



PENGANTAR PHP

- Pengertian PHP
- Hubungan PHP dengan HTML
- Kelebihan PHP

Pengertian PHP

PHP merupakan singkatan dari "PHP: Hypertext Preprocessor", adalah sebuah bahasa scripting yang terpasang pada HTML. Sebagian besar sintaks mirip dengan bahasa C, Java dan Perl, ditambah beberapa fungsi PHP yang spesifik. Tujuan utama bahasa ini adalah untuk memungkinkan perancang web menulis halaman web dinamik dengan cepat.

Hubungan PHP dengan HTML

Halaman web biasanya disusun dari kode-kode html yang disimpan dalam sebuah file berekstensi .html. File html ini dikirimkan oleh server (atau file) ke browser, kemudian browser menerjemahkan kode-kode tersebut sehingga menghasilkan suatu tampilan yang indah. Lain halnya dengan program php, program ini harus diterjemahkan oleh web-server sehingga menghasilkan kode html yang dikirim ke browser agar dapat ditampilkan. Program ini dapat berdiri sendiri ataupun disisipkan di antara kode-kode html sehingga dapat langsung ditampilkan bersama dengan kode-kode html tersebut. Program php dapat ditambahkan dengan mengait program tersebut di antara tanda <? dan ?>. Tanda-tanda tersebut biasanya disebut tanda untuk *escaping* (kabur) dari kode html. File html yang telah dibubuhi program php harus diganti ekstensi-nya menjadi **.php** atau **.php3**.

PHP merupakan bahasa pemrograman web yang bersifat *server-side HTML=embedded scripting*, di mana script-nya menyatu dengan HTML dan berada di server. Artinya adalah sintaks dan perintah-perintah yang kita berikan akan sepenuhnya dijalankan di server tetapi disertakan HTML biasa. PHP dikenal sebagai bahasa scripting yang menyatu dengan tag HTML, dieksekusi di server dan digunakan untuk membuat halaman web yang dinamis seperti ASP (active Server Pages) dan JSP (Java Server Pages).

PHP pertama kali dibuat oleh Rasmus Lerdorf, seorang programmer C. Semula PHP digunakannya untuk menghitung jumlah pengunjung di dalam webnya. Kemudian ia mengeluarkan *Personal Home Page Tools* versi 1.0 secara gratis. Versi ini pertama kali keluar pada tahun 1995. Isinya adalah sekumpulan script PERL yang dibuatnya untuk membuat halaman webnya menjadi dinamis. Kemudian pada tahun 1996 ia mengeluarkan PHP versi 2.0 yang kemampuannya telah mampu mengakses database dan dapat terintegrasi dengan HTML.

Pada tahun 1998 tepatnya pada tanggal 6 Juni 1998 keluarlah PHP versi 3.0 yang dikeluarkan oleh Rasmus sendiri bersama kelompok pengembang softwarena..

Versi terbaru yaitu PHP 4.0 keluar pada tanggal 22 Mei 2000 merupakan versi yang lebih lengkap lagi dibandingkan dengan versi sebelumnya. Perubahan yang paling mendasar pada PHP 4.0 adalah terintegrasinya Zend Engine yang dibuat oleh Zend Suraski dan Andi Gutmans yang merupakan penyempurnaan dari PHP3 scripting engine. Yang lainnya adalah build in HTTP session, tidak lagi menggunakan library tambahan seperti pada PHP3. Tujuan dari bahasa scripting ini adalah untuk membuat aplikasi-aplikasi yang dijalankan di atas teknologi web. Dalam hal ini, aplikasi pada umumnya akan memberikan hasil pada web browser, tetapi prosesnya secara keseluruhan dijalankan web server.

Kelebihan PHP

Ketika e-commerce semakin berkembang, situs-situs yang statis pun semakin ditinggalkan karena dianggap sudah tidak memenuhi keinginan pasar

karena situs tersebut harus tetap dinamis selama setiap hari. Pada saat ini bahasa PERL dan CGI sudah jauh ketinggalan jaman sehingga sebagian besar designer web banyak beralih ke bahasa server-side scripting yang lebih dinamis seperti PHP.

Seluruh aplikasi berbasis web dapat dibuat dengan PHP. Namun kekuatan yang paling utama PHP adalah pada konektivitasnya dengan system database di dalam web. Sistem database yang dapat didukung oleh PHP adalah :

1. Oracle
2. MySQL
3. Sybase
4. PostgreSQL
5. dan lainnya

PHP dapat berjalan di berbagai system operasi seperti windows 98/NT, UNIX/LINUX, solaris maupun macintosh.

PHP merupakan software yang *open source* yang dapat anda download secara gratis dari situs resminya yaitu <http://www.php.net>, ataupun dari situs-situs yang menyediakan software tersebut seperti di <ftp://gerbang.che.itb.ac.id>.

Software ini juga dapat berjalan pada web server seperti PWS (Personal Web Server), Apache, IIS, AOLServer, fhttpd, phttpd dan sebagainya. PHP juga merupakan bahasa pemrograman yang dapat kita kembangkan sendiri seperti menambah fungsi-fungsi baru.

Keunggulan lainnya dari PHP adalah PHP juga mendukung komunikasi dengan layanan seperti protocol IMAP, SNMP, NNTP, POP3 bahkan HTTP.

PHP dapat diinstal sebagai bagian atau modul dari apache web server atau sebagai CGI script yang mandiri. Banyak keuntungan yang dapat diperoleh jika menggunakan PHP sebagai modul dari apache di antaranya adalah :

1. Tingkat keamanan yang cukup tinggi
2. waktu eksekusi yang lebih cepat dibandingkan dengan bahasa pemrograman web lainnya yang berorientasi pada server-side scripting.
3. Akses ke system database yang lebih fleksibel. seperti MySQL.

Dalam modul ini kita akan mempelajari PHP sebagai server-side scripting yang menggunakan apache sebagai webserver. Versi PHP yang kita gunakan adalah PHP4 untuk windows.

INSTALASI WEBSERVER

- Pendahuluan
- Modul PHP Pada Apache
- Modul Perl
- Instalasi
- Konfigurasi PHP

Pendahuluan

Web server merupakan server internet yang mampu melayani koneksi transfer data dalam protocol HTTP. Web server merupakan hal yang terpenting dari server di internet dibandingkan server lainnya seperti e-mail server, ftp server ataupun news server. Hal ini di sebabkan web server telah dirancang untuk dapat melayani beragam jenis data, dari text sampai grafis 3 dimensi. Kemampuan ini telah menyebabkan berbagai institusi seperti universitas maupun perusahaan dapat menerima kehadirannya dan juga sekaligus menggunakannya sebagai sarana di internet.

Web server juga dapat menggabungkan dengan dunia mobile wireless internet atau yang sering di sebut sebagai WAP (wireless Access Protocol) yang banyak digunakan sebagai sarana handphone yang memiliki fitur WAP. Dalam kondisi ini, webserver tidak lagi melayani data file HTML tetapi telah melayani WML (wireless Markup Language)

Salah satu software yang biasa digunakan oleh banyak web master di dunia adalah apache. Software tersebut dapat kita download secara gratis dari web resmi apache yaitu <http://www.apache.org>. Apache merupakan software *open source* yang sekarang ini sudah merebut pasar dunia lebih dari 50%. Web server ini

fleksibel terhadap berbagai system operasi seperti windows9x/NT ataupun unix/linux. Dalam penggunaannya.

Apache merupakan turunan dari webserver yang dikeluarkan oleh NCSA yaitu NCSA HTTPd pada sekitar tahun 1995.

Kelebihan web server Apache :

- *Freeware* (software gratisan)
- Mudah diinstall.
- Mampu beroperasi pada berbagai *platform* sistem operasi .
- Mudah mengonfigurasinya.
- Apache Web server mudah dalam menambahkan periferal lainnya ke dalam *platform* web servernya, misalnya : untuk menambahkan modul, cukup hanya menset file konfigurasinya agar mengikutsertakan modul itu kedalam kumpulan modul lain yang sudah dioperasikan.
- Dapat dijadikan pengganti bagi NCSA web server.
- Perbaikan terhadap kerusakan dan error pada NCSA 1.3 dan 1.4
- Merespon client ebih cepat daripada server NCSA.
- Mampu di *kompilasi* sesuai dengan spesifikasi HTTP yang sekarang.
- Menyediakan feature untuk *multihomed* dan *virtual server*.
- Kita dapat men-set respon error yang akan dikirim web server dengan menggunakan file atau skrip.
- Secara otomatis menjalankan file *index.html*, halaman utamanya, untuk ditampilkan secara otomatis pada clientnya.
- Lebih aman karena memiliki level-level pengamanan
- Apache mempunyai komponen dasar terbanyak di antara webserver-webserver lain, yang berarti bahwa webserver Apache termasuk salah satu dari webserver yang lengkap.
- Performansi dan konsumsi sumberdaya (*resource*) dari webserver apache tidak terlalu banyak, hanya seandar 20 MB untuk file-file dasarnya dan setiap *daemon*nya hanya memerlukan seandar 950 KB memory per-*child*.

- Mendukung transaksi yang aman (*secure transaction*) menggunakan SSL (*Secure Socket Layer*).
- Mempunyai dukungan teknis melalui web.
- Mempunyai kompatibilitas *platform* yang tinggi.
- Mendukung *third party* berupa modul-modul tambahan.

Modul PHP Pada Apache

Apache web server mendukung penambahan modul-modul. Diantara modul yang sering dipakai adalah modul php. PHP (Personal Home Page Tool) adalah salah satu jenis alat yang digunakan untuk membuat halaman web anda menjadi lebih menarik, lebih aman, dan lebih dinamik. Pada dasarnya PHP mirip dengan bahasa script yang lainnya seperti asp, Javascript, Visual Basic atau yang lainnya, namun keuntungannya adalah pada PHP tidak diperlukan tambahan pada sisi webclient seperti halnya Javascript dan Visual Basic script, sehingga lebih luas penggunaannya.

Beberapa keunggulan PHP dibandingkan dengan yang lain :

- Autentikasi http menggunakan PHP menggunakan fungsi *header()* yang ada dalam modul apache PHP.
- Pembuatan file GIF menggunakan library GD yang dikompilasi saat menginstall php.
- PHP dapat menerima metoda upload file
- Mendukung penggunaan cookie
- Mendukung integrasi dengan database
- Mendukung ekspresi regular seperti *ereg()*, *ereg_replace()* dan lainnya.
- Penanganan kesalahan berdasarkan tingkat kesalahan.

Modul Perl

Disamping modul php, apache juga mendukung modul perl. Integrasi modul perl ke apache didapatkan dengan menghubungkan “Perl runtime library” ke server dan menyediakan interface Perl yang berorientasi objek ke server. Ini memungkinkan anda untuk menuliskan module-module Apache dalam lingkungan Perl. Selain itu melekatnya interpreter dalam server anda akan menghindarkan penggunaan interpreter eksternal dan akan menghemat waktu start-up Perl. Script dalam modul perl bekerja hampir menyamai script under CGI, sehingga programmer bisa mengubah script yang bekerja under CGI ke under Perl tanpa banyak perubahan. Kelebihan mod_perl untuk server apache antara lain ia mempunyai kecepatan akses lebih tinggi untuk HTTP request. Selain itu mod_perl punya kelebihan dalam header parser, uri translate, otentikasi, otoritas, akses, type check, fix up, dan pencatat request.

Instalasi

Setelah kita berhasil mendownload dan apache tersebut, maka kita dapat menginstalnya seperti biasa. Apache yang kita gunakan pada modul ini adalah apache_1_3_12_win32. Software ini bentuknya application, sehingga kita bisa langsung menginstalnya seperti software lainnya.

Setelah kita berhasil menginstalnya, jika kita tidak melakukan perubahan directory ketika menginstalnya maka software tersebut akan otomatis tersimpan di directory “*c:\program files\Apache Group*”.

Agar apache dapat berjalan dengan baik dan benar, maka kita harus mengedit suatu file yang berfungsi mengkoordinir dan menghubungkan antara webserver dengan modul PHP yang telah ada. File tersebut adalah “*httpd.conf*”. File tersebut berada di folder “*c:\program files\Apache Group\Apache\conf\httpd.conf*”.

Isi file tersebut dan apa saja yang harus di edit agar apache kita dapat menjadi sebuah web server yang benar akan di jelaskan di bawah ini :

```
ServerType <standalone|inetd>
```

Disini anda menentukan apakah akan menjalankan Apache sebagai http daemon sendiri atau lewat inetd. Pilihan inetd hanya untuk apache berbasis Unix. Defaultnya adalah standalone, dengan pilihan ini apache dapat memberikan performanya secara maksimal. Bila ingin diubah menjadi inetd maka tinggal mengganti baris standalone menjadi inetd.

```
ServerRoot "C:/Program Files/Apache Group/Apache"
```

Perintah ini menentukan letak direktori yang berisi semua file-file konfigurasi maupun file binari (eksekusi) apache. Defaultnya akan menyesuaikan pada saat konfigurasi.

```
LockFile logs/httpd.lock
```

Directory untuk lock file. Defaultnya ia akan bertanda pagar (tidak diaktifkan).

```
PidFile logs/httpd.pid
```

File yang digunakan server untuk menyimpan nomor identifikasi proses (pid) saat apache dijalankan dan aktif jika menggunakan server type standalone

```
ScoreBoardFile logs/httpd.scoreboard
```

File yang digunakan untuk menyimpan informasi internal dari proses server seperti komunikasi antar child dan parent process yang mungkin dibutuhkan oleh beberapa arsitektur computer.

```
ResourceConfig conf/srm.conf
```

```
AccesConfig conf/acces.conf
```

Pilihan untuk membaca file srm.conf dan access.conf secara berurutan. Apache secara default telah memproses kedua file secara berurutan sehingga dapat anda biarkan keduanya ditandai pagar.

```
Timeout 300
```

Jumlah waktu (dalam detik) yang diberikan server untuk menunggu klien saat mengakses server.

```
KeepAlive On
```

Berlaku untuk server standalone untuk menentukan apakah server memperbolehkan lebih dari satu permintaan dalam satu koneksi. Untuk non-aktif isikan `Off`.

```
MaxKeepAliveRequest 100
```

Berlaku untuk server standalone. Menentukan jumlah maksimum permintaan dalam satu koneksi yang bersamaan. Set `0` jika tidak dibatasi.

```
KeepAliveTimeOut 20
```

Berlaku untuk server standalone yaitu jumlah waktu (dalam detik) untuk menunggu permintaan berikutnya dari klien yang sama dan koneksi yang bersamaan.

```
MinSpareServer 5
```

```
MaxSpareServer 10
```

Berlaku untuk server standalone dan digunakan untuk menentukan minimum atau maksimum spare (idle) yang digunakan saat koneksi dilakukan.

```
StartServer 6
```

Berlaku untuk server standalone yang menentukan jumlah server yang berjalan ditandai saat diaktifkan (ditandai dengan jumlah PID).

```
MaxClients 100
```

Berlaku untuk server standalone yang menentukan jumlah maksimal klien yang dapat mengakses secara simultan.

```
MaxRequestPerChild 40
```

Jumlah permintaan maksimal untuk mempergunakan childprocess. Bila diset `0` maka childprocess akan akan berlangsung terus dan tidak akan mati.

```
Listen 12.34.56.78:80
```

Setting server untuk melayani port dan atau alamat IP .Hal ini sangat berguna jika anda memakai satu mesin yang punya beberapa nomor IP dan atau nama DNS samaran.

```
BindAddress *
```

Dukungan virtual hosts dengan memerintahkan apache untuk mengikat beberapa alamat. Bisa diisi dengan *, alamat IP atau nama domain internet yang memenuhi syarat.

```
Port 80
```

Port yang digunakan oleh server. Standarnya memakai port 80.

```
User nobody
```

```
Group nobody
```

Nama user dan group yang menjalankan httpd. Defaultnya adalah user nobody dan group nobody.

```
ServerAdmin firdaus@tpb.itb.ac.id
```

Alamat email untuk mengirimkan/memberitahu halaman yang dibuat secara otomatis oleh server. Misalnya untuk mengirim pesan error.

```
ServerName www.firdausadnan.com
```

Untuk setting nama server anda. dengan menggunakan standar fqdn (fully qualified domain name) dan harus merupakan nama yang ada dalam DNS (Domain Name Service) jaringan anda. Jika tidak yakin isikan nomor IP.

```
DocumentRoot "C:\wwwroot\html"
```

Perintah untuk setting lokasi direktori web yang akan diterbitkan sebagai *webpage*.

```
UserDir public_html
```

Menyatakan letak direktori homepage bagi masing-masing user pada sistem Unix tempat Webserver anda di install. Jika perintah ini diset, maka tiap user tinggal membuat direktori public_html di direktori homenya.

```
AccessFileName .htaccess
```

Digunakan untuk untuk menentukan nama file untuk keperluan control akses direktori.

```
UseCanonicalName Off
```

Digunakan untuk menentukan apakah server akan menggunakan self-referencing URL address atau tidak. Beberapa klien yang mengakses server memerlukannya namun kebanyakan tidak memerlukan sama sekali. Sebaiknya di-offkan saja.

```
DefaultType text/plain
```

Diguanakn untuk menentukan tipe content default apabila server gagal menentukan tipe MIME tertentu.

```
HostnameLookups Off
```

Digunakan untuk menentukan apakah DNS akan selalu dilihat apabila klien mengakses server.

```
ErrorLog logs/error_log
```

Digunakan untuk menentukan nama file yang digunakan untuk mencatat semua error yang terjadi selama operasional.

```
LogLevel warn
```

Digunakan untuk menentukan level log yang akan disimpan pada file yang telah ditentukan oleh ErrorLog. Optionnya cukup banyak <emerg|alert|crit|errors|warn|debug>. Semakin tinggi level yang ditentukan semakin banyak pesan yang dapat dibaca pada file log, hanya saja kerja server akan menurun.

```

LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combine
LogFormat "%h %l %u %t \"%r\" %>s %b" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent

```

Digunakan untuk mendefinisikan format yang akan digunakan oleh log file. Defaultnya seperti sintak di atas.

Berikut dijelaskan sedikit mengenai formatnya :

- o %h = remot host
- o %l = remote Logname
- o %u = remote user
- o %t = Time
- o %r = First Line Of Request.
- o %s = Status. Menunjukkan status terakhir dari request.
- o %b = Jumlah Byte yang dikirim client, dalam format CLF ("- " bila tidak ada byte yang dikirim).
- o %{Referer}I = Isi dari Tag Referer pada Header
- o %{User-Agent}I = Isi dari Tag User-Agent pada Header.
- o %U = URL yang diminta (direquest)
- o combined, common, referer dan agent yang tertulis pada bagian belakang directive adalah nickname untuk memudahkan proses pemanggilan oleh directive CustomLog.

```
Alias /icons/ "C:/Program Files/Apache Group/Apache/icons/"
```

Digunakan untuk menentukan alias dari suatu direktori. Misalnya direktori "C:/Program Files/Apache Group/Apache/icons/" hanya menjadi /icons/ saja.

```
IndexOptions FancyIndexing
```

Digunakan untuk menentukan letak direktori index yang akan mengelompokkan file-file tertentu yang sejenis.

```
AddIcon /icons/tar.gif .tar
```

Digunakan menentukan nama icon serta extensionnya. Sintaknya :

```
AddIcon <icon-name> <extension>
```

```
AddIconByEncoding (CMP,/icons/compressed.gif) x-compress x-gzip
```

Digunakan untuk menentukan nama icon untuk keperluan MIME_Encoding. Sintaknya :

```
AddIconByEncoding <icon> <MIME-type>
```

```
AddIconByType (TXT,/icons/text.gif) text/*
```

Digunakan untuk menentukan nama icon untuk keperluan FancyIndexing.

```
ReadmeName README
```

```
HeaderName HEADER
```

Digunakan untuk menentukan file yang berfungsi sebagai footer dan header.

```
AddEncoding x-compress Z
```

```
AddEncoding x-gzip gz tgz
```

Digunakan untuk keperluan MIME encoding. Sebaiknya dibiarkan apa adanya.

```
AddLanguage en .en
```

```
AddLanguage et .ee
```

```
AddLanguage fr .fr
```

Digunakan untuk menentukan language-MIME encoding dengan file extension yang digunakan.

```
AddType <content-type> <extension>
```

Digunakan untuk menambahkan module-module yang digunakan dalam server apache. Misalnya untuk mengaktifkan (meload) modul php maka ditambahkan :

```
AddType application/x-httpd-php .php
AddType application/x-httpd-php-source .phps
LoadModule php4_module c:/php/sapi/php4apache.dll
AddType application/x-httpd-php .php4
ScriptAlias /php4/ "C:/php/"
Action application/x-httpd-php4 "/php4/php.exe"
AddType application/x-httpd-php4 .php .php3 .phtml
AddType application/x-tar .tgz
```

Pada baris ke 6 terakhir ini kita harus menambahnya sendiri jika kita ingin menginstal PHP sebagai modul web di webserver. Script ini juga dapat dilihat ketika kita mengekstrak software PHP setelah kita mendownloadnya.

Konfigurasi PHP

Setelah kita mendownload modul PHP dari situs resminya maka kita harus mengkonfigurasi agar modul PHP dapat di baca oleh apache sebagai web server kita.

Caranya adalah :

1. Modul PHP sebaiknya di letakkan di directory c dengan nama php yaitu "c:\php
2. File php4ts.dll yang ada di dalam modul tersebut harus di pindahkan ke folder system(32) yang ada di folder windows yaitu "c:/windows/system (32)"

3. Supaya apache dapat membaca modul PHP4 maka harus ditambahkan

```
LoadModule php4_module c:/php/sapi/php4apache.dll
```

```
AddType application/x-httpd-php .php4
```

pada bagian bawah baris

```
AddType application/x-httpd-php .php
AddType application/x-httpd-php-source .phps
```

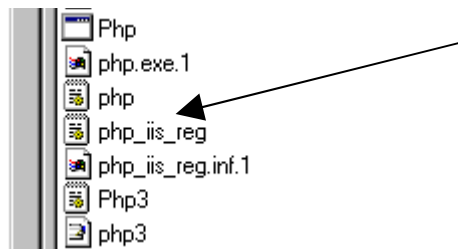
Yang ada di file httpd.conf

4. Untuk menjalankan script CGI maka di bawahnya juga harus di tambahkan script:

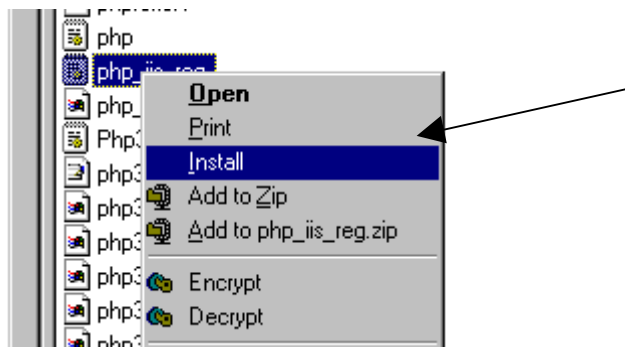
```
ScriptAlias /php4/ "C:/php/"
```

```
Action application/x-httpd-php4 "/php4/php.exe"  
AddType application/x-httpd-php4 .php
```

- Langkah terakhir adalah merubah ekstension `php.ini-dist` yang ada di dalam modul tersebut menjadi `php.ini` dan selanjutnya di simpan di dalam folder windows.
- Namun sebelumnya jangan lupa untuk mengkopi file `php_iis_reg` yang ada di dalam ekstrak software `php-3.0.17` ke folder `c:\php` yang telah kita tentukan sebelumnya. Software tersebut dapat di download dari web resmi php.

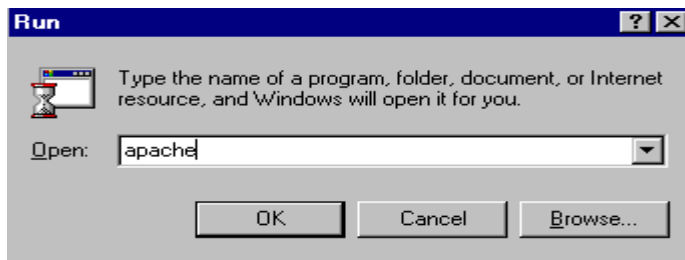


Setelah menemukan file tersebut, maka klik kanan tepat pada file tersebut sehingga tampak tampilan :

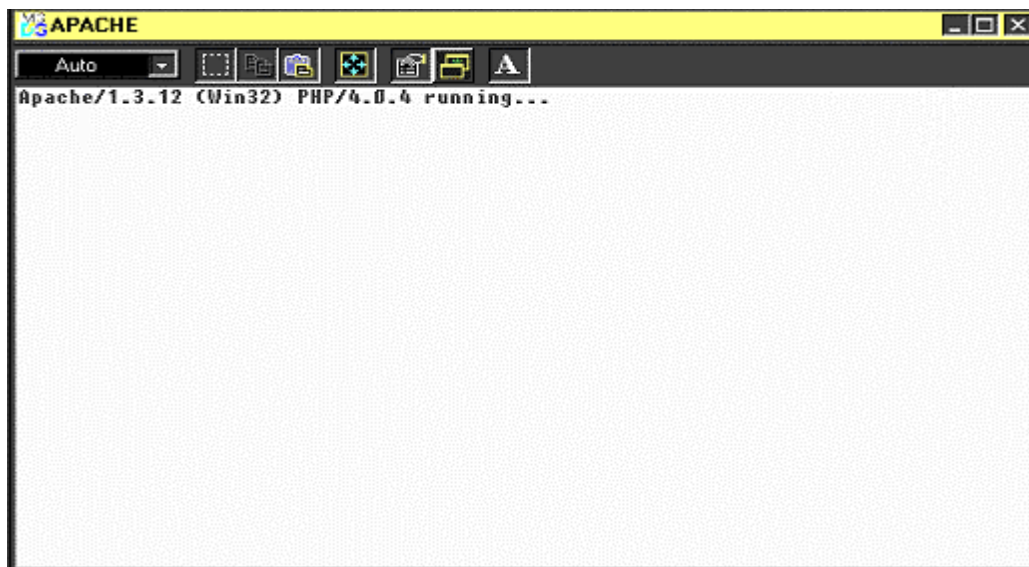


Klik pada kata `Install` tersebut.

