong)
- Penambahan atribut PRIMARY KEY berfungsi agar data pada kolom **id** tersebut bersifat unik (tidak ada yang sama). Atribut ini berlaku pada kolom yang bersifat not null.

- Attribut AUTO_INCREMENT fungsinya yakni apabila ketika pemasukkan data, jika nilai kolom id kosong, maka MySQL akan secara otomatis membuat nilai yang unik yang bertambah secara otomatis (auto_increment) pada kolom id tersebut. Atribut ini hanya berlaku pada kolom yang bertipe integer dan bersifat primary key.

Struktur Tablenya sekarang menjadi sebagai berikut:

```
mysql> desc feedback;
+-----+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key  | Default | Extra       |
+-----+-----+-----+-----+-----+-----+
| id   | int(11) | YES  | PRI  | NULL    | auto_increment |
| nama | varchar(40) | YES  |       | NULL    |             |
| email | varchar(30) | YES  |       | NULL    |             |
| isi  | text    | YES  |       | NULL    |             |
| waktu | datetime | YES  |       | NULL    |             |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

Adapun jika dari awal kita ingin membuat tabel dengan struktur di atas, maka perintah pembuatan tabel (CREATE) –nya akan seperti ini:

```
CREATE TABLE feedback (
id int NOT NULL PRIMARY KEY AUTO_INCREMENT,
name varchar(40),
email varchar(30),
isi text,
waktu datetime);
```

Untuk pemasukkan data di kolumn yang menyandang atribut **primary key auto increment** itu nilainya boleh dikosongkan karena system MySQL akan memasukkan nilainya secara **otomatis** dan menaik.

Jadi perintah pemasukkan datanya boleh sebagai berikut:

```
INSERT INTO feedback VALUES(' ', 'Yudi Gandana',
, 'yudi@yahoo.com', 'Hai saya numpang mampir
lho!', '2010-03-02 14:00:00');
```

Menghapus Database atau Table: Drop

Dengan statement drop, Anda bisa menghapus sebuah tabel atau sebuah database

Contoh :

```
DROP TABLE nama_tabel;  
DROP DATABASE nama_database;
```

Hati-hatilah memakai statement ini. Bisa-bisa data Anda bisa hilang semua.

So What Next?

Kali ini kita telah mengenal perintah-perintah fundamental dalam administrasi MySQL. Hal ini cukup penting untuk dapat diketahui sebagai dasar Anda berinteraksi lebih dalam dengan MySQL. Adapun untuk memudahkan dan mempercepat administrasi MySQL, anda bisa menggunakan software HeidiSQL (<http://www.heidisql.com>) atau PHPMyAdmin.

Untuk PHPMyAdmin sudah terinstall secara otomatis oleh XAMPP dan alamat untuk administrasinya ada di <http://localhost/phpmyadmin> - tentunya dengan username root dan password yang ada create tadi. Berikutnya kita akan mempelajari hubungan antara PHP dan MySQL. Kita akan belajar cara mengakses MySQL melalui fungsi-fungsi yang telah disediakan PHP. Tapi sebelumnya mari kita rehat kopi dulu sejenak

Bab 9

Konektivitas PHP dengan MySQL

Pada bab ini kita akan berusaha untuk menggabungkan kode PHP dengan perintah-perintah yang ada di MySQL. Tujuannya agar kita dapat membuat halaman web yang bisa menyimpan data dari form ke database kemudian menampilkannya.

Akses Database

Secara umum akses ke database ada tiga tahapan

1. Koneksi ke database (persiapan).
2. Query/permintaan data.
3. Penutupan koneksi.

Konsep ini akan selalu diterapkan ketika ketika PHP berurusan dengan MySQL.

Fungsi-fungsi PHP untuk Mengakses MySQL

Di bawah ini adalah sejumlah fungsi PHP yang sering digunakan untuk mengakses database MySQL.

a. Membuka Koneksi: `mysql_connect()`

`mysql_connect()` digunakan untuk membuka koneksi ke MySQL. Prototipenya adalah:

```
int mysql_connect(string host, string user, string password)
```

Fungsi ini akan berusaha membuka koneksi ke host dimana MySQL berada. Biasanya nama `host`-nya adalah `localhost`. Kemudian parameter `user` adalah nama user yang terdaftar di MySQL. Kemudian `password` adalah password dari user tersebut. Nilai kembalian fungsi ini adalah integer positif (true) yakni apabila proses koneksi berhasil. Jika proses koneksi gagal, maka akan memberikan nol (false).

Berikut contoh nyata pemakaian fungsi ini:

```
<?php  
$host="localhost";  
$user="root";  
$pass="hallo";  
  
$sambung=mysql_connect($host, $user, $pass);  
if ($sambung)  
    echo "koneksi ke database berhasil";  
else  
    echo "Koneksi ke database gagal";  
?>
```

Kode 1: `sambung.php` – Test penyambungan ke database

b. Memilih Database: mysql_select_db()

Sebelum kita bermain dengan database, kita diharuskan untuk memilih database mana yang akan kita pakai/aktifkan (di mysql dipakai perintah USE). Fungsi `mysql_select_db()` digunakan untuk memilih database yang aktif.

Prototipenya:

```
int mysql_select_db(string nama_database)
```

Berikut contoh nyata pemakaian fungsi ini:

```
<?php  
$host="localhost";  
$user="root";  
$pass="hallo";  
$database="palasari";  
  
$sambung=mysql_connect($host, $user, $pass);  
$pilih=mysql_select_db($database);  
if ($sambung AND $pilih)  
    echo "Koneksi dan pengaktifan database berhasil. ";  
else  
    echo "Koneksi dan pengaktifan database gagal. ";  
?>
```

Kode 2: pilih.php – Test koneksi dan pemilihan database

c. Mengeksekusi Perintah SQL: mysql_query()

Fungsi `mysql_query` berguna untuk mengeksekusi perintah SQL pada tabel. Prototipenya:

```
int mysql_query(string query)
```

Sebelum melakukan proses ini, alangkah baiknya untuk mendefinisikan string query yang akan dijalankan, misal :

```
$sql= "INSERT INTO feedback (nama,email) VALUES ('Faisal', 'isal79@yahoo.com')";
```

Berikut adalah contoh penerapannya:

```
<?php  
$host="localhost";  
$user="root";  
$pass="hallo";  
$database="palasari";  
  
mysql_connect($host, $user, $pass); //koneksi ke database  
mysql_select_db($database); //aktifkan database yang akan dipakai  
$sql= "INSERT INTO feedback (nama,email) VALUES ('Faisal', 'isal@yahoo.com')";  
$hasil=mysql_query($sql);  
  
if ($hasil)  
    echo "Pemasukkan data berhasil dilakukan!";  
else  
    echo "Pemasukkan data gagal dilakukan!";  
?>
```

Kode 3:input.php – Test pemasukan data ke database

Ke depan kita akan berusaha memasukkan data dari form feedback ke database dengan menggunakan fungsi ini.

d. Mengambil data dari database: mysql_fetch_row()

Fungsi `mysql_fetch_row()` berguna untuk mengambil data hasil query secara baris perbaris (row). Pengambilan pertama adalah baris data yang paling atas. Data yang diambil dalam bentuk array, dimana elemen dari array adalah field-field dari tabel data. Prototipnya:

```
array mysql_fetch_row(int hasil)
```

Argumen hasil diperoleh dari proses query untuk mengambil data (SELECT). Contoh string query untuk mengambil data:

```
$sql="SELECT* FROM feedback";
$hasil=mysql_query($sql);
```

Apabila kita mempunyai data seperti ada pada tabel di bawah ini:

```
mysql> SELECT * FROM feedback;
+----+-----+-----+-----+-----+
| id | nama | email | isi   | waktu |
+----+-----+-----+-----+-----+
| 1  | Guntur Saputra SH | guntur@yahoo.com | Hallo barang di Palasari murah ya | NULL  |
| 2  | Tina Talisa    | tina@gmail.com   | Boleh gak COD ke kantor?          | NULL  |
| 3  | Yudi Gandana   | yudi@yahoo.com  | Hai saya numpang mampir lho!      | 2010-03-02 14:00:00 |
| 4  | Faizal          | faizal@yahoo.com | NULL                           | NULL  |
+----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

Sintaks pengambilan data :

```
$baris=mysql_fetch_row($hasil)
```

Hasilnya jika kita menggunakan loop while :

```
While ($baris=mysql_fetch_row($hasil)) {
...
}
```

Maka pada:

Pembacaan data yang pertama:

```
$baris[0]="""
$baris[1]=""Guntur Saputra SH"""
$baris[2]=""Guntur@yahoo.com"""
$baris[3]=""Hallo barang di Palasari murah ya"""
$baris[4]=""
```

\$baris[0] mempunyai arti *elemen baris pada kolom yang ke-nol*

Pembacaan data yang kedua:

```
$baris[0]=""
$baris[1]=""Tina Talisa"""
$baris[2]=""tina@gmail.com"""
$baris[3]=""Boleh gak COD ke kantor?"""
$baris[4]=""
```

dan seterusnya.

Berikut adalah contoh penggunaan fungsi mysql_fetch_row():

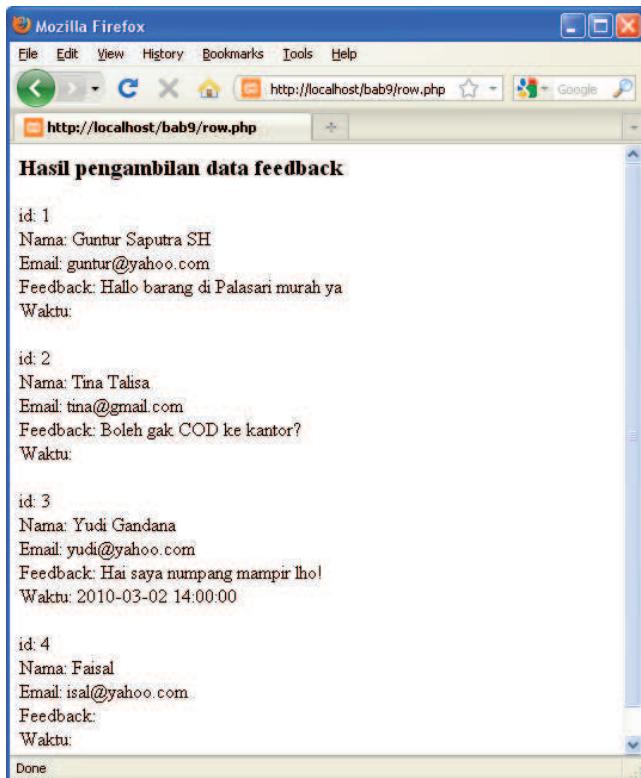
```
<?php
$host="localhost";
$user="root";
$pass="hallo";
$database="palasari";

mysql_connect($host, $user, $pass); //koneksi ke database
mysql_select_db($database); //aktifkan database yang akan dipakai
$sql="SELECT* FROM feedback";
$hasil=mysql_query($sql);

echo "<h3>Hasil pengambilan data feedback</h3>";
while($baris=mysql_fetch_row($hasil) ) {
    echo "id: $baris[0] <br>";
    echo "Nama: $baris[1] <br>";
    echo "Email: $baris[2] <br>";
    echo "Feedback: $baris[3] <br>";
    echo "Waktu: $baris[4]<br><br>";
}

?>
```

Kode 4: row.php – Mengambil data dari database



Gb.1 Tampilan `row.php` dimana data feedback dari database ditampilkan

e. Menghitung Jumlah Baris: `mysql_num_rows()`

Fungsi `mysql_num_row` digunakan untuk menghitung jumlah record dari hasil query. Prototipenya:

```
int mysql_num_rows (int hasil)
```

Untuk contoh penggunaannya, tambahkan skrip di bawah ini pada `row.php`:

```
$jum_data=mysql_num_rows($hasil)  
echo "Jumlah data: $jum_data";
```

Studi Kasus 1: Menyimpan Data Feedback ke Database

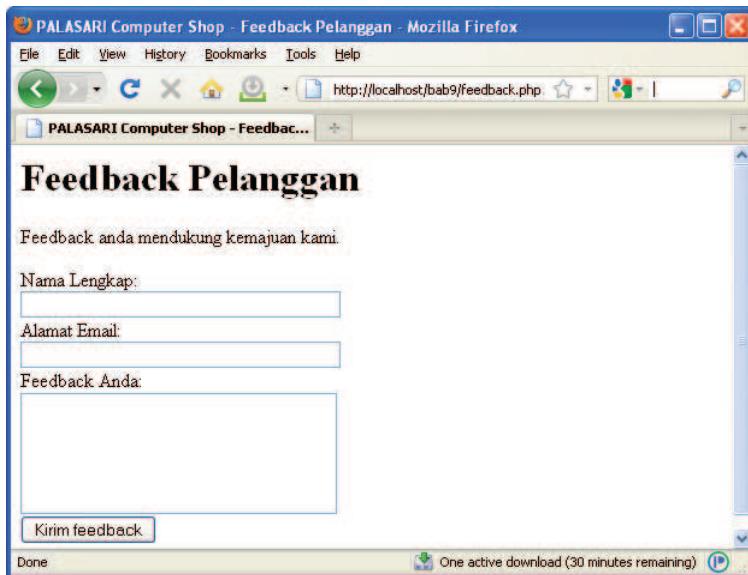
Kita akan berusaha untuk mempercanggih form feedback yang kita buat agar selain bisa dikirim via email, datanya juga bisa disimpan di dalam database. Data feedback akan ditampung dalam tabel feedback yang sudah dibuat pada bab sebelumnya. Berikut ditampilkan kembali atribut yang terdapat pada tabel feedback:

```
mysql> desc feedback;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	YES	PRI	NULL	auto_increment
nama	varchar(40)	YES		NULL	
email	varchar(30)	YES		NULL	
isi	text	YES		NULL	
waktu	datetime	YES		NULL	

5 rows in set (0.00 sec)

Kemudian halaman form feedback yang ada di bab 7 terlihat seperti di bawah:



Gb.2 Tampilan halaman feedback.php

Nah, agar data dari form di atas dapat disimpan di database, maka kita harus memodifikasi kode PHP yang ada di halaman `feedback_proses.php`. Tambahkan kode berikut pada halaman tersebut (tentunya di dalam area tag php):

```
*****proses pemasukkan ke database *****
$host="localhost";
$user="root";
$pass="hallo";
$database="palasari";

//koneksi ke database
mysql_connect($host,$user,$pass);
mysql_select_db($database);

$waktu=date("Y-m-d H:m:s"); //format data sekarang

//perintah sql untuk pemasukkan data ke tabel feed-
```

```
back
$sql="INSERT INTO feedback VALUES('',
'$nama','$email','$feedback', '$waktu')";
$hasil=mysql_query($sql);
if ($hasil)
    echo "<p>Data Telah dimasukan ke database kami";
else
    echo "<p>Maaf database dalam perbaikan";
//*****
```

Berikut adalah hasil tampilan `feedback_proses.php`, dimana datanya selain dikirim via email juga disimpan dalam database:



Gb.3 Halaman hasil proses feedback

Kita dapat mengecek hasil penambahan datanya di shell MySQL:

```
mysql> SELECT * FROM feedback;
+----+-----+-----+-----+
| id | nama | email | isi      | waktu        |
+----+-----+-----+-----+
| 1  | Guntur Saputra SH | guntur@yahoo.com | Hallo barang di Palasari murah ya | NULL         |
| 2  | Tina Talisa       | tina@gmail.com   | Boleh gak COD ke kantor?           | NULL         |
| 3  | Yudi Gandana      | yudii@yahoo.com | Hai saya numpang mampir lho!       | 2018-03-02 14:00:00 |
| 4  | Faisal            | isal@yahoo.com  | NULL                                | NULL         |
| 5  | Saiful Hanif      | sahanif@live.com | Saya mau donk buku anak-anak       | 2018-04-23 12:04:58 |
+----+-----+-----+-----+
5 rows in set (0.00 sec)
```

Studi Kasus 2: Menampilkan Data Feedback di Halaman Web

Setelah kita berhasil menyimpan data feedback dari form ke database, sekarang kita akan membuat halaman yang dapat menampilkan data feedback dari database. Intinya kita akan menggunakan perintah SELECT untuk menampilkannya. Berikut adalah kodennya:

```
<html>
<head>
    <title>Data Feedback Pelanggan</title>
</head>

<body>
<?php
//inisialisasi parameter koneksi
$host="localhost";
$user="root";
$pass="hallo";
$database="palasari";

//koneksi ke database
mysql_connect($host,$user,$pass);
mysql_select_db($database);

//Perintah SQL
$sql="SELECT * FROM feedback ORDER BY waktu DESC";

//eksekusi perintah sql
$hasil=mysql_query($sql);

echo "<h3>Data Feedback Pelanggan PALASARI</h3>";
//gunakan loop untuk menampilkan data
//kita gunakan style tabel
echo "<TABLE border=1 cellpadding=\"3\" cellspacing=\"1\">";
echo "<TR bgcolor="#cccccc">";
echo "    <TH>Waktu</TH><TH>Nama</TH><TH>Email</TH><TH>Feedback</TH>";
echo "    </TR>";
while ($baris=mysql_fetch_row($hasil)) {
    //inisialisasi variable untuk elemen kolom tabel
    //feedback (agak mudah dibaca)
```

■ Aplikasi Berbasis Web dengan PHP & MySQL *Learning By Doing and Make it Simple!*

```
$id=$baris[0];
$nama=$baris[1];
$email=$baris[2];
$feedback=$baris[3];
$waktu=$baris[4];
echo "<TR><TD>$waktu</TD><TD>$nama</TD><TD>$email</
TD><TD>$feedback</TD></TR>";
}
echo "</TABLE>";

//Tampilkan jumlah total baris/record
$jumlah_baris=mysql_num_rows($hasil);
echo "<b>Total: $jumlah_baris baris</b>";

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

Kode 5: feedback_tampil.php – Kode untuk menampilkan data feedback di halaman web

Hasil tampilannya adalah sebagai berikut:



Waktu	Nama	Email	Feedback
2010-04-23 12:04:53	Saiful Hanif	sahanhf@live.com	Saya mau donk buku anak-anak
2010-03-02 14:00:00	Yudi Gandana	yudi@yahoo.com	Hai saya numpang mampir lho!
	Guntur Saputra SH	guntur@yahoo.com	Hallo barang di Palasari murah ya
	Tina Talisa	tina@gmail.com	Boleh gak COD ke kantor?
	Faisal	isal@yahoo.com	

Gb.4 Tampilan data feedback dalam bentuk tabel

Bagaimana? Mudah bukan! Nah jika style penampiliannya ingin seperti yang terdahulu (**seperti gb.x**), it's up to

you mana yang lebih disukai. Sekarang setidaknya setelah belajar cara pengambilan data ini, Anda bisa menerapkannya untuk aplikasi buku tamu (*guestbook*) yang sekarang lagi tren di hampir semua website pribadi. Just try it ok!