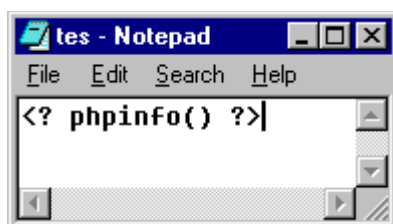
Sampai di sini, konfigurasi PHP4 telah selesai. Untuk menjalankannya, jalankan apache dari tombol start windows dan tekan tombol Run. Tulis di dalam kolomnya `apache` untuk menjalankan apache di windows. Tampilannya sebagai berikut :



Tampilan apachenya adalah :

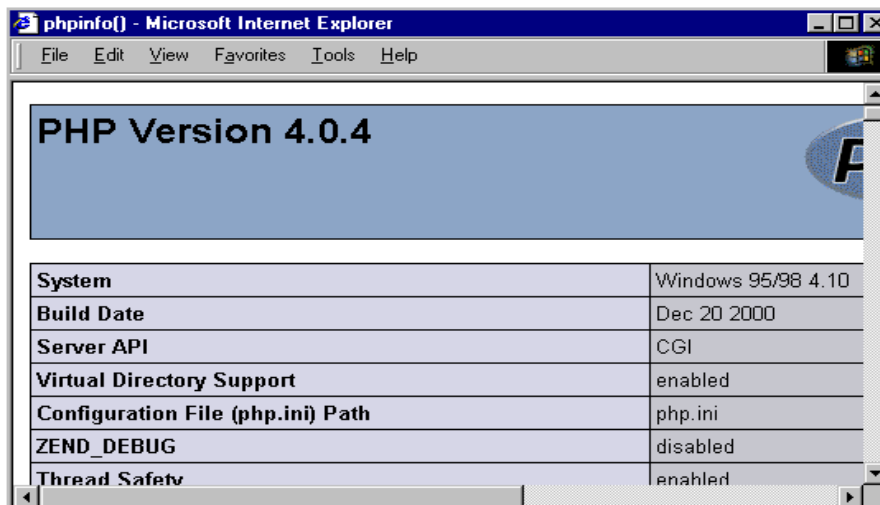


Untuk memastikan kalau modul PHP4 kita sudah berjalan dengan benar maka kita harus menulis sintak di notepad dengan isi :



dan di simpan dengan nama file *tes.php*

Kemudian kita buka browser yang ada dan tulis localhost/tes.php pada addressnya, maka akan tampil suatu tampilan :



Setelah gambar di atas tampil di browser kita, maka kita telah berhasil menginstal web server di komputer.

Dasar-dasar Pemrograman PHP

- Pendahuluan
- Sintaks Dasar (Keluar dari kode HTML)
- Pemisah instruksi
- Komentar

Pendahuluan

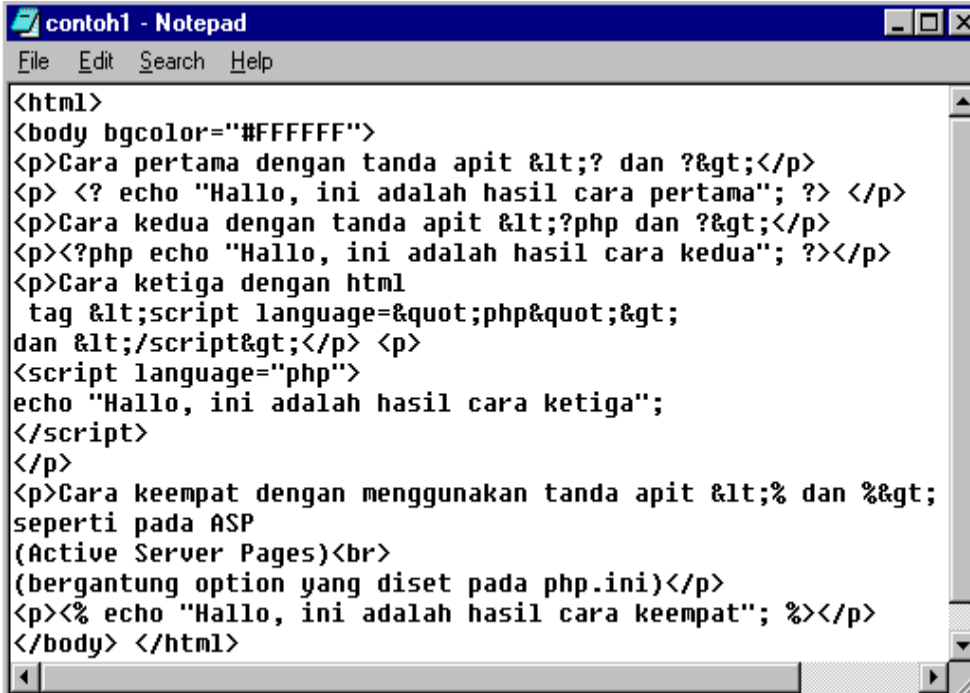
Sintak pemrograman PHP dapat di satukan dengan sintak HTML dalam suatu file. Kode PHP diawali dengan sintak “<?” dan diakhiri dengan sintak “>”. File tersebut diberi ekstension “.php” atau ekstension yang telah kita atur webserver.

Berdasarkan ekstension tersebut, maka saat file tersebut diakses, server akan mendeteksi bahwa file tersebut mengandung sintak PHP. Server kemudian menterjemahkan kode tersebut dan menghasilkan output berupa sintak HTML yang kemudian akan dikirim ke browser user yang mengakses web dari web server.

Sebagaimana layaknya bahasa pemrograman umumnya, PHP memiliki sintaks program, jenis variabel, bagaimana mendefinisikan variabel, operator, struktur kontrol dan fungsi. Selain itu, PHP juga memiliki sejumlah fungsi yang sangat bermanfaat dalam melakukan pemrosesan html.

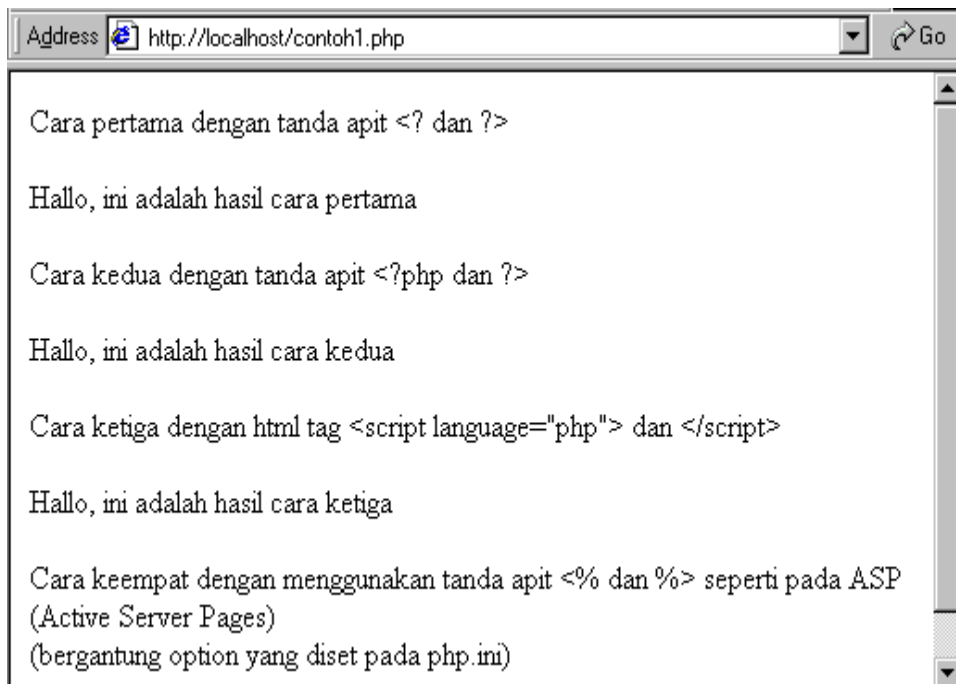
Sintaks Dasar (Keluar dari kode HTML)

Terdapat 4 cara untuk keluar dari HTML, perhatikan contoh di bawah ini yang menerangkan keempat cara tersebut :



```
<html>
<body bgcolor="#FFFFFF">
<p>Cara pertama dengan tanda apit &lt;? dan ?&gt;</p>
<p><? echo "Hallo, ini adalah hasil cara pertama"; ?> </p>
<p>Cara kedua dengan tanda apit &lt;?php dan ?&gt;</p>
<p><?php echo "Hallo, ini adalah hasil cara kedua"; ?></p>
<p>Cara ketiga dengan html
tag &lt;script language="php"&gt;
dan &lt;/script&gt;</p> <p>
<script language="php">
echo "Hallo, ini adalah hasil cara ketiga";
</script>
</p>
<p>Cara keempat dengan menggunakan tanda apit &lt;% dan %&gt;
seperti pada ASP
(Active Server Pages)<br>
(bergantung option yang diset pada php.ini)</p>
<p>% echo "Hallo, ini adalah hasil cara keempat"; %</p>
</body> </html>
```

Tampilan di browsernya adalah :

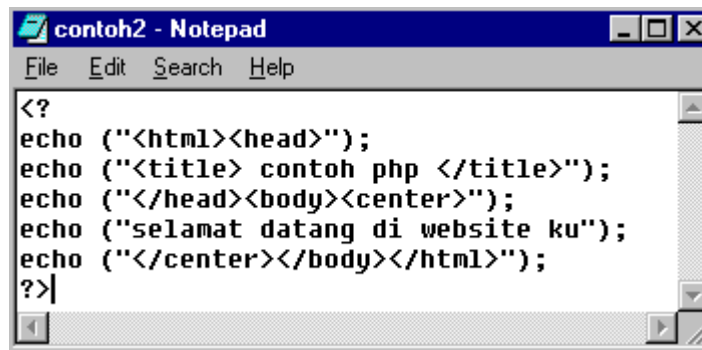


Sedangkan source kodenya yang dapat kita lihat dari browser tersebut adalah :

```
File Edit Search Help
<html>
<head>
<title>contoh2</title>
</head>
<body bgcolor="#FFFFFF">
<p>Cara pertama dengan tanda apit &lt;? dan ?&gt;</p>
<p> Hallo, ini adalah hasil cara pertama </p>
<p>Cara kedua dengan tanda apit &lt;?php dan ?&gt;</p>
<p>Hallo, ini adalah hasil cara kedua</p>
<p>Cara ketiga dengan html tag &lt;script language=&quot;php&quot;
&gt;
&gt; dan &lt;/script&gt;</p>
<p>
Hallo, ini adalah hasil cara ketiga
</p>
<p>Cara keempat dengan menggunakan tanda apit &lt;% dan %&gt;
seperti pada ASP
(Active Server Pages)<br>
(bergantung option yang diset pada php.ini)</p>
<p><% echo "Hallo, ini adalah hasil cara keempat"; %></p>
</body>
</html>
```

Dari kedua gambar di atas, perhatikan, ada beberapa kalimat yang tidak terlihat oleh source tersebut. Kalimat yang tidak kelihatan tersebut adalah kalimat yang sifatnya server-side scripting, yaitu script yang hanya dapat di baca oleh server sehingga di browser tidak dapat tampil secara keseluruhan.

Sebuah file PHP dapat juga keseluruhannya berisi perintah-perintah PHP. Sebagai contoh perhatikan contoh di bawah ini :



```
<?
echo (<html><head>");
echo (<title> contoh php </title>");
echo (</head><body><center>");
echo ("selamat datang di website ku");
echo (</center></body></html>");
?>
```

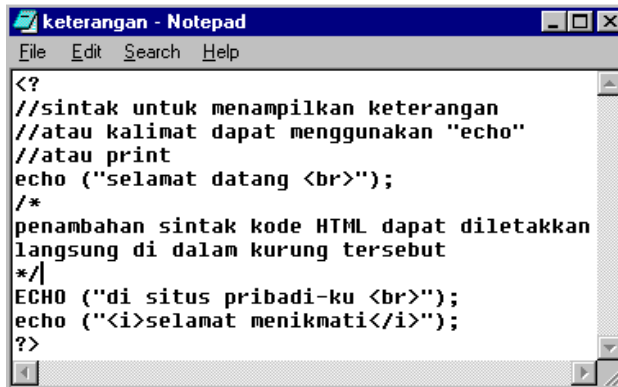
Pemisah instruksi

Instruksi-instruksi dipisahkan oleh tanda semi-colon (;). Dengan demikian setiap instruksi yang panjang seperti mendefinisikan/mencetak tulisan, mendefinisikan array dapat dilakukan pada beberapa baris.

Komentar

Komentar dapat dibubuhkan pada program sebagai keterangan yang berguna. Tanda # dapat mengabaikan teks setelahnya hingga tanda ganti baris. Selain itu, tanda apit /* dan */ juga dapat mengapit komentar/keterangan. Seperti layaknya tanda apit lain, tanda apit ini juga berlaku untuk beberapa baris komentar.

Dari segi penulisannya, PHP membedakan antara huruf kecil dan huruf besar untuk penulisan variable, sedangkan untuk penulisan fungsi-fungsi, PHP tidak membedakan antara huruf kecil dan huruf besar. Contohnya :



```
<?
//sintak untuk menampilkan keterangan
//atau kalimat dapat menggunakan "echo"
//atau print
echo ("selamat datang <br>");
/*
penambahan sintak kode HTML dapat diletakkan
langsung di dalam kurung tersebut
*/
ECHO ("di situs pribadi-ku <br>");
echo ("<i>selamat menikmati</i>");
?>
```

dengan menggunakan sintak di atas maka akan menghasilkan pada browser sebagai berikut :



selamat datang
di situs pribadi-ku
selamat menikmati

TIPE DAN KONVERSI DATA

- Pengantar
- Identifier
- Tipe Data

Pengantar

PHP mengenal beberapa tipe data. Tipe data tersebut merupakan sebuah variable yang dapat ditentukan secara otomatis oleh PHP yang bergantung pada operasi yang sedang dilakukan menggunakan variable tersebut. Kelima tipe data tersebut antara lain.

1. Integer
2. Floating Point
3. String
4. Array
5. Objects.

Floating Float biasanya di namakan dengan double. Tipe data double selalu dalam bentuk desimal.

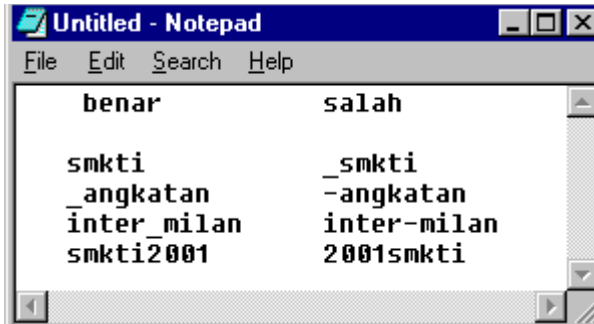
Namun sebelum kita membahas kelima tipe data di atas, kita akan membahas identifier terlebih dahulu.

Identifier

Identifier dalam PHP terdiri dari *fungsi, variable dan classes*. Aturan dalam identifier yang diperuntukkan pada PHP adalah :

1. Tidak boleh menggunakan tanda baca dalam mendefinisikan sebuah identifier.
2. Harus dimulai dengan huruf atau under_score (_)
3. variable selalu diawali dengan tanda \$ (dollar).
4. Identifier dalam PHP sangat sensitive, kecuali fungsi-fungsi yang telah disediakan oleh PHP.

5. Nama fungsi yang dibuat tidak boleh sama dengan nama fungsi yang telah tersedia dalam bahasa PHP.
6. Nama variable seharusnya tidak boleh sama dengan nama fungsi yang telah di tetapkan dalam PHP.



Tipe Data

Integer

Tipe data integer yang meliputi semua bilangan bulat, besar range data integer pada PHP sama dengan range tipe data long pada bahasa C yaitu antara

-2.147.483.648 sampai 2.147.483.647 pada platform 32-bit. PHP akan secara otomatis mengkonversi data integer menjadi floating point jika berada di luar range di atas. Integer dapat dinyatakan dengan bentuk octal, desimal dan heksadesimal.

Contoh :

```
$biaya = 1000;
```

```
$a = 1234
```

```
$a = -123
```

`$a = 0123` (bilangan oktal)

`$a = 0x12` (bilangan heksadesimal)

Floating point Numbers (Bilangan Riil)

Floating float merupakan representasi dari bilangan desimal atau pecahan. Range tipe data floating point sama dengan range tipe data double pada bahasa C yaitu $1.7E-308$ sampai $-1.7E-308$. Floating point atau bilangan riil dapat dinyatakan sbb. :

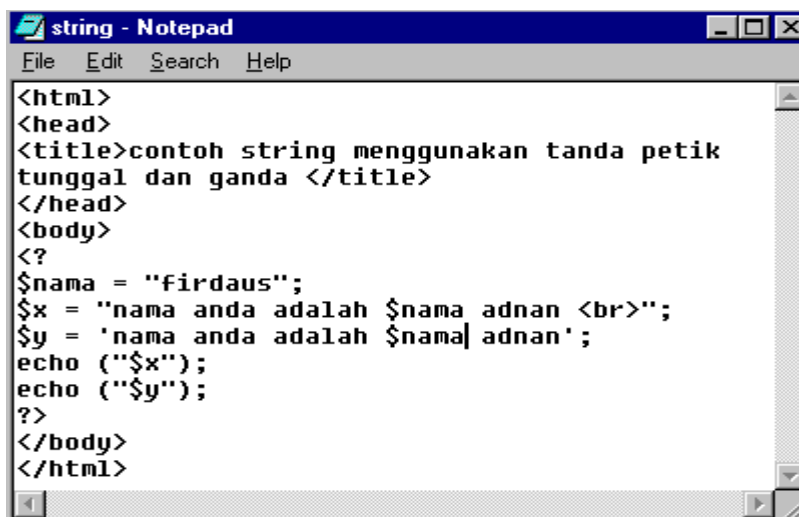
`$a = 1.234`; `$a = 1.2e3`;

String

String atau teks dapat dibuat dengan menuliskan teks di antara tanda kutip tunggal (') atau tanda kutip ganda (").

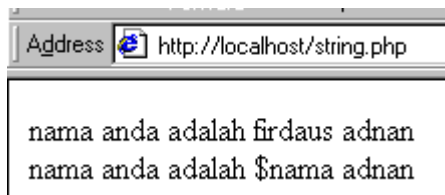
`$a = "Hallo world"`; `$a = 'Apa khabar'`;

Perbedaannya adalah jika menggunakan tanda petik tunggal, maka pada string itu tidak dapat kita masukkan variable dan escape sequence handling. Contoh :



```
<html>
<head>
<title>contoh string menggunakan tanda petik
tunggal dan ganda </title>
</head>
<body>
<?
$nama = "firdaus";
$x = "nama anda adalah $nama adnan <br>";
$y = 'nama anda adalah $nama| adnan';
echo ("$x");
echo ("$y");
?>
</body>
</html>
```

Jika kita jalankan browser dengan address `localhost/string.php` maka akan menghasilkan suatu tampilan :



Dari gambar di atas dapat terlihat bahwa baris atas yang menggunakan string ganda dapat mengeluarkan value dari variable \$nama yang telah kita definisikan di atas, sedangkan pada baris yang di bawah tidak dapat tampil.

Perhatikan table di bawah ini, yang berisi tanda-tanda di dalam PHP beserta artinya.

Tanda	Arti
\n	Baris baru
\r	Carriage
\t	Tab horizontal
\\	Backslash
\\$	Tanda dollar
\"	Tanda kutip
\[0-7]{1-3}	Urutan karakter yang sesuai dengan ekspresi reguler adalah karakter yang berada dalam notasi oktal
\x[0-9A-Fa-f]{1,2}	Urutan karakter yang sesuai dengan ekspresi reguler adalah karakter yang berada dalam notasi heksa

Array

Array merupakan sebuah data yang mengandung satu atau lebih data, dan dapat diindex berdasarkan numeric maupun string. Array atau jajaran merupakan variabel berindeks yang dapat dimanipulasi dengan mudah seperti layaknya tabel dalam sebuah spreadsheet. Untuk mendefinisikan nilai array, dapat dilakukan dengan mencantumkan index dari array tersebut.

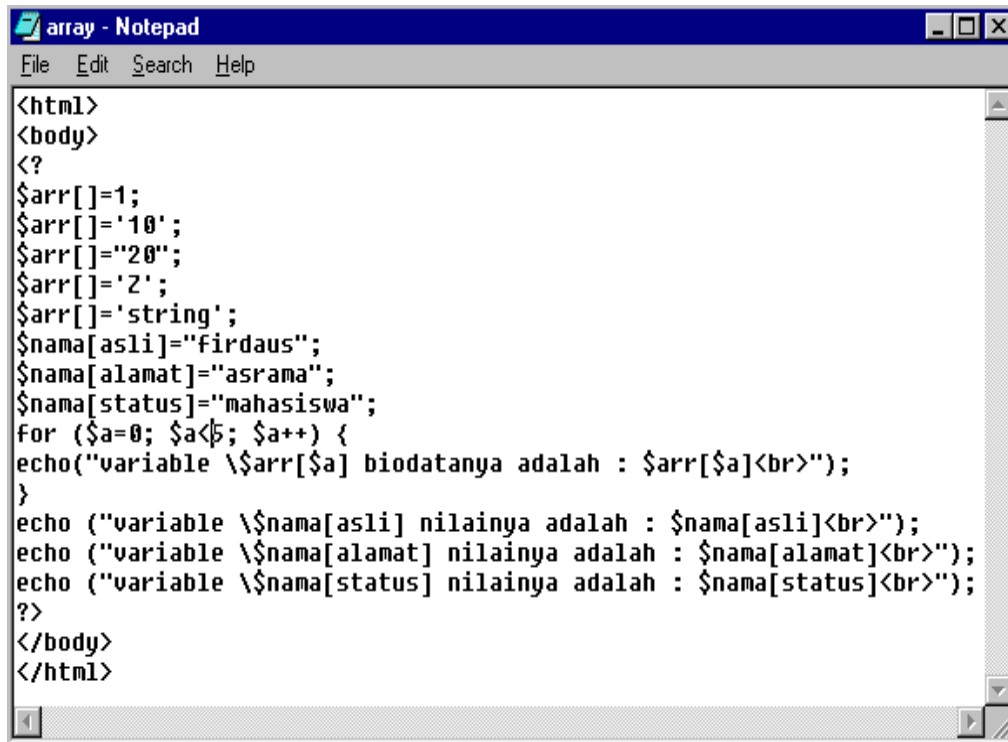
```
$a[0] = "abc";
```

```
$a[1] = "def";
```

Indeks array dapat juga berupa string (*associative array*) seperti :

```
$a['mobil'] = "TOYOTA";
```

Perhatikan contoh di bawah ini :



```
<html>
<body>
<?
$arr[]=1;
$arr[]='10';
$arr[]="20";
$arr[]='Z';
$arr[]='string';
$nama[asli]="firdaus";
$nama[alamat]="asrama";
$nama[status]="mahasiswa";
for ($a=0; $a<5; $a++) {
echo("variable \$$arr[$a] biodatanya adalah : $arr[$a]<br>");
}
echo ("variable \$$nama[asli] nilainya adalah : $nama[asli]<br>");
echo ("variable \$$nama[alamat] nilainya adalah : $nama[alamat]<br>");
echo ("variable \$$nama[status] nilainya adalah : $nama[status]<br>");
??
</body>
</html>
```

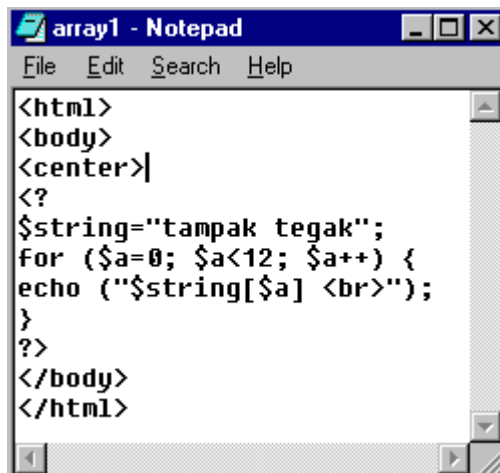
Perhatikan gambar di atas, array di deklarasikan dengan memasukkan nilai. Cara lain adalah dengan menggunakan fungsi `array()` atau `list()`. Dengan cara ini angka index yang tidak dinyatakan maka secara otomatis akan diisi dengan angka index berikutnya sesuai urutan pengisiannya.

Maka hasilnya yang tampak di browser adalah :

Address  http://localhost/array.php

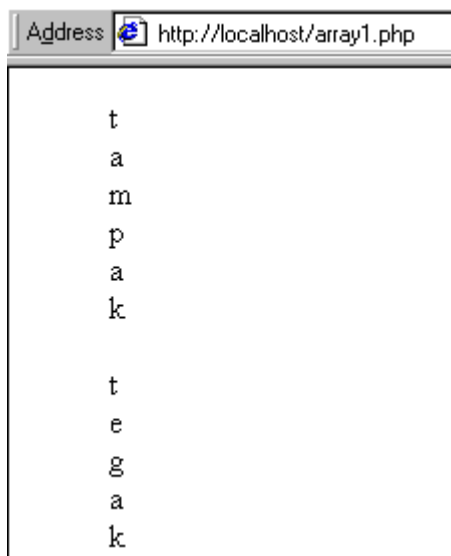
```
variable $arr[0] biodatanya adalah : 1  
variable $arr[1] biodatanya adalah : 10  
variable $arr[2] biodatanya adalah : 20  
variable $arr[3] biodatanya adalah : Z  
variable $arr[4] biodatanya adalah : string  
variable $nama[asli] nilainya adalah : firdaus  
variable $nama[alamat] nilainya adalah : asrama  
variable $nama[status] nilainya adalah : mahasiswa
```


Perhatikan sintak di bawah ini :



```
array1 - Notepad  
File Edit Search Help  
<html>  
<body>  
<center>  
<?  
$string="tampak tegak";  
for ($a=0; $a<12; $a++) {  
echo ("string[$a] <br>");  
}  
>  
</body>  
</html>
```

Tampilan di browsernya adalah :

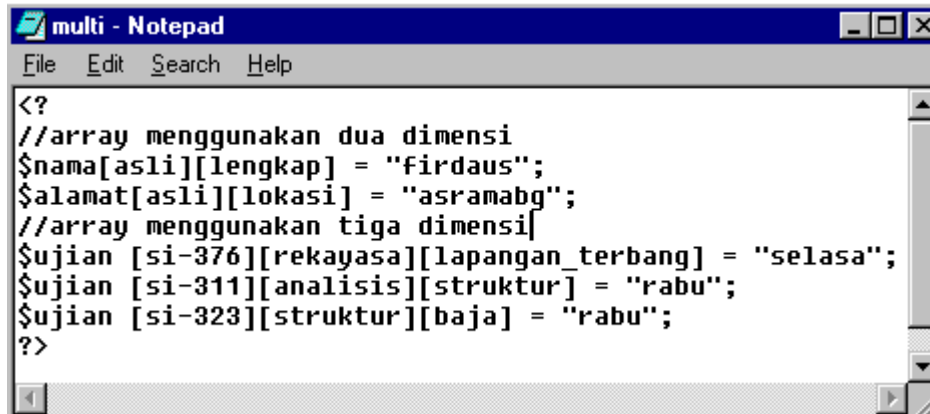


Address  http://localhost/array1.php

```
t  
a  
m  
p  
a  
k  
  
t  
e  
g  
a  
k
```

Array multi dimensi adalah array yang elemennya juga berupa array.

Contohnya adalah :



```
<?
//array menggunakan dua dimensi
$nama[asli][lengkap] = "firdaus";
$salamat[asli][lokasi] = "asramabg";
//array menggunakan tiga dimensi
$ujian [si-376][rekayasa][lapangan_terbang] = "selasa";
$ujian [si-311][analisis][struktur] = "rabu";
$ujian [si-323][struktur][baja] = "rabu";
?>
```

TIPE DATA (lanjutan) DAN VARIABLE

- Tipe Data Object
- Nilai Boolean
- Konversi Tipe Data

Variable

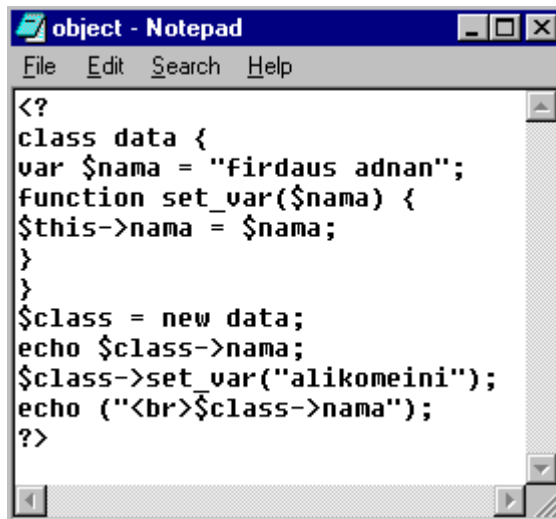
- Pengantar
- Definisi Variabel & Array
- Kompatibilitas Variabel

Ekspresi

Tipe Data Object

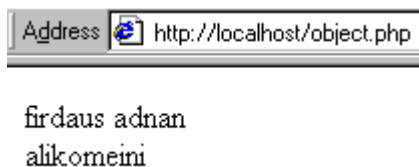
Object adalah sebuah tipe data yang dapat berupa sebuah bilangan, variable bahkan sebuah fungsi. Object dibuat dengan maksud untuk membantu programmer yang telah terbiasa dengan OOP (Object Oriented Programming), walaupun fasilitas OOP yang disediakan PHP masih sangat terbatas.

Perhatikan contoh di bawah ini :



```
<?
class data {
var $nama = "firdaus adnan";
function set_var($nama) {
$this->nama = $nama;
}
}
$class = new data;
echo $class->nama;
$class->set_var("alikoineini");
echo ("<br>$class->nama");
?>
```

Pada tampilan browser akan terlihat seperti gambar di bawah ini :



Dalam program object.php di atas terdapat sebuah class dengan nama data, kemudian dibuat sebuah data object data dari class data, data yang di cetak dan keluar adalah Firdaus adnan. Class data mempunyai sebuah metode dengan nama set_var. Metode ini digunakan untuk memberi nilai pada variable \$nama. Data object data yang dibuat dari class data akan mempunyai semua properties dari class data, termasuk method-metohod-nya, sehingga sebuah data bertipe object juga dapat mengandung sebuah method (fungsi). Pada program di atas method set_var pada object data kita pakai untuk mengubah nilai variable \$nama menjadi alikoineini kemudian mencetaknya.

Nilai Boolean

Nilai Boolean biasanya digunakan pada struktur kontrol program, seperti if atau if-else. Ada beberapa kondisi nilai Boolean harus di gunakan, yaitu :

- Untuk integer dan floating point, nilai booleannya adalah false jika nilainya 0 selainnya adalah true
- Untuk string, nilai booleannya false jika string kosong (“”), selainnya adalah true.
- Untuk tipe data array, akan bernilai false jika elemennya kosong dan sebaliknya true.
- Untuk tipe data object, bernilai false jika tidak ada variable atau fungsi/method yang terdefinisi di dalamnya, dan sebaliknya bernilai true.
- Untuk variable yang belum dapat dibuat atau didefinisikan, nilainya adalah false.

Di dalam PHP, ada dua keyword yaitu true dan false. Di mana true merepresentasikan nilai integer 1 dan false merepresentasikan string kosong, keyword true dan false tidak case sensitive.

Konversi Tipe Data

Tipe data tidak di tentukan secara eksplisit di dalam pemograman PHP karena tipe data ditetapkan sesuai konteksnya pada saat digunakan. Sebagai contoh, sebuah variable \$var di deklarasikan bertipe integer, tetapi akan berubah secara otomatis menjadi bertipe floating point jika diberi nilai bilangan pecahan.

Perhatikan contoh di bawah ini :

```

konversi - Notepad
File Edit Search Help
<?
$var=1;
echo "\$var=" . $var . " dan bertipe " . gettype($var);
$var=1+1.0;
echo "<br>\$var=" . $var . " dan bertipe " . gettype($var);
?>

```

Perubahan pada tipe data dapat juga dilakukan secara eksplisit, yang biasanya disebut type casting. Casting pada sebuah variable dilakukan dengan menuliskan jenis tipe data yang diinginkan di depan nilai dari variable tersebut.

Di bawah ini adalah table dari PHP casting operators.

Operator	Function
(int), (integer)	Cast to an integer
(real), (double), (float)	Cast to a floating point number
(string)	Cast to a string
(array)	Cast to an array
(object)	Cast to an object

Casting ke bentuk tipe data array akan mengubah nilai semula menjadi elemen pertama dari variable array. Casting ke bentuk tipe data object akan mengubah variable semula menjadi property dari data object yang baru yaitu dengan nama “scalar”.

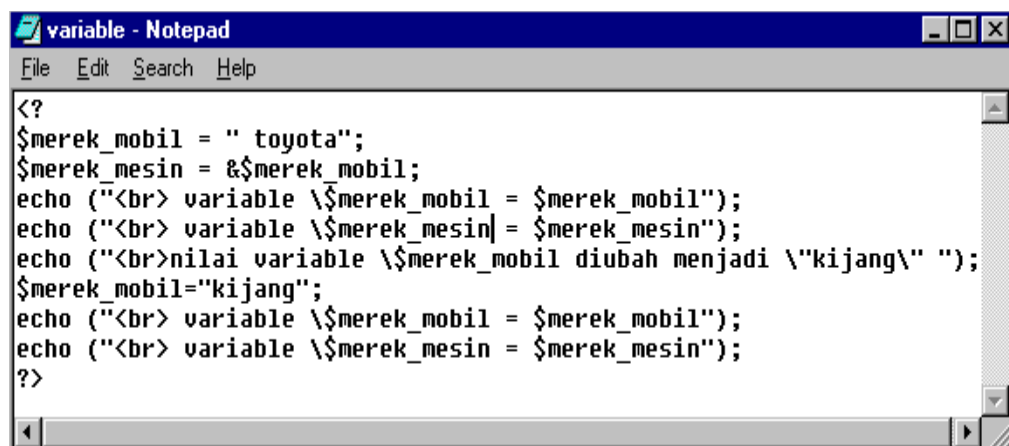
VARIABLE

Pengantar

Di dalam bahasa pemrograman PHP, variable dapat dinyatakan dengan tanda \$ di belakang nama variable. Nama variable dapat terdiri atas angka, huruf dan underscore. Penamaan variable bersifat case sensitive artinya penggunaan huruf kecil dan besar berbeda. Deklarasi sebuah variable selalu diikuti dengan pemberian nilai dari variable tersebut, sehingga jika sebuah variable belum memiliki nilai, tidak perlu di deklarasikan dahulu.

Variabel adalah sebuah tempat untuk menyimpan data yang nilainya dapat berubah-ubah. Variable dalam PHP tidak harus dideklarasikan sebelum variable tersebut digunakan, tidak seperti halnya dengan bahasa pemrograman lainnya yang harus mendeklarasikan variable terlebih dahulu.

Pemberian nilai dari suatu variable dapat juga mengacu pada nilai dari variable lain yang menjadi referensi, pemberian nilai berdasar referensi ini dinyatakan dengan tanda & di depan nama variable reference-nya. Perhatikan contoh di bawah ini :



```
<?
$merek_mobil = " toyota";
$merek_mesin = &$merek_mobil;
echo ("<br> variable \ $merek_mobil = $merek_mobil");
echo ("<br> variable \ $merek_mesin = $merek_mesin");
echo ("<br> nilai variable \ $merek_mobil diubah menjadi \"kijang\" ");
$merek_mobil="kijang";
echo ("<br> variable \ $merek_mobil = $merek_mobil");
echo ("<br> variable \ $merek_mesin = $merek_mesin");
?>
```

Definisi Variabel & Array

Sebuah variabel tunggal dapat didefinisikan melalui tanda sama-dengan (=).

Array dapat didefinisikan dengan fungsi 'array'.

```
$a = 1;
```

```
$s = "Ini sebuah string";
```

```
$a = $b = 4;  
$p = array(1,2,3,4,5);  
$q = array("satu","dua","tiga","empat","lima");  
$r = array("tomat"=>3,"jeruk"=>4,"apel"=>5);
```

Kompatibilitas Variabel

Dalam PHP, variabel dapat secara otomatis ditaksir jenisnya tanpa harus didefinisikan terlebih dahulu. Misalnya `$a = "5"` (huruf lima) yang merupakan sebuah string dapat dijumlahkan dengan sebuah integer lain (mis. `$b = $a + 2;`). Dalam hal ini, `$a` dianggap sebagai sebuah variabel integer. Sebaliknya, sebuah variabel integer atau float dapat dijumlahkan dengan string sehingga membentuk suatu string lain.

EKSPRESI

Increment dan Decrement

Ekspresi increment dapat berupa :

`$b = $a++` : definisikan nilai `$b` sama dengan nilai `$a`, dan tingkatkan (increment) nilai `$a`.

`$b = ++$a` : tingkatkan nilai `$a`, dan definisikan nilai `$b` sama dengan nilai `$a`.

Ekspresi kondisional

Ekspresi singkat untuk melakukan kendali program :

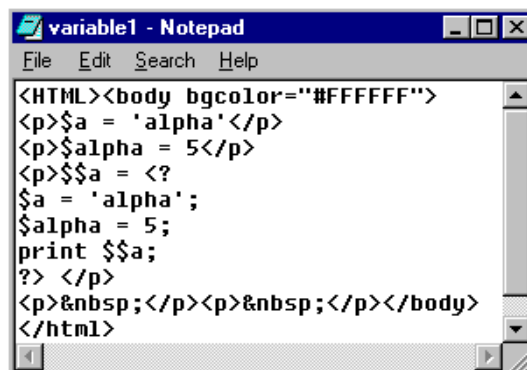
`$first ? $second : $third` (jika `$first` benar, maka `$second` menjadi nilai ekspresi, sedangkan jika `$first` salah, maka `$third` menjadi nilai ekspresi).

VARIABLE (lanjutan)

- Batas Variabel/variable scope (Lokal dan Global)
- Variabel dari luar PHP

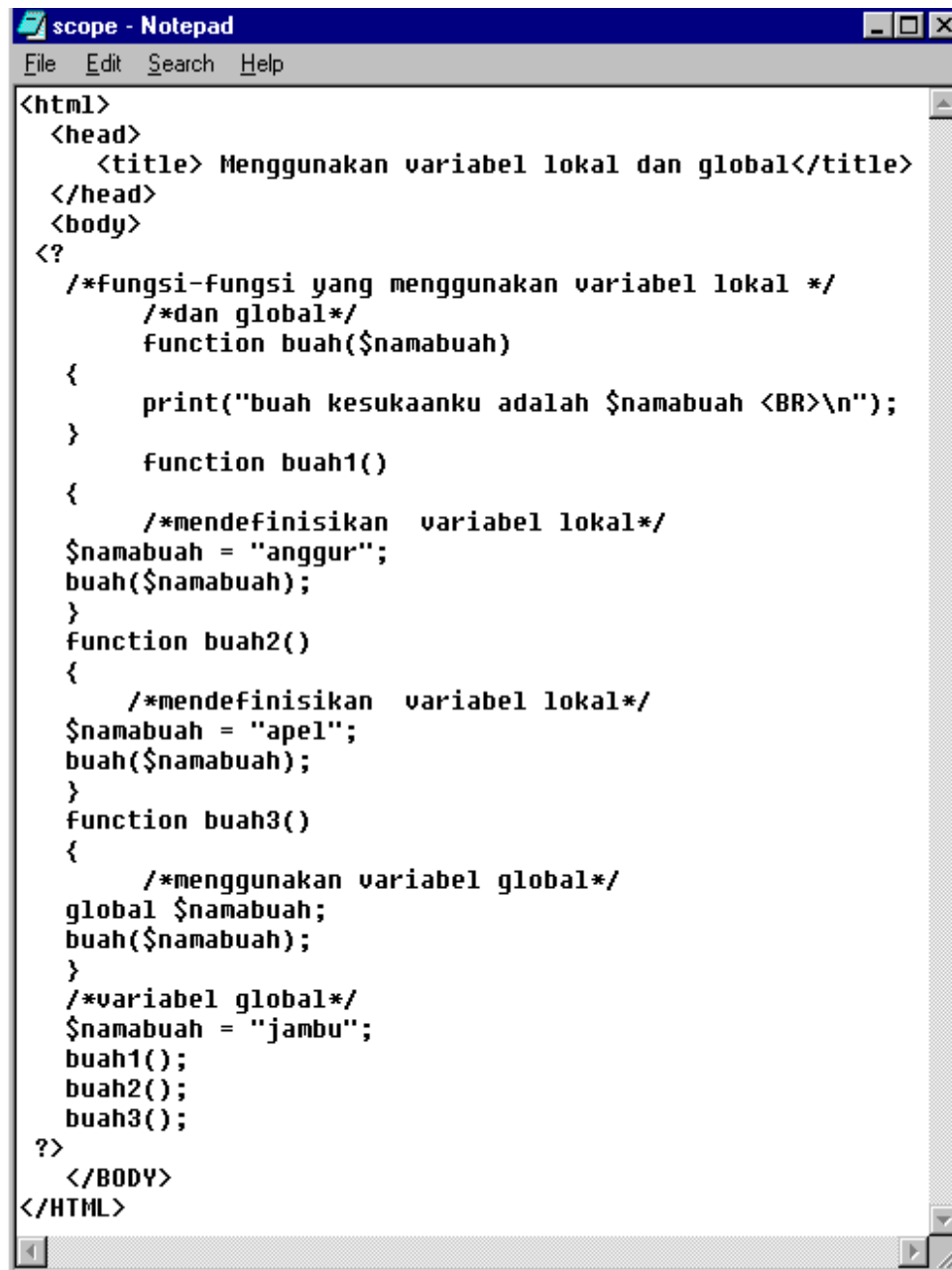
Batas Variabel/variable scope (Lokal dan Global)

Variable Scope bergantung pada konteks di mana variable tersebut dinyatakan. Variable yang dinyatakan di dalam suatu fungsi akan memiliki scope local hanya di dalam fungsi itu saja. Variable yang dinyatakan di dalam bagian sebuah script program akan memiliki scope global dalam script tersebut, variable dalam sebuah file yang dimasukkan akan juga memiliki sifat global. Sebuah variable \$a memiliki nilai hanya setelah variable tersebut didefinisikan. variable ini berlaku padaprogram sisipan (include dan require). Apabila menemui sebuah fungsi, variable ini tidak berlaku lagi di dalam fungsi kecuali variable tersebut dilewatkan sebagai argumen fungsi tersebut. Demikian sebaliknya, sebuah variable yang didefinisikan di dalam fungsi tidak berlaku di luar fungsi kecuali variable tersebut dikembalikan ke pemanggilnya. Agar variable tersebut berlaku baik di program utama maupun di dalam fungsi, maka variable dapat dinyatakan sebagai variable global, artinya variable tersebut berlaku baik di dalam fungsi maupun tidak di dalam fungsi Perhatikan contoh di bawah ini :



```
<HTML><body bgcolor="#FFFFFF">
<p>$a = 'alpha'</p>
<p>$alpha = 5</p>
<p>$$a = <?
$a = 'alpha';
$alpha = 5;
print $$a;
?> </p>
<p>&nbsp;</p><p>&nbsp;</p></body>
</html>
```

Perhatikan contoh di bawah ini :



```
<html>
  <head>
    <title> Menggunakan variabel lokal dan global</title>
  </head>
  <body>
  <?
    /*fungsi-fungsi yang menggunakan variabel lokal */
    /*dan global*/
    function buah($namabuah)
    {
      print("buah kesukaanku adalah $namabuah <BR>\n");
    }
    function buah1()
    {
      /*mendefinisikan variabel lokal*/
      $namabuah = "anggur";
      buah($namabuah);
    }
    function buah2()
    {
      /*mendefinisikan variabel lokal*/
      $namabuah = "apel";
      buah($namabuah);
    }
    function buah3()
    {
      /*menggunakan variabel global*/
      global $namabuah;
      buah($namabuah);
    }
    /*variabel global*/
    $namabuah = "jambu";
    buah1();
    buah2();
    buah3();
  ?>
  </BODY>
</HTML>
```

Variabel dari luar PHP

Apabila sebuah form di-submit, maka variabel dalam form (didefinisikan dengan name sebagai nama variabelnya dan value sebagai nilai variabelnya) akan secara otomatis berlaku di dalam program php yang dituju. Hal ini akan dibahas lebih lanjut di bagian variable.

Untuk lebih jelasnya lagi perhatikan contoh dibawah ini :

```
submit - Notepad
File Edit Search Help
<html>
<head>
<title>Contoh 2</title>
</head>
<body bgcolor="#FFFFFF">
<p><font size="6">formulir registrasi</font></p><hr>
<form method="post" action="submit.php">
<p>Masukkan data diri anda : </p>
<p>Nama : <input type="text" name="nama"><br>
Alamat : <input type="text" name="alamat"><br>
Kota : <input type="text" name="kota"><br>
Kode Pos : <input type="text" name="kodepos"><br>
Tanggal Lahir : <input type="text" name="tgllahir"><br>
Pekerjaan : <input type="text" name="pekerjaan"><br>
<input type="submit" name="Submit" value="Submit">
<input type="reset" name="Submit2" value="Reset">
</p>
</form>
<hr>
<p>&nbsp;</p>
</body>
</html>
```

Tampilan di browsernya adalah :

Address <http://localhost/latihan%20php/DECURE/TES/submit.html>

formulir registrasi

Masukkan data diri anda :

Nama :

Alamat :

Kota :

Kode Pos :

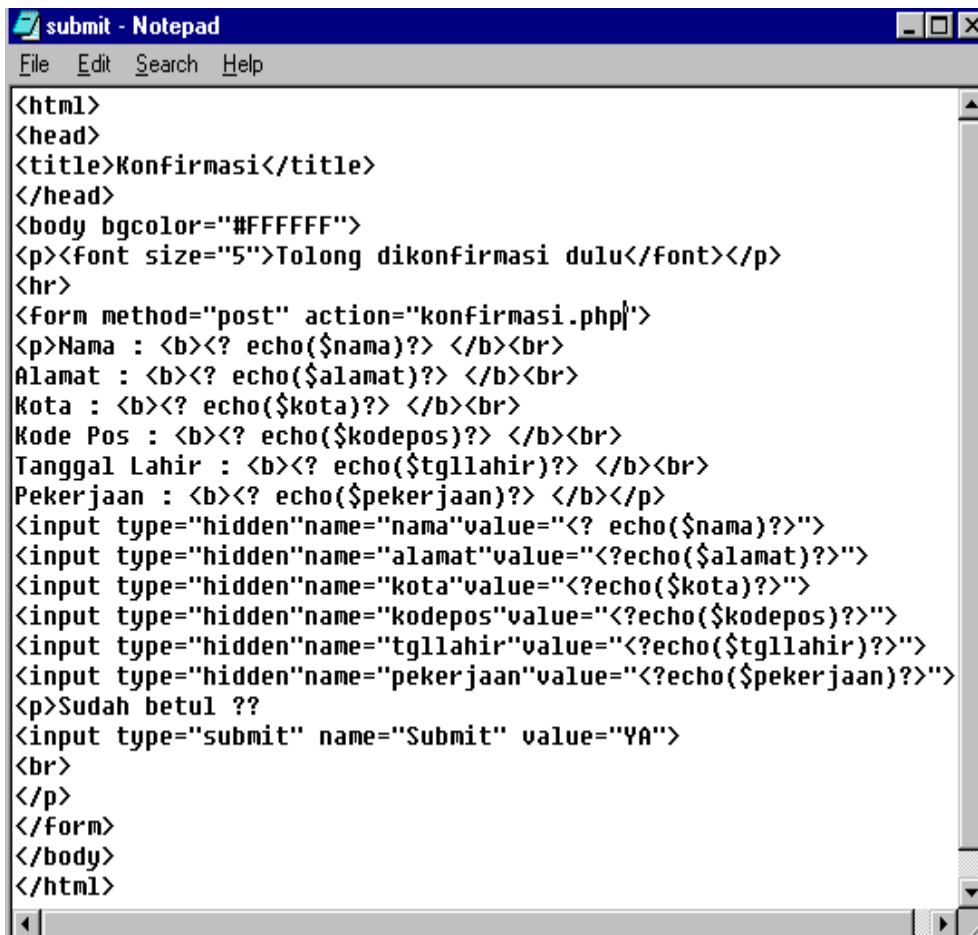
Tanggal Lahir :

Pekerjaan :

Sintak action pada script di atas merupakan tanda bahwa input dari form ini akan di kirim ke suatu file dan akan ditunjukkan pada file tersebut dengan

bantuan antar variable dari file submit.html ke file submit.php yang di letakkan pada suatu folder.

Tampilan file submit.php adalah :



```
<html>
<head>
<title>Konfirmasi</title>
</head>
<body bgcolor="#FFFFFF">
<p><font size="5">Tolong dikonfirmasi dulu</font></p>
<hr>
<form method="post" action="konfirmasi.php">
<p>Nama : <b><? echo($nama)?> </b><br>
Alamat : <b><? echo($alamat)?> </b><br>
Kota : <b><? echo($kota)?> </b><br>
Kode Pos : <b><? echo($kodepos)?> </b><br>
Tanggal Lahir : <b><? echo($tgllahir)?> </b><br>
Pekerjaan : <b><? echo($pekerjaan)?> </b></p>
<input type="hidden" name="nama" value="<? echo($nama)?>">
<input type="hidden" name="alamat" value="<?echo($alamat)?>">
<input type="hidden" name="kota" value="<?echo($kota)?>">
<input type="hidden" name="kodepos" value="<?echo($kodepos)?>">
<input type="hidden" name="tgllahir" value="<?echo($tgllahir)?>">
<input type="hidden" name="pekerjaan" value="<?echo($pekerjaan)?>">
<p>Sudah betul ??
<input type="submit" name="Submit" value="YA">
<br>
</p>
</form>
</body>
</html>
```

Tampilan di browsernya menjadi :


Address  http://localhost/latihan%20php/DECURE/TES/submit.php3

Tolong dikonfirmasi dulu

Nama : **firdaus**
Alamat : **asramabg**
Kota : **bandung**
Kode Pos : **40135**
Tanggal Lahir : **120580**
Pekerjaan : **mahasiswa**

Sudah betul ??