Another Case: Order Barang

Untuk memperlancar permainan kita di database, sekarang coba Anda terapkan proses penyimpanan data order yang ada di bab 4 ke dalam database kemudian menampilkannya. Formnya ada seperti pada gambar di bawah. Silahkan analisis tipe data apa saja yang nantinya ada pada tabel order. Kemudian barulah kita membuat kodennya seperti yang telah dicontohkan di atas. Selamat mencoba!

The screenshot shows a Mozilla Firefox browser window with the title "PHP Computer Shop - Mozilla Firefox". The address bar displays the URL "http://localhost/bab4/shop.php". The main content area is a form titled "Form Pemesanan Barang". The form has several input fields:

- A table with two columns: "Item" and "Banyaknya". It contains three rows: "Monitor", "Speaker", and "Keyboard". Each row has a text input field next to it.
- A dropdown menu labeled "Mengetahui Toko PHP dari". The options are "Langsung datang", "Dari teman", and "Dari iklan". "Langsung datang" is currently selected.
- A dropdown menu labeled "Dari". The options are "Submit", "Iklan", "Buku telepon", and "Word of mouth". "Langsung datang" is currently selected.
- A "Done" button at the bottom left.

Gb.4 Form pemesanan barang dimana datanya akan dimasukkan ke database MySQL

Bab 10

Memahami Fungsi Lebih Dalam

Anda sebenarnya sudah masuk ke tingkatan level PHP dengan skill menengah. Syukurlah! Namun jangan berpuas telebih dahulu, karena skill yang Anda punya sekarang tidak akan adanya gunanya kalau tidak disertai dengan latihan dan pengembangan intuisi yang terus menerus. Oleh karena itu sekarang kita akan berusaha untuk beralih ke bahasan yang lebih advanced.

Memeriksa keberadaan suatu variable : `isset()`

Fungsi `isset` digunakan untuk memeriksa keberadaan suatu variable. Prototipenya:

```
int isset(variable)
```

Nilai kembalinya akan integer positif (TRUE) apabila *variable* eksis, serta nol (FALSE) jika variable tidak ada. Kita akan mengaplikasikan fungsi ini untuk mengintegrasikan proyek-proyek yang telah kita buat dalam satu halaman web. Berikut ini adalah skripnya:

```
<html>
<head>
    <title>PALASARI Computer Shop</title>
</head>

<body>
<h1 align="center">Welcome To PALASARI Computer Shop</h1>
<h2 align="center">We always deliver the best Product for You!</h2>

<p align="center">
<a href="index.php"> Home </a> |
<a href="index.php?hal=order">Order</a> |
<a href="index.php?hal=tarif">Tarif Pengiriman</a> |
<a href="index.php?hal=feedback">Feedback</a>
</p>

<TABLE align="center" border="0"><TR><TD>
<?php
if ( isset($_GET[hal]) AND $_GET[hal]=="order")
    include ("shop.php");
elseif ( isset($_GET[hal]) AND $_GET[hal]=="tarif")
    include ("pengiriman.php");
elseif ( isset($_GET[hal]) AND $_GET[hal]=="feedback")
    include ("feedback.php");
else
    include ("gambar.php");
?>

</TD></TR></TABLE>

</body>
</html>
```

kode 10.1: index.php – Contoh penggunaan fungsi `isset` untuk memeriksa keberadaan variable

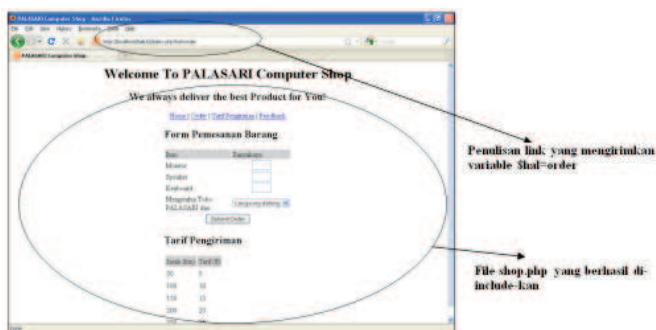
Perhatikan penggunaan hyperlink: `index.php?hal=order`

Style penulisan link tersebut banyak digunakan dalam pemrograman web, tujuannya agar web yang kita buat tidak rumit dan lebih sederhana. Link tersebut akan mengirimkan variable `$hal` yang mempunyai value `order`. Nah jika kita mengklik link ini, maka PHP akan mem-follow-upinya dengan skrip:

```
if ( isset($_GET[hal]) AND $_GET[hal]=="order")
    include ("shop.php");
```

Dari sini PHP akan berusaha memeriksa keberadaan variable `$hal` (dengan fungsi `isset`) kemudian jika variable `$hal` tersebut bernilai string `order`, maka PHP akan memunculkan (meng-include) file `shop.php` di halaman `index.php`. Prefix `$_GET[]` ditambahkan pada variable `$hal` karena ia berfungsi untuk mengambil nilai variable yang dikirimkan via hyperlink URL (misal : `index.php?hal=order`).

Hasilnya akan terlihat seperti pada gambar di bawah ini:



Gb. 10.1 Tampilan hasil peng-includan yang variablenya dikirim via hyperlink

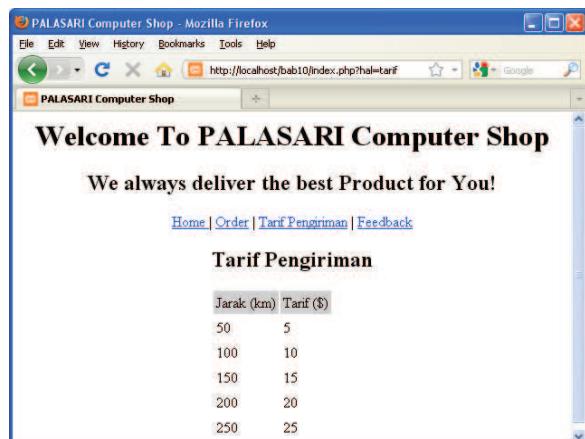
Demikian seterusnya. PHP akan melakukan pemeriksaan variable yang dikirimkan via hyperlink. Jika tidak

■ Aplikasi Berbasis Web dengan PHP & MySQL *Learning By Doing and Make it Simple!*

ada, maka PHP akan menampilkan file **gambar.php**, dimana ia akan dimunculkan di halaman depan (index.php).



Gb. 10.2 Penampilan halaman depan (index.php), dimana PHP akan mengincludekan file gambar.php



Gb. 10.3 Link index.php?hal=tarif. Disini halaman index.php akan meng-include-kan file pengiriman.php

Function dalam PHP

Function ada hampir di setiap bahasa pemrograman. Dalam membuat sebuah program seringkali kita membutuhkan sekumpulan perintah yang akan digunakan berkali-kali. Untuk menghindari penulisan perintah yang sama berkali-kali, kita dapat menggunakan function. Hal ini membuat kode program kita lebih mudah dibaca dan dipelihara.

a. Built -In Function

Buit-In Function adalah function yang sudah disediakan oleh PHP. PHP memiliki built-in function yang luar biasa banyaknya, sehingga sangat tidak mungkin jika semuanya harus dijabarkan dalam modul ini. Sebenarnya kita telah banyak menggunakan beberapa *built-in function* PHP, diantaranya `echo()`, `date()`, `fopen()`, `mysql_connect()`, dan masih banyak lagi. Adalah mustahil untuk bisa menghafalkan semua fungsi built-in di PHP. Oleh sebab itu, bawalah teman setia Anda kemanapun Anda pergi, yaitu manual PHP!

Namun meskipun begitu, adalah penting bagi kita untuk memahami cara membaca built-in function. Sebagai contoh di manual PHP, mari kita lihat protipe function `number_format()` :

```
string number_format ( float $number , int $decimals , string $dec_point , string $thousands_separator )
```

Manual PHP mengatakan: bahwa fungsi tersebut mempunyai parameter masukkan yaitu

- Parameter pertama : \$number yang bertipe float
- Parameter kedua : \$decimals yang bertipe integer
- Parameter ketiga : \$dec_point yang bertipe string
- Parameter keempat: \$thousand_separator yang bertipe sting

<?php

```
$number = 1234.56;

// english notation (default) dengan pemisah ribuan
$english_format_number = number_format($number);
// hasilnya 1,235

// Indonesia notation
$format_number_indonesia = number_format($number, 2,
',', '.');
//hasilnya 1.234,56

$number = 1234.5678;

// english notation tanpa pemisah ribuan
$english_format_number = number_format($number, 2,
'.', '');
// hasilnya 1234.57

?>
```

b. User Defined Function (UDF)

PHP telah banyak menyediakan kita function built-in untuk berbagai keperluan. Namun dalam pekerjaan memogram suatu waktu kita pasti membutuhkan function yang memang tidak disediakan oleh PHP. Untungnya, Anda tidak dibatasi untuk menggunakan fungsi built-in karena

Anda dapat menulis function yang sesuai dengan keperluan yang Anda hadapi. Function buatan Anda mungkin saja adalah campuran fungsi-fungsi built-in yang ada digabung dengan alur logika yang sesuai dengan keperluan.

Jika Anda menulis serangkai blok kode dan ternyata blok kode tersebut terulang digunakan di halaman yang lain, maka alangkah baiknya Anda menjadikan blok tersebut sebagai function. Anda dapat mendefinisikan parameter apa saja yang dijadikan masukkan dan tipe keluaran apa yang diinginkan dari fungsi tersebut.

1. Struktur Function

Sintaks untuk pendeklarasian function dalam PHP adalah sebagai berikut:

```
function my_function() {  
    echo "Hehehe.. Ini function-ku! ";  
}
```

pendeklarasian function selalu diawali oleh kata `function`. Nah nama function di atas adalah `my_function`. Untuk memanggil function di atas, Anda tinggal memanggilnya dengan pernyataan sebagai berikut:

```
my_function();
```

Sebagaimana dapat kita tebak, function tersebut akan menampilkan text di browser sebagai berikut:

```
Hehehe... Ini function-ku!
```

2. Penamaan Function

Beberapa hal yang harus diperhatikan dalam memberi

nama function:

- Nama function yang Anda bikin tidak boleh sama dengan function yang sudah ada
- Nama function hanya diperbolehkan mengandung huruf, angka atau underscore.
- Nama function tidak boleh didahului oleh angka

Contoh penamaan function yang legal

- nama()
- nama2()
- nama_tiga()
- _nama4()

Contoh penamaan function yang ilegal (tidak dibenarkan):

- 5nama()
- nama-6()
- strlen()

3. Parameter

Biasanya function memerlukan satu atau lebih parameter/ argumen, hal ini bergantung kasus yang kita temui. Parameter mengizinkan kita untuk melewatkkan data ke dalam function. Berikut ini adalah contoh sebuah function `tulis_tebal()` yang memiliki sebuah parameter masukan bernama `$teks`.

```
function tulis_tebal($teks) {  
    echo "<b>$teks</b>";  
}
```

Kita dapat memanggil function di atas dengan melewatkkan

sebuah teks di parameternya:
tulis_tebal("Halo PALASARI!");

Maka hasil keluarannya akan seperti berikut:

Halo **PALASARI!**

Function di atas adalah function yang cukup sederhana, dimana parameternya hanya satu. Sekarang mari kita lihat function perkalian antara dua bilangan di bawah ini, dimana function ini memiliki dua parameter, yakni \$input1 dan \$input2 :

```
Function kali($input1, $input2){  
    $hasil=$input1*$input2;  
    echo "Hasil perkalian $input1 dengan $input2 adalah:  
$hasil";  
}  
$a=6;  
$b=7;  
//panggil function  
kali($a,$b);
```

Keluaran kode di atas adalah sebagai berikut:

Hasil perkalian 6 dengan 7 adalah: 42

Berikut ini adalah contoh *real-world* penggunaan function

kali() pada form:

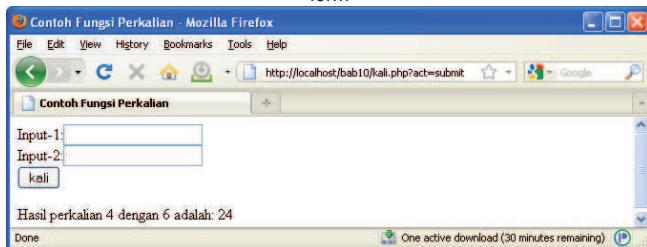
```
<html>  
<head> <title>Contoh Fungsi Perkalian </title> </  
head>  
<body>  
<Form method="post" action="?act=submit">  
Input-1:<input type=text name=input1><br>  
Input-2:<input type=text name=input2><br>  
<input type=submit value=kali>  
</Form>  
  
<?php  
function kali ($a, $b) {  
    $hasil = $a*$b;
```

```
echo "Hasil perkalian $a dengan $b adalah: $hasil";
}

if (isset($_GET[act]) AND $act=="submit"){
    $input1=$_POST[input1];
    $input2=$_POST[input2];
    kali($input1,$input2);
}

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

Kode 10.2: kali.php – Penerapan function yang parameternya dilewatkan melalui input form



Gb. 1.4 Tampilan kali.php dimana variable dari input dilewatkan ke dalam function

Dari bahasan parameter ini dapat disimpulkan:

1. Parameter adalah data masukan yang akan diproses di badan function
2. Parameter dalam function boleh ada atau tiada, ini bergantung pada kasus yang kita temui serta keluaran apa yang kita inginkan
3. Nama paramater pada saat function dipanggil boleh berbeda atau boleh juga sama dengan nama parameter pada saat function didefinisikan.
4. Apabila jumlah parameter pada saat function dipanggil tidak sesuai dengan parameter yang ada

pada saat function didefinisikan, maka program akan memberikan *Warning missing argument*.

4. Nilai Balik (Return Value)

Umumnya function memberikan nilai balik. Misal, ketika kita memanggil function `strlen()`, function ini memberikan nilai balik (return value) bertipe integer. Nah bagaimanakah cara membuat function yang dapat memberikan nilai balik ketika dipanggil? Contoh berikut memberikan gambaran tentang function yang memberikan nilai balik.

```
function kali ($a, $b) {  
    $hasil = $a*$b;  
    return $hasil;  
}
```

Tugas yang dilakukan oleh function `kali()` di atas sangatlah sederhana, yakni menghasilkan nilai perkalian antara dua parameter. Selanjutnya pernyataan:

```
return $hasil;
```

berfungsi untuk memberikan nilai balik function.

Dengan dieksekusinya pernyataan `return`, eksekusi terhadap kode pada function akan diakhiri.

Berikut ini adalah contoh pemanggilan function `kali()` di atas:

```
$a=4;  
$b=7;  
echo "Hasil kali antara $a dan $b adalah: ".kali($a,$b);
```

Hasil keluarannya:

Hasil kali antara 4 dan 7 adalah: 28

Penggunaan pernyataan `return` juga digunakan untuk

menghentikan sebuah kondisi error dari suatu function. Misal, jika kita menulis sebuah function untuk mencari mana angka yang terbesar (dari dua angka), maka kita harus membuat suatu pernyataan dimana jika angka yang dibandingkan belum terbentuk maka **eksekusi** akan dihentikan. Berikut adalah kodenya:

```
function banding( $x, $y )
{
    if (!isset($x) || !isset($y)){
        echo "Function ini memerlukan dua bilangan";
        return;
    }

    if ($x>$y)
        echo $x;
    else
        echo $y;
}
```

Function built-in `isset()` digunakan untuk memeriksa keberadaan suatu variable, apakah sudah terbentuk dan ada nilainya atau belum. Pada kode di atas kita akan memberikan pesan error jika parameter function belum diberikan nilainya. Kita memeriksanya dengan `!isset()`, yang berarti “NOT `isset()`”. Jadi pernyataan `if` dapat dibaca “jika x belum di-set atau y belum di-set” maka function akan menghentikan eksekusi. Penggunaan kodenya adalah sebagai berikut:

```
$a=4;
$b=9.5;
banding($a,b);
banding($a,$c);
```

Keluarannya adalah:

```
9,5
Function ini memerlukan dua bilangan
```

Berikut ini adalah contoh salah satu function yang berguna untuk melakukan pemilihan nilai di form. Function ini bisa digunakan untuk menggenerate pemilihan tanggal dan bulan. Berikut adalah kodennya:

```
function get_pilihan($jumlah,$nama) {  
    $pilihan = "<select name=$nama>";  
    $pilihan .= "<option value=\"$\">$nama";  
    for ($i=1; $i<=$jumlah; $i++) {  
        $pilihan .= "<option value=\"$i\">$i";  
    }  
    $pilihan .= "</select>";  
    return $pilihan;  
}  
//Panggil Function  
echo "Tanggal: ".get_pilihan(31,"tanggal")." Bulan:  
".get_pilihan(12,"bulan");
```

Kode 10.3: pilihan.php – function untuk menampilkan pilihan tanggal dan bulan

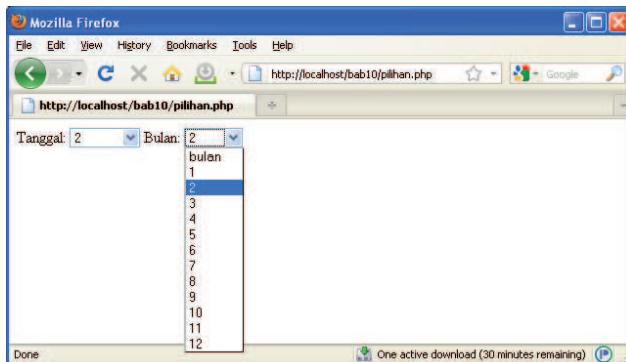
Untuk penulisan menu select di function di atas, sengaja tidak memakai echo. Tujuannya agar hasil akhrinya dapat ditempatkan setelah pernyataan return. Pernyataan .= digunakan sebagai operator penggabung string. Konsep dasarnya:

```
$nama = "Sarini ";  
$nama .= "Arifah ";  
$nama .= "Asli Garut";  
echo $nama;
```

keluarannya akan menghasilkan

Sarini Arifah Asli Garut

Kembali kepada function `get_pilihan()`, hasil keluaran kode pilihan.php akan sebagai berikut:



Gb. 10.5 Tampilan form select dari function get_pilihan()

5. Pass By Value Versus Pass By Reference

Secara default parameter dalam sebuah function bersifat *pass by Value*, artinya hanya nilainya saja yang dibutuhkan. Sehingga perubahan nilai parameter yang ada pada badan function tidak akan berpengaruh ketika function dipanggil.

Jika suatu waktu kita ingin mengubah nilai sebuah parameter dalam function sehingga pemanggilnya juga merasakan perubahan tersebut, maka parameter tersebut dinamakan *pass by reference*. Untuk membuat parameter *pass by reference* digunakan karakter "&" di depan nama parameter.

Contoh sederhananya ada pada contoh di bawah ini:

```
function kali ($a, $b) {  
    $hasil = $a*$b;  
    $a = 500;  
    return $hasil;  
}  
$x=7;  
$y=9;  
echo "Isi variable $x = $x <br>";
```

```
$z = kali($x,$y);  
echo "Hasil perkalian \$x dan \$y : $z <br>";  
echo "Isi variable \$x = $x ";
```

Hasil keluarannya adalah:

```
Isi variable $x = 7  
Hasil perkalian $x dan $y : 63  
Isi variable $x = 7
```

Terlihat setelah function dipanggil tidak terjadi perubahan nilai variable \$a yang telah ditetapkan di dalam function. Agar setelah function dipanggil terjadi perubahan nilai pada variable, maka harus kita tambahkan tanda “&” di parameter \$a.