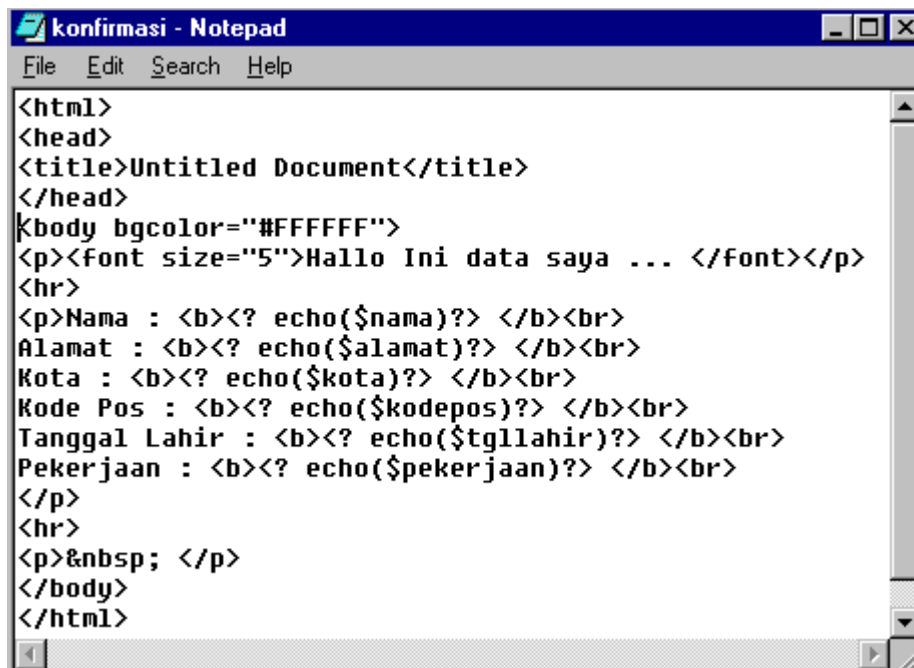
Ketika kita menekan button “YA” pada halaman submit.php3 maka akan tampil di browser seperti di bawah ini :

Address  http://localhost/latihan%20php/DECURE/TES/konfirmasi.php

Hallo Ini data saya ...

Nama : **firdaus**
Alamat : **asramabg**
Kota : **bandung**
Kode Pos : **40135**
Tanggal Lahir : **120580**
Pekerjaan : **mahasiswa**

Tampilan di atas berasal dari script php seperti di bawah ini :



```
<html>
<head>
<title>Untitled Document</title>
</head>
<body bgcolor="#FFFFFF">
<p><font size="5">Hallo Ini data saya ... </font></p>
<hr>
<p>Nama : <b><? echo($nama)?> </b><br>
Alamat : <b><? echo($alamat)?> </b><br>
Kota : <b><? echo($kota)?> </b><br>
Kode Pos : <b><? echo($kodepos)?> </b><br>
Tanggal Lahir : <b><? echo($tgllahir)?> </b><br>
Pekerjaan : <b><? echo($pekerjaan)?> </b><br>
</p>
<hr>
<p>&nbsp; </p>
</body>
</html>
```

Contoh seperti di atas merupakan konfigurasi variable PHP dengan sintak HTML biasa. Yang perlu diingat adalah sintak bagian action pada bagian form-nya. Karena pada bagian inilah yang akan menunjukkan ke halaman mana browser akan menampilkan suatu file lainnya dengan pengalamatan yang di letakkan pada action tersebut.

KONSTANTA DAN OPERATOR

- Konstanta
- Operator

Konstanta

Konstanta merupakan variable yang nilainya tetap. Konstanta hanya diberi nilai pada awal program dan nilainya tidak pernah berubah selama program berjalan. Konstanta juga tidak memerlukan sintak “\$” untuk menyatakannya.

Untuk menyatakannya, PHP memiliki fungsi tertentu yaitu *define()*. PHP juga telah mendefinikan beberapa konstanta , misalnya PHP_VERSION yaitu konstanta yang memberikan informasi tentang versi PHP yang digunakan; konstanta TRUE telah diberi nilai 1 dan FALSE dengan nilai 0.

Sintak untuk membuat konstanta sendiri adalah :

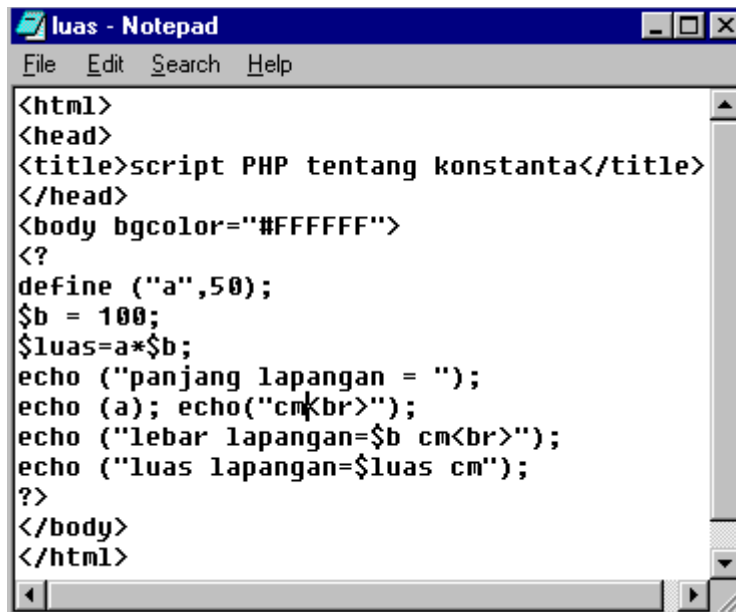
Define (“nama_konstanta,”nilai konstanta”);

Konstanta dideklarasikan hanya satu kali dan nilai tidak dapat diubah atau didefinisikan lagi pada keseluruhan program.

Perhatikan contoh di bawah ini :

```
<html>
<head>
<title>script PHP tentang konstanta</title>
</head>
<body bgcolor=\"#FFFFFF\">
<?
define (\"konstanta\", \"sedang belajar konstanta\");
echo| (konstanta);
?>
</body>
</html>
```

contoh penggunaan pada aplikasi aritmatika :



```
<html>
<head>
<title>script PHP tentang konstanta</title>
</head>
<body bgcolor="#FFFFFF">
<?
define ("a",50);
$b = 100;
$luas=a*$b;
echo ("panjang lapangan = ");
echo (a); echo("cm<br>");
echo ("lebar lapangan=$b cm<br>");
echo ("luas lapangan=$luas cm");
?>
</body>
</html>
```

Operator

Operator adalah symbol yang digunakan untuk memanipulasi data, seperti penambahan dan pengurangan. Ada operator yang menggunakan satu operan dan ada pula yang menggunakan dua operan.

Operator dapat dikelompokkan dalam empat kelompok, yaitu :

1. Operator aritmatika, adalah operator yang berhubungan dengan fungsi matematika
2. Operator logika adalah operator yang membandingkan antar true dan false
3. Operator bitwise adalah operator yang membandingkan binary
4. Operator jenis lain.

Operator Aritmatika

Operator aritmatika merupakan operator yang berhubungan dengan fungsi matematika.

contoh	Nama	Hasil
\$a + \$b	Penjumlahan	Jumlah dari \$a dan \$b
\$a - \$b	Pengurangan	Selisih dari \$a dan \$b

\$a * \$b	Perkalian	Hasil kali dari \$a dan \$b
\$a / \$b	Pembagian	Pembagian dari \$a dan \$b
\$a % \$b	Modulus	Sisa dari \$a dibagi dengan \$b

Contoh script PHP yang menggunakan operator aritmatika adalah :

```

<HTML>
<HEAD>
  <TITLE>Operator Aritmatika</TITLE>
</HEAD>
<BODY>
<?
$Harga = 50;
$Pengunjung = 4;
print("Jumlah pengunjung hari ini $Pengunjung <BR>\n");
print("Total pendapatan : ");
print($Harga * $Pengunjung);
?>
</BODY>
</HTML>

```

Operator Assignment

Assignment adalah operasi pemberian/pendefinisian nilai. Operator assignment antara lain =, +=, -=, *=, .=.

\$a = (\$b = 4) + 5 : \$a sama dengan sembilan sedangkan \$b sama dengan 4.

Operator Perbandingan

Operator perbandingan ini digunakan pada struktur kontrol program, seperti if, elseif, di mana dilakukan perbandingan antar dua nilai.

contoh	Nama	Hasil
\$a == \$b	sama dengan	benar jika \$a sama dengan \$b
\$a === \$b	Identik	benar jika \$a sama dengan \$b dan memiliki jenis besaran yang sama
\$a != \$b	tidak sama	benar jika \$a tidak sama dengan \$b
\$a < \$b	lebih kecil	benar jika \$a lebih kecil \$b
\$a > \$b	lebih besar	benar jika \$a lebih besar \$b
\$a <= \$b	lebih kecil atau sama dengan	benar jika \$a lebih kecil atau sama dengan \$b

$\$a \geq \b	lebih besar atau sama dengan	benar jika \$a lebih besar atau sama dengan \$b
----------------	------------------------------	---

Operator Logikal

contoh	Nama	Hasil
$\$a \text{ and } \b	AND	benar jika \$a dan \$b benar
$\$a \text{ or } \b	OR	benar jika \$a atau \$b benar
$\$a \text{ xor } \b	OR	benar jika \$a atau \$b benar tetapi tidak keduanya
$!\$a$	NOT	benar jika \$a tidak benar
$\$a \&\& \b	AND	benar jika \$a dan \$b benar
$\$a \ \ \b	OR	benar jika \$a atau \$b benar

Operator String

Penjumlahan string dapat dilakukan dengan operator (.).

Operator Bitwise

Operator ini digunakan untuk operasi-operasi bilangan binary. Operator ini untuk memanipulasi bit-bit dari nilai data.

Jenis-jenis operator ini ada di dalam tabel operator di bawah ini, adalah :

Operator	Operasi
$\$a \& \b	And
$\$a \b	Or
$\$a \wedge \b	Xor
$\sim \$a$	Not
$\$a \ll \b	Pergeseran bit ke kiri
$\$a \gg \b	Pergeseran bit ke kanan

Operator Increment/Decrement

Tabel operatr Increment/decrement adalah :

Operator	Operasi	Contoh
$++\$a$	Pre-increment	Nilai \$a ditambah satu, kemudian operasi berjalan
$\$a++$	Post-increment	Operasi dilaksanakan, kemudian \$a ditambah satu

--\$a	Pre-decrement	Nilai \$a dikurang satu, kemudian operasi dilaksanakan satu
\$a--	Post-decrement	Operasi dilaksanakan, kemudian \$a dikurangi satu

Perhatikan contoh di bawah ini :

```

<html>
<body>
<?
$var=1;
echo ("(operasi++\var) nilai\var=" . ++$var);
echo ("(operasi\var++) nilai\var=" . $var++);
echo ("(operasi--\var) nilai\var=" . --$var);
echo ("(operasi\var--) nilai\var=" . $var--);
?>
</body>
</html>

```

Operator Precedence (ekspresi)

Precedence adalah urutan atau level dari operator, operator dengan level yang lebih tinggi akan dieksekusi lebih dahulu. Contoh :

$$\$hasil = 8 - 4 * 2$$

Dari contoh di atas, maka variable \$hasil akan memiliki hasil 2 bukan 10, hal ini disebabkan operator (*) memiliki level yang lebih tinggi dari operator (-).

Perhatikan contoh di bawah ini :

```

<html>
<body>
<?
//perintah yang menghasilkan angka 12
print ((4+2*4) . "<br>");
//perintah yang menghasilkan angka 24
print(((4+2)*4));
?>
</body>
</html>

```

PERNYATAAN (STATEMENT)

- **Pembagian Statement**

Struktur kontrol (statement) yang dimiliki PHP berfungsi hampir sama dengan statement program lainnya yaitu berfungsi sebagai rangka badan program yang mengatur aliran program. Rangkaian program yang ditulis harus memenuhi kondisi sebagai berikut :

- Melanjutkan sebuah pernyataan bila kondisi terpenuhi.
- Mengulang suatu perintah bila kondisi telah terpenuhi
- Memiliki sebuah pilihan dari beberapa alternatif bila kondisi telah terpenuhi.

Pembagian Statement

1. Statement if

Statement if digunakan jika satu atau lebih operasi akan dilaksanakan jika syaratnya telah terpenuhi.

Sintak dari statement tersebut adalah :

```
<?
If (persyaratan) {
Operasi program;
}
```

?>

Operasi program akan berjalan apabila persyaratan terpenuhi atau bernilai true, jika tidak terpenuhi maka operasi program akan diabaikan.

Perhatikan contoh di bawah ini :



```
<HTML>
<HEAD>
  <TITLE>Struktur Kontrol IF</TITLE>
</HEAD>
<BODY>
<?
$X=100;
$Y=10;
if ($X > $Y)
{
  print('X lebih besar dari Y');
}
?>
</BODY>
</HTML>
```

2. Statement if...else

Statement ini hampir sama dengan statement “if”, tetapi pada statement ini digunakan untuk banyak blok perintah. Atau dengan kata lain, jika pernyataan if memiliki beberapa permasalahan seperti memiliki dua alternatif, dengan syarat jika persyaratan dipenuhi dilakukan operasi 1, jika tidak, dilakukan oleh operasi 2.

Kondisi di atas dapat di gambarkan dengan sintak :

If (kondisi1)

{ pernyataan 1 akan berjalan jika kondisi 1 bernilai benar }

elseif (kondisi2)

{ pernyataan 2 akan berjalan jika kondisi 2 bernilai benar dan kondisi 1 bernilai salah }

else

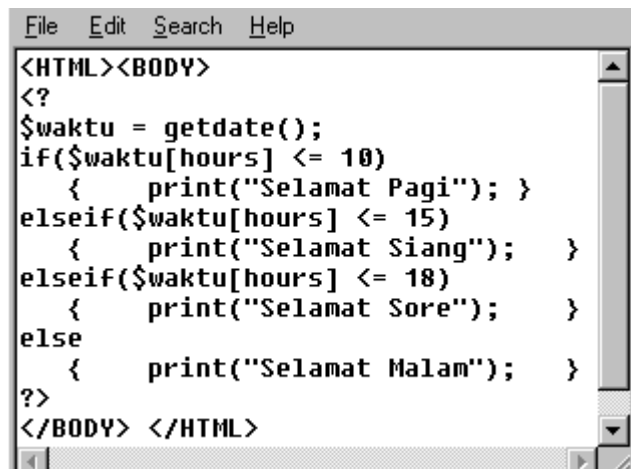
{ Pernyataan ini akan berjalan, jika kondisi 1 dan 2 bernilai salah }

Untuk lebih jelasnya perhatikan contoh di bawah ini :



```
if - Notepad
File Edit Search Help
<html>
<head>
<title>Untitled Document</title>
</head>
<body bgcolor="#FFFFFF">
<p>$a = 5</p>
<p>$b = 3</p>
<p>$c = 7</p>
<?
$a = 5;
$b = 3;
$c = 7;
?>
<p>if ($a == $b) adalah : <? if ($a==$b)
{ print "benar"; } else { print "salah"; } ?></p>
<p>if ($a &lt; $c) adalah : <? if ($a<$c)
{ print "benar"; } else { print "salah"; } ?></p>
<p>if ($a &gt; $b) benar menghasilkan BENAR,
selain itu jika ($b &lt; $c) maka
menghasilkan MERAH, selain itu BIRU.</p>
<p>Hasilnya : <? if ($a>$b) { print "BENAR"; } elseif
($b < $c) { print "MERAH"; } else { print "BIRU"; } ?></p>
<p>if ($a &lt; $b) benar menghasilkan BENAR,
selain itu jika ($b &lt; $c) maka
menghasilkan MERAH, selain itu BIRU.</p>
<p>Hasilnya : <? if ($a<$b) { print "BENAR"; } elseif
($b < $c) { print "MERAH"; } else { print "BIRU"; } ?></p>
</body>
</html>
```

Perhatikan juga contoh di bawah ini :

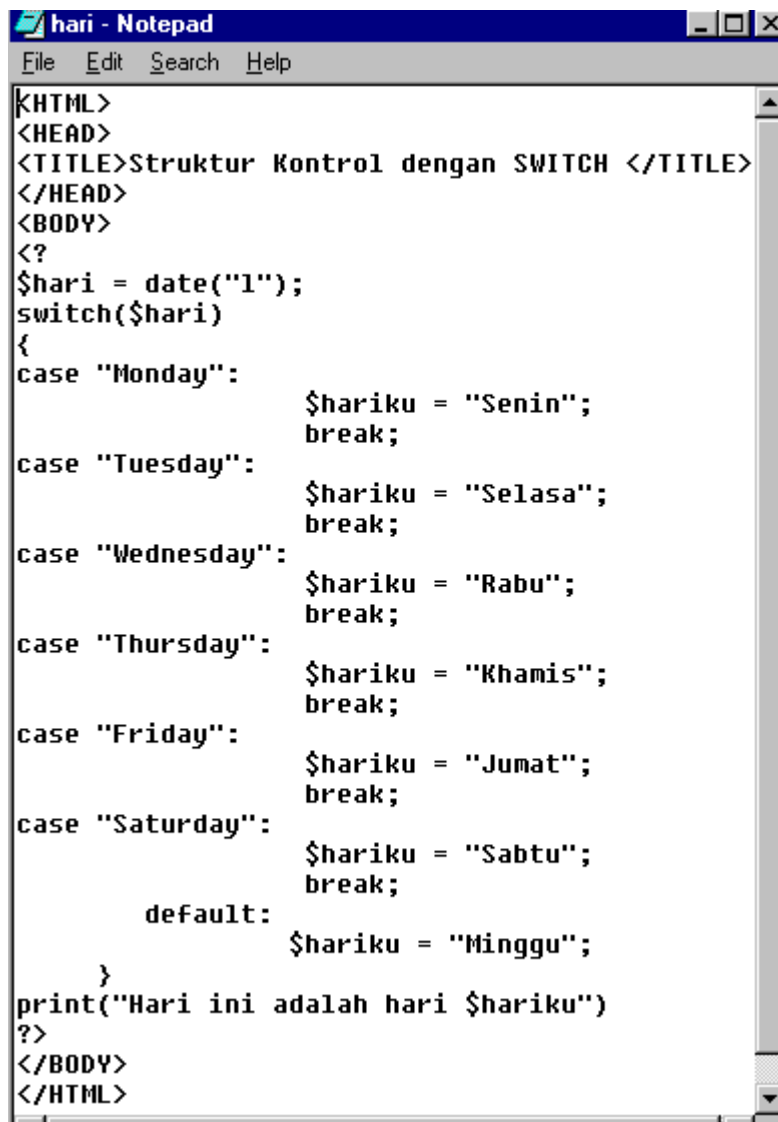
A screenshot of a text editor window with a menu bar containing 'File', 'Edit', 'Search', and 'Help'. The main area contains PHP code wrapped in HTML tags. The code uses the 'getdate()' function to get the current time and then uses a series of 'if', 'elseif', and 'else' statements to print different greetings based on the hour of the day: 'Selamat Pagi' (10:00-15:00), 'Selamat Siang' (15:00-18:00), 'Selamat Sore' (18:00-21:00), and 'Selamat Malam' (21:00-00:00).

```
<HTML><BODY>
<?
$waktu = getdate();
if($waktu[hours] <= 10)
{   print("Selamat Pagi"); }
elseif($waktu[hours] <= 15)
{   print("Selamat Siang"); }
elseif($waktu[hours] <= 18)
{   print("Selamat Sore"); }
else
{   print("Selamat Malam"); }
?>
</BODY> </HTML>
```

3. Switch

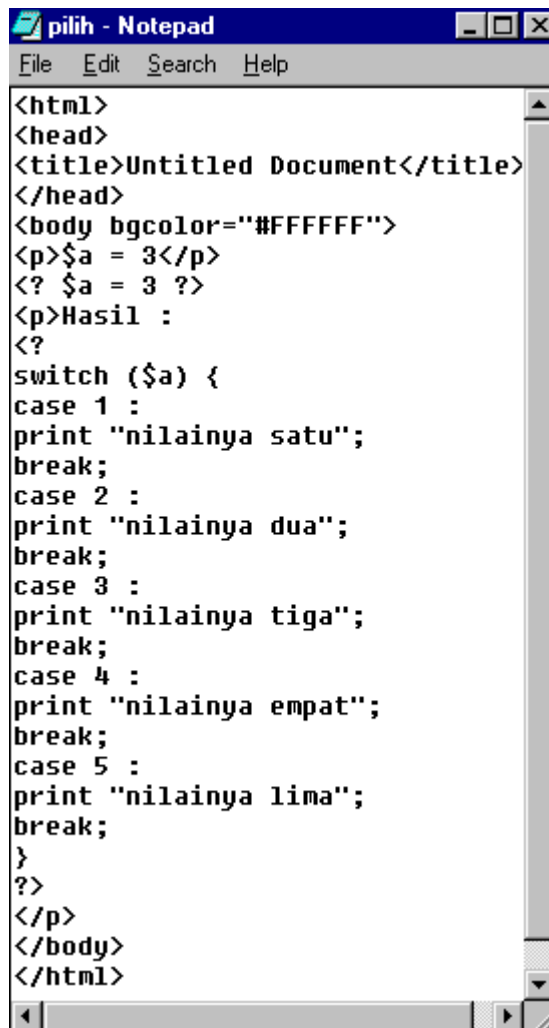
Statement ini digunakan sebagai alternatif pengganti dari struktur if...else dengan else lebih dari satu. Dengan menggunakan statement ini, program akan semakin mudah dibuat dan digunakan karena peran switch adalah digunakan pada saat ditemukannya suatu masalah antara membandingkan suatu variable dengan berbagai nilai.

Perhatikan contoh di bawah ini :



```
<HTML>
<HEAD>
<TITLE>Struktur Kontrol dengan SWITCH </TITLE>
</HEAD>
<BODY>
<?
$hari = date("l");
switch($hari)
{
case "Monday":
    $hariku = "Senin";
    break;
case "Tuesday":
    $hariku = "Selasa";
    break;
case "Wednesday":
    $hariku = "Rabu";
    break;
case "Thursday":
    $hariku = "Khamis";
    break;
case "Friday":
    $hariku = "Jumat";
    break;
case "Saturday":
    $hariku = "Sabtu";
    break;
    default:
    $hariku = "Minggu";
}
print("Hari ini adalah hari $hariku")
?>
</BODY>
</HTML>
```

Perhatikan contoh di bawah ini :



```
<html>
<head>
<title>Untitled Document</title>
</head>
<body bgcolor="#FFFFFF">
<p>$a = 3</p>
<? $a = 3 ?>
<p>Hasil :
<?
switch ($a) {
case 1 :
print "nilainya satu";
break;
case 2 :
print "nilainya dua";
break;
case 3 :
print "nilainya tiga";
break;
case 4 :
print "nilainya empat";
break;
case 5 :
print "nilainya lima";
break;
}
?>
</p>
</body>
</html>
```

STATEMENT (lanjutan)

- Pengulangan

Pengulangan

Loop merupakan proses eksekusi operasi program secara berulang-ulang sampai ditemui kondisi untuk mengakhiri eksekusi tersebut. Dalam hal ini kita tidak perlu menulis ulang sebuah perintah sebanyak pengulangan yang diinginkan.

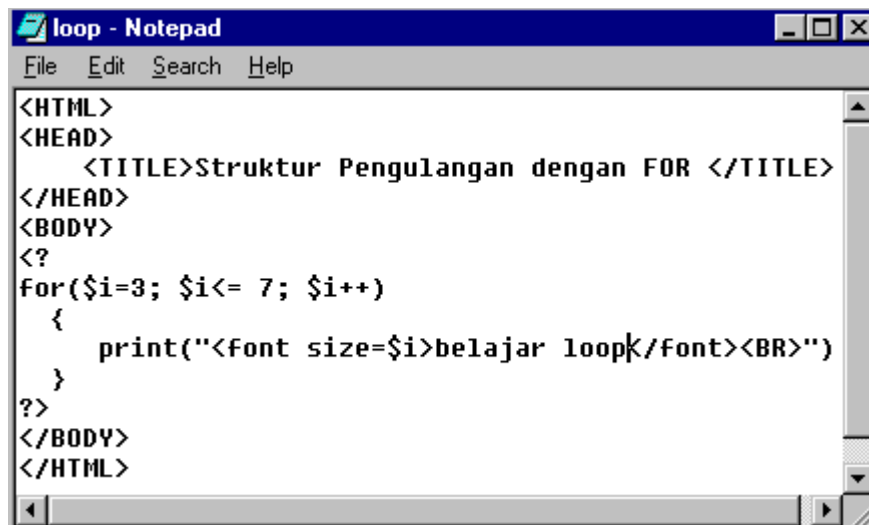
PHP memiliki 3 jenis perintah loop (perulangan), yaitu :

1. for

Perintah FOR difungsikan untuk mengulangi perintah dengan jumlah perulangan yang sudah diketahui. Pada perintah ini kita tidak perlu menuliskan sebuah kondisi untuk diuji. Kita hanya menuliskan nilai awal dan akhir variabel perhitungan. Nilai ini akan secara otomatis bertambah dan berkurang tiap kali sebuah pengulangan dilaksanakan. Sintaknya adalah :

```
for (nilai_awal, nilai_akhir, peningkatan/penurunan)  
{  
    pernyataan yang akan dijalankan  
}
```

Untuk lebih jelasnya lagi, perhatikan contoh di bawah ini :



```
<HTML>
<HEAD>
  <TITLE>Struktur Pengulangan dengan FOR </TITLE>
</HEAD>
<BODY>
<?
for($i=3; $i<= 7; $i++)
{
  print("<font size=$i>belajar loop</font><BR>")
}
?>
</BODY>
</HTML>
```

Jika kita simak dengan seksama, maka script ini hampir sama dengan script pada modul javascript yang lalu.