```
function kali (&$a, $b) {  
    $hasil = $a*$b;  
    $a = 500;  
    return $hasil;  
}  
$x=7;  
$y=9;  
echo "Isi variable \$x = $x <br>";  
$z = kali($x,$y);  
echo "Hasil perkalian \$x dan \$y : $z <br>";  
echo "Isi variable \$x = $x ";
```

Hasil keluarannya:

```
Isi variable $x = 7  
Hasil perkalian $x dan $y : 63  
Isi variable $x = 500
```

6. Function yang bersifat rekursif

Pada PHP, suatu function dapat memanggil dirinya sendiri. Function seperti ini disebut sebagai function rekursif. Salah satu contoh mudahnya adalah function untuk menghitung faktorial. Suatu faktorial $N!$ didefinisikan seb-

agai berikut:

1, Jika N=0

Misal hasil dari 4! adalah :

$$4 \times 3! = 4 \times 3 \times 2! = 4 \times 3 \times 2 \times 1! = 4 \times (N \times (N-1) \times (N-2) \times \dots \times 1) \text{ jika } N \geq 0 \\ 3 \times 2 \times 1 \times 1 = 24$$

Maka function dari faktorial adalah sebagai berikut:

```
Function faktorial ($input){  
    if ($input==0)  
        return 1;  
    else  
        return $input* faktorial($input-1);  
}  
  
$a=4;  
echo "$a! Adalah : ".faktorial($a);
```

Kode 10.4: faktorial.php – function rekursif terlihat pada kasus faktorial

Jalannya ekskusi function faktorial(4) akan memanggil dirinya beberapa kali dengan parameter sebagai berikut:

```
4*faktorial(3);  
3*faktorial(2);  
2*faktorial(1);  
1*faktorial(0);  
1
```

Setiap pemanggilan function, maka di memory server akan tercipta salinan baru function namun dengan parameter yang berbeda. Dari setiap kali pemanggilan, function akan mengembalikan nilai balik berupa nilai \$input yang kemudian nilai ini akan diperiksa apakah mencapai nol atau belum. Ketika input mencapai angka nol, maka kondisi nilai balik akan memberikan nilai 1. Akhirnya function akan mengembalikan return value berupa

perkalian nilai balik di setiap kali pemanggilan.

Dari bahasan di atas, terlihat proses rekursif dapat digunakan sebagai alternatif pengganti iterasi karena keduanya memungkinkan kita untuk melakukan sesuatu secara berulang. Proses rekursif lebih lamban dan memerlukan lebih banyak memory dari pada proses iterasi. Jadi Anda bebas memilih cara mana yang lebih mudah untuk digunakan. Bila cara rekursif lebih dianggap mudah, Just Do It! Berikut ini adalah function faktorial alternatif yang menggunakan proses iterasi.

```
function factorial($nilai){  
    $fak=1;  
    for ($i=1; $i<=$nilai; $i++) {  
        $fak=$fak*$i;  
    }  
    return $fak;  
}  
//panggil function  
$a=3;  
echo "$a! Adalah : ".factorial($a);
```

kode 10.5 factorial.php – function untuk menghitung faktorial dengan cara iterasi

Studi Kasus: Perpustakaan Rakyat

- ☺ Sebuah perpustakaan rakyat menyewakan 3 golongan buku, yaitu A, B, dan C. harga sewa per 7 hari adalah:

Golongan	Harga Sewa 7 Hari
A	1000
B	750
C	500

Jika meminjam lebih dari 7 hari, setiap harinya denda Rp.200. Buatlah program untuk menghitung pembayaran dengan

■ Aplikasi Berbasis Web dengan PHP & MySQL Learning By Doing and Make it Simple!

- Masukan: Golongan Buku, Lama Pinjam
- Keluaran: Bayar

Petunjuk: Buatlah 3 function

1. Function untuk menghitung harga sewa
2. Function untuk menghitung denda
3. Function untuk menghitung total pembayaran

Berikut adalah protipe tampilannya:

The screenshot shows a Microsoft Internet Explorer window titled "Perpustakaan Rakyat Cerdas - Microsoft Internet Explorer". The address bar shows "http://localhost/adv/1/perpus.php?act=submit". The main content area has a title "Perpustakaan Rakyat "CERDAS"" and a subtitle "Sistem Informasi Peminjaman Buku". Below this, there is a form with two input fields: "Golongan Buku" (selected as "Golongan") and "Lama Pinjam" (set to "9"). There are also "Submit" and "reset" buttons. Below the form, a section titled "Detail Pembayaran" displays the following calculations:
Golongan Buku : A
Harga Sewa : 1000
Lama Pinjam : 9
Denda : Rp. 400
Total Bayar : Rp. 1,400

Gb. 10.6 Tampilan form studi kasus Perpustakaan Rakyat

Kesimpulan

Pada pertemuan kali ini setidaknya kita telah mempunyai bekal mengenai konsep dasar function. Pada bahasan ke depan kita akan mempelajari konsep objek oriented dalam PHP dimana di dalamnya akan dibahas class. Di dalam class kita pasti akan banyak berinteraksi dengan function. Tapi sebelumnya mari kita rehat sejenak!

Bab 11

MySQL Database Design

Kali ini kita akan berusaha membahas materi database MySQL yang lebih advanced. Pada bab ini kita tidak akan menjelaskan kembali sintax-sintax dasar yang berhubungan dengan MySQL. Namun kita akan berusaha untuk merancang bagaimana membuat desain database yang baik. Fokusnya agar kita dapat terbiasa dengan design database yang diterapkan pada aplikasi PHP. Untuk bahasan bab ini akan mengambil studi kasus database pada suatu toko buku online.

Skema Database

Kumpulan design tabel database dikenal dengan skema database. Skema ini akan menentukan cetak biru database. Suatu skema harus memperlihatkan tabel-tabel

beserta kolom-kolomnya, kemudian tipe data dari kolom, dan menunjukkan tipe data tambahan lainnya seperti primary key.

Sebuah skema umumnya tidak memasukkan data apapun , namun Anda bisa saja memperlihatkan contoh data dengan skemanya sehingga bisa lebih jelas. Skemanya dapat dibuat dengan model *entity relationship diagram* (tidak dibahas dalam modul ini) atau dapat dibuat dalam bentuk teks. Misal kita ingin membuat database toko buku online dengan skema berbentuk teks sebagai berikut:

pelanggan (pelID, nama, alamat, kota, kodepos, phone)
pesanan (pesananID, pelID, total, tgl)
buku (isbn, pengarang, judul, harga)

Skema database di atas dibuat dalam bentuk teks. Penyataan yang digaris bawah menujukkan kolom tersebut bersifat primary key (tidak diperkenankan ada data yang sama) . Berikut ini adalah gambaran *real world* tabel-tabel di atas:

Pelanggan

pelID	Nama	alamat	kota	Kodepos	Phone
1	Ade Gartinach	Jl. Kebayoran Lama 10	Jakarta	11001	7202548
2	Sarini Arifah	Jl. Tarogong 42	Garut	41254	558474
3	Bisma Santabudi	Jl. Kosambil 7E	Bandung	40235	25847854
4	Penta Handayani	Jl. Tagog 10	Cimahi	40584	5847841

Pesanan

pesananID	pelID	total	Tgl
1	1	42000	2003-12-20 17:55
2	1	120000	2003-12-22 09:44
3	3	50000	2004-01-07 23:01
4	2	150000	2004-01-15 06:14
5	2	200000	2004-01-16 20:11

Buku

isbn	pengarang	judul	harga
979-543-082-3	Drs Surayin	Perfect English Grammar	25000
979-20-3778-0	Sudantha Wirija	Microsoft Windows 2000 Server	40000
979-533-913-3	Edi S. Mulyanta	Menggunakan Corel Draw 11	30000

Tabel 2.1. Skema awal database terdiri dari tabel pelanggan, order, dan buku

Bagaimana Mendesign Database

Dalam mendesain sebuah database, kita mungkin harus sedikit melakukan corat coret dulu di atas kertas. Di situ kita merancang skema tabel data serta relasi seperti apa yang dibutuhkan. Hal ini sangat penting untuk kita lakukan agar database kita lebih efektif dan efisien.

Mengenai design database, Anda dapat membaca secara panjang lebar tentang *entity relationship diagram* dan *normalisasi* database di buku khusus tentang database. Namun Anda dapat mengikuti beberapa prinsip dasar design database yang akan kita bahas di depan *base on* kasus database toko buku online di atas.

a. Renungkan objek-objek apa yang akan dimodelkan

Ketika Anda membuat sebuah database, Anda biasanya memodelkan aspek-aspek di dunia nyata serta hubungannya. Umumnya, setiap kelas objek dunia nyata yang dimodelkan membutuhkan tabel tersendiri.

Pada database buku online di atas, kita ingin menyimpan informasi tentang pelanggan, buku yang kita jual dan informasi detil order. Order mempunyai informasi ten-

tang tanggal, total harga yang harus dibayar, dan buku yang dipesan. Buku mempuai informasi tentang ISBN, pengarang, judul dan harga.

Kasus ini mengharuskan kita membuat setidaknya tiga tabel dalam database: pelanggan, order dan buku. Skema awalnya dapat Anda lihat pada tabel 1.

b. Hindari Data yang bersifat berlebihan (rendundant)

Anda mungkin bertanya, “mengapa tidak menempatkan nama dan alamat Sarini Arifah pada tabel order?”. Jawabnya, jika sarini memesan buku beberapa kali, maka kita akan menyimpan datanya dalam beberapa baris. Jika demikian, maka tabel order akan berbentuk seperti tabel di bawah ini:

	total	tgl	nama	alamat	Kota
1	42000	2003-12-20 17:55	Sarini Arifah	Jl. Tarogong 42	Garut
2	120000	2003-12-22 09:44	Sarini Arifah	Jl. Tarogong 42	Garut
3	40000	2004-01-07 23:14	Sarini Arifah	Jl. Tarogong 42	Garut

Tabel 11.2. Sebuah desain database yang memerlukan ruang ekstra

Di sini terdapat dua permasalahan, pertama adalah terdapat ruang data yang berlebihan (sampah). Mengapa menyimpan data detil Sarini hingga tiga kali jika kita bisa menyimpannya dengan cukup sekali saja?

Masalah yang kedua adalah bahwa desain ini akan membawa kita pada keadaan anomali, yakni situasi dimana jikalau kita melakukan perubahan pada database maka akan berakhir dengan data yang inkonsisten. Integritas data dilanggar dan kita tidak lagi tahu data yang mana

yang tidak benar. Umumnya ini akan menyebabkan hilangnya informasi.

Jika Sarini pindah rumah ketika dia sedang menunggu pesanan, pasti kita diharuskan untuk mengubah alamatnya sebanyak tiga kali. Jika kita hanya melakukan satu kali perubahan saja (sesuai dengan tanggal order), maka ini akan menyebabkan data yang inkonsisten.

Sebagai solusi , kita harus mendesign database sehingga masalah di atas tidak terjadi. Caranya yakni dengan menempatkan nomor pelanggan yang bersifat unik (`pelID`) di tabel order sehingga dari situ kita bisa melakukan link ke tabel pelanggan yang memuat data detail pelanggan. Dengan demikian apabila terjadi perubahan misal alamat, maka tinggal mengubahnya di dalam satu tabel, yakni tabel pelanggan. (lihat tabel 1).

c. Gunakan Satu Data Spesifik Dalam Satu Kolom

Hal ini berarti di setiap kolom pada masing masing baris, disarankan hanya menyimpan satu hal saja secara spesifik. Berikut ini contoh design tabel data yang kurang baik:

pesananiD	pellID	total	Tgl	pesanan
1	3	100000	2003-12-20 17:55	979-543-081-3
2	1	124500	2003-12-25 17:00	979-543-012-3, 979-541-001-3
3	2	100000	2003-12-27 19:00	979-543-082-1
4	3	650000	2004-01-20 18:00	979-543-082-4, 979-743-082-3, 979-543-082-0

Tabel 11.3. Dengan design ini, atribut buku yang dipesan mempunyai nilai yang yang sulit dianalisis

Design tabel dan pemasukkan data di atas bukanlah ide yang bagus untuk beberapa alasan. Apa yang kita lakukan di atas adalah menyarangkan beberapa item pesanan ke dalam sebuah kolom (kolom pesanan). Ketika Anda melakukan design seperti ini, maka pertanyaan berikut akan sulit dijawab: “Berapa banyak buku yang berjudul *PHP Beginner* telah terjual?” System tidak akan sanggup menghitungnya dengan cepat dan tepat.

Hal ini disebabkan dalam satu kolom data terdapat lebih dari satu item pesanan. Solusinya kita harus membuat satu tabel lagi yang dapat menampung item order yang dipesan secara lebih spesifik. Dengan demikian system akan bekerja lebih efektif dan tepat untuk menjawab pertanyaan di atas. Berikut adalah tablenya:

Item_pesanan

pesananiD	isbn	Jumlah
1	0-672-31697-8	1
2	0-672-31487-7	3
2	0-672-35847-8	2
3	0672-854744-1	1

Tabel 11.4 Dengan design seperti ini akan lebih mudah untuk mencari buku yang telah dipesan

Tabel ini memberikan link antara tabel order dan tabel buku. Tipe tabel memiliki hubungan many-to-many antara dua objek—pada kasus ini, satu order bisa terdiri dari banyak buku, dan setiap buku dapat diorder oleh banyak orang.

d. Ajukan pertanyaan-pertanyaan kritis

Lanjutan dari bagian sebelumnya, yang harus kita lakukan adalah mengajukan pertanyaan-pertanyaan kritis yang harus database jawab. Dengan pengajuan pertanyaan ini, design database kita akan menjadi lebih terencana serta menjadi lebih kokoh dan stabil. Apabila ada kasus yang ditemui di tengah jalan, database yang kita buat sudah mempunyai upaya preventif untuk menjawab kasus tersebut. Misal petanyaan: "Buku apa yang menjadi best seller?". Yakinkan bahwa design database yang kita buat mengandung semua data yang diperlukan, dan link yang tepat untuk menjawab pertanyaan di atas.

e. Hindari design yang mengandung banyak atribut kosong

Jika kita ingin menambahkan ulasan buku ke dalam database, setidaknya terdapat dua cara yang dapat kita lakukan. Dua pendekatan ini tergambar pada tabel sebagai berikut:

Buku

isbn	pengarang	judul	harga	ulasan
979-543-082-3	Drs Surayin	Perfect English Grammar	25000	
979-20-3778-0	Sudantha Wirija	Microsoft Windows 2000 Server	40000	
979-533-913-3	Edi S. Mulyanta	Menggunakan Corel Draw 11	30000	

Ulasan_buku

isbn	ulasan

Tabel 11.5. Untuk menambah ulasan, kita dapat menambahkan sebuah kolom baik itu di tabel buku maupun dengan membuat tabel khusus satu lagi

Cara pertama berarti menambah sebuah kolom ulasan ke dalam tabel buku. Dengan cara ini, akan terdapat sebuah field untuk ulasan yang harus ditambahkan ke dalam setiap buku. Jika banyak buku ada dalam database, dan para pengulas tidak berencana untuk mengulaskannya semua, maka akan banyak baris yang tidak mempunyai nilai di atribut ini. Hal ini dinamakan mempunyai nilai null.

Mempunyai nilai null pada database adalah ide yang buruk. Hal ini akan memboroskan ruang penyimpanan dan menyebabkan masalah ketika dilakukan pengolahan data. Ketika pengguna melihat sebuah null di tabel, mereka tidak mengetahui apakah atribut ini irrelevant, apakah ada kesalahan di dalam database, atau apakah datanya belum dimasukkan.

Umumnya, Anda dapat menghindari masalah yang berkenaan dengan nilai null dengan menggunakan design alternatif. Pada kasus ini kita dapat menggunakan design yang kedua (*ulasan_buku*). Di sini, hanya buku yang diulas terlist di dalam tabel *ulasan_buku* beserta dengan ulasannya.

f. Kelompokkan data berdasarkan kategori

Pengelompokkan data berdasarkan kategori akan sangat mempermudah dalam pengolahan data. Pada umumnya di suatu toko buku online , setiap buku yang dijual diberikan suatu kategorisasi, misalkan ada buku berkategori komputer, rohani, manajemen, sosial, teknik, musik, dan sebagainya. Dengan penambahan kategorisasi ini,

para pembeli akan lebih mudah untuk mencari buku yang mereka inginkan. Kita dapat menambahkan sebuah field kategori di tabel buku. Agar lebih efektif, langkah baiknya kita membuat satu table kategori seperti di bawah:

Kategori

katID	keterangan
1	Komputer
2	Teknik
3	Rohani
4	Bahasa

Dengan penambahan kategori, tabel buku akan menjadi sebagai berikut:

Buku

isbn	pengarang	judul	harga	katID
979-543-082-3	Drs Surayin	Perfect English Grammar	25000	4
979-20-3778-0	Sudantha Wirija	Microsoft Windows 2000 Server	40000	1
979-533-913-3	Edi S. Mulyanta	Menggunakan Corel Draw 11	30000	1

Table 11.6. Kategorisasi akan mempermudah administrasi dan pengolahan data

Implementasi di MySQL

Setelah kita melakukan perencanaan design database untuk toko buku online di atas, maka dapat disimpulkan design databasenya berisi tabel-tabel beserta atributnya sebagai berikut:

```
pelanggan (pelID, nama, alamat, kota, kodepos, phone)
pesanan (pesananID, pelID, total, tgl)
item_pesanan(pesananID, isbn, jumlah)
buku (isbn, pengarang, judul, penerbit, harga, katid,
deskripsi)
```

kategori (katid, keterangan)
ulasan_buku (isbn, ulasan)

Dengan skema database di atas, maka tinggal implementasikan skrip SQL untuk membuat database di atas.

Berikut ini adalah skriptnya:

```
create database toko_buku;
use toko_buku;

create tabel pelanggan (
    pelID int unsigned not null auto_increment primary
key,
    nama varchar(40) not null,
    alamat varchar(40) not null,
    kota varchar(20) not null,
    kodepos char(6)
);

create tabel pesanan(
    pesananID int unsigned not null auto_increment pri-
mary key,
    pelID int unsigned not null,
    total float,
    tgl datetime not null
);

create tabel item_pesanan (
    pesananID int unsigned not null,
    isbn char(13) not null,
    jumlah tinyint unsigned not null,
    primary key (pesananID, isbn)
);

create tabel buku (
    isbn char(13) not null primary key,
    pengarang varchar(30),
    judul varchar(60),
    penerbit varchar(60),
    harga float not null,
    katid int unsigned,
    deskripsi varchar(255)
);

create tabel kategori(
```

```
    katid int unsigned not null auto_increment primary
key,
    keterangan char(40) not null
);

create tabel ulasan_buku(
isbn char(13) not null primary key,
ulasan text
);
```

Kode 11.1: tokobuku.sql – Skrip SQL untuk database toko buku online

Ekseskusikan skrip di atas dengan menggunakan MySQL Front ataupun PHPMyAdmin. Hasilnya database `toko_buku` beserta tabel-tabelnya akan segera tercipta. Sekarang database `toko_buku` sudah siap menampung data.

Kesimpulan

Dalam bab ini kita telah membahas tips fundamental dalam mendesain suatu database. Memang contoh yang kita ambil adalah untuk database toko buku online. Namun setidaknya ini dapat memberikan kita gambaran dasar bagaimana cara mendesain database untuk kasus-kasus yang lainnya.

Bab 12

Menggunakan Control Session di PHP

Anda mungkin sudah pernah mendengar bahwa “HTTP adalah protokol tanpa keadaan”. Artinya protokol tersebut tidak mempunyai cara built-in untuk memelihara keadaan antara dua transaksi. Ketika seorang user *me-request* satu halaman, diikuti oleh yang lainnya, nah HTTP tidak mempunyai cara untuk memberitahukan bahwa kedua *request* tersebut datang dari user yang sama.

Ide kontrol session digunakan untuk melacak seorang user ketika sedang melakukan kunjungan pada suatu website. Dengan session, server memberikan sebuah identitas tertentu kepada user (pengunjung) dan mengingat identitas tersebut sehingga halaman yang ia buka dapat disesuaikan dengan level otorisasi maupun setting personalnya. Dengan demikian kita dapat melacak

kelakuan user. Kita dapat menerapkannya di shopping cart maupun di form login otentifikasi halaman.

Dasar Kegunaan Session

Session dalam PHP dikendalikan oleh sebuah session ID unik berupa sebuah nomor acak yang bersifat kriptografis (terenkripsi). Session ID ini digenerate oleh PHP dan disimpan pada sisi client untuk sesi waktu tertentu. Ia dapat disimpan di komputer user dalam bentuk cookie, ataupun dilewatkan melalui URL.

Session ID berlaku seperti sebuah kunci yang mengizinkan Anda untuk mendatarkan variable-variable tertentu yang juga dinamakan variable session. Isi dari variabel ini disimpan di server. Session ID adalah satu-satunya informasi yang terlihat di sisi client. Jika, pada waktu hubungan dengan situs Anda, session ID-nya terlihat baik itu melalui sebuah cookie atau URL, Anda dapat mengakses variable session yang tersimpan di server untuk sesi tersebut. Secara default, varibale session disimpan dalam flat file di Server. Barangkali Anda pernah mengunjungi sebuah website yang menyimpan session ID di URL. Jika terlihat ada data acak di URL, maka itu mirip sebagai bentuk kendali session.

Implementasi Session

Langkah dasar dalam menggunakan session adalah:

- Memulai Session
- Mendaftarkan dan menggunakan variable session

- Menghapus variable dan menghancurkan session

Ingat bahwa langkah-langkah ini tidak perlu semuanya terjadi dalam skrip yang sama, dan beberapa diantaranya akan terjadi dalam skrip yang berbeda. Mari kita bahas satu persatu langkah-langkah di atas.

a. Memulai Session

Salah satu cara untuk mengawali sebuah session adalah dengan memanggil function bernama `session_start()`. Anda cukup memanggil function ini sebagai berikut:

```
session_start();
```

b. Mendaftarkan dan menggunakan variable Session

Untuk memerintahkan agar variable dapat dilacak dari satu skrip ke skrip yang lain, Anda perlu mendaftarkannya dan menggunakan dengan memanggilnya dengan `$_SESSION['variable_session']`.

```
$myvar=7;  
$_SESSION['var_ses']=$myvar;
```

c. Menghapus Variable dan Menghancurkan Session

Ketika Anda selesai dengan sebuah variable session, Anda dapat menghapusnya menggunakan function `unset()` seperti berikut:

```
Session_unset('var_ses');
```

Lagi-lagi, function ini membutuhkan nama dari variable yang ingin Anda hapus tanpa simbol \$. Function ini hanya dapat menghapus sebuah variable session sekali saja. Namun Anda dapat menggunakan function `session_unset()` untuk menghapus seluruh variabale-variable session.

Ketika Anda selesai dengan sebuah session, pertama-tama Anda harus menghapus seluruh variable session dan kemudian memanggil function:

```
session_destroy();
```

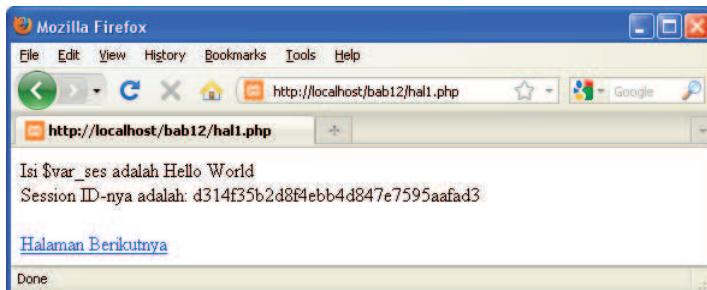
untuk membersihkan session ID.

Contoh Session Sederhana

Berikut ini kita akan mengimplementasikan contoh penggunaan session sederhana. Kita akan menerapkannya pada tiga halaman. Listing kodennya tercantum seperti di bawah:

```
<?php  
session_start();  
  
$_SESSION['var_ses'] = 'Hello World';  
echo "Isi \$_var_ses adalah ". $_SESSION['var_ses']."<br>";  
echo "Session ID-nya adalah: ".session_id()."<br><br>";  
?>  
<a href="hal2.php">Halaman Berikutnya</a><a href =  
"hal2.php">Halaman Berikutnya</a>
```

Kode 12.1: hal1.php – Memulai session dan mendaftarkan variable

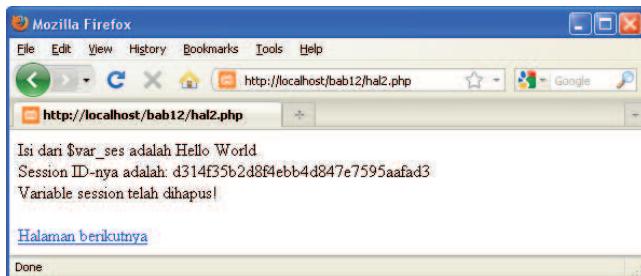


Gb.12.1 Nilai awal dari variable session nampak pada hal1.php

Contoh di atas, kita menset nilainya setelah variabel session telah didaftarkan. Kita juga dapat melakukan yang sebaliknya – menset nilainya kemudian mendaftarkan variablenya. Nilai variable session pada halaman adalah tetap yang akan berlaku pada halaman berikutnya melalui pemanggilan `session_start()`. Skrip berikutnya (hal2.php) ditampilkan dengan kode berikut ini:

```
<?php
session_start();
echo "Isi dari \$var_ses adalah ".$_SESSION['var_ses']."<br>";
echo "Session ID-nya adalah: ".$_SESSION['id']."<br>";
session_unset('var_ses');
echo "Variable session telah dihapus!<br><br>";
?>
<a href = "hal3.php">Halaman berikutnya</a>
```

Kode 12.2: hal2.php – Mengakses variable session kemudian menghapusnya (`session_unset`).

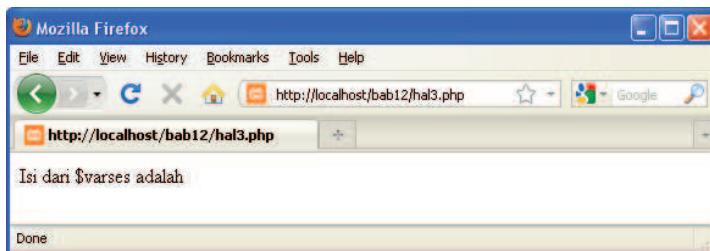


Gb.12.2 Nilai dari variable session telah dilewatkan ke hal2.php kemudian dihapus (deregister).

Setelah menggunakan variablenya, kita memanggil `session_unset()` untuk menghapus variable. Melalui cara ini, session masih exist, namun variable `$var_ses` tidak lagi menjadi variable yang didaftarkan. Akhirnya kita melewatkannya session ke `hal3.php`. Kodenya adalah sebagai berikut:

```
<?php
session_start();
echo "Isi dari \$var_ses adalah ".$_SESSION['var_ses']."<br>";
session_destroy();
?>
```

Kode 12.3: hal3.php – Variable yang telah dihapus sudah tidak berlaku lagi



Gb.12.3 Variable yang telah dihapus sudah tidak berlaku lagi

Studi Kasus: Implementasi Otentifikasi Melalui Kontrol Session

Akhirnya, kita kan melihat contoh yang lebih penting dalam penggunaan kontrol session. Mungkin penggunaan yang paling umum dari kontrol session adalah untuk menjaga jejak user setelah mereka terautentifikasi melalui mekanisme login. Pada contoh ini, kita akan mengkombinasikan otentifikasi dari database MySQL dengan menggunakan session.

Contoh ini terdiri dari tiga skrip sederhana. Pertama adalah `login.php` berupa sebuah form login dan otentifikasi untuk anggota website. Kedua adalah `member.php` yang menampilkan informasi kepada anggota yang telah berhasil login. Yang ketiga adalah `logout.php` berupa skrip yang digunakan untuk menghancurkan session.

Namun sebelumnya kita harus membuat sebuah database otentifikasi yang berisi tabel keanggotaan user yang memiliki kewenangan login. Skrip SQL-nya adalah sebagai berikut:

```
create database otentik;
use otentik;
create tabel user (
    nama varchar(10) not null,
    pass varchar(30) not null,
    primary key (nama)
);
insert into user values ('user', password ('pass'));
```

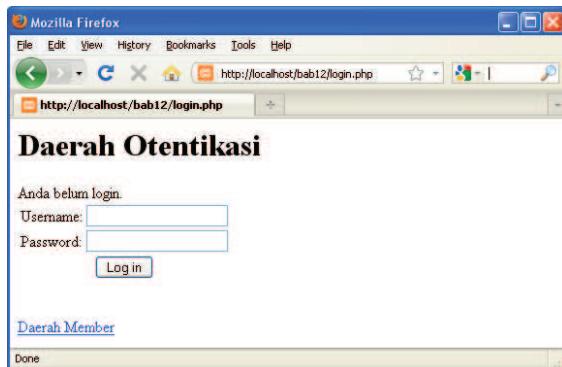
Kode 12.4: `otentik.sql` – Skrip untuk membuat database otentifikasi yang berisi tabel
user

Skrip di atas, kita telah membuat sebuah username bernama “user” dengan password “pass”. Apabila kita melihat tablenya, maka password telah terenkripsi (karena kita menggunakan function `password` ketika melakukan proses insert.

nama	pass
user	29bad1457ee5e49e

Gb.x Table user, dimana passwordnya telah terenkripsi

Untuk memperjelas intuisi kita, berikut ditampilkan halaman login.php:

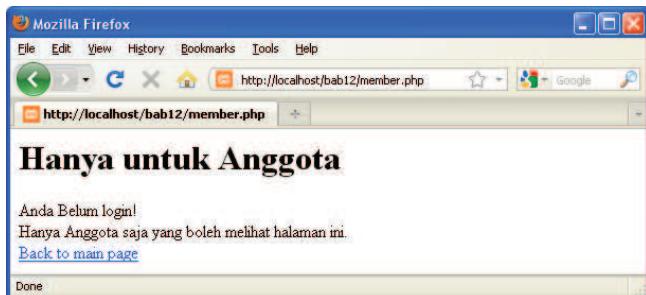


Gb. 12.4 Halaman otentifikasi login

Jika username dan paswordnya invalid ataupun kosong, maka akan nampak peringatan bahwa username dan password kita invalid.

Jika kita mengklik daerah member (member.php) tanpa login terlebih dahulu, maka akan muncul halaman

■ Aplikasi Berbasis Web dengan PHP & MySQL *Learning By Doing and Make it Simple!*
sebagai berikut:



Gb. 12.5 Tampilan halaman ketika belum terjadi login atau username dan password keliru

Apabila username dan password cocok, maka ia akan otomatis menampilkan link ke halaman member.



Gb.12.6 Halaman ketika user berhasil login

Kemudian dengan megklik link daerah member, kita bisa masuk ke dalam halaman member seperti di bawah ini:



Gb.12.7 Hasil otentikasi mengalihkan ke halaman untuk member

Sekarang mari kita lihat kode aplikasinya.