2. foreach

Perintah ini digunakan untuk melakukan iterasi pada sebuah nilai array.

Sintaknya adalah :

```
foreach (ekspresi_array as $nilai/value)
{
  operasi program;
}
```

Perhatikan contoh di bawah ini :

```
foreach - Notepad
File Edit Search Help

<html>
<head>
<title>Untitled Document</title>
</head>
<body bgcolor="#FFFFFF">
<p>Cetak angka 1 s/d 100 :</p>
<p><?
for ($i=1; $i<=100; $i++) {
print "$i ";
}
?> </p>
<p>Cetak angka 1 s/d 100 dengan kenaikan 2 angka :</p>
<p><?
for ($i=1; $i<=100; $i+=2) {
print "$i ";
}
?></p>
<p>$p = array(&quot;senin&quot;,&quot;selasa&quot;
,&quot;rabu&quot;,&quot;kamis&quot;,&quot;jumat&quot;
,&quot;sabtu&quot;,&quot;minggu&quot;);
</p>
<p>cetak hari-hari dalam seminggu :</p>
<p><?
$p=array("senin","selasa","rabu","kamis","jumat","sabtu","minggu");
foreach ($p as $hari) {
print "$hari <br>";
}
?>&nbsp; </p>
</body>
</html>
```

3. while

Perintah while digunakan untuk mengulangi sebuah perintah sampai pada jumlah tertentu. Untuk menghentikan pengulangan digunakan suatu kondisi tertentu. Nilai kondisi ini memiliki hasil akhir berupa false dan true seperti layaknya perintah *if...else*. Pengulangan akan terus berlanjut jika kondisi masih menunjukkan nilai benar.

Sintaknya adalah :

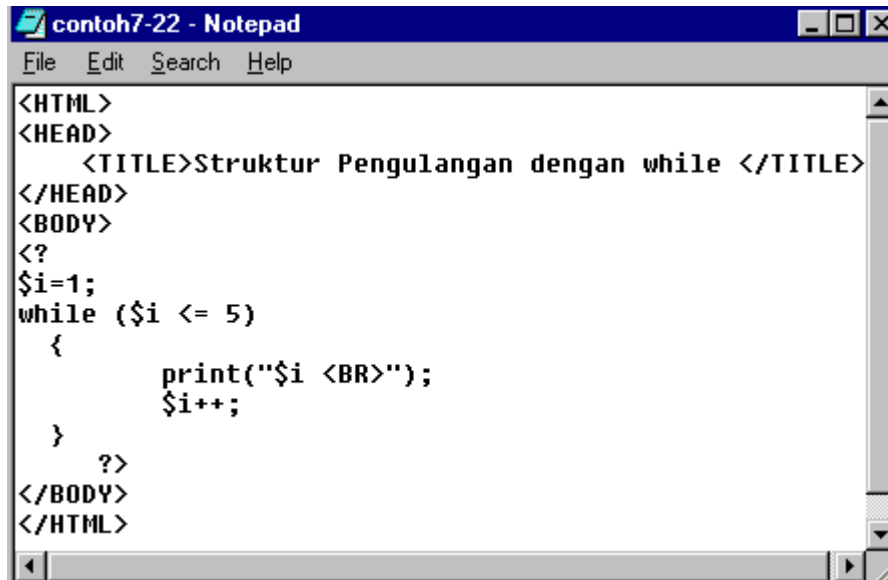
While (kondisi)

{

Pernyataan yang akan dijalankan

}

Perhatikan contoh dibawah ini :

A screenshot of a Notepad window titled "contoh7-22 - Notepad". The window contains the following HTML code:

```
<HTML>
<HEAD>
  <TITLE>Struktur Pengulangan dengan while </TITLE>
</HEAD>
<BODY>
<?
$i=1;
while ($i <= 5)
{
    print("$i <BR>");
    $i++;
}
?>
</BODY>
</HTML>
```

Untuk bias masuk ke dalam system pengulangan (loop) while, kondisi harus dibuat true terlebih dahulu. Untuk menghentikan kondisi while maka di akhir pernyataan harus dibuat kondisi false (salah), hal ini dapat terlihat dari contoh di atas, di mana nilai variable i selalu berubah sehingga menyebabkan kondisi menjadi tidak terpenuhi dan loop akan berhenti.

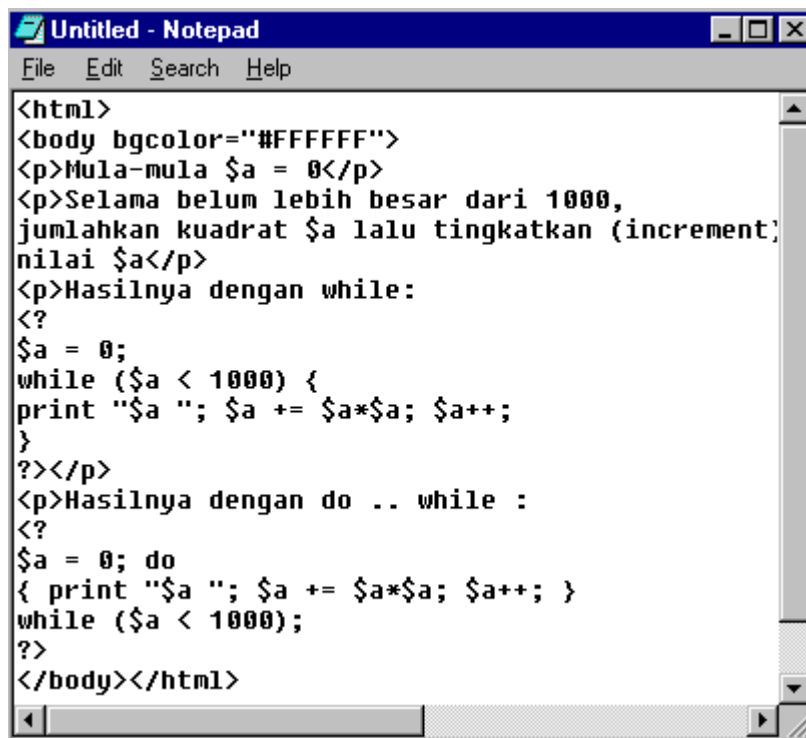
4. do..while

Pada prinsipnya, metode menggunakan do..while dengan while hampir sama. Tetapi yang membedakannya adalah terletak pada eksekusi operasi program. Pada pernyataan while, maka persyaratan terlebih dahulu diperiksa, jika persyaratan bernilai true maka proses eksekusi bias dijalankan. Tetapi proses pada do..while adalah, operasi program dieksekusi terlebih dahulu baru kemudian persyaratannya diperiksa. Jika persyaratannya true, maka loop akan terus dijalankan, loop hanya akan berhenti jika proses memiliki hasil false. Oleh karena itu operasi program minimal dapat dieksekusi sekali.

Sintak dasarnya adalah :

```
do
{
    pernyataan yang akan dijalankan
}
while (kondisi)
```

Untuk lebih jelasnya lagi, perhatikan contoh di bawah ini :



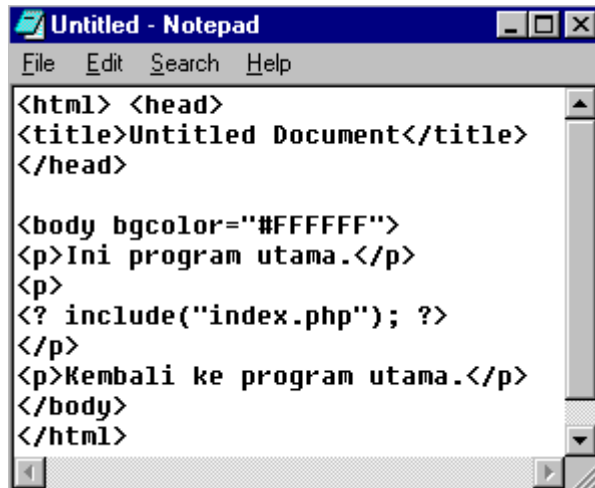
```
Untitled - Notepad
File Edit Search Help
<html>
<body bgcolor="#FFFFFF">
<p>Mula-mula $a = 0</p>
<p>Selama belum lebih besar dari 1000,
jumlahkan kuadrat $a lalu tingkatkan (increment)
nilai $a</p>
<p>Hasilnya dengan while:
<?
$a = 0;
while ($a < 1000) {
print "$a "; $a += $a*$a; $a++;
}
?></p>
<p>Hasilnya dengan do .. while :
<?
$a = 0; do
{ print "$a "; $a += $a*$a; $a++; }
while ($a < 1000);
?>
</body></html>
```

5. require dan include :

Include dan require digunakan untuk memanggil dan mengeksekusi file yang ditentukan. Dengan kedua statement ini dapat dibuat fungsi-fungsi, konstanta ataupun perintah operasi biasa dalam sebuah file terpisah yang dapat dipanggil dari file lainnya. Maksudnya adalah jika kita memiliki suatu script PHP yang lengkap dan script tersebut ingin kita panggil dari sebuah file lainnya dengan tujuan untuk menampilkan atau

menggunakan script yang ada di file tersebut maka kita harus menggunakan sintak require atau include.

Perhatikan contoh dibawah ini :

A screenshot of a Notepad window titled "Untitled - Notepad". The window has a menu bar with "File", "Edit", "Search", and "Help". The text area contains the following HTML code:

```
<html> <head>
<title>Untitled Document</title>
</head>

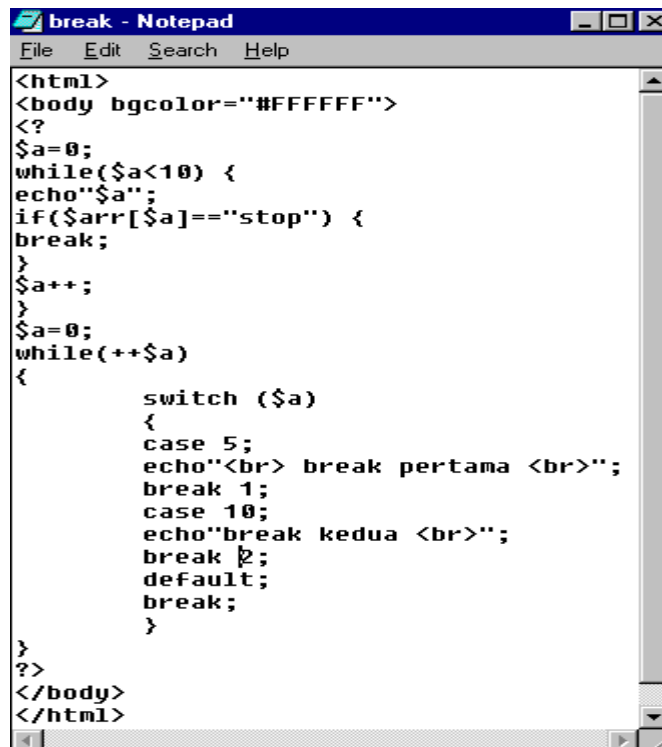
<body bgcolor="#FFFFFF">
<p>Ini program utama.</p>
<p>
<? include("index.php"); ?>
</p>
<p>Kembali ke program utama.</p>
</body>
</html>
```

Perhatikan sintak include di atas, di dalam tanda kurung yang menyertai sintak include ada nama file index.php. Jika saja script yang ada di atas (contoh penggunaan pernyataan do..while) kita simpan dengan nama file index.php dan kita letakkan dalam satu folder dengan script yang di atas, maka isi tampilan dan juga sintak-sintak yang dapat digunakan pada file di atas akan terintegrasi dengan sendirinya.

6. Break

Break digunakan untuk menghentikan iterasi dari sebuah loop. Break dapat diikuti dengan numeric argumen yang menunjukkan berapa tingkatan loop yang dihentikan iterasinya.

Perhatikan contoh di bawah ini :

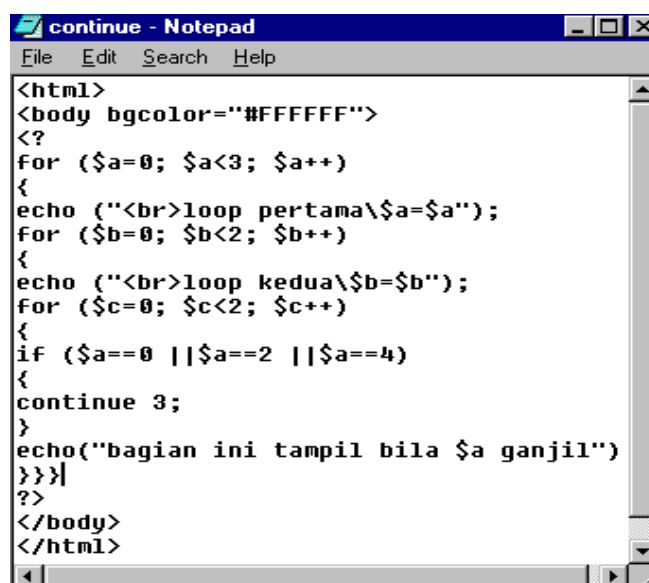


```
<html>
<body bgcolor="#FFFFFF">
<?
$a=0;
while($a<10) {
echo"$a";
if($arr[$a]=="stop") {
break;
}
$a++;
}
$a=0;
while(++$a)
{
    switch ($a)
    {
        case 5;
        echo"<br> break pertama <br>";
        break 1;
        case 10;
        echo"break kedua <br>";
        break 2;
        default;
        break;
    }
}
?>
</body>
</html>
```

7. Continue

Continue digunakan untuk kembali ke awal loop dan sisa operasi program di bawahnya akan diabaikan. Continue diikuti numeric argumen menunjukkan di mana loop akan dimulai kembali.

Contoh :



```
<html>
<body bgcolor="#FFFFFF">
<?
for ($a=0; $a<3; $a++)
{
echo ("<br>loop pertama\\$a=$a");
for ($b=0; $b<2; $b++)
{
echo ("<br>loop kedua\\$b=$b");
for ($c=0; $c<2; $c++)
{
if ($a==0 ||$a==2 ||$a==4)
{
continue 3;
}
}
}
echo("bagian ini tampil bila $a ganjil")
}}|
?>
</body>
</html>
```

FUNGSI

- Pengantar Fungsi
- Fungsi buatan
- Fungsi dengan Argumen
- Variable Fungsi
- Fungsi Rekursif

Pengantar Fungsi

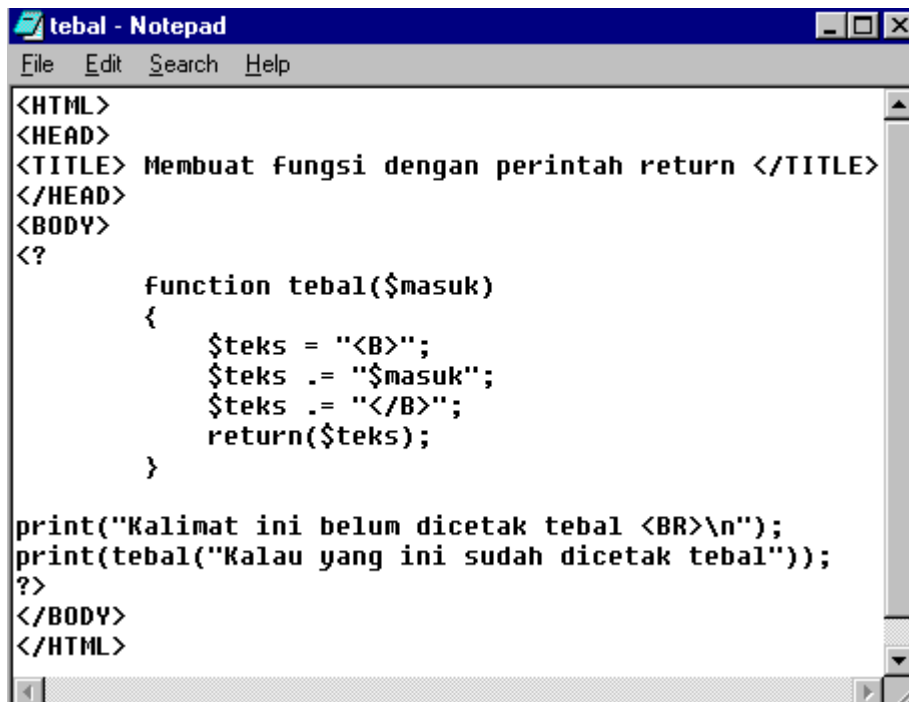
Fungsi merupakan sekumpulan perintah operasi program yang dapat menerima argumen input dan dapat memberikan hasil output berupa sebuah nilai atau sebuah hasil operasi. Fungsi dideklarasikan dengan statement function diikuti nama fungsi dan beberapa variable input jika ada. PHP kaya dengan fungsi-fungsi untuk mengolah string, kode html, file dan terutama akses terhadap berbagai macam database. Dokumentasi lengkap dapat ditemukan dalam manual PHP.

Dalam pemograman, ada dua macam subrutin yaitu prosedur dan fungsi. Perbedaannya adalah jika fungsi berguna untuk mengembalikan nilai tertentu sedangkan prosedur tidak mengembalikan nilai tersebut.

Sintak fungsi dalam PHP adalah :

```
Function nama_function (argumen)  
{  
kode perintah  
}
```

Untuk lebih paham, perhatikan contoh di bawah ini :



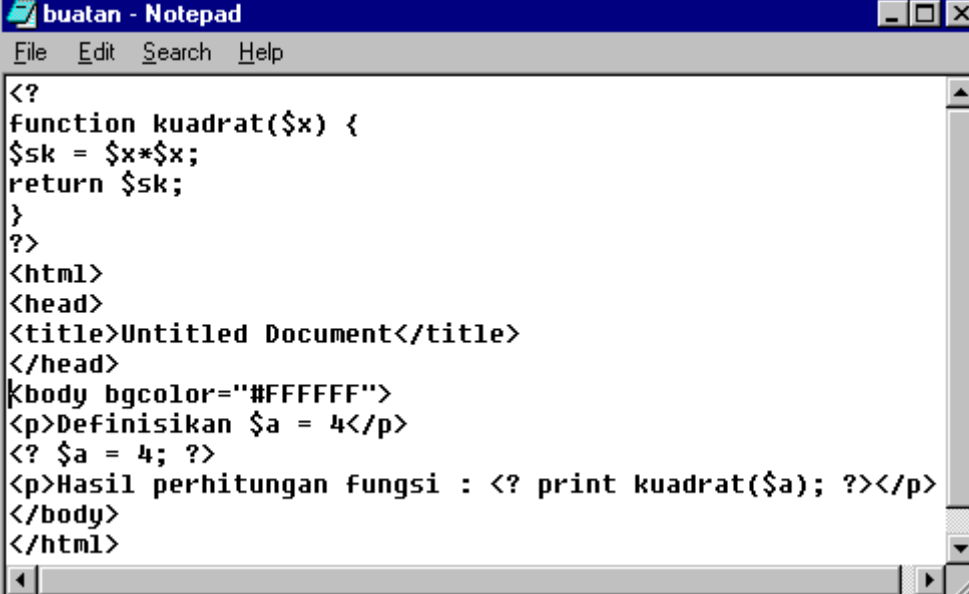
```
<HTML>
<HEAD>
<TITLE> Membuat fungsi dengan perintah return </TITLE>
</HEAD>
<BODY>
<?
    function tebal($masuk)
    {
        $teks = "<B>";
        $teks .= "$masuk";
        $teks .= "</B>";
        return($teks);
    }

print("Kalimat ini belum dicetak tebal <BR>\n");
print(tebal("Kalau yang ini sudah dicetak tebal"));
?>
</BODY>
</HTML>
```

Fungsi buatan

PHP memungkinkan seseorang membuat fungsi dan menggunakannya dalam program, baik ditulis bersamaan dengan program maupun secara terpisah (file yang berlainan dan diakses menggunakan `include` atau `require`). Sintaks fungsi tersebut ditunjukkan di bawah ini :

Nilai suatu variabel dapat dilewatkan ke dalam fungsi melalui argumen fungsi (variabel tersebut juga dapat dinyatakan sebagai variabel global, namun akan cukup sulit apabila jumlah variabel yang harus dilewatkan ke dalam fungsi-fungsi cukup banyak). Hasil operasi fungsi juga dapat dikeluarkan dari fungsi dengan menyatakannya dalam fungsi (dengan *return*). Perhatikan contoh di bawah ini



```
<?
function kuadrat($x) {
$sk = $x*$x;
return $sk;
}
?>
<html>
<head>
<title>Untitled Document</title>
</head>
<body bgcolor="#FFFFFF">
<p>Definisikan $a = 4</p>
<? $a = 4; ?>
<p>Hasil perhitungan fungsi : <? print kuadrat($a); ?></p>
</body>
</html>
```

Fungsi dengan Argumen

Argumen Merupakan suatu nilai tertentu yang dimasukkan ke dalam sebuah fungsi. Secara default sebuah argumen bersifat pass by value. Artinya adalah nilainya saja yang dibutuhkan sehingga nilai pada variable tersebut tidak mengalami perubahan. Jika nilai tersebut diubah di dalam operasi fungsi tersebut, nilai asli pada variable tidak berubah, jika kita ingin mengubah nilai tersebut maka kita harus melakukan pengiriman argumen ke fungsi sebagai reference.

Perhatikan contoh argumen yang menggunakan sifat pass by value di bawah ini :

```
argumen - Notepad
File Edit Search Help
<HTML>
<BODY>
<?
function XpangkatY($X, $Y)
{
    if($Y == 0)
    {
        $hasil = 1;
    }
    elseif($Y > 0)
    {
        $hasil = $X;
        for($I = 1;$I < $Y; $I++)
        {
            $hasil *= $X;
        }
    }
    elseif($Y < 0)
    {
        $hasil = 1;
        for($I = 1;$I < $Y; $I ++)
        {
            $hasil /= $X;
        }
    }
}
return($hasil);
print(XpangkatY(2,10));
?>
</BODY>
</HTML>
```

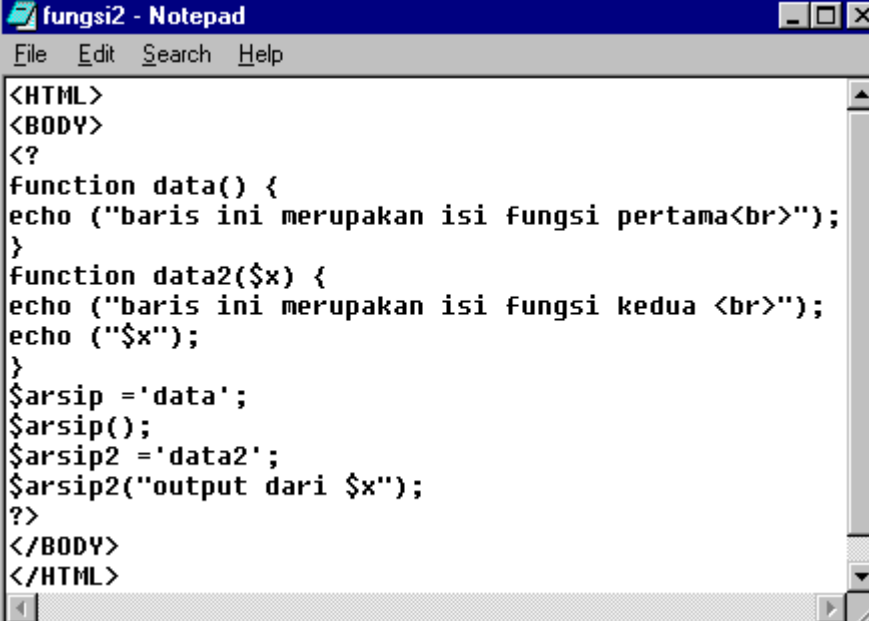
Contoh argumen yang menggunakan sifat pass by reference adalah :

```
reference - Notepad
File Edit Search Help
<HTML>
<BODY>
<?
function arsip(&$data)
{
    $data .= "salman ITB";
}
$unit = "PUSTENA unit dari ";
arsip($unit);
print($unit);
?>
</BODY>
</HTML>
```

Variable Fungsi

Variable fungsi merupakan sebuah variable yang berisi sebuah fungsi di dalam sebuah script PHP.