```
<?php
session_start();

if ( isset($_POST[username]) && isset($_POST[sandi]))
{
    // jika user berusaha untuk login
    // lakukan koneksi ke database
    mysql_connect("localhost", "root", "hallo");
    mysql_select_db("otentik");

    //cek ke tabel apakah username dan sandinya cocok
    $sql = "select * from user where nama='".$_POST[username]' and pass=password('$_POST[sandi]')";
    $hasil = mysql_query($sql);
    if (mysql_num_rows($hasil) >0 ){
        // jika ia ada di dalam database, daftarkan username sebagai varibel session
        $_SESSION['valid_user'] = $_POST[username];
    }

}
?>
<html>
<body>
<h1>Daerah Otentikasi</h1>
<?
    if (isset($_SESSION['valid_user'])){
        echo "Anda Login Sebagai: ".$_SESSION['valid_user']."<br>";
        echo "<a href=\"logout.php\">Log out</a><br>";
    }else{

```

```
if (isset($_POST[username])){
    // jika mereka berusaha masuk dan ternyata ga-
gal
    echo "Username dan Password Invalid!";

}

else{
    //mereka belum berusaha masuk atau sudah
logout
    echo "Anda belum login.<br>";
}

// Tampilkan Form Login
echo "<form method=post action=\"login.php\">";
echo "<tabel>";
echo "<tr><td>Username:</td>";
echo "<td><input type=text name=username></td></
tr>";
echo "<tr><td>Password:</td>";
echo "<td><input type=password name=sandi></td></
tr>";
echo "<tr><td colspan=2 align=center>";
echo "<input type=submit value=\"Log in\"></td></
tr>";
echo "</tabel></form>";
}
?>
<br>
<a href="member.php">Daerah Member</a>
</body>
</html>
</html>
```

Kode 12.5: login.php – Bagian utama dari aplikasi otentifikasi

Aktivitas skrip berkisar pada variable session \$valid_user. Ide dasarnya adalah jika seseorang masuk dengan sukses (username dan passwordnya cocok), kita akan mendaftarkan sebuah varibale session \$val-id_user yang mengandung usernamenya. Setelah itu pada halaman tersebut akan diberikan link untuk masuk ke

dalam halaman member (member.php). Kodenya adalah sebagai berikut:

```
<?
    session_start();

    echo "<h1>Hanya untuk Anggota</h1>";
    // check variable session
    if (!isset($_SESSION['valid_user'])){
        echo "Anda Belum login!<br>";
        echo "Hanya Anggota saja yang boleh melihat
halaman ini.<br>";
        echo "<a href=\"login.php\">Back to main page</
a>";
        exit;
    }

    echo "Anda Login sebagai <b>".$_SESSION['valid_
user']."</b><br>";
    echo "<a href=\"logout.php\">Logout</a>";

?>
```

Kode 12.6: member.php – Halaman ini akan mengecek variable session yang dikirimkan sebelumnya

Kode di atas cukup sederhana. Semuanya bermula dengan memulai sebuah session, dan memeriksa jika session yang sekarang mengandung user yang terdaftar dengan menggunakan function `!isset($_SESSION['valid_user'])`. Jika usernya berhasil login, kita memperlihatkan keterangan keanggotaannya. Sebaliknya jika tidak/belum login, akan ditampilkan informasi bahwa dia tidak mempunyai hak untuk masuk ke halaman tersebut.

Akhirnya kita membuat skrip logout.php yang akan memproses penghapusan variable session dan mengakhiri session yang ada. Berikut adalah kodenya:

```
<?php  
session_start();  
  
$old_user = $_SESSION['valid_user']; // untuk men-  
guji apakah mereka pernah login  
session_unset('valid_user');  
session_destroy();  
?>  
  
<html>  
<body>  
<h1>Log out</h1>  
<?  
if (!empty($old_user)){  
    echo "User <b>$old_user</b> telah Logout!.<br>";  
  
} else{  
    // Jika tiba-tiba saja datang ke halaman logout  
    // tanpa login dulu  
    echo "Anda belum login dan juga tidak juga  
    logout!<br>";  
}  
?>  
<a href="login.php">Kembali ke halaman Login</a>  
</body>  
</html>
```

Kode 12.7: logout.php – Skrip untuk menghapus variable session dan mengakhiri (destroy) session

Kode aplikasi di atas cukup sederhana. Kita mulai sebuah session, menyimpan usernamenya di variable \$old_user, menghapus variable session dan mengakhiri session. Kita kemudian memberikan user sebuah pesan bahwa ia telah log out.

Kesimpulan

Pada bab ini kita telah mempelajari satu topik yang cukup fundamental, sebab kontrol session ini banyak digu-

nakan pada aplikasi web seperti webmail, shopping cart, halaman otentikasi, personalisasi dan masih banyak lagi.

Contoh aplikasi otentikasi dapat kita ambil manfaatnya untuk melindungi halaman website yang bersifat sensitif, seperti halaman administrasi untuk pemasukan berita. Ini merupakan satu keharusan bagi sebuah situs berita online agar para wartawannya dapat memasukkan berita kapanpun dan dimana pun ia berada.

Bab 13

Object Oriented dalam PHP

Bahasa pemrograman modern biasanya mendukung atau bahkan membutuhkan suatu pendekatan berorientasi objek. Pengembangan object-oriented (OO) berusaha untuk menggunakan klasifikasi, relasi dan sifat dari objek suatu sistem untuk membantu pengembangan program.

Pada umumnya proyek aplikasi web skalanya kecil, sehingga banyak programmer mengerjakannya dengan cara *straightforward* atau berorientasi pada function. Namun pada proyek yang berskala besar, yang melibatkan banyak programmer, cara *straightforward* akan mengakibatkan hasilnya acak-acakan.

Ini dapat diibaratkan antara proyek membuat rak sepatu dengan proyek membuat rumah. Untuk membuat

sebuah rak, Anda tinggal gergaji sana-sini, paku-paku kemudian jadi deh. Nah beda dengan membuat rumah, dimana Anda tidak akan sembarangan membuatnya dimana harus dibuat aturan dan perencanaan yang matang.

Alternatif yang ditawarkan adalah programming yang berorientasi object. Orientasi objek dapat membantu untuk memanage kompleksitas dari proyek Anda, meningkatkan kinerja serta reusabilitas kode sehingga dengan cara yang demikian ini akan mempermudah dalam penambahan fitur serta biaya pemeliharaan. Cara ini sangat mendukung proyek *Open Source* yang sekarang lagi ngetrend.

Konsep dasar OOP: Abstraksi, Encapsulasi dan Inheritance

Terminologi Object Oriented mengharuskan kita berfikir proses sebagai satu kesatuan. Dengan kata lain, cara kita berfikir mengenai objek sehari-hari. Misalnya sebuah handphone. Untuk menggunakannya, kita tidak perlu mengetahui bagaimana alur kerja di dalamnya. Yang harus kita ketahui adalah bagaimana cara menggunakan interfacenya, misal cara untuk menggunakan keypad, bagian mana untuk bicara dan bagian mana untuk mendengar serta nomor handphone teman yang akan kita hubungi.

Hal ini mirip dengan penggunaan function-function internal PHP. Kita mungkin saja tidak tahu secara pasti alur kerja function tersebut. Kita hanya perlu tahu parameter apa yang harus dilewatkan ke dalam function, dan tipe nilai kembalian (return value) apa yang dihasilkan. Inlah yang

disebut sebagai **abstraksi**.

Di lain pihak, orang yang kita hubungi mempunyai tipe handphone yang berbeda dengan kita. Handphone yang berbeda tersebut dapat berjalan dengan baik meskipun alur kerja internalnya beda. Handphone yang berbeda tersebut tidak perlu tahu alur kerja internal masing-masing, namun hanya perlu tahu bagaimana cara ber-interface satu sama lain. Konsep ini dikenal sebagai **encapsulasi**. Dengan kata lain, implementasi alur internal tetap terpisah dari interface public.

Sebagai tambahan, kita bisa saja membeli sebuah handphone yang ada PDA-nya. Barangkali user interface serta fungsi dasarnya tetap sama, namun ia mempunyai fungsionalitas yang baru. Handphone PDA mewarisi sifat handphone biasa (**inherit**) namun fungsionalitasnya diperluas. Class dan Object diperkenalkan untuk menggabungkan tiga konsep ini yakni abstraksi, encapsulasi dan inheritance ke dalam programming.

Alasan penting untuk mengimplementasikan OOP adalah *encapsulasi* mempermudah koordinasi antar programmer yang bekerja dalam satu proyek. Programmer tidak perlu khawatir mengenai nama variable namun yang lebih penting adalah mereka harus memperhatikan interface dengan class yang lain. Untuk alasan yang sama, encapsulasi dan abstraksi memungkinkan programmer untuk berbagi kode dengan yang lainnya, sebuah praktik yang umum dalam komunitas open source. Lebih jauh,

inheritance memberikan kita kemampuan untuk menambah fungsionalitas pada sebuah class dengan mudah. Jadi ibaratnya tidak perlu lagi membuat “roda” dari awal, hanya tinggal memakaikan “ban”. Jadi deh..

Class dan Object

Object (Objek) adalah representasi sesuatu di dalam dunia nyata. Sesuatu ini bisa apa saja, sebuah benda, sebuah aktivitas dan lain-lain. Kalau di dunia software, objek ini bisa berupa *text input area* ataupun sebuah file. Sedangkan Class adalah cetakan (template) dari sebuah objek. Class mendefinisikan implementasi detail object. Object-object yang sama berada dalam satu class. Sebuah object dibuat berdasarkan definisi class. Sebagai contoh handphone NOKIA adalah sebuah object yang dibuat berdasarkan definisi dari class handphone.

Sebuah class adalah cetakan (template) untuk objek. Ini termasuk properti dan metode yang menggambarkan sebuah objek dan apa fungsi kerjanya. Jika kita mendefinisikan sebuah class untuk handphone, properti class tersebut akan memasukkan informasi spesifikasi batre, serta informasi tambahan seperti jenis layar ataupun ringtone. Properti-properti ini akan digunakan oleh metode class untuk menjalankan tugasnya seperti mengirim/menerima sms, berdering ketika ada yang menelpon dan menyambungkan percakapan.

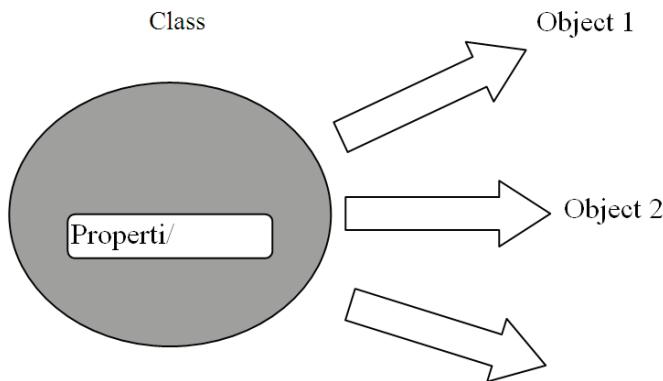
Properti yang lainnya dapat memasukkan karakteristik fisik dari telepon seperti dimensi dan berat. Sekali

template (cetakan) ini didefinisikan, kita dapat membuat handphone sesuai permintaan. Handphone dan objek yang lainnya, dibuat oleh suatu cetakan class yang dikenal dengan **instans**.

Suatu pabrik mungkin menginginkan style handphone yang berbeda yang mempunyai karakteristik fungsional sama. Dalam kasus ini template yang sama dapat digunakan. Namun, jika fungsionalitas tambahan diperlukan seperti halnya sebuah handphone PDA, kita harus menuliskan class yang baru. Dengan menggunakan inheritance kita dapat mengambil properti dan metode yang telah didefinisikan dalam handphone tradisional kemudian tinggal menambahkannya metode dan properti yang diperlukan untuk fungsionalitas tambahan dari sebuah handphone PDA.

Mendefinisikan Class

Sebuah Class mencakup sejumlah variable (properti) yang dimiliki oleh objek dan sejumlah function (metode) yang digunakan untuk mengakses properti.



Gb.13.1 Gambaran Class dan Object

Dalam terminologi pemrograman berorientasi objek, Variable-variable yang berada dalam sebuah class disebut properti, sedangkan function yang digunakan untuk mengakses properti disebut metode.

Bentuk pendefinisan class adalah sebagai berikut:

```
Class nama_kelas{  
    [ var deklarasi_variable ]  
    [ deklarasi function ]  
}
```

Contoh berikut memberikan gambaran pendefinisan class

handphone:

```
<?php  
Class handphone {  
    //properti  
    var $berat;  
    var $ringtone;  
    var $layar;  
  
    //metode  
    function set_karakter($brt, $ring, $lay) {
```

```
$this->berat = $kg;  
$this->ringtones=$ring;  
$this->layar=$lay;  
}  
}  
?>
```

Kode 13.1: hp.php – Pendefinisian class handphone

Pada contoh di atas, `$this` berarti objek dari kelas itu sendiri (kelas handphone). Tanda `->` digunakan untuk mengakses properti dan metode. Sebagai contoh:

`$this->berat`

dibaca sebagai properti berat milik objek itu sendiri.

Sedangkan pernyataan:

`$this->berat = $brt;`

merupakan instruksi untuk mengisikan parameter `brt` ke properti `berat`.

Menciptakan Objek (Instansiasi)

Setelah class didefinisikan, objek (seringkali disebut instan class) dapat dibentuk melalui pernyataan `new`.

Sebagai contoh:

```
$nokia_3350 = new handphone();  
$siemens_A52= new handphone();
```

pernyataan di atas merupakan instruksi untuk menciptakan dua buah objek berkelas handphone. Objek pertama disimpan pada variable `nokia_3350` dan objek kedua pada variable `siemens_A52`. melalui kedua variable tersebut kelak Anda dapat mengakses metode atau properti yang dimiliki masing-masing objek.

Mengakses Properti dan Metode

Setelah suatu objek berkelas handphone diciptakan, Anda bisa mengisi properti \$berat, \$ringtone dan \$layar dengan memanggil metode set_karakter seperti berikut:

```
$nokia_3350->set_karakter (100, "Monophonic", "Hitam Putih");
```

Dengan menggunakan pernyataan di atas, amaka:

- Properti berat berisi 100
- Properti ringtone berisi Monophonic
- Properti layar berisi hitam putih]

Tanda -> juga dapat digunakan untuk mengakses properti.

Sebagai contoh

```
echo $nokia_3350->berat;
```

akan menampilkan properti berat milik objek yang disimpan pada variable nokia_3350.

Skrip berikut memperlihatkan contoh penciptaan objek berkelas hanphone, pengaksesan metode set_karakter dan juga properti dalam objek.

```
<?php  
include "hp.php";  
  
$nokia_3350 = new handphone();  
$nokia_3350->set_karakter(100,"monophonic","Hitam Putih");  
  
echo "<h3>Karakteristik Nokia 3350</h3>";  
echo "Berat : ". $nokia_3350->berat ." gram <br>";  
echo "Ringtone: ". $nokia_3350->ringtone. "<br>";  
echo "Layar: ". $nokia_3350 -> layar;  
?>
```

Kode 13.2: call.php – Pemanggilan metode dan properti objek nokia_3350

Hasil keluarannya adalah sebagai berikut:

Karakteristik Nokia 3350

Berat : 100 gram

Ringtones: monophonic

Layar: Hitam Putih

Coba perhatikan kembali class handphone. Sepintas function set_karakter() isinya sepele atau tanpa nilai. Di dalam class function yang isinya seperti ini dinamakan function pengakses. Alasan untuk membuat function pengakses adalah sederhana, yakni: Kita akan mempunyai metode (berupa function) yang dapat mengakses nilai properti tertentu di luar class.

Alasan lainnya yakni adalah bahwa bukan ide yang baik jika kita mengakses properti secara langsung di luar class. Inilah salah satu keuntungan pendekatan objek oriented adalah bahwa ia menganjurkan encapsulasi.

Berikut ini adalah perbandingan class tanpa function pengakses dan dengan memakai function pengakses.

```
class handphone{  
    var $berat; //properti  
}  
  
//membuat objek  
$nokia_3350 = new hanphone();  
$nokia_3350->berat=100; //bukan cara yang baik men-  
gakses properti di luar class  
echo $nokia_3350->berat;
```

Kode 13.3 : no_akses.php – pendekrasian class tanpa function pengakses (not recommended)

```
class handphone{  
    var $berat;  
  
    //Metode (function pengakses)
```

```
function set_berat($brt) {  
    $this->berat=$brt;  
}  
}  
//membuat objek  
$nokia_3350= new handphone();  
echo $nokia_3350->set_berat(100); //cara yang baik  
dgn memanggil function  
                                            //yang dapat mengak-  
ses properti berat
```

kode 13.4: akses.php –pendeklarasian class dengan function pengakses (recommended)

Setelah melihat kode di atas, Kita dapat merasakan bahwa class dengan function pengakses lebih mudah untuk digunakan.

Konstruktor

Pada umumnya class akan mempunyai metode khusus yang dinamakan konstruktor. Sebuah konstruktor dipanggil ketika sebuah objek diciptakan, dan secara normal ia menampilkan inisialisasi awal seperti pemasangan nilai bawaan pada properti atau membuat objek lain yang diperlukan oleh objek kelas.

Sebuah konstruktor dideklarasikan dengan cara yang sama seperti metode yang lain, namun mempunyai nama yang sama dengan class. Meskipun kita dapat memanggil konstruktor secara manual, maksud utamanya adalah ia akan dipanggil secara otomatis ketika sebuah objek dibuat. Kode berikut mendeklarasikan sebuah class dengan sebuah konstruktor:

```
<?php  
class handphone {
```

```
//properti
var $berat;
var $ringtone;
var $layar;

//Konstruktor
function handphone($gram=100, $ring="polyphonic",
$lay="warna"){
    $this->berat = $gram;
    $this->ringtone = $ring;
    $this->layar = $lay;
    echo "<p>Disini konstruktor dijalankan!</p>";
}

//metode
function set_karakter($gram, $ring, $lay) {
    $this->berat = $gram;
    $this->ringtone = $ring;
    $this->layar = $lay;
}

?>
```

Kode 13.5: hp2.php – Class dengan konstruktor

Bila anda bermaksud tidak memberikan nilai bawaan, maka judul function akan ditulis seperti berikut:

```
function handphone($gram, $ring, $lay)
```

Berikut akan terlihat bagaimana konstruktor di dalam class bekerja:

```
<?php
include "hp2.php";

$nokia_3530 = new handphone();
echo "<h3>Karakteristik Nokia 3530</h3>";
echo "Berat : ". $nokia_3530->berat ." gram <br>";
echo "Ringtone: ". $nokia_3530->ringtone. "<br>";
echo "Layar: ". $nokia_3530 -> layar;

$samsung_220 = new handphone(150,"Monophonic","Hitam
Putih");
```

```
echo "<h3>Karakteristik Samsung 220</h3>";
echo "Berat : ". $samsung_220->berat ." gram <br>";
echo "Ringtone: ".$samsung_220->ringtone. "<br>";
echo "Layar: ".$samsung_220 -> layar;
?>
```

Kode 13.6: call2.php – Penciptaan objek class yang ada konstruktornya

Hasil keluarannya adalah seperti berikut:

Disini konstruktor dijalankan!

Karakteristik Nokia 3530

Berat : 100 gram

Ringtone: polyphonic

Layar: warna

Disini konstruktor dijalankan!

Karakteristik Samsung 220

Berat : 150 gram

Ringtone: Monophonic

Layar: Hitam Putih

Inheritance dalam PHP

Salah satu kelebihan pemrograman berorientasi objek adalah adanya sifat pewarisan (*inheritance*). Sebagai contoh, Anda bisa membuat class baru bernama handphone_pda yang mewarisi class handphone akan mencakup properti berat, ringtone, layar yang diperluas oleh properti prosesor dan os (operating system) .

Dengan melakukan pendekatan seperti ini Anda tidak perlu menulis kembali kode-kode yang telah ada pada class handphone, tetapi cukup memperluas class tersebut pada class baru. Contohnya dapat dilihat di bawah ini:
<?php

```
include_once "hp2.php";

class hp_pda extends handphone{
    //properti tambahan
    var $prosesor;
    var $os;

    //konstruktor
    function handphone_pda($gram, $ring, $lay, $pros,
    $os) {
        $this->berat = $gram;
        $this->ringtone = $ring;
        $this->layar = $lay;
        $this->prosesor = $pros;
        $this->os = $os;
        echo "<p>Di sini konstruktor dijalankan!</p>";
    }

    //Metode
    function set.hardware() {
        $hard = "Berat : $this->berat gram <br>";
        $hard .= "Prosesor : $this->prosesor <br>";
        $hard .= "Layar : $this->layar";
        return $hard;
    }

    function set.software() {
        $soft = "Operating System : $this->os <br>";
        $soft .= "Ringtone : $this->Ringtone";
        return $soft;
    }
}

?>
```

Kode 13.7 : hp_pda.php – class handphone_pda diturunkan dari class handphone

Pada skrip di atas :

```
class handphone_pda extends handphone
digunakan untuk mewariskan class handphone ke class
handphone_pda. Di dalam class handphone_pda :
    var $prosesor;
    var $os;
```

merupakan pendefinisian dua buah properti baru. Tentu saja, selain kedua properti tambahan ini, class `handphone_pda` dengan sendirinya memiliki properti bernama berat, ringtone dan layar.

Berikut ini adalah tampilan kode bagaimana objek class di atas diciptakan:

```
<?php  
include "hp_pda.php";  
  
$sony_p800 = new hp_pda (350,"Polyphonic","Warna",  
"AMD X-80", "Symbian OS");  
echo "<h3>Karakteristik HP PDA Sony P800</h3>";  
echo $sony_p800->set.hardware();  
echo "<br>";  
echo $sony_p800->set.software();  
  
$siemens_Z45 = new hp_pda(400,"Polyphonic","Warna",  
"Crusoe M3","Windows CE");  
echo "<h3>Karakteristik HP Siemens Z45 </h3>";  
echo $siemens_Z45->set.hardware();  
echo "<br>";  
echo $siemens_Z45->set.software();  
  
?>
```

Kode 13.8 : call3.php – Objek hasil inheritance

Berikut ini adalah hasil keluarannya:

Karakteristik HP PDA Sony P800

Hardware

Berat : 350 gram

Prosesor : AMD X-80

Layar : Warna

Software

Operating System : Symbian OS

Ringtone : Polyphonic

Karakteristik HP Siemens Z45

Hardware

Berat : 400 gram

Prosesor : Crusoe M3

Layar : Warna

Software

Operating System : Windows CE

Ringtone : Polyphonic

Kesimpulan

Dasar-dasar OOP yang telah dibahas sebelumnya adalah merupakan bekal yang harus kita fahami. Setidaknya sekarang Anda sudah bisa mengerti jika membaca suatu skrip yang menerapkan OOP. Kita pasti bisa menhami alur skrip OOP tersebut, yakni mana Class, Property, Method dan Objek. Pada bab berikutnya akan dibahas penerapan OOP untuk membuat sebuah template sederhana. Let's Go!

Bab 14

Kasus OOP: Template

Kalau Anda perhatikan dengan seksama, dalam sebuah situs yang memiliki banyak halaman, maka Anda akan melihat walaupun halamannya berbeda tetapi format dasarnya tetap sama. Misalkan dalam sebuah situs hanya ada satu format dasar, dan situs tersebut memiliki 20 halaman. Karena alasana tertentu, setelah semua halaman selesai dibuat, format tampilannya harus diubah, misalkan logo perushaaan yang pada awalnya diletakkan di kiri atas harus diubah menjadi di tengah atas. Apa yang Anda lakukan? Anda harus mengedit ke-20 file yang ada agar semuanya menampilkan logo sesuai format baru.

Untuk mengatasi masalah tersebut biasanya situs-situs besar menggunakan *template*. Dengan *template*, desain tampilan dan skrip dipisahkan. Bentuk tampilan bi-

asanya disimpan sebagai file template atau *theme*. Setiap halaman situs akan merujuk pada theme (format tampilan) tertentu sehingga jika diperlukan adanya perubahan pada tampilan, cukup mengubah pada satu file saja yaitu file theme bersangkutan, dan otomatis semua halaman situs yang merujuk ke theme tersebut akan berubah sesuai dengan bentuk tampilan theme yang baru.

Membuat Template Sederhana

Di internet Anda bisa menjumpai banyak skrip template yang bisa didapat secara gratis seperti fastTemplate, Xtemplate, flexTemplate, Smarty, dan lain-lain. Namun pada proyek ini kita akan menggunakan template buatan sendiri yang sangat sederhana. Skrip template biasanya dibuat dengan konsep OOP karena lebih mudah diimplementasikan dibanding tanpa OOP. Walaupun PHP tidak mendukung OOP 100%, tapi sudah lebih dari cukup untuk keperluan pemrograman bergaya OOP. Pada proyek ini skrip yang dimaksud diberi nama **template.php**. Skripnya adalah sebagai berikut:

```
<?php
//-----
//File template.php
//File untuk keperluan pemisahan desain tampilan,
script dan data
//-----

if (ereg("template.php", $PHP_SELF)) {
    header("location : index.php");
    die;
}

class template {
    var $TAGS = array();
```

```
var $THEME;
var $CONTENT;
function define_tag($tagname, $varname) {
    $this->TAGS[$tagname] = $varname;
}

function define_theme($themename) {
    $this->THEME = $themename;
}

function parse() {
    $this->CONTENT = file($this->THEME);
    $this->CONTENT = implode("", $this-
>CONTENT);
    while(list($key, $val) = each($this-
>TAGS)) {
        $this->CONTENT = ereg_
replace($key, $val, $this->CONTENT);
    }
}

function cetakproses() {
    echo $this->CONTENT;
}

?
?>
```

Kode 14.1 template.php – contoh format template sederhana

Pada skrip di atas terdapat tiga properti yaitu \$TAGS, \$THEME, dan \$CONTENT. Kemudian disertai empat metode berupa function define_tag(), define_theme(), parse(), dan cetakproses(). Anda mungkin lupa apa itu kegunaan \$this->. Suatu function yang menggunakan \$this disebut function pengakses. Jadi gunanya adalah untuk mengakses properti yang ada di dalam class (misal \$THEME) dengan melewatkannya/mengisikannya sebuah variable (misal \$themename). Meth-

ode-method yang ada pada class di atas akan dijelaskan berikutnya.

Membuat Theme

Situs yang menerapkan template memerlukan file theme atau desain tata letak content. Kali ini akan diberikan contoh file theme untuk keperluan halaman situs toko buku online. Anda bisa membuat file theme lainnya untuk halaman yang memerlukan tata letak yang berbeda. Berikut ini adalah salah satu contoh skrip untuk theme:

```
<html>
<head>
    <title>PALASARI.COM - Toko Buku Online Terbesar di Indonesia</title>
    <link href="palasari.css" rel="stylesheet" type="text/css">
</head>

<body leftmargin="0" topmargin="0">
<TABLE border="0" width="100%" align="center">
<TR><TD>
    <TABLE border="0" width="760" align="center">
        <TR><TD>
            <DIV class="header" align="center">::: Citarum Online Shopping :::</DIV>
            <DIV class="menu" align="center" style="width:760px">
                <a href="index.php">HOME</a> | <a href="berita.php">BERITA</a> | <a href="shop.php">KERANJANG BELANJA</a>
            </DIV><p>
            <TABLE class="bodytext" border="0" width="760" align="center">
                <TR>
                    <TD valign="top" width="200"> {MENUKIRI} </TD>
                    <TD valign="top" width="500"> <DIV style="text-align:justify">{ISI}</DIV>
                </TR>
            </TABLE>
        </TD>
    </TR>
</TABLE>
</body>
```

```
</TD>
</TR>
</TABLE>
<TR><TD>
<DIV align="center">&copy;2004</DIV>
</TD></TR>
</TD></TR>
</TABLE>
</TD></TR>
</TABLE>

</body></html>
```

Kode 14.2 theme1.htm – File theme untuk tata letak content situs

Jika diperhatikan pada file theme tersebut meyertakan file CSS bernama citarum.css. Isi file tersebut adalah :

```
.bodytext{
    font-family: verdana;
    font-size: 11px;
}

a{
    text-decoration: none;
    color: navy;
}

a:hover{
    text-decoration: underline;
    color: red;
}

.header{
    font-family: verdana;
    font-size: 18px;
    border: solid Navy 1px;
    text-align: center;
    vertical-align: middle;
    background-color: #84D7FF;
    color: Black;
    font-weight: bold;
}

.footer{
```

```
    font-family:verdana;
    font-size: 12px;
    border: solid Navy 1px;
    text-align: center;
    background-color: #84D7FF;
    color: Black;
}

.menu{
    font-family: verdana;
    font-size: 12px;
    color:Navy;
    font-weight:bold;
}

.boxtext{
    font-family: verdana;
    font-size: 12px;
    border: 1px solid navy; }
```

Kode 14.3 palasari.css – Skrip untuk CSS theme

Jika Anda perhatikan pada file theme1.htm di atas, ada tag yang formatnya sebagai berikut:

```
{nama_tag}
```

Contoh : {MENUKIRI}

Tag-tag tersebut nantinya akan diganti dengan data yang bersesuaian. Contoh tag {MENUKIRI} nantinya akan diganti dengan data yang berisi daftar menu kategori. Sedangkan tag {ISI} nantinya akan diisi dengan suatu content misal berita. Ingat sekali lagi bahawa Anda bisa membuat file theme lainnya untuk halaman yang memerlukan tata letak yang berbeda

Tampilan akhir dari theme1.htm:



Gb.14.1 Tampilan nyata file theme1.htm

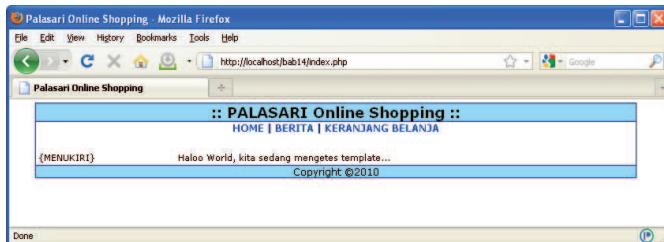
Menggunakan Template

Fungsi skrip template.php adalah untuk menganti tag-tag {nama_tag} menjadi skrip HTML yang sesuai dengan keinginan. Berikut ini adalah contoh sedehana bagaimana menggunakan file template dengan theme:

```
<?php  
//File index.php  
include "template.php";  
  
$test = "Haloo World, kita sedang mengetes tem-  
plate...";  
$title="Palasari Online Shopping";  
  
$tpl = new template;  
$tpl->define_theme("themel.htm");  
$tpl->define_tag("{ISI}", $test);  
$tpl->parse();  
$tpl->cetakproses();  
?>
```

Kode 14.4 index.php – Penggunaan class template beserta themenya

Skrip di atas akan menghasilkan tampilan sebagai berikut:



Gb. 14.2 Tampilan hasil pengetesan class template

Jika kita juga ingin mengisi TAG {MENUKIRI}, maka di skrip index.php tinggal ditambahkan pendefinisian TAG {MENUKIRI}. Skriptnya akan seperti berikut:

```
<?php
//File index.php
include "template.php";

$test = "Haloo World, kita sedang mengetes tem-
plate...";
$kiri ="<b>KATEGORI</b><li>Komputer<li>Ruhani<li>Dew
asa";
$title="PALASARI Online Shopping";

$tpl = new template;
$tpl->define_theme("theme1.htm");
$tpl->define_tag("{TITLE}", $title);
$tpl->define_tag("{ISI}", $test);
$tpl->define_tag("{MENUKIRI}", $kiri);
$tpl->parse();
$tpl->cetakproses();
?>
```

Kode 14.5 index.php – yang telah ditambahkan pendefinisian TAG {MENUKIRI}



Gb.14.3 TAG {MENUKIRI} telah diisi oleh daftar kategori

Penjelasan Skrip

Proses yang terjadi pada skrip-skrip di atas merupakan pekerjaan dari skrip template.php. Pada skrip template.php terdapat sebuah class bernama template. Class template tersebut akan menjadi cetakan bagi halaman-halaman situs. Bentuk cetakannya bergantung pada theme yang dipakai. Oleh karenanya Anda disarankan untuk membuat theme yang lain yang sesuai dengan tata letak situs yang diperlukan.

Perhatikan cuplikan skrip berikut:

```
$tpl = new template;
```

Penyataan tersebut menyatakan bahwa variable \$tpl adalah objek yang dicetakkan dari class template. Untuk selanjutnya objek \$tpl inilah yang akan diproses.

```
$tpl->define_theme ("theme00.htm");
```

Pernyataan di atas memanggil metode define_theme() yang fungsinya untuk menentukan file theme yang akan digunakan. Dalam skrip di atas file theme yang digunakan

adalah file **theme1.htm**.

```
$tpl->parse();
```

Pernyataan di atas memanggil metode `parse()` yang berfungsi untuk membaca file theme untuk diproses lebih lanjut. Pada objek `$tpl` terdapat satu metode yaitu `define_tag()`. Metode `define_tag()` menentukan hubungan antara tag-tag `{nama_tag}` dengan variable-variable yang nantinya akan diproses oleh metode `parse()`. Pendefinisian metode `define_tag()` tersebut, akan menentukan di lokasi mana skrip yang telah diproses akan diletakkan. Hasilnya bisa dilihat pada Gb. 14.3 di atas.

Pada metode `parse()`, skrip:

```
$this->CONTENT = file($this->THEME);
```

Berfungsi untuk mengisi properti `CONTENT` dengan array yang dihasilkan dari pembacaan file theme yang telah didesiniskan oleh properti `THEME`. Function `file()` sendiri berfungsi untuk membaca seluruh isi file menjadi sebuah array.

```
$this->CONTENT = implode("", $this->CONTENT);
```

Function `implode()` berfungsi untuk menggabungkan elemen-elemen array dengan string perekat “”.

```
while(list($key, $val) = each($this->TAGS)) {
```

```
$this->CONTENT = ereg_replace($key, $val,  
$this->CONTENT);  
}
```

Function `list()` dan `each()` digunakan untuk mengakses key dan value dari array asosiatif `TAGS`. Karena kita memanggil `each()` dalam loop while, maka ia terus menerus mengembalikan key dan value setiap elemen array dan berhenti ketika posisi akhir array tercapai.

Function `ereg_replace` berfungsi mencari string (dari `$this->CONTENT`) yang mempunyai pola tertentu (`$key`) untuk digantikan dengan string pengganti (`$val`). Jika tidak ditemukan kecocokan pola, maka string yang dihasilkan tidak akan berubah.

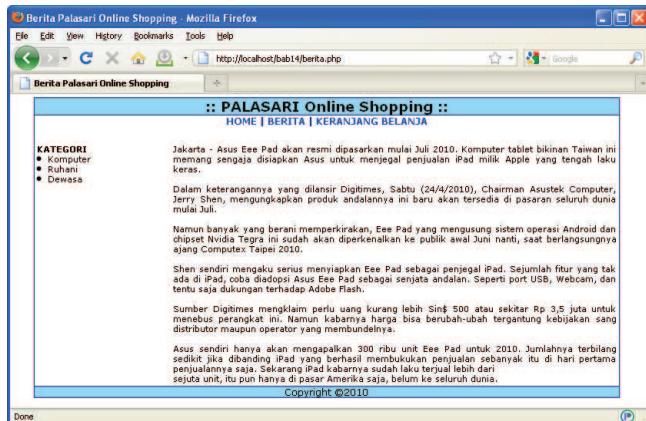
Intinya skrip dia tas digunakan untuk menganti `{nama_tag}` dengan skrip hasil proses, kemudian menempatkannya pada `CONTENT`.

Studi Kasus

Buatlah halaman-halaman dari class template di atas untuk menu HOME, BERITA dan KERANJANG BELANJA sehingga bisa terlihat seperti di bawah ini:



Gb. 14.4 Halaman index.php



Gb. 14.5 Halaman berita.php

Kesimpulan

Bahasan OOP memang memerlukan suatu intuisi dalam mendesain suatu class sesuai dengan kasus yang ditemui. Latihan yang terus-menerus akan sangat membantu kita membiasakan diri dengan OOP. Dengan OOP project yang kita hasilkan akan lebih mudah dipelihara untuk pengembangan selanjutnya. OOP yang kita pelajari pada bab ini lebih kepada konsep penggeneratan halaman. Anda bisa mengembangkan metode-metode yang ada di class tersebut sesuai dengan keperluan. Harap diingat, template yang kita buat pada bab ini akan terus dilakukan pengembangan sesuai dengan bahasan pada bab-bab berikutnya, seperti pada teknik pemasukkan dan pengambilan data serta pada aplikasi Shopping Cart.

Bab 15

Teknik Pemasukan dan Pengambilan Data

Aktivitas pemasukan dan pengambilan data adalah suatu hal yang lumrah dan biasa kita temukan dalam halaman web. Kita pasti sudah pernah mengisi suatu form buku tamu, kemudian melihat isi buku tamu tersebut. Nah itu adalah salah satu contoh nyata aktivitas pemasukan dan pengambilan data. Jika kita bekerja menjadi suatu wartawan portal berita, pasti kita akan sering berinteraksi dengan aktivitas pemasukan data berita ke database, sehingga dengan demikian berita tersebut akan dapat dengan cepat ditampilkan dan dibaca oleh jutaan pengunjung.

Selama ini mungkin aplikasi pemasukan data yang kita buat cukup simple dan apa adanya. Asalkan sudah masuk ke dalam database, ya sudah beres. Tapi ingat, sebuah aplikasi pemasukan data yang baik harus mem-

perhatikan segala aspek yang akan terjadi, misalkan apakah formnya sudah diisi atau belum, kemudian data yang diisikan sudah valid atau tidak, serta pemberian konfirmasi kebenaran data yang dimasukkan.

Dari sisi pengambilan data, kita akan mengupas tuntas tentang paging untuk penampilan data yang banyak. Dengan paging, data yang diambil dari database akan lebih mudah ditampilkan serta lebih enak untuk diakses karena disitu disediakan menu untuk mengakses data-data sebelumnya atau sesudahnya. Penerapan paging pada halaman web merupakan salah satu hal yang mutlak dikehendaki oleh para programmer PHP.

Teknik Pemasukan data

Kita akan mencoba untuk membuat sebuah proyek real untuk memasukkan data, misal berupa form aplikasi buku tamu. Pada aplikasi tersebut, akan digunakan form input text, select dan tombol submit. Akan dijabarkan pula bagaimana mengambil data dari database untuk diisikan ke dalam form. Kemudian kita akan berusaha untuk membuat trap untuk mengecek validitas data yang diisikan user. Seperti biasa, kita akan tetap menggunakan sistem template yang telah dibuat sebelumnya. Langkah yang pertama kita akan membuat databasenya terlebih dahulu.

a. Membuat database buku tamu

Database yang kita buat ini digunakan untuk menampung data yang dimasukkan melalui form. Berikut ini adalah

listing statemen SQL-nya:

```
USE DATABASE palasari;
CREATE TABLE bukutamu (
    id int primary key auto_increment not null,
    tgl date default '0000-00-00',
    pukul varchar(10),
    nama varchar(50),
    lokasi varchar(50),
    email varchar(50),
    pekerjaan varchar(50),
    komentar text
);

CREATE TABLE lokasi (
    id int primary key auto_increment not null,
    kota varchar(50)
);

INSERT INTO lokasi (kota) VALUES ('Jakarta');
INSERT INTO lokasi (kota) VALUES ('Jawa Barat');
INSERT INTO lokasi (kota) VALUES ('Jawa Tengah');
INSERT INTO lokasi (kota) VALUES ('Jawa Timur');
INSERT INTO lokasi (kota) VALUES ('Sumatra');
INSERT INTO lokasi (kota) VALUES ('Bali');
INSERT INTO lokasi (kota) VALUES ('Nusa Tenggara');
INSERT INTO lokasi (kota) VALUES ('Kalimantan');
INSERT INTO lokasi (kota) VALUES ('Sulawesi');
INSERT INTO lokasi (kota) VALUES ('Maluku');
INSERT INTO lokasi (kota) VALUES ('Papua');
INSERT INTO lokasi (kota) VALUES ('Foreign');
```

Kode 15. Skrip gbook.sql – Digunakan untuk membuat database untuk menyimpan data dari form

Dari skrip gbook.sql di atas, kita membuat dua buah tabel, yang pertama adalah tabel bukutamu dan yang kedua adalah tabel lokasi. Untuk tabel lokasi kita sekalian memasukkan data-data lokasi seperti jakarta, jawa barat, dsb. Nantinya kita akan membuat sebuah form select dimana datanya diambil dari tabel lokasi.

Setelah menjalankan skrip sql di atas, langkah berikutnya adalah membuat sebuah function untuk koneksi ke database di atas. Berikut ini adalah function `konek_db()`, dimana penempatanya dipisahkan pada satu file terpisah yang benama `dll.php`.

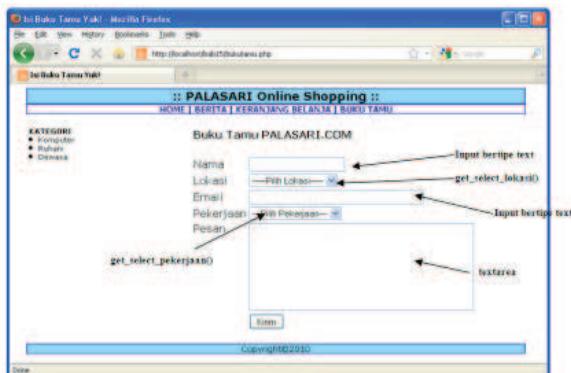
```
<?php
function konek_db() {
    $host="localhost";
    $database="palasari";
    $username="root";
    $password="hallo";
    $koneksi = mysql_connect ($host,$username,$password);
    $pilih = mysql_select_db($database);
    if ($koneksi && $pilih)
        return 1;
    else
        return 0;
}
?>
```

Kode 15.2 Function `konek_db()` – Ada di dalam file `dll.php`

Function di atas akan memproses koneksi ke database. Jika koneksi berlangsung sukses maka return valuenya adalah 1, jika gagal return valuenya akan 0.

b. Membuat Form

Langkah pertama adalah kita akan membuat form masukan untuk aplikasi buku tamu yang akan dibuat. Karena kita menggunakan template, jadi semua format formnya diisikan dalam sebuah variable yang bernama `$isi`. Dengan demikian variable `$isi` tersebut bisa di-parse ke objek yang dihasilkan dari class template. Berikut adalah gambaran form yang hendak akan kita buat:



Gb.15. Form buku tamu yang akan dibuat

Form aplikasi buku tamu di atas melibatkan input bertipe text, textarea dan select. Input bertipe select sangat cocok digunakan untuk membatasi pemasukan dari user karena kita telah menyediakan pilihannya. User tinggal memilihnya saja. Kita akan menggunakan input select ini pada field lokasi dan pekerjaan dengan menuliskannya dalam bentuk function. Untuk field lokasi, datanya akan diambil dari database sedangkan untuk field pekerjaan akan ditulis secara manual dalam function. Berikut adalah skrip halaman buku tamu di atas :

```
<?php  
  
include "template.php";  
include "dll.php";  
koneksi_db();  
  
function get_select_lokasi () {  
    $select="";  
    $s = "<select name=lokasi>";  
    $s .= "<option value=\"\"><b>-----Pilih Lokasi-----</b></option>";  
    $sql ="SELECT * FROM lokasi";
```

```
if ( ($hasil=mysql_query($sql)) && mysql_numrows($hasil) >0 ) {
    while ($row=mysql_fetch_row($hasil)) {
        if ( (isset($_POST['lokasi'])) && ($_POST['lokasi']== "$row[1]" ) ){
            $select="SELECTED";
        }
        $s .= "<option value=\"$row[1]\" $select>$row[1]";
        $select="";
    }
}
$s .="</SELECT>";
return $s;
}

function get_select_pekerjaan(){
$select="";
$ss = "<select name=pekerjaan size=1>";
$ss .= "<option value=\"\" ---Pilih Pekerjaan---";
if ( (isset($_POST[pekerjaan])) && ($_POST[pekerjaan]=="pelajar") )
    $select = "SELECTED";
$ss.= "<option value=pelajar $select>Pelajar";
$select = "";
if ( (isset($_POST[pekerjaan])) && ($_POST[pekerjaan]=="karyawan") )
    $select = "SELECTED";
$ss.= "<option value=karyawan $select>Karyawan";
$select = "";
if ( (isset($_POST[pekerjaan])) && ($_POST[pekerjaan]=="wirausaha") )
    $select = "SELECTED";
$ss.= "<option value=wirausaha $select>Enterpreneur";
$select = "";
$ss.= "</select>";
return $s;
}

$form_buka      ="<FORM method=post action=\"$PHP_SELF?act=1\"><TABLE align=center>";
$form_buka     .="<TR><TD colspan=\"2\"><h4>Buku Tamu PALASARI.COM</h4></TD></TR>";
$form_tutup     ="<TR><TD colspan=\"2\"><INPUT type=submit value=Kirim></TD></TR></TABLE></FORM>";
```

```
//-----//  
//**Di baris ini akan dilakukan cek validitas data  
//-----//  
  
$isi  = $form_buka;  
$isi .= "<TR><TD>Nama </TD><TD><INPUT type=\"Text\" name=\"nama\"></TD></TR>";  
$isi .= "<TR><td>Lokasi </td><td class=kanan>".get_select_lokasi()."</td></tr>";  
$isi .= "<TR><TD>Email</TD><TD><INPUT type=text name=email size=\"40\"></TD></TR>";  
$isi .= "<TR><TD>Pekerjaan</TD><TD>".get_select_pekerjaan()."</TD></TR>";  
$isi .= "<TR><TD valign=top>Pesan</TD>";  
$isi .= "<TD><textarea name=pesan cols=40 rows=7></textarea></TD></TR>";  
$isi .= $form_tutup;  
  
$kiri  = "<b>KATEGORI</b><li>Komputer<li>Ruhani<li>Dew  
asa";  
$title="Buku Tamu PALASARI.Com";  
  
$tpl = new template;  
$tpl->define_theme("themel.htm");  
$tpl->define_title($title);  
$tpl->define_tag("{ISI}", $isi);  
$tpl->define_tag("{MENUKIRI}", $kiri);  
$tpl->parse();  
$tpl->cetakproses();  
  
?>
```

Kode.15.3 bukutamu.php – skrip awal halaman formulir buku tamu

c. Submit Form dan Cek Validitas Data

Form yang kita buat di atas, actionnya mempunyai pernyataan action `$PHP_SELF?act=1`. Artinya jika tombol submit di tekan, maka form akan diproses di halaman yang sama (`$PHP_SELF`) dengan mengirimkan variable `act=1` ke URL, oleh karena itu variable `act` ini akan diakses me-

lalui metode `$_GET[]`. Variable-variable hasil kiriman dari form input juga akan dikirimkan. Variable-variable ini nantinya akan diakses melalui metode `$_POST`.

Setelah pernyataan submit tersebut, maka tugas kita adalah memeriksa validitas data yang telah dikirimkan. Built-in function `empty` akan kita gunakan untuk mengecek apakah field nama, lokasi, pekerjaan dan pesan apakah sudah diisi atau belum. Sedangkan untuk mengecek validitas email, function `cek_email()` akan digunakan. Function `cek_email()` intinya akan menggunakan regular expression untuk mengecek pola alamat email yang dimasukkan user. Berikut ini adalah skripnya:

```
function cek_email($email) {
    if (ereg("^[a-zA-Z0-9_]+@[a-zA-Z0-9\-\-]+\.[a-zA-Z0-9\-\-\.]+\$",
             $email))
        return 1;
    else
        return 0;
}
```

kode 15.4 Function `cek_email()` – Digunakan untuk mencek validitas email. Ditempatkan di `dll.php`

OK, sekarang coba kita tempatkan skrip berikut di baris yang bertanda bintang (**):

```
//... (Daerah skrip sebelumnya)
```

```
$form_kirim="";
$isi ="";
$pesan_error ="";

//Jika form disubmit
if (isset($_GET[act]) && $_GET[act]==1) {

//cek validitas data
    if (empty($_POST[nama])) {
        $pesan_error .= "<li>Nama Belum Diisi</li>";
```

```
        }
        $form_kirim .= "<TR><TD>Nama    </TD><TD><INPUT
type=Text name=nama
value=$_POST[nama]></TD></TR>";

if (empty($_POST[lokasi])) {
    $pesan_error .="<li>Lokasi belum dipilih</li>";
}
$form_kirim .= "<TR><td>Lokasi </td><td>".get_se-
lect_lokasi()."</td></TR>";

if (!cek_email($_POST[email])) {
    $pesan_error .="<li>Email Anda invalid!</li>";
}
$form_kirim .= "<TR><TD>Email</TD><TD><INPUT
type=text name=email size=40
value=$_POST[email]></TD></TR>";

if (empty($_POST[pekerjaan])){
    $pesan_error .="<li>Pekerjaan Anda belum dipi-
lih</li>";
}
$form_kirim.= "<TR><TD>Pekerjaan</TD>
<TD>".get_select_pekerjaan()."</TD></
TD></TR>";

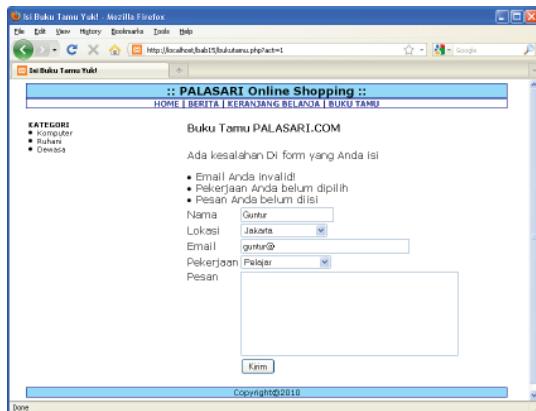
if (empty($_POST[pesan])){
    $pesan_error .="<li>Pesanan Anda belum diisi</
li>";
}
$form_kirim .= "<TR><TD  valign=top>Pesanan </
TD><TD><textarea name=pesan cols=40
rows=7>$_POST[pesan]</textarea></TD></TR>";

//jika pesan errornya sudah tiada, maka proses data
ke database
if (empty($pesan_error)){
    //-----
    //***di baris ini data hasil input akan di insert
ke database
    //-----
    $isi .= "Pesanan sudah Diproses";
} else {
    //jika masih ada pesan error, maka tampilkan pesan
error-nya
    $isi .= $form_buka;
$isi .="<TR><TD colspan=\"2\">Ada kesalahan Di form
```

```
yang Anda isi                                     <P>" . $pesan_er-
ror."</P></TD></TR>";
    $isi .= $form_kirim;
    $isi .= $form_tutup;
}
} else {
//. . . (form utama)
}
```

Kode 15.5 bukutamu.php – halaman bukutamu yang telah dimodifikasi dengan pengecekan validitas data

Dengan penambahan kode untuk mengecek validitas data, maka data yang dimasukkan pengunjung akan secara ketat diperiksa validitasnya apakah sah atau tidak. Di bawah ini adalah contoh pengecian data yang tidak lengkap dimana email dan pesan belum diisi :



Gb.15.2 Contoh hasil penambahan kode untuk mengecek validitas data yang dimaksudkan

Pemasukkan data ke database

Skrip pemasukkan data ke dalam database ditempatkan ketika semua pesan error sudah tidak ada. Kita bisa memasukkannya pada baris yang bertanda ***. Berikut adalah skripnya:

```
//. . . (Bagian cek validasi data)

//jika pesan errornya sudah tiada, maka masukkan data
ke database
if (empty($pesan_error)){
    $pukul = date ("H:i:s");
    $sql_str="INSERT INTO bukutamu
                VALUES ('', curdate(), '$pukul', '$_
POST[nama]', '$_POST[lokasi]', '$_POST[email]', ' $_
POST[pekerjaan]', '$_POST[pesan]' )";

    if (mysql_query($sql_str)){
        $isi .= "Terima Kasih $_POST[nama], pesan Anda
sudah diproses.<br>";
        $isi .= "Kembali ke halaman <a href=\"index.
php\">utama</a>";

        //kirimkan data yang disubmit ke email
        @mail("info@citarum.com","Pengisian buku tamu
dari $_POST[nama]",
            date("d-m-Y H:i:s"). "\n Lokasi : $_POST[lokasi]\n
Pekerjaan : $_POST[pekerjaan] \n Pesan : $_
POST[pesan]",
            "From: $_POST[email]");
    } else {
        $isi .= "Mohon Maaf Database masih dalam per-
baikan";
    }
} else {
    //tampilkan pesan error
    $isi .= $form_buka;
    $isi .="<TR><TD colspan=\"2\">Ada kesalahan Di form
yang Anda isi
<P>". $pesan_er-
ror."</P></TD></TR>";
    $isi .= $form_kirim;
    $isi .= $form_tutup;
}

} else {
```

■ Aplikasi Berbasis Web dengan PHP & MySQL *Learning By Doing and Make it Simple!*

```
//. . . (Bagian form utama)  
}
```

Kode 15.6 bukutamu.php – data yang disubmit lewat form di-insert ke dalam database
dan dikirimkan via email

Pada proses pemasukkan data di atas, data dimasukkan ke dalam database lewat perintah SQL yang mana value dari datanya diambil dari variable-variable yang dikirimkan lewat form. Apabila data-datanya sukses dimasukkan ke dalam database, selanjutnya akan dikirim via email dan pesan TERIMA KASIH akan ditampilkan. Selain itu juga disediakan pilihan link untuk ke halaman utama (index.php) dan ke halaman untuk melihat isi buku tamu (lihat.php).



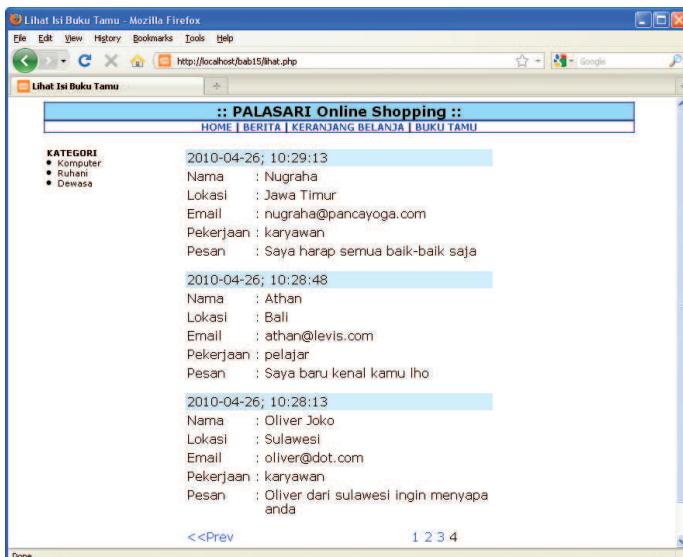
Gb.15.3 Pesan terima kasih apabila buku tamu selesai diisi

Kode di atas kita sekali lagi menggunakan function mail() yang merupakan function built-in PHP yang digunakan untuk mengirim email. Struktur functionnya adalah sebagai berikut:

```
mail (tujuan, subjek email, opsi header);
```

Menampilkan data buku tamu dari database

Untuk menampilkan isi dari buku tamu dari database, kita akan membuat satu halaman lagi yang diberi nama lihat.php. Pada halaman tersebut akan ditampilkan data-data pengisi buku tamu seperti yang terlihat pada gambar di bawah ini :



Gb.15.4 Hasil pengisian buku tamu yang ditampilkan pada halaman lihat.php

Adapun kode untuk halaman lihat.php adalah sebagai berikut:

```
<?php
include "dll.php";
koneksi_db();
$isi ="";
$sql ="SELECT * FROM bukutamu ORDER BY id DESC ";
$hasil = mysql_query($sql);

//tampilkan data
```

■ Aplikasi Berbasis Web dengan PHP & MySQL *Learning By Doing and Make it Simple!*

```
while ($baris = mysql_fetch_row($hasil)) {
    $isi .= "<table width=\"400\" class=\"boxtext\" cellpadding=\"2\">;
    $isi .="<tr><td colspan=\"3\" bgcolor=\"#EFEFEF\">$baris[1];
    $baris[2]</td></tr>";