Perhatikan contoh di bawah ini :



```
<HTML>
<BODY>
<?
function data() {
echo ("baris ini merupakan isi fungsi pertama<br>");
}
function data2($x) {
echo ("baris ini merupakan isi fungsi kedua <br>");
echo ("$x");
}
$arsip = 'data';
$arsip();
$arsip2 = 'data2';
$arsip2("output dari $x");
?>
</BODY>
</HTML>
```

Fungsi Rekursif

Fungsi rekursif merupakan fungsi yang memanggil dirinya sendiri secara berulang-ulang. Untuk lebih memahaminya, perhatikan contoh di bawah ini :

Tampilan di browsernya adalah :



```
Apakah 0 bilangan Integer?Tepat
Apakah 5 bilangan Integer?Tepat
Apakah -5 juga bilangan Integer?Tepat
Apakah 2.5 bilangan Integer?Bukan
```

Gambar di atas berasal dari script PHP di bawah ini :

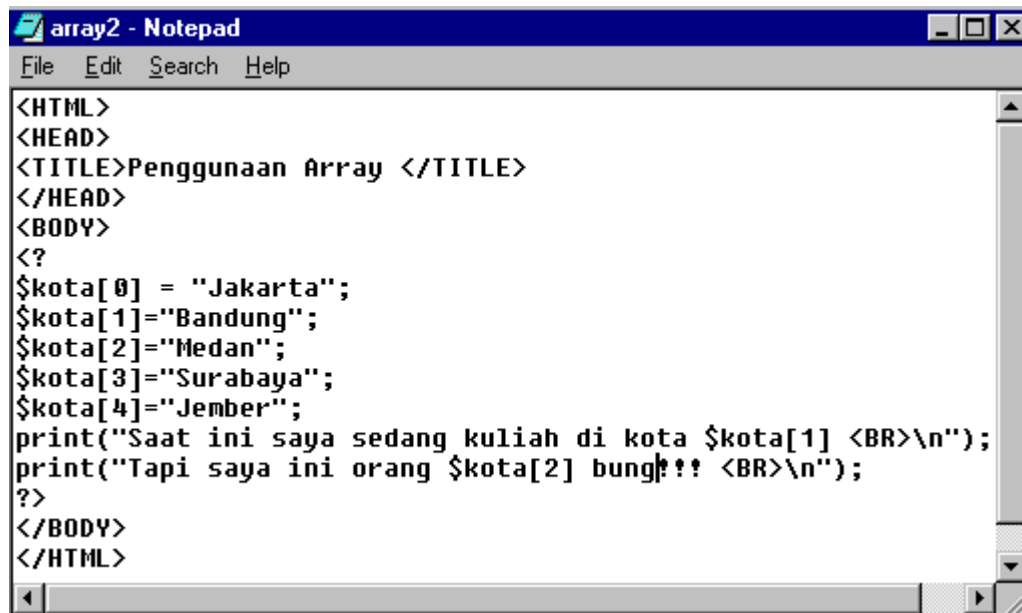
```
rekursif - Notepad
File Edit Search Help
<HTML>
<HEAD>
  <TITLE> Fungsi Rekursif </TITLE>
</HEAD>
<BODY>
<?
function cekInteger($nilai)
{
  if($nilai > 1)
  { return(cekInteger($nilai - 1));
  }
  elseif($nilai < 0)
  { return(cekInteger((- 1)* $nilai - 1));
  }
  else
  { if(($nilai > 0)and ($nilai < 1))
    { return("Bukan");
    }
    else
    { return("Tepat");
    }
  }
}
print("Apakah 0 bilangan Integer?" .cekInteger(0)."<BR>");
print("Apakah 5 bilangan Integer?" .cekInteger(5)."<BR>");
print("Apakah -5 juga bilangan Integer?" .cekInteger(-5)."<BR>");
print("Apakah 2.5 bilangan Integer?" .cekInteger(2.5)."<BR>");
?>
</BODY>
</HTML>
```

ARRAY

- Pengantar Array
- Array Multidimensi
- Urutan Elemen
- Mengambil dan Menggabungkan Elemen
- Fungsi `array_walk`

Pengantar Array

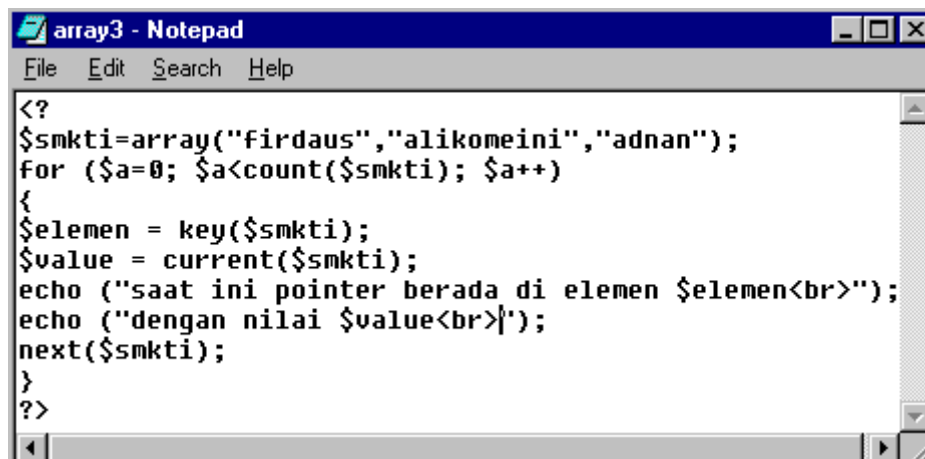
Data dalam suatu array disebut sebagai elemen-elemen array. Posisi elemen array ditunjukkan oleh suatu indeks. Nomor index diawali dengan angka 0 (nol) jika nomor index tidak di deklarasikan secara explicit sedangkan elemen selanjutnya diberi nomor sesuai urutan secara berurut. Data di dalam array akan tersimpan di antara batas atas dan batas bawah. Dari segi pendimensian, array di dalam PHP memiliki kemampuan membuat pendimensian satu, dua atau lebih. Dimensi satu di dalam array PHP mewakili vector, sedangkan dimensi dua array mewakili bentuk matriks atau table, dan dimensi tiga mewakili bentuk ruang. Perhatikan contoh di bawah ini :



```
<HTML>
<HEAD>
<TITLE>Penggunaan Array </TITLE>
</HEAD>
<BODY>
<?
$kota[0] = "Jakarta";
$kota[1]="Bandung";
$kota[2]="Medan";
$kota[3]="Surabaya";
$kota[4]="Jember";
print("Saat ini saya sedang kuliah di kota $kota[1] <BR>\n");
print("Tapi saya ini orang $kota[2] bung?!! <BR>\n");
?>
</BODY>
</HTML>
```

Untuk mengetahui nomor index elemen yang sedang aktif maka digunakan fungsi *key()* dan jika kita ingin mengetahui nilai (value) dari elemen tersebut, maka digunakan fungsi *current()*. Untuk menaikkan posisi pointer, digunakan fungsi *next()* dan jika untuk menurunkannya, gunakan fungsi *prev()*. Untuk meletakkan pointer pada posisi akhir, gunakan fungsi *end()* dan jika ingin mengembalikan posisi pointer ke posisi awal, maka gunakan fungsi *reset()*.

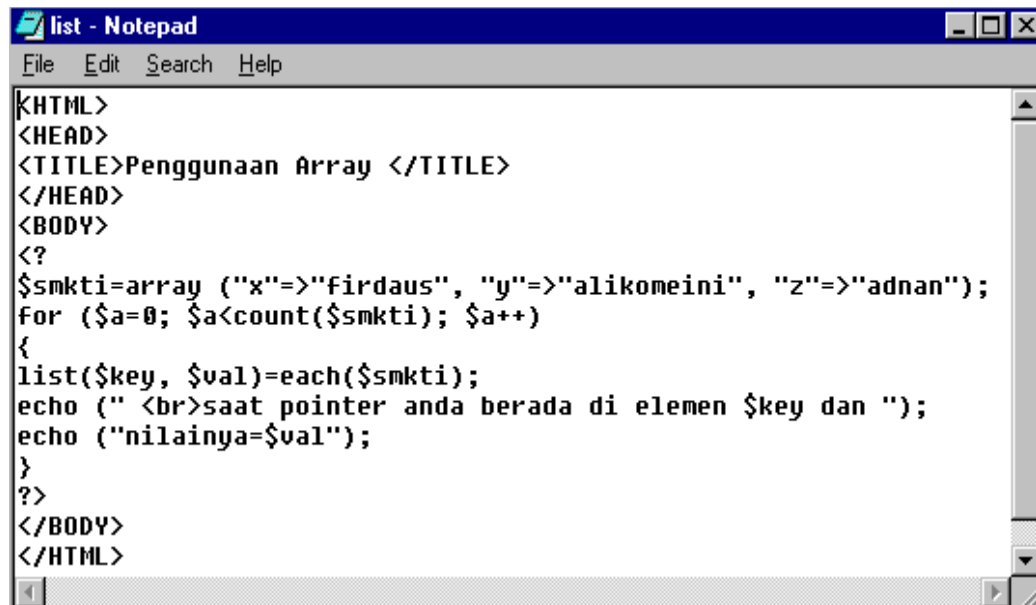
Untuk lebih jelas, perhatikan contoh di bawah ini ;



```
<?
$smkti=array("firdaus","alikota","adnan");
for ($a=0; $a<count($smkti); $a++)
{
$elemen = key($smkti);
$value = current($smkti);
echo ("saat ini pointer berada di elemen $elemen<br>");
echo ("dengan nilai $value<br>");
next($smkti);
}
?>
```

Pada contoh di atas, index element berupa integer. Untuk merunut array dengan index string, maka digunakan fungsi *each()*. Fungsi *each()* mengambil nilai dari array kemudian memajukan pointer satu elemen. Hasilnya adalah 0,1,key

dan val. Di mana angka 0 dan key menunjukkan index elemen, nilai 1 dan val menunjukkan nilai elemen. Untuk lebih jelasnya, perhatikan contoh di bawah ini :

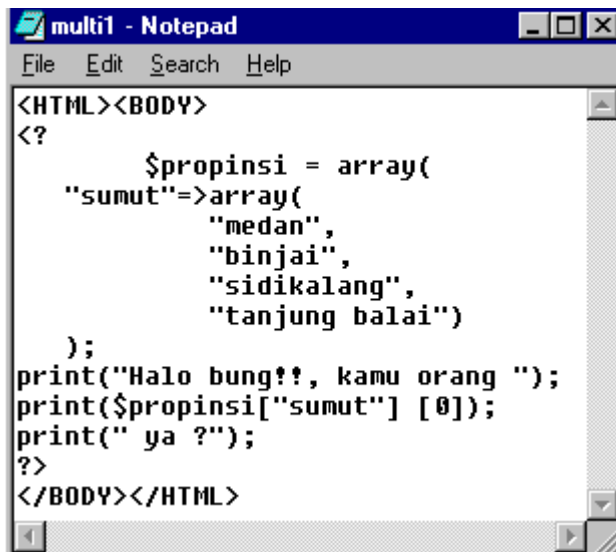


```
<HTML>
<HEAD>
<TITLE>Penggunaan Array </TITLE>
</HEAD>
<BODY>
<?
$smkti=array ("x"=>"firdaus", "y"=>"alikomeini", "z"=>"adnan");
for ($a=0; $a<count($smkti); $a++)
{
list($key, $val)=each($smkti);
echo (" <br>saat pointer anda berada di elemen $key dan ");
echo ("nilainya=$val");
}
?>
</BODY>
</HTML>
```

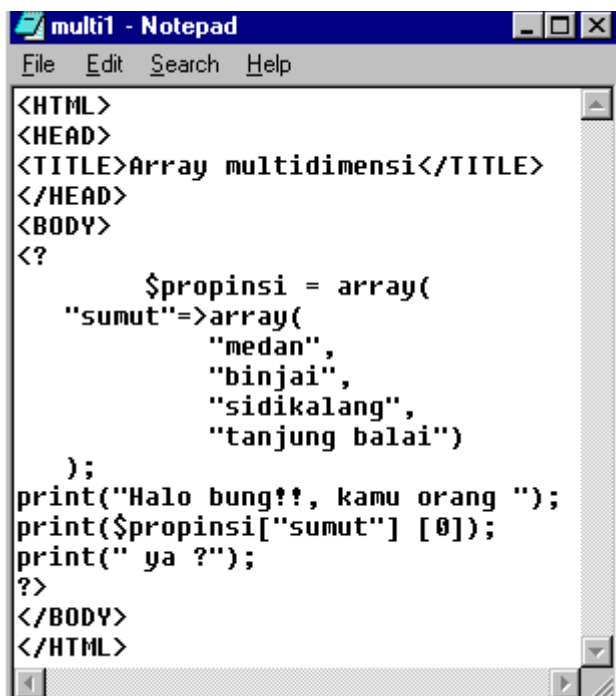
Array Multidimensi

Pada penjelasan di atas, kita telah mempelajari dan membahas fungsi array dengan satu dimensi. Array juga dapat di olah dengan beberapa dimensi seperti 2 dimensi ataupun tiga dimensi, di mana dalam satu pendeklarasian array kita bisa membuat menu dan sub menu di deklarasi tersebut. Maksudnya adalah pada bagian menu, disebut dimensi pertama, sedangkan pada bagian sub menu, disebut dimensi 2.

perhatikan contoh dibawah ini :



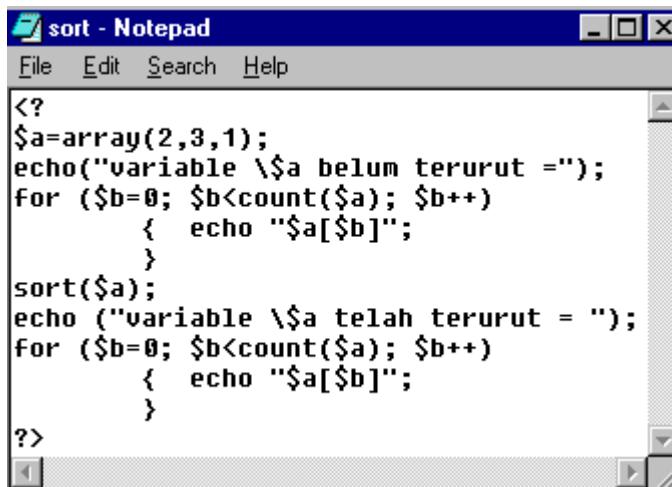
```
multi - Notepad
File Edit Search Help
<HTML><BODY>
<?
    $propinsi = array(
        "sumut"=>array(
            "medan",
            "binjai",
            "sidikalang",
            "tanjung balai")
    );
print("Halo bung??, kamu orang ");
print($propinsi["sumut"] [0]);
print(" ya ?");
?>
</BODY></HTML>
```



```
multi - Notepad
File Edit Search Help
<HTML>
<HEAD>
<TITLE>Array multidimensi</TITLE>
</HEAD>
<BODY>
<?
    $propinsi = array(
        "sumut"=>array(
            "medan",
            "binjai",
            "sidikalang",
            "tanjung balai")
    );
print("Halo bung??, kamu orang ");
print($propinsi["sumut"] [0]);
print(" ya ?");
?>
</BODY>
</HTML>
```

Urutan Elemen

Ada beberapa cara mensorting data dari operasi array, salah satunya dengan cara bubblesort. Cara lainnya adalah dengan menggunakan fungsi sort(). Jika kita menggunakan fungsi sort, contohnya adalah :



```
<?
$a=array(2,3,1);
echo("variable \$a belum terurut =");
for ($b=0; $b<count($a); $b++)
    { echo "$a[$b]";
    }
sort($a);
echo ("variable \$a telah terurut = ");
for ($b=0; $b<count($a); $b++)
    { echo "$a[$b]";
    }
?>
```

Mengambil dan Menggabungkan Elemen

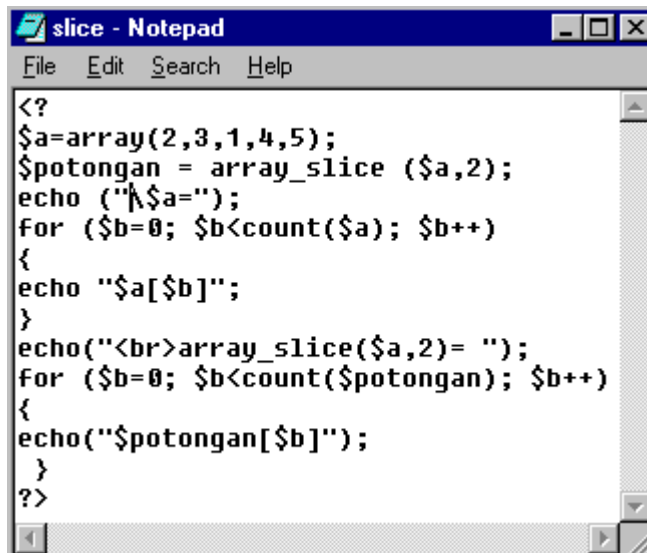
Fungsi `array_slice()` digunakan untuk mengambil potongan elemen dari suatu array yang ditunjukkan oleh parameter offset dan parameter length.

Sintaknya adalah

Array_slice (variable, parameter offset, parameter length)

Parameter offset bernilai positif, jika elemen awal menunjukkan nilai positif, dengankan bernilai negatif jika elemen akhirnya negatif. Sedangkan parameter length bernilai positif jika elemen yang diambil bernilai positif, sedangkan bernilai negatif, jika elemen akhir bernilai negatif. Jika parameter tidak dicantumkan, maka pemotongan dilakukan sampai elemen akhir.

Untuk lebih jelasnya, perhatikan contoh di bawah :

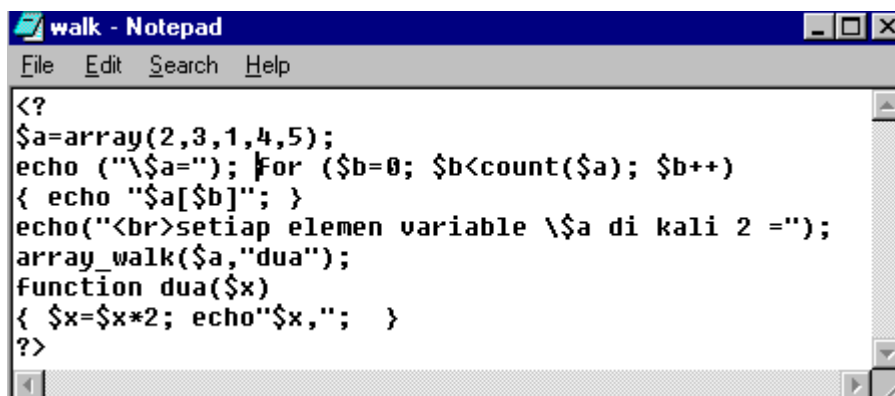


```
<?
$a=array(2,3,1,4,5);
$potongan = array_slice ($a,2);
echo ("\$a=");
for ($b=0; $b<count($a); $b++)
{
echo "$a[$b]";
}
echo("<br>array_slice($a,2)= ");
for ($b=0; $b<count($potongan); $b++)
{
echo("$potongan[$b]");
}
?>
```

Fungsi array_walk

Fungsi ini digunakan untuk melakukan operasi pada elemen bukan array.

Contoh :



```
<?
$a=array(2,3,1,4,5);
echo ("\$a="); for ($b=0; $b<count($a); $b++)
{ echo "$a[$b]"; }
echo("<br>setiap elemen variable \$a di kali 2 =");
array_walk($a,"dua");
function dua($x)
{ $x=$x*2; echo"$x,"; }
?>
```

MEMBUAT HALAMAN DINAMIK

- Halaman Dinamik
- Fungsi Tanggal dan Waktu
- Membuat Tabel Otomatis
- Membuat Tabel Dasar untuk Bulan Berjalan
- Membuat Kalender dengan Nama Hari

Halaman Dinamik

Halaman dinamik adalah halaman web yang dapat berubah secara otomatis tanpa ada intervensi rutin. Contoh halaman dinamik seperti halaman-halaman web yang memuat kurs mata uang, prakiraan cuaca, berita terkini dls. Halaman ini

memiliki sebuah program yang dapat mengambil data yang diperlukan secara online baik dari peralatan ataupun database yang tersedia.

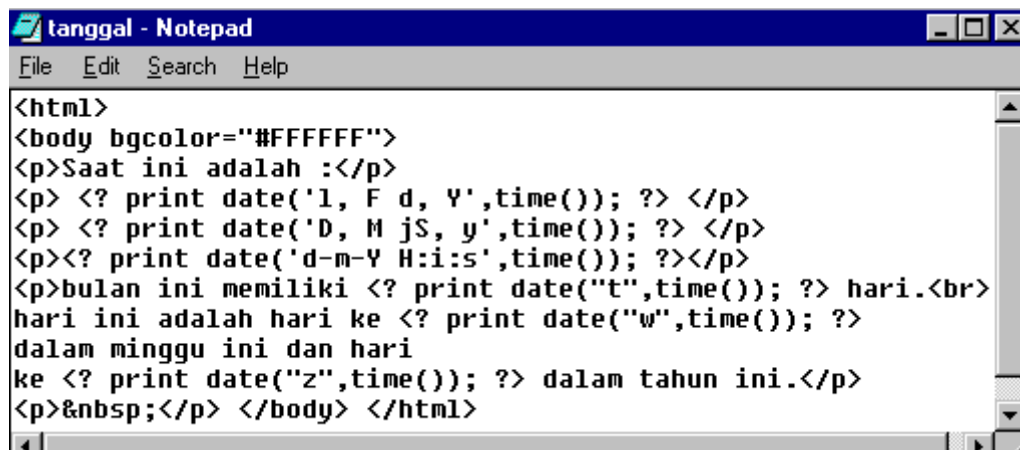
Berikut disajikan beberapa contoh tentang penggunaan fungsi tanggal dan waktu, serta bagaimana membuat tabel secara otomatis menggunakan iterasi. Selanjutnya pengetahuan tersebut digunakan untuk membangun sebuah program untuk menyusun kalender bulan berjalan.

Fungsi Tanggal dan Waktu

Fungsi tanggal adalah salah satu fungsi bawaan PHP untuk melakukan operasi-operasi tanggal dan waktu. Dokumentasi tentang fungsi-fungsi ini dapat dilihat pada manual PHP, dan beberapa di antaranya adalah sbb. :

- **date** - memformat tanggal/waktu setempat.
- **getdate** - mengambil informasi tanggal/waktu.
- **mktime** - membuat UNIX timestamp dari tanggal/waktu yang diberikan.
- **time** - mengambil UNIX timestamp untuk waktu sekarang.

Perhatikan contoh di bawah ini :



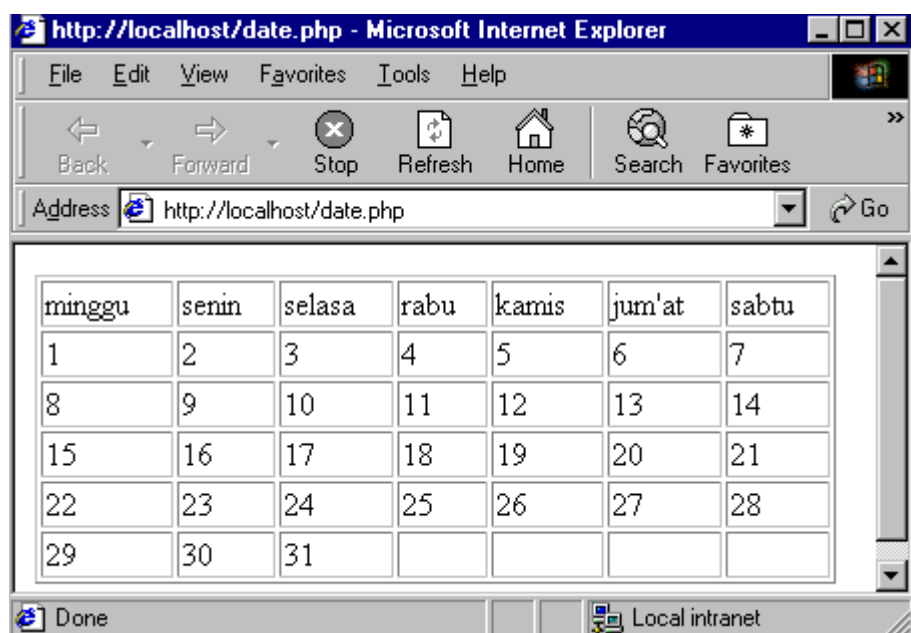
```
<html>
<body bgcolor="#FFFFFF">
<p>Saat ini adalah :</p>
<p> <? print date('l, F d, Y',time()); ?> </p>
<p> <? print date('D, M jS, y',time()); ?> </p>
<p><? print date('d-m-Y H:i:s',time()); ?></p>
<p>bulan ini memiliki <? print date("t",time()); ?> hari.<br>
hari ini adalah hari ke <? print date("w",time()); ?>
dalam minggu ini dan hari
ke <? print date("z",time()); ?> dalam tahun ini.</p>
<p>&nbsp;</p> </body> </html>
```

contoh di atas menggambarkan fungsi-fungsi tersebut.

Membuat Tabel Otomatis

- Buatlah baris-baris berikutnya secara iteratif. Gunakan perintah while untuk baris karena jumlah baris belum diketahui dengan pasti, sedangkan untuk kolom digunakan perintah for karena sudah terdefinisi jumlahnya. Sambil membuat baris dan kolom, tuliskan nilai-nilai tanggal berurutan.
- Lakukan hingga semua tanggal dalam array tertuliskan seluruhnya.

Kalender sederhana ini ditunjukkan gambar di bawah ini :



Script PHP yang membuat gambar di atas adalah :

```

date - Notepad
File Edit Search Help
<html>
<body bgcolor="#FFFFFF">
<table width="400" border="1">
<tr>
<td>minggu</td> <td>senin</td> <td>selasa</td> <td>rabu</td>
<td>kamis</td> <td>jum'at</td> <td>sabtu</td>
</tr>
<?
$jh = date('t',time());
$tgl = array();
for ($i=1; $i<=$jh; $i++) { $tgl[$i] = $i; }
$i = 1;
while ($i < count($tgl)) {
print "<tr>";
for ($j=1; $j<=7; $j++) {
if (!empty($tgl[$i])) {
print "<td>$tgl[$i]</td>";
} else {
print "<td>&nbsp;</td>";
} $i++;
} print "</tr>";
}
?>
</table> </body> </html>

```

Membuat Kalender dengan Nama Hari

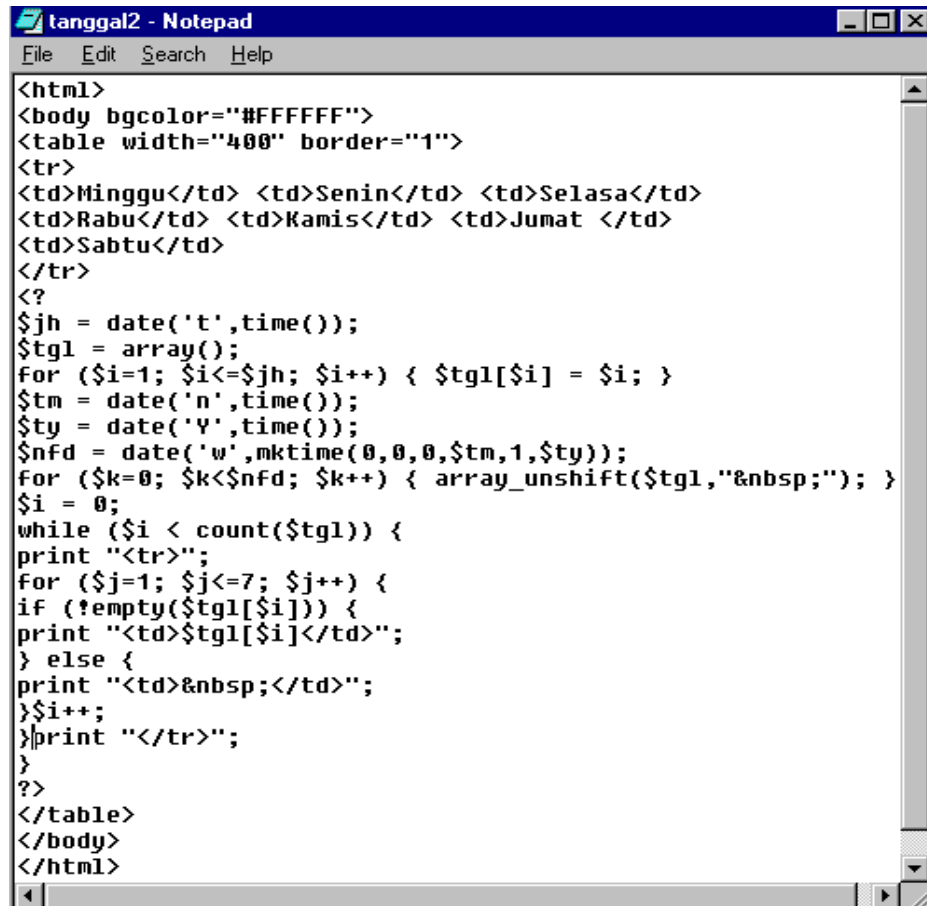
Pada contoh lalu, tanggal pertama bulan berjalan berada di kolom pertama. Hal ini belum tentu benar karena awal bulan tidak selalu hari pertama dalam minggu. Untuk menentukan posisi awal bulan, dapat dilakukan dengan mencari bulan dan tahun saat ini, kemudian menentukan hari untuk awal bulan dengan fungsi date. Lihat gambar di bawah ini :

```

hari2 - Notepad
File Edit Search Help
<html> <body bgcolor="#FFFFFF">
<p>Tanggal Hari ini : <? print date("l, j F Y",time()); ?> </p>
<p>Bulan ini adalah bulan ke : <? print $m=date("n",time()); ?>
- tahun ke : <? print $y=date("Y",time()); ?></p>
<p>Nama hari awal bulan :
<? print date("l",mktime(0,0,0,$m,1,$y));?>
- hari ke : <? print date("w",mktime(0,0,0,$m,1,$y)); ?></p>
</body> </html>

```

Dari pengetahuan posisi hari untuk awal bulan, maka array tanggal bisa disisipi sejumlah offset posisi hari awal bulan terhadap awal minggu. Penyisipan array dilaksanakan melalui perintah *array_unshift*. Lihat contoh di bawah ini :



```
<html>
<body bgcolor="#FFFFFF">
<table width="400" border="1">
<tr>
<td>Minggu</td> <td>Senin</td> <td>Selasa</td>
<td>Rabu</td> <td>Kamis</td> <td>Jumat </td>
<td>Sabtu</td>
</tr>
<?
$jh = date('t',time());
$tgl = array();
for ($i=1; $i<=$jh; $i++) { $tgl[$i] = $i; }
$tm = date('n',time());
$ty = date('Y',time());
$nfd = date('w',mktime(0,0,0,$tm,1,$ty));
for ($k=0; $k<$nfd; $k++) { array_unshift($tgl,"&nbsp;"); }
$i = 0;
while ($i < count($tgl)) {
print "<tr>";
for ($j=1; $j<=7; $j++) {
if (!empty($tgl[$i])) {
print "<td>$tgl[$i]</td>";
} else {
print "<td>&nbsp;</td>";
}$i++;
}print "</tr>";
}
?>
</table>
</body>
</html>
```

Beda antara kalender pertama dengan kalender kedua ini adalah kalender kedua ini mengikuti penanggalan waktu yang ada di komputer sedangkan kalender pertama tidak.

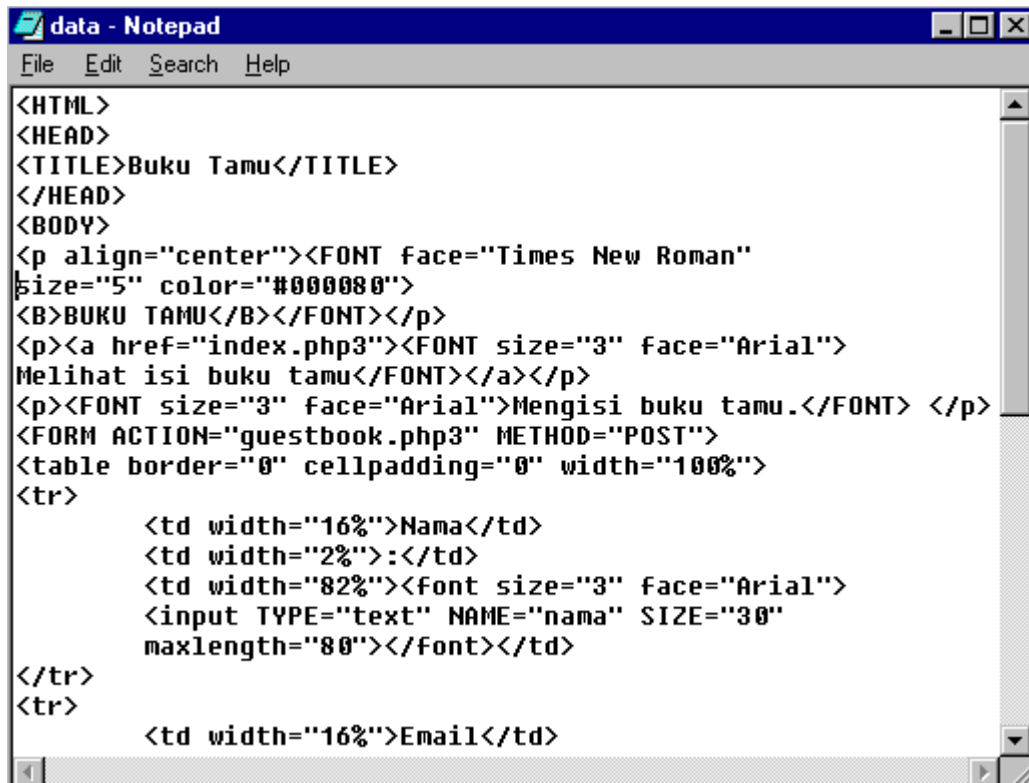
CONTOH APLIKASI PHP

Buku Tamu

Untuk pembuatan aplikasi ini, ada beberapa yang harus diperhatikan sebelum membuat yaitu Dari segi pembuatannya, minimal kita harus membuat 4 file yang isinya tentu berbeda-beda, yaitu file utama, yang berisikan halaman

utama tempat pengisian data (input) yang nantinya kita beri nama *index.html*. Kemudian kita harus membuat tempat script PHP yang menjadi wahana pengiriman variable-variable yang nantinya akan kita gunakan, nama filenya adalah *guestbook.php3*. File untuk membuka hasil masukan data (input) harus disertai pula, nantinya akan kita beri nama *index.php3*. Dan yang terakhir adalah suatu file txt yang menjadi gudang data penyimpanan (arsip) agar data tidak mudah hilang begitu saja, nama filenya *data.txt*. Untuk lebih memahaminya, perhatikan script file-file tersebut :

- a. file utama, *index.html* :



```
<HTML>
<HEAD>
<TITLE>Buku Tamu</TITLE>
</HEAD>
<BODY>
<p align="center"><FONT face="Times New Roman"
size="5" color="#000080">
<B>BUKU TAMU</B></FONT></p>
<p><a href="index.php3"><FONT size="3" face="Arial">
Melihat isi buku tamu</FONT></a></p>
<p><FONT size="3" face="Arial">Mengisi buku tamu.</FONT> </p>
<FORM ACTION="guestbook.php3" METHOD="POST">
<table border="0" cellpadding="0" width="100%">
<tr>
<td width="16%">Nama</td>
<td width="2%">:</td>
<td width="82%"><font size="3" face="Arial">
<input TYPE="text" NAME="nama" SIZE="30"
maxlength="80"></font></td>
</tr>
<tr>
<td width="16%">Email</td>
```

```
data - Notepad
File Edit Search Help

<td width="2%":</td>
<td width="82%"><font size="3" face="Arial">
<input TYPE="text" NAME="email" SIZE="30"
maxlength="80"></font></td>
</tr>
<tr>
<td width="16%" valign="top">Komentar</td>
<td width="2%" valign="top">:</td>
<td width="82%"><font size="3" face="Arial">
<textarea NAME="komentar" ROWS="10" COLS="30">
</textarea></font></td>
</tr>
<tr>
<td width="16%"><font size="2" face="Arial">
<input TYPE="submit" NAME="gb"
VALUE="OKE! Kirim"></font></td>
<td width="2%"></td>
<td width="82%"></td>
</tr>
</table>
</FORM>
</BODY>
</HTML>
```

b. file guestbook.php3

```
variable - Notepad
File Edit Search Help

<HTML>
<BODY>
<?
    $formulir = "index.html";
    function isEmpty($nama)
    {
        return !empty($nama);
    }
    function isValid($email)
    {
        return !empty($email);
    }
    function isGoodComment($komentar)
    {
        return !empty($komentar);
    }
    if ( isEmpty($nama) AND isValid($email)
        AND isGoodComment($komentar))
    {
        $nama1 =htmlentities($nama);
        $email1 =htmlentities($email);
        $komentar1=htmlentities($komentar);
```

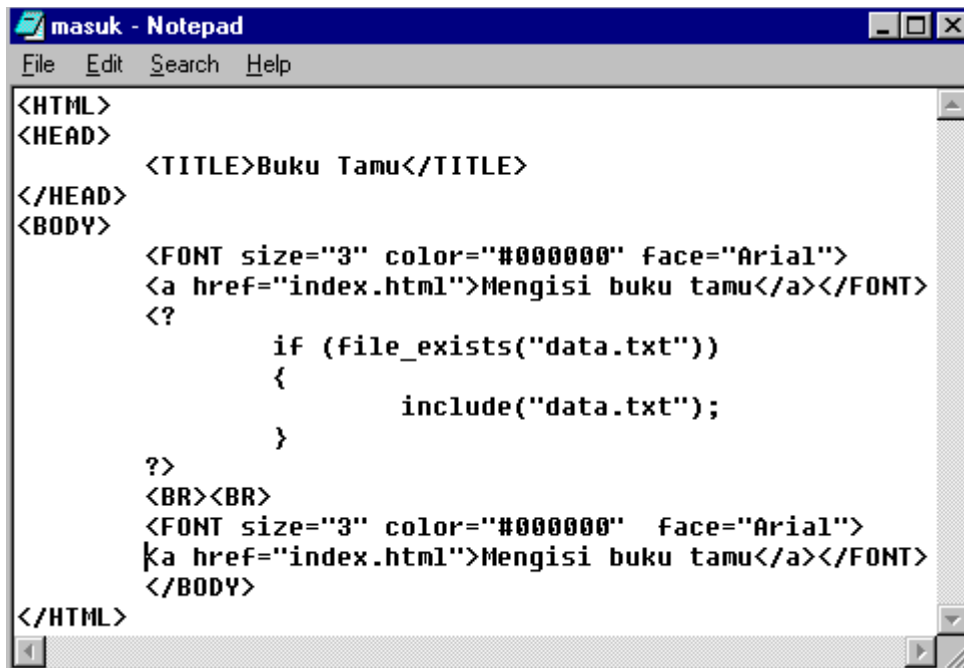
```
variable - Notepad
File Edit Search Help

$komentar1 = ereg_replace("\n","<BR>",$komentar1);
$data = "data.txt";
$stampilan = "index.php3";
$page = $data;
$filename = "data.txt";
$fd = fopen( $filename, "r" );
$current = fread( $fd, filesize( $filename ) );
fclose( $fd );
$fileMessage = "<P><b><font size='2' color='#000000'
face='Arial'>Nama: </b></font><font size='2'
color='#000000' face='Arial'>$nama1</font>\n";
$fileMessage .= "<br><font size='2' color='#000000'
face='Arial'><b>Tanggal: </b></font>";
$fileMessage .= ("<font size='2' color='#000000'
face='Arial'>\n");
$fileMessage .= (date("dS of F Y h:i:s A"));
$fileMessage .= ("</font>\n");
$fileMessage .= "</font><font size='2' color='#000000'
face='Arial'><br><b>E-mail:</b></font>\n";
$fileMessage .= "<font size='2' color='#000000'
face='Arial'>$email1</font>\n";
$fileMessage .= "<br><font size='2' color='#000000'
face='Arial'><b>Komentar:</b></font>\n";
$fileMessage .= "<br><font size='2' color='#000000'
face='Arial'>$komentar1</font>\n";
$fileMessage .= "$current\n";
if (file_exists("$page"))
{ $gabung = fopen("$page","w+");
fputs($gabung,$fileMessage);
fclose($gabung);
} else {
$gabung = fopen("$page","w");
fputs($gabung,$fileMessage);
fclose($gabung);
} include($stampilan);
} else {
$salah = "<font size='3' color='red' face='Arial'>
<STRONG>Ada kesalahan pengisian form</STRONG>
</font><br>";
echo $salah;
include($formulir);
} ?> </BODY></HTML>
```

c. File data.txt

Merupakan file kosong dalam extension txt, dimana seluruh input data akan tercatat di file tersebut.

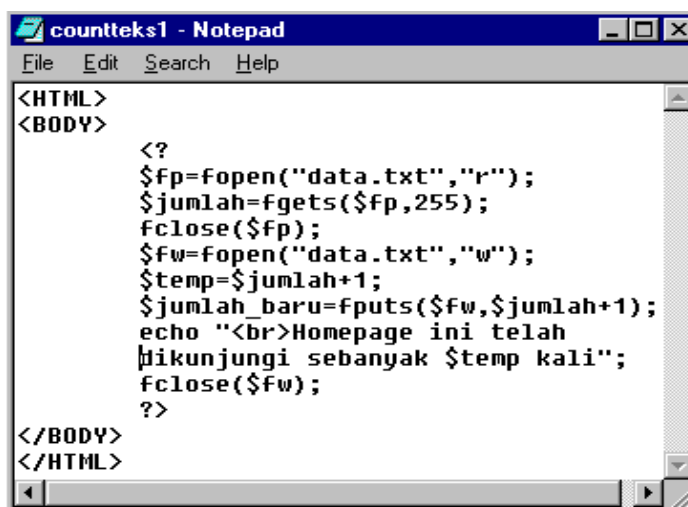
d. file index.php3



```
<HTML>
<HEAD>
  <TITLE>Buku Tamu</TITLE>
</HEAD>
<BODY>
  <FONT size="3" color="#000000" face="Arial">
  <a href="index.html">Mengisi buku tamu</a></FONT>
  <?
    if (file_exists("data.txt"))
    {
      include("data.txt");
    }
  ?>
  <BR><BR>
  <FONT size="3" color="#000000" face="Arial">
  <a href="index.html">Mengisi buku tamu</a></FONT>
</BODY>
</HTML>
```

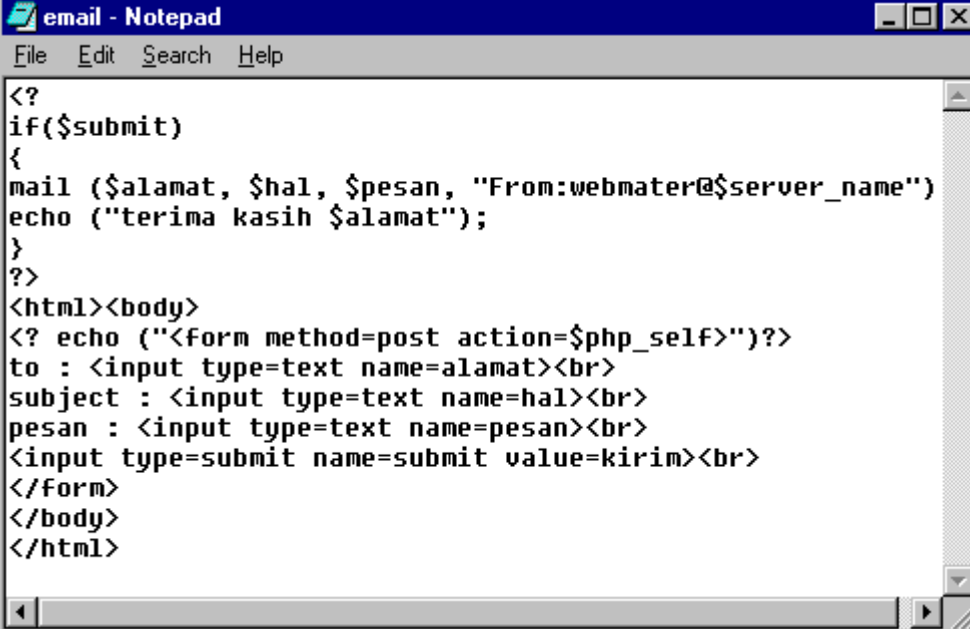
Counter

Counter digunakan dalam PHP untuk menghitung jumlah pengunjung yang telah mengakses website kita. Counter ada beberapa jenis, yaitu dengan text dan gambar. Dalam modul ini, kita akan mempelajari counter dengan text. Selain file dalam extension php, kita juga harus buat suatu file txt, dalam hal ini namanya data.txt. Perhatikan contoh di bawah ini :



```
<HTML>
<BODY>
  <?
    $fp=fopen("data.txt","r");
    $jumlah=fgets($fp,255);
    fclose($fp);
    $fw=fopen("data.txt","w");
    $temp=$jumlah+1;
    $jumlah_baru=fputs($fw,$jumlah+1);
    echo "<br>Homepage ini telah
    dikunjungi sebanyak $temp kali";
    fclose($fw);
  ?>
</BODY>
</HTML>
```

Aplikasi Pengiriman Email



```
<?
if($submit)
{
mail ($alamat, $hal, $pesan, "From:webmater@$server_name")
echo ("terima kasih $alamat");
}
?>
<html><body>
<? echo ("<form method=post action=$php_self>")?>
to : <input type=text name=alamat><br>
subject : <input type=text name=hal><br>
pesan : <input type=text name=pesan><br>
<input type=submit name=submit value=kirim><br>
</form>
</body>
</html>
```

sintak [webmaster@server_name](#) bisa diganti dengan alamat email anda sendiri tidak mesti seperti di atas.

PHP DAN KONEKSI DATABASE MYSQL

- Mengirimkan input suatu form
- Sintak Dasar Konektivitas PHP ke MySQL

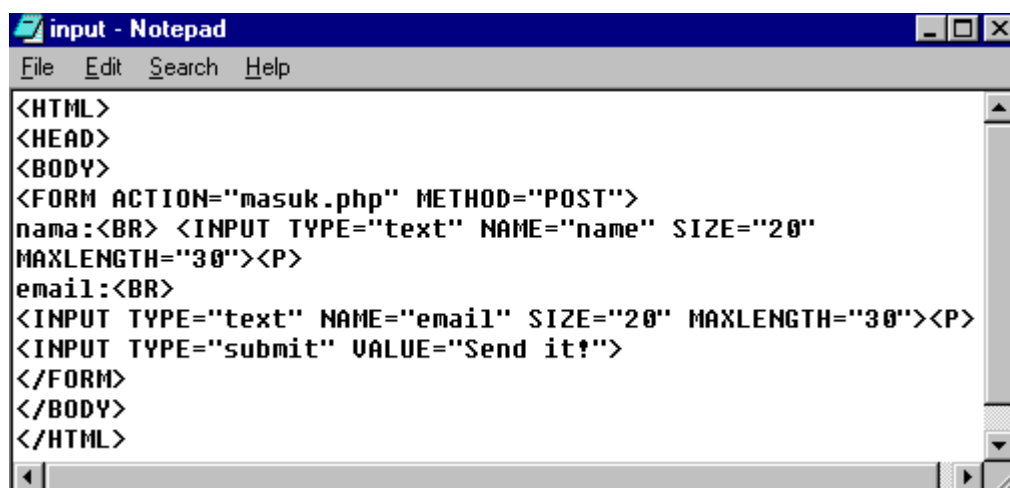
Seperti yang telah diungkapkan di atas, kekuatan utama php sebenarnya adalah pada kemudahan akses database yang dibentuknya. Dan untuk modul ini, kita akan memfokuskan isi modul ini pada pengaksesan database dengan server MySQL .

Data akan diambil pada form yang akan kita letakkan pada halaman input.html. Selanjutnya, bila data yang diisi telah terkirim, maka akan dipanggil halaman proses.phtml yang akan menjalankan proses php untuk mengolah data tersebut, baik itu menyimpan maupun mengambil data dari dan ke database.

Mengirimkan input suatu form

Dari pengetahuan dasar yng kita miliki tentang script html, kita dapat membuat form yang akan kita gunakan untuk mengambil input dari user.

praktikkan contoh berikut !



```
<HTML>
<HEAD>
<BODY>
<FORM ACTION="masuk.php" METHOD="POST">
nama:<BR> <INPUT TYPE="text" NAME="name" SIZE="20"
MAXLENGTH="30"><P>
email:<BR>
<INPUT TYPE="text" NAME="email" SIZE="20" MAXLENGTH="30"><P>
<INPUT TYPE="submit" VALUE="Send it!">
</FORM>
</BODY>
</HTML>
```

Hal baru yang perlu diperhatikan di sini adalah setelah tag ACTION, maka kita harus memasukkan nama file tempat di mana input yang kita peroleh akan diproses, dalam hal ini yaitu file masuk.php.

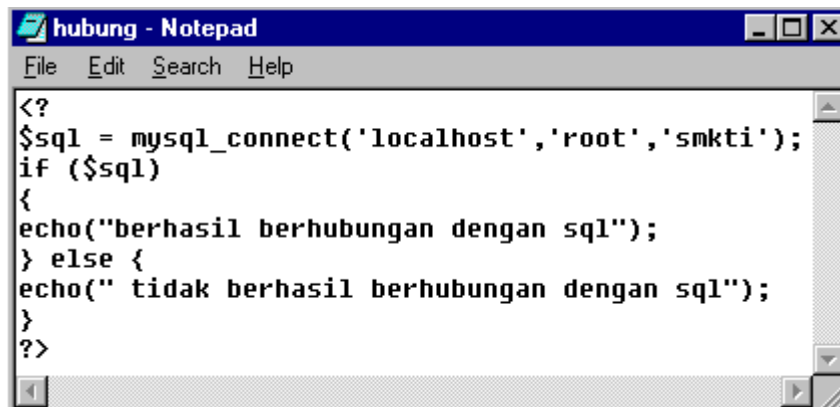
Sintak Dasar Konektivitas PHP ke MySQL

Membuka Koneksi Ke Database

`mysql_connect()` adalah fungsi yang memungkinkan kita untuk masuk ke suatu sistem database. syntaxnya;

```
int mysql_connect(string [hostname [:port] [:/path/to/socket] ] , string [username] , string [password] );
```

contoh :



```
<?
$sql = mysql_connect('localhost','root','smkti');
if ($sql)
{
echo("berhasil berhubungan dengan sql");
} else {
echo(" tidak berhasil berhubungan dengan sql");
}
?>
```

Semua argumen di dalamnya adalah bersifat optional dan bila tidak diisi maka akan memberi default ('localhost', nama user owner, password kosong).

Satu hal yang perlu diperhatikan bahwa untuk berhasil berhubungan dengan MySQL, maka program database MySQL-nya harus terlebih dahulu dijalankan. Cara menjalankannya bisa anda lihat pada modul MySQL.