    $isi .="<tr><td>Nama</td><td>:</td><td width=\"335\">$baris[3]</td></tr>";
    $isi .="<tr><td>Lokasi</td><td>:</td><td>$baris[4]</td></tr>";
    $isi .="<tr><td>Email</td><td>:</td><td>$baris[5]</td></tr>";
    $isi .="<tr><td>Pekerjaan</td><td>:</td><td>$baris[6]</td></tr>";
    $isi .="<tr><td width=\"60\" valign=\"top\">Pesan</td><td valign=\"top\">:</td>
    <td>$baris[7]</td></tr>";

    $isi .= "</table><p>";
}

$kiri ="<b>KATEGORI</b><li>Komputer<li>Ruhani<li>Dewasa";
include "template.php";

$tpl = new template;
$tpl->define_title("Lihat Isi Buku Tamu");
$tpl->define_theme("themel.htm");
$tpl->define_tag("{ISI}",$isi);
$tpl->define_tag("{MENUKIRI}",$kiri);
$tpl->parse();
$tpl->cetakproses();
```

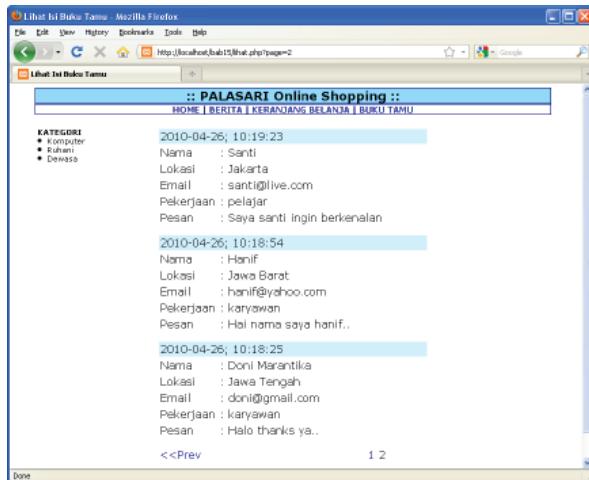
Kode 15.7 lihat.php – Halaman untuk melihat data buku tamu dari database

Paging Buku tamu

Halaman untuk melihat isian buku tamu di atas apabila yang mengisinya semakin banyak, maka pergeseran (scrolling) halaman akan semakin ke bawah. Dan ini tidak baik karena akan membebankan server jika datanya banyak sekali. Apa jadinya jika data buku tamunya mencapai

1000 entry. Pasti loading halamannya akan lama dan kita akan pusing untuk melihatnya.

Nah solusinya kita bisa terapkan sistem paging pada halaman web tersebut sehingga entry yang ditampilkan bisa kita batasi misal cukup 10 entry per halaman. Selain itu juga kita bisa sediakan menu navigasi previous – next untuk beralih ke entry halaman berikutnya. Contoh aplikasinya dapat dilihat pada gambar di bawah ini :



Gb.15.5 Penerapan paging untuk membatasi pengambilan data. Contoh paging 3 entry per halaman

Dan berikut ini adalah modifikasi kode lihat.php dengan penambahan fitur paging :

```
<?php
include "dll.php";
koneksi_db();
$isi = "";
// ======Koding untuk pag-
```

```
ing=====//  
  
//banyaknya post per page  
$per_hal= 3;  
  
// Menghitung jumlah baris di tabel bukutamu  
$sql = "SELECT COUNT(*) FROM bukutamu";  
  
// Lakukan query  
$res=mysql_query($sql);  
  
// ambil jumlah baris dari array pengambilan row tabel  
list($total_post)=mysql_fetch_row($res);  
  
// hitung total banyaknya halaman  
$total_hal = ceil ($total_post / $per_hal);  
  
// default hal: page terakhir  
$page=$_GET[page]; //ambil variable page dari URL  
if (empty($page)) {  
    $page = $total_hal;  
}  
  
// record awal di hal ini  
$batas_awal = ($total_hal-$page)*$per_hal;  
  
// -----NAVIGASI-----//  
$menu = "<tabel align=\"center\" width=\"100%\"><tr>";  
if ($page>1){  
    $menu .= "<td align=\"left\" width=\"10%\">  
              <a href=\"lihat.php?page=".($page-1)."\">&lt;&lt;Prev</a></td> ";  
}  
  
//menu navigasi untuk nomor page  
$menu .=" for ($i=1;$i<=$total_hal;$i++){     if ($page!=$i)         $menu.= " <a href=\"lihat.php?page=$i\">";      $menu.= "$i";      if ($page!=$i)         $menu.= "</a> "; } |
```

■ Aplikasi Berbasis Web dengan PHP & MySQL *Learning By Doing and Make it Simple!*

```
$menu.= "</td>";  
  
if ($page<$total_hal) {  
    $menu.= "<td align=\"right\" width=\"10%\">  
        <a href=\"lihat.php?page=".($page+1)."\">Next&gt;&gt;</a></td> ";  
}  
  
$menu.= "</tr></table>";  
  
//-----Terapkan paging di sintaks SQL -----//  
  
$sql ="SELECT * FROM bukutamu ORDER BY id DESC LIMIT  
$batas_awal,$per_hal ";  
$hasil = mysql_query($sql);  
  
  
  
//tampilkan data  
while ($baris = mysql_fetch_row($hasil)){  
    $isi .= "<table width=\"400\" class=\"boxtext\" cellpadding=\"2\">";  
    $isi.= "<tr><td colspan=\"3\"";  
    $isi.= "<tr><td>$baris[1]; $baris[2]</td></tr>";  
    $isi.= "<tr><td>Nama</td><td>:</td><td width=\"335\">$baris[3]</td></tr>";  
    $isi.= "<tr><td>Lokasi</td><td>:</td><td>$baris[4]</td></tr>";  
    $isi.= "<tr><td>Email</td><td>:</td><td>$baris[5]</td></tr>";  
    $isi.= "<tr><td>Pekerjaan</td><td>:</td><td>$baris[6]</td></tr>";  
    $isi.= "<tr><td width=\"60\" valign=\"top\">Pesan</td><td valign=\"top\">:$baris[7]</td></tr>";  
  
    $isi .= "</table><p>";  
}  
  
//tampilkan menu navigasi  
$isi.=$menu;  
  
$kiri = "<b>KATEGORI</b><li>Komputer<li>Ruhani<li>Dewasa";
```

```
include "template.php";
$tpl = new template;
$tpl->define_title("Lihat Isi Buku Tamu");
$tpl->define_theme("themel.htm");
$tpl->define_tag("{ISI}", $isi);
$tpl->define_tag("{MENUKIRI}", $kiri);
$tpl->parse();
$tpl->cetakproses();
```

?>

Kode 15.8 lihat.php – Telah dilengkapi dengan fitur menu navigasi

a. Konsep Dasar Paging

Mungkin anda bingung mengenai algoritma di atas. Mari kita jelaskan satu per satu. Konsep paging itu sendiri sebenarnya kita menggunakan teknik manipulasi SELECT dari sintaks MySQL dimana kita menentukan batasan dari record yang akan ditampilkan. Untuk mempermudah pemahaman, berikut ini diberikan ilustrasi mengenai konsep paging.

Misal kita mempunyai 10 baris record data di tabel buku tamu sebagai berikut:

id	tgl	pukul	nama	lokasi	email	pekerjaan
1	2003-01-26	10:02:06	Ade	Jawa Barat	ade@yahoo.com	pelajar
2	2003-01-26	10:02:55	Yudi	Jawa Barat	yudi@metro.com	pelajar
3	2003-01-26	10:08:09	Devi	Jawa Barat	Isal@telkom.net	pelajar
4	2003-01-26	10:09:04	Yuyu	Jakarta	ujang@telkom.net	pelajar
5	2003-01-26	13:33:10	Tintin Sumartini	Jakarta	tintin@yahoo.com	wirousaha
6	2003-01-26	13:34:26	Amorita Kusuma Dewi	Jawa Tengah	amor@yahoo.com	wirousaha
7	2003-01-26	13:40:26	Damaryadi	Jawa Barat	damar@telkom.net	karyawan
8	2003-01-12	14:00:12	Dona Harun	Jakarta	dona@detik.com	karyawan
9	2003-01-14	14:00:00	Dina Lorenza	Foreign	dina@yahoo.com	karyawan
10	2003-01-23	11:00:12	Vira Yuniar	Jakarta	vira@yahoo.com	pelajar

Gb.15.6 Listing data dalam tabel buku tamu

Nah jika kita ingin menampilkan datanya sebanyak 3 re-

cord per halaman, maka sintaks SQL yang harus kita gunakan adalah:

```
SELECT * FROM bukutamu LIMIT $batas_awal, $jumlah_pengambilan
```

Untuk pengambilan baris record dari baris ke 1 sampai dengan 3, maka settingan sintaks SQL-nya adalah:

```
SELECT * FROM bukutamu LIMIT 0, 3
```

Atau jika diset lewat variable :

```
$batas_awal =0;  
$jumlah_pengambilan=3;  
$sql = "SELECT * FROM bukutamu LIMIT $batas_awal,  
$jumlah_pengambilan";
```

Hasilnya :

id	tgl	pukul	nama	lokasi	email	pekerjaan
1	2003-01-26	10:02:06	Ade	Jawa Barat	ade@yahoo.com	pelajar
2	2003-01-26	10:02:55	Yudi	Jawa Barat	yudi@metro.com	pelajar
3	2003-01-26	10:08:09	Devi	Jawa Barat	Isal@telkom.net	pelajar

Gb. 15.7 Hasil pengambilan data dengan LIMIT 0,3

Begin juga jika kita selanjutnya ingin mengambil data dari baris ke 4 , maka settingannya adalah :

```
$batas_awal =3;  
$jumlah_pengambilan=3;  
$sql = "SELECT * FROM bukutamu LIMIT $batas_awal,  
$jumlah_pengambilan";
```

Hasilnya :

id	tgl	pukul	nama	lokasi	email	pekerjaan
4	2003-01-26	10:09:04	Yuyu	Jakarta	ujang@telkom.net	pelajar
5	2003-01-26	13:33:10	Tintin Sumartini	Jakarta	tintin@yahoo.com	wirausaha
6	2003-01-26	13:34:26	Amorita Kusuma Devi	Jawa Tengah	amor@yahoo.com	wirausaha

Gb. 15.8 Hasil pengambilan data dengan LIMIT 3,3

Untuk pengambilan data dari baris ke 7, settingannya:

```
$batas_awal =6;  
$jumlah_pengambilan=3;  
$sql = "SELECT * FROM bukutamu LIMIT $batas_awal,  
$jumlah_pengambilan";
```

ID	Tgl	Pukul	Nama	Lokasi	Email	Pekerjaan
7	2003-01-26	13:40:26	Damaryadi	Jawa Barat	damar@telkom.net	karyawan
8	2003-01-12	14:00:12	Dona Harun	Jakarta	dona@detik.com	karyawan
9	2003-01-14	14:00:00	Dina Lorenza	Foreign	dina@yahoo.com	karyawan

Gb. 15.9 Hasil pengambilan data dengan LIMIT 6,3

Untuk pengambilan data dari baris ke 10 dan seterusnya, settingannya:

```
$batas_awal =9;  
$jumlah_pengambilan=3;  
$sql = "SELECT * FROM bukutamu LIMIT $batas_awal,  
$jumlah_pengambilan";
```

ID	Tgl	Pukul	Nama	Lokasi	Email	Pekerjaan
10	2003-01-23	11:00:12	Vira Yuniar	Jakarta	vira@yahoo.com	pelajar

Gb. 15.0 Hasil pengambilan data dengan LIMIT 9,3

dari pola di atas berarti setting batas awal untuk misal untuk baris ke 5 adalah baris – 1 = 5 – 1 = 4.

Dengan demikian jika kita mempunyai kasus :

Jumlah data =10

dan

Jumlah pengambilan data per halaman =3,

maka jumlah total halaman yang ada adalah = ceil (10/3) = 4.

Fungsi ceil() digunakan untuk menaikkan angka desimal ke atas dimana yang tadinya $10/3 = 3,3$ setelah di-ceil() akan menjadi 4.

Nah dalam kode di atas, konsep ini diberikan oleh baris kode :

```
$total_hal = ceil ($total_post / $per_hal);
```

b. Peran Variable \$Page

Untuk mendapatkan batas awal pengambilan data, maka kita memanfaatkan variable **\$page** dari link URL (misal [lihat.php?page=1](#)).

Rumusnya adalah :

Batas awal = (total halaman – page) / jumlah pengambilan

Jadi jika ingin mengambil data dari halaman pertama (page=1), maka batas awalnya adalah :

Batas awal = $(4 - 1) * 3 = 9$. Hasil halaman webnya



Gb.15.1 Penampilan data pada halam pertama mempunyai batas awal 9

Jika kita ingin mengambil data dari halaman kedua (page=2), maka batas awalnya adalah :

Batas awal = (4 - 2) * 3 = 6. Hasil halaman webnya :

:: PALASARI Online Shopping ::
HOME | BERITA | KERANJANG BELANJA | BUKU TAMU

KATEGORI:
• Komputer
• Buku
• Olahraga

2010-04-26; 10:19:23
Nama : Santi
Lokasi : Jakarta
Email : santi@live.com
Pekerjaan : pelajar
Pesan : Saya santi ingin berkenalan

2010-04-26; 10:18:54
Nama : Hanif
Lokasi : Jawa Barat
Email : hanif@yahoo.com
Pekerjaan : karyawan
Pesan : Hallo nama saya hanif..

2010-04-26; 10:18:25
Nama : Doni Marantika
Lokasi : Jawa Tengah
Email : doni@mail.com
Pekerjaan : karyawan
Pesan : Halo thanks ya..

<<Prev 1 2 3 4 Next>>

Gb.15.2 Penampilan data pada halaman kedua mempunyai batas awal 6

Demikian juga seterusnya.

Jadi penetuan batas awal ini diberikan oleh baris kode :

```
$batas_awal = ($total_hal-$page)*$per_hal;
```

Ketika kita mengklik menu buku tamu pertama kali, maka link-nya adalah **lihat.php** tanpa ada parameter page. Nah ini dimaksudkan untuk mengambil data dari halaman terakhir. Maka untuk maksud tersebut, baris kodennya diberikan sebagai berikut :

```
if (empty($page)) {  
    $page = $total_hal;  
}
```

Jadi untuk kasus diatas, halaman terakhir bernilai 4. Maka

nilai page akan sama dengan jumlah total halaman seperti pada kode di atas. Jadi

$$\text{Batas awal} = (4-4) * 3 = 0.$$

c. Menu Navigasi

Menu navigasi adalah merupakan hal yang sangat esensial dalam paging. Di sini diberikan menu navigasi untuk link Previous – Next dan nomor halaman.

Untuk menu navigasi diberikan oleh kode sebagai berikut:

```
// -----NAVIGASI-----//
$menu = "<tabel align=\"center\" width=\"100%\"><tr>";

//Jika Page lebih besar dari 1 maka menu Previous akan
ditampilkan
if ($page>1) {
    $menu .= "<td align=\"left\" width=\"10%\">
        <a href=\"lihat.php?page=".($page-
1)."\">&lt;&lt;Prev</a></td> ";
}

//menu navigasi untuk nomor halaman/page
$menu .= "<td align=\"center\" width=\"80%\">";
for ($i=1;$i<=$total_hal;$i++) {

    //jangan pasang hiperlink jika page sama dengan i
    if ($page!=$i)
        $menu.= " <a href=\"lihat.php?page=$i\">";
    $menu.= "$i";

    if ($page!=$i)
        $menu.= "</a> ";
}

$menu.= "</td>";

//Jika page lebih kecil dari total halaman, maka menu
Next ditampilkan
```

■ Aplikasi Berbasis Web dengan PHP & MySQL *Learning By Doing and Make it Simple!*

```
if ($page<$total_hal) {  
    $menu.= "<td align=\"right\" width=\"10%\">  
            <a href=\"lihat.php?page=".($page+1)."\">Next>&gt;</a></td> ";  
}  
  
$menu.= "</tr></table>";
```

Kode 15.9 Listing untuk menu navigasi

d. Penerapan di SQL

Nah setelah semua konfigurasi di atas, langkah berikutnya adalah menerapkan setiap variable dari hasil proes ke dalam sintaks SQL. Berikut adalah sintaksnya:

```
$sql ="SELECT * FROM bukutamu ORDER BY id DESC LIMIT  
$batas_awal,$per_hal ";
```

yang artinya kita mengambil entry record dari tabel bukutamu yang diurutkan secara menurun (DESC) berdasarkan kolom id dengan dibatasi oleh setting batas awal dan jumlah pengambilan per halaman.

Kesimpulan

Setelah kita mempelajari konsep pemasukkan dan pengambilan data beserta paging, maka kini kita sudah bisa mengaplikasikannya pada aplikasi lainnya. Anda dapat menerapkan teknik pemasukkan data beserta validasinya pada aplikasi data entry semacam news, forum dan sebagainya.

Tentang Penulis



Faisal, Ssi telah berkecimpung di dunia IT lebih dari 8 tahun dan juga merupakan seorang IT Hobbiest. Semenjak kuliah di Institut Teknologi Bandung dan lulus tahun 2003 telah aktif dalam mengerjakan berbagai project dan training permrograman web berbasis PHP dan MySQL.

Saat ini beliau bekerja sebagai System Administrator di salah satu perusahaan konsultan di Jakarta selatan. Aktivitas sehari-hari banyak berinteraksi Firewall berbasis Linux, Mail Server berbasis MS Exchange, Aplikasi push mail berbasis Blackberry Enterprise Server (BES), Virtualisasi berbasis VmWare dan pengembangan aplikasi berbasis PHP & MySQL untuk internal kantor.

Di waktu senggang hobi yang beliau lakoni yakni berenang dan men-gotak-atik aplikasi open source berbasis PHP MySQL favorit yakni lain Wordpress dan Drupal. Faisal dapat dikontak via email di faisal@smart.co.id