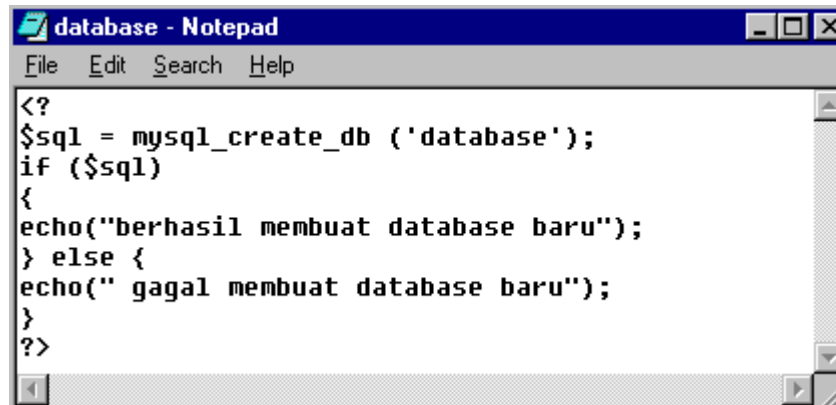
Membuat Database Baru

Untuk membuat suatu database di MySQL, ada beberapa cara, di antaranya adalah membuatnya secara manual di program MySQL atau membuatnya melalui software PHPMYADMIN yang pada modul terdahulu telah kita bahas. Cara lainnya adalah menulis sintak PHP di text editor dengan script khusus sehingga dengan menulis sintak tersebut maka database yang kita inginkan akan terbentuk dengan sendirinya.

Sintak tersebut adalah :

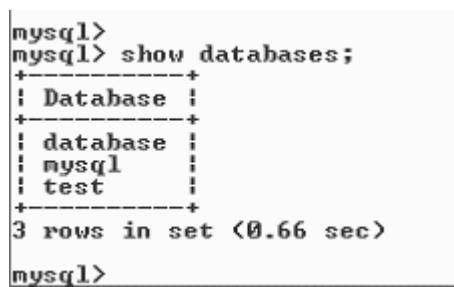
int mysql_create_db(string database name, int [link_identifier]);

contohnya :



```
<?
$sql = mysql_create_db ('database');
if ($sql)
{
echo("berhasil membuat database baru");
} else {
echo(" gagal membuat database baru");
}
?>
```

Jika keluar pernyataan berhasil, coba di cek kembali di MySQL, apakah telah terbentuk suatu database dengan nama *database*. Perhatikan gambar di bawah ini :



```
mysql>
mysql> show databases;
+-----+
| Database |
+-----+
| database |
| mysql    |
| test     |
+-----+
3 rows in set (0.66 sec)

mysql>
```

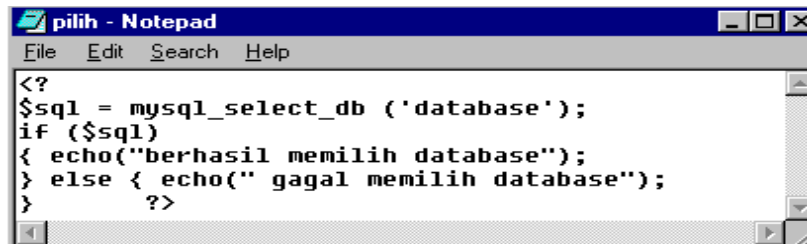
Sintak `mysql_create("nama_database")` sama halnya dengan sintak manual pada MySQL yang ditulis secara manual yaitu `create nama_database;` . Tampak di gambar atas ada sebuah nama database yaitu *database*, artinya kita telah berhasil membuat suatu nama database tanpa harus mengetik sintaknya secara manual seperti pada modul terdahulu.

Memilih Database Yang akan Digunakan

`mysql_select_db()` adalah fungsi yang digunakan untuk membuka sebuah database yang akan digunakan. Syntaxnya;

int mysql_select_db(string database_name, int [link_identifier]);

contohnya :



```
<?
$sql = mysql_select_db ('database');
if ($sql)
{ echo("berhasil memilih database");
} else { echo(" gagal memilih database");
}
?>
```

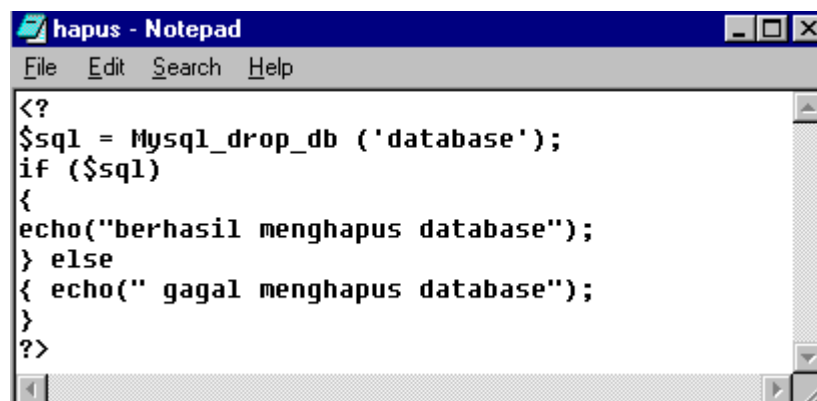
Sintak `mysql_select_db(nama_database)` sama halnya dengan sintak dasar pada MySQL yaitu `use nama_database;`

Menghapus Database

`mysql_drop_db` adalah fungsi yang digunakan untuk mengeluarkan data dari database. Sintaknya ;

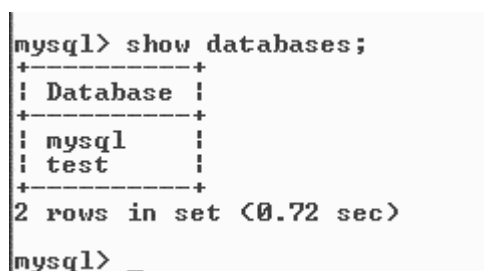
Int mysql_drop_db (string database_name, int [link_identifier]);

Contohnya :



```
<?
$sql = Mysql_drop_db ('database');
if ($sql)
{
echo("berhasil menghapus database");
} else
{ echo(" gagal menghapus database");
}
?>
```

Kemudian cek kembali MySQL dari DOS seperti gambar di bawah ini :



```
mysql> show databases;
+-----+
| Database |
+-----+
| mysql    |
| test     |
+-----+
2 rows in set (0.72 sec)

mysql>
```

Tampak di atas, database dengan nama *database* telah hilang. Sintak tersebut sama halnya dengan sintak di MySQL yaitu `drop nama_database;`

Memasukkan Data Ke Database

Untuk memasukkan data ke database, perhatikan contoh di bawah ini :



```
<html>
<body>
<?
if ($submit)//$submit adalah variable untuk tombol enter di bawah
{
    /*
proses form
*/
    $db = mysql_connect("localhost", "root","firdaus");
    mysql_select_db("data",$db);
    $sql = "insert into data (nama,alamat,pekerjaan,telepon) values
(' $nama', '$alamat', '$pekerjaan', '$telepon')";
    /*
nama-nama field di atas sebelumnya sudah ada di database 'data'
*/
    $result = mysql_query($sql);
    echo "terimakasih, data anda telah masuk ke database\n";
}
else
{
    /* menampilkan tampilan form di webpage
dalam contoh ini action yang dibuat adalah PHP_SELF;
berarti setelah data dimasukkan kedalam form di webpage
maka halaman tersebut akan me loop halaman itu juga
dengan membaca pernyataan yang telah kita buat seperti di atas
*/
    ?>
    <form method="post" action="<?php echo $PHP_SELF?>"
        nama :<input type="text" name="nama"><br>
        alamat      :<input type="text" name="alamat"><br>
        pekerjaan   :<input type="text" name="pekerjaan"><br>
        telepon     :<input type="text" name="telepon"><br>
        <input type="Submit" name="submit" value="enter">
    </form>
    <?php
}
// akhir form
?>
</body>
</html>
```

Untuk diperhatikan, bahwa sebelum kita menjalankannya di localhost, database beserta kolom-kolom yang diperlukan harus sudah ada terlebih dahulu berada di system database MySQL sehingga bila file ini dijalankan dan input dimasukkan, maka data yang masuk akan langsung terkirim ke MySQL, jika databasenya belum ada atau belum lengkap maka input tersebut akan hilang begitu saja. Untuk lebih lengkap lagi perhatikan contoh di bawah ini :

```
masuk - Notepad
File Edit Search Help

<html>
<body>
<?php
$db = mysql_connect("localhost", "root", "firdaus");
mysql_select_db("data",$db);
if ($submit)
{
    if ($nama)
    {
        $sql = "insert into data (nama,alamat,pekerjaan,telepon)
        values('$nama','$alamat','$pekerjaan','$telepon)";
        $result = mysql_query($sql);
        echo "telah masuk database<p>";
    } elseif ($delete)
    {
        $sql = "delete from data where nama='$nama'";
        $result = mysql_query($sql);
        echo "telah terdelete!<p>";
    } else
    {
        if (!$nama)
        {
            $result = mysql_query("select*from data",$db);
        }
    }
}
?>
<P> <a href="<?php echo $PHP_SELF?>">tambah data</a> <P>
<form method="post" action="<?php echo $PHP_SELF?>"
nama :<input type="Text" name="nama"
value="<?php echo $nama ?>"><br>
alamat :<input type="Text" name="alamat"
value="<?php echo $alamat ?>"><br>
pekerjaan:<input type="Text" name="pekerjaan"
value="<?php echo $pekerjaan ?>"><br>
telepon :<input type="Text" name="telepon"
value="<?php echo $telepon ?>"><br>
<input type="Submit" name="submit" value="Enter">
<input type="Submit" name="delete" value="delete">
</form>
<?php
}
?>
</body>
</html>
```

EDITING DATA

- Memasukkan perintah ke dalam database
- Menampilkan data dari database
- Editing Data
- Menutup Koneksi ke database

Memasukkan Perintah Ke Dalam Database

`mysql_query()` adalah fungsi yang mengirimkan perintah SDL ke database yang kita masuki, sebagaimana layaknya bila kita memasukkan perintah MySQL langsung dari mesin server.

a. `int mysql_query(string query, int [link_identifier]);`

`mysql_db_query` adalah hampir sama halnya dengan `mysql_query` yaitu yang berfungsi mengirimkan pernyataan SQL ke server data base MYSQL selama nama database masih aktif.

b. `int mysql_db_query(string query, int [link_identifier]);`

Menampilkan data dari database

Untuk hasil data yang cukup besar adakalanya kita harus melakukan operasi tambahan data yang diambil agar dapat diolah dengan efektif dan efisien. Antara lain:

Setelah data dimasukkan melalui contoh-contoh pada modul 14, dan data telah dipastikan masuk kedalam system database MySQL, maka untuk menampilkan hasil masukan (input) di tampilan browser maka sintaknya adalah :

a. `array mysql_fetch_array(int result, int [result_type]);`

`mysql_fetch_array()` adalah fungsi yang memberikan hasil array dari data yang kita contohnya :

`array mysql_fetch_array ($result);`

b. array mysql_fetch_row(int result, int [result_type]);

Pemakaian sintak ini hampir sama dengan sintak *array mysql_fetch_array (int result, int [result_type]);*

Dalam modul ini, kita akan menggunakan sintak *array mysql_fetch_array (int result, int [result_type]);* sebagai acuan selanjutnya.

c. int mysql_num_fields(int result);

mysql_num_fields() adalah fungsi yang memberikan jumlah field dari data yang diambil. Contohnya :

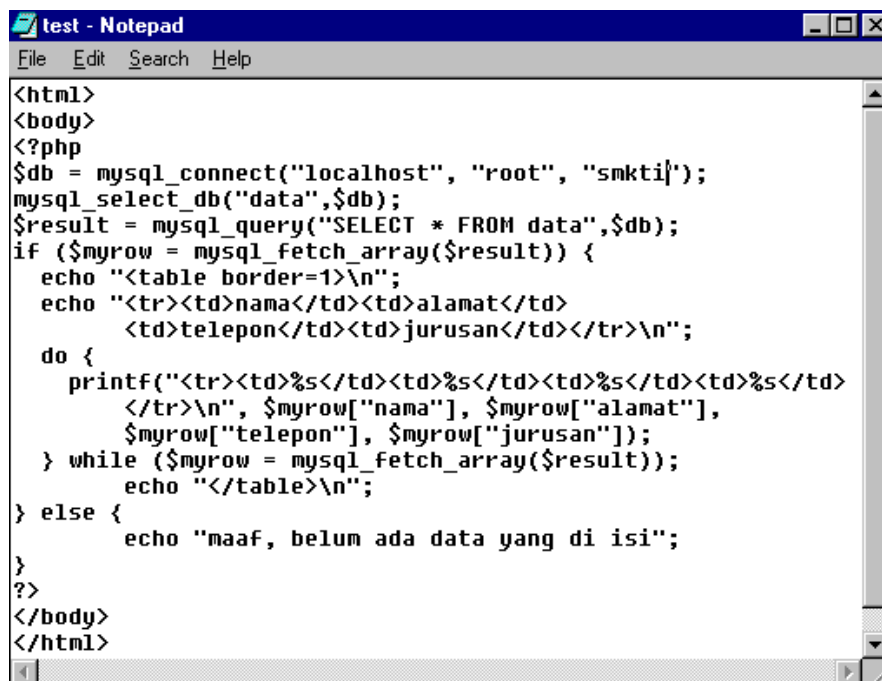
Mysql_num_field (\$result);

d. int mysql_num_row(int result);

mysql_num_row() adalah fungsi yang memberikan jumlah baris dari data yang diambil. Contohnya :

Mysql_num_row (\$result);

Misalkan nama database yang telah di buat adalah *data* dan nama tabelnya adalah *data* beserta beberapa kolom sebagai tempat pengisian data yaitu *nama*, *alamat*, *telepon* dan *jurusan*. Untuk lebih jelasnya perhatikan contoh di bawah ini



```
<html>
<body>
<?php
$db = mysql_connect("localhost", "root", "smkti");
mysql_select_db("data",$db);
$result = mysql_query("SELECT * FROM data",$db);
if ($myrow = mysql_fetch_array($result)) {
    echo "<table border=1>\n";
    echo "<tr><td>nama</td><td>alamat</td>
        <td>telepon</td><td>jurusan</td></tr>\n";
    do {
        printf("<tr><td>%s</td><td>%s</td><td>%s</td><td>%s</td>
            </tr>\n", $myrow["nama"], $myrow["alamat"],
                $myrow["telepon"], $myrow["jurusan"]);
    } while ($myrow = mysql_fetch_array($result));
    echo "</table>\n";
} else {
    echo "maaf, belum ada data yang di isi";
}
?>
</body>
</html>
```

Perhatikan gambar di atas, tampak bahwa sintak *array mysql_fetch_array (\$result)* sebagai pemacu keluarnya data dari MySQL sehingga dapat tampil di

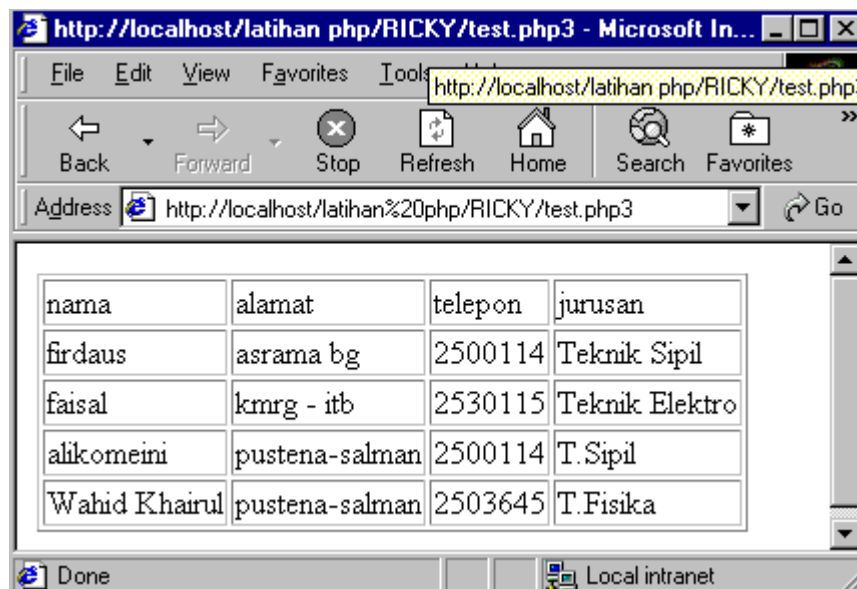
browser. Di dalam sintak tersebut ada tag ‘%s’, sitnak tersebut menjadi variable dari setiap input yang telah masuk ke database, dan untuk menampilkannya, variable tersebut harus diletakkan agar input yang telah masuk tadi, dapat keluar secara otomatis dan sesuai dengan kolom yang telah dibuat sebelumnya.

Jika kita telah memasukan data sebanyak 4 data, jika di MySQL telah tampil seperti

```
mysql>
mysql> use data;
Database changed
mysql> select*from data;
+-----+-----+-----+-----+
| nama      | alamat      | telepon | jurusan  |
+-----+-----+-----+-----+
| firdaus   | asrama bg   | 2500114 | Teknik Sipil |
| faisal    | kmrg - itb  | 2530115 | Teknik Elektro |
| alikomeini | pustena-salman | 2500114 | T.Sipil |
| Wahid Khairul | pustena-salman | 2503645 | T.Fisika |
+-----+-----+-----+-----+
4 rows in set (0.17 sec)

mysql>
```

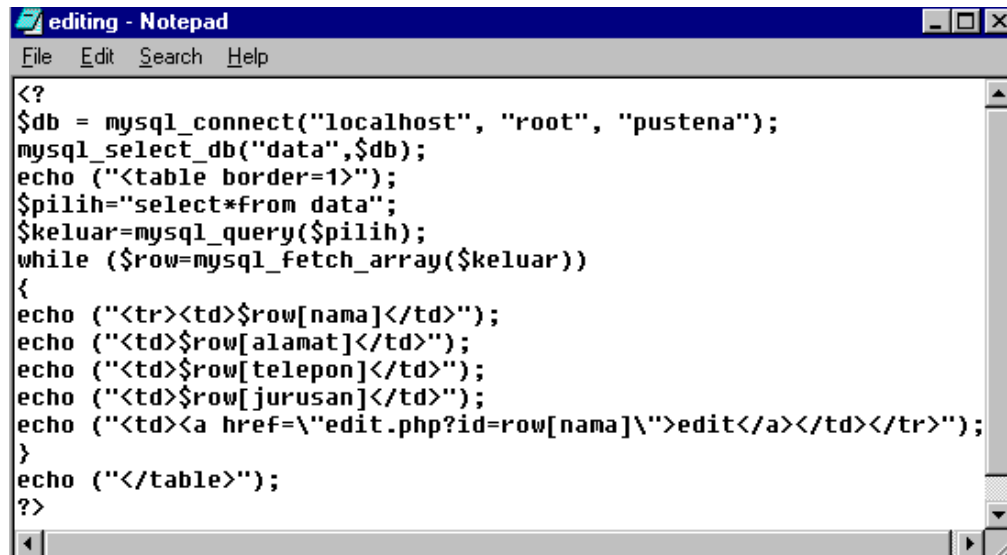
Maka tampilan di browsernya akan menjadi :



Editing Data

Setelah browser berhasil menampilkan data dari MySQL seperti gambar di atas, jika kita ingin merubah data di atas, tanpa merubahnya langsung dari

MySQL, PHP telah menyediakan sintak khusus untuk mengeditnya langsung di browser. Perhatikan contoh dibawah ini :

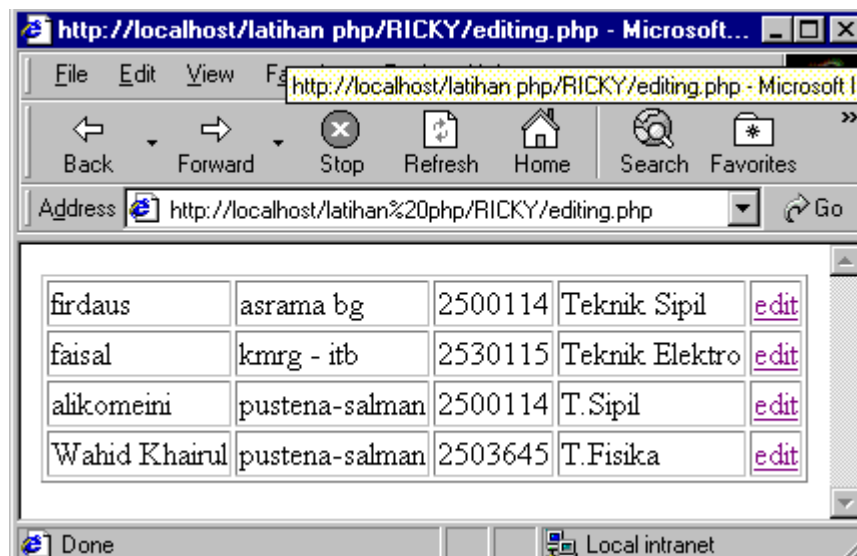


```

<?
$db = mysql_connect("localhost", "root", "pustena");
mysql_select_db("data",$db);
echo ("<table border=1>");
$pilih="select*from data";
$keluar=mysql_query($pilih);
while ($row=mysql_fetch_array($keluar))
{
echo ("<tr><td>$row[nama]</td>");
echo ("<td>$row[alamat]</td>");
echo ("<td>$row[telepon]</td>");
echo ("<td>$row[jurusan]</td>");
echo ("<td><a href=\"edit.php?id=row[nama]\">edit</a></td></tr>");
}
echo ("</table>");
?>

```

Password di atas bias anda ganti dengan smkti, yang jelas password-nya sesuai dengan password ketika menginstal MySQL. Setelah menulis sintak tersebut di text editor, maka di browser akan tampil seperti di bawah ini :



Perhatikan sintak di atas, pada bagian akhir sintak, ada bagian yang mengalamatkan ke file lainnya yaitu *edit.php*. Sintak tersebut adalah :
`echo ("<td>edit</td></tr>");`

Sintak tersebut bermaksud agar ketika kita mengklik kata *edit* yang ada di dalam table, maka browser akan menampilkan suatu tampilan dengan nama *edit.php* yang digunakan untuk mengedit data yang kita inginkan.

Isi file *edit.php* tersebut adalah :

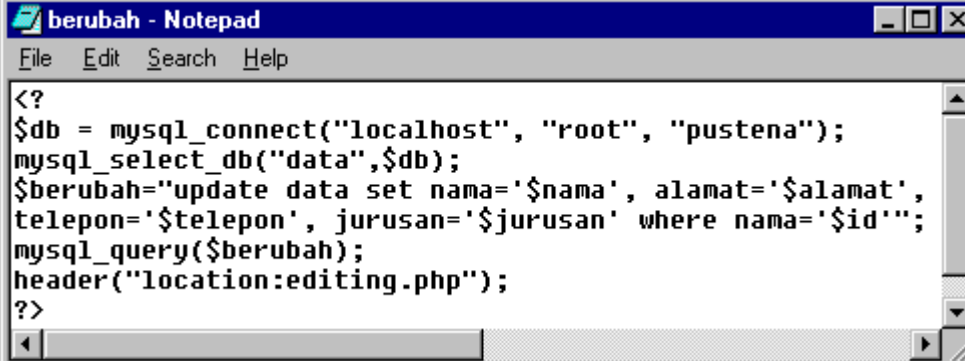


A screenshot of a web browser window. The address bar shows "http://localhost/latiha". The main content area displays a form with four text input fields and one submit button. The fields are labeled "nama", "alamat", "telepon", and "jurusan". The values entered are "firdaus", "asrama bg", "2500114", and "Teknik Sipil" respectively. Below the fields is a button labeled "edit".

Tampilan di atas berasal dari script PHP yang ada di bawah ini :

```
edit - Notepad
File Edit Search Help
<html>
<body>
<table border=1>
<?
$db = mysql_connect("localhost", "root", "pustena");
mysql_select_db("data",$db);
$pilih="select*from data where nama='$id'";
$keluar=mysql_query($pilih);
$row=mysql_fetch_array($keluar);
?>
<form method=post action=berubah.php>
<tr>
<input type=hidden name="id" value
="<?echo ("{$row[nama]}) ?>">
<td>
nama</td><td><input type=text name=nama
value="<? echo ("{$row[nama]}) ?>"></td></tr>
alamat</td><td><input type=text name=alamat
value="<? echo ("{$row[alamat]}) ?>"></td></tr>
telepon</td><td><input type=text name=telepon
value="<? echo ("{$row[telepon]}) ?>"></td></tr>
jurusan</td><td><input type=text name=jurusan
value="<? echo ("{$row[jurusan]}) ?>"></td></tr>
<tr>
<td><input type=submit name=submit value=edit></td>
</tr>
</table>
</body>
</html>
```

Setelah kita mengedit data yang kita inginkan dari contoh di atas, maka file yang mengolah data setelah input di masukkan adalah file *berubah.php* yang ada di bawah ini :



```
<?
$db = mysql_connect("localhost", "root", "pustena");
mysql_select_db("data",$db);
$berubah="update data set nama='$nama', alamat='$alamat',
telepon='$telepon', jurusan='$jurusan' where nama='$id'";
mysql_query($berubah);
header("location:editing.php");
?>
```

Perhatikan pada gambar di atas, tampak pada baris terakhir ada sintak *header("location:editing.php");*

Sintak tersebut menyatakan bahwa setelah kita mengedit data dan menekan button *edit* pada file *edit.php*, seperti pada gambar di atas, walaupun action-nya ke file *berubah.php* seperti yang ada di atas, tetapi dengan adanya sintak di atas, maka browser akan mengarahkan file tersebut ke file *editing.php*. Hal ini di sebabkan file *berubah.php* hanya berisi script murni PHP tanpa adanya sintak atau pernyataan kalau data yang kita edit sebelumnya telah kita rubah, sehingga pengalamanan action dari *berubah.php* yang ada pada file *edit.php* berubah langsung kembali ke file utama yaitu *editing.php*.

Menutup Koneksi ke Database

Untuk mengakhiri koneksi ke database dapat dipakai syntax;

```
int mysql_close(int [link_idenfier] );
```