

PHP

Apakah PHP Itu?

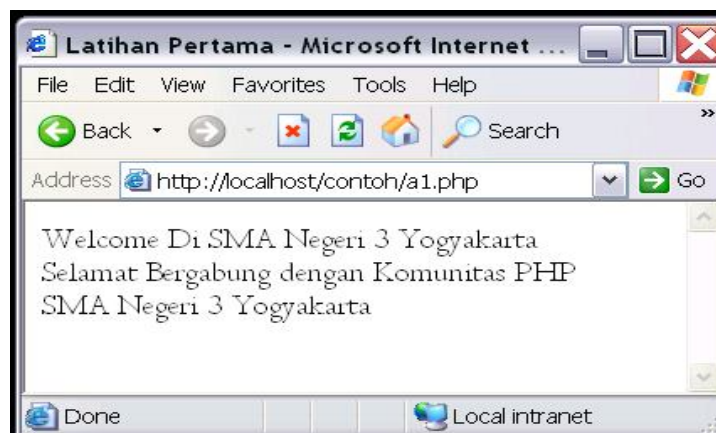
- Merupakan bahasa berbentuk skrip yang ditempatkan dalam server dan diproses di server.
- Hasilnyalah yang dikirimkan ke klien, tempat pemakai menggunakan *browser*.
- Secara khusus, PHP dirancang untuk membentuk web dinamis. Artinya, ia dapat membentuk suatu tampilan berdasarkan permintaan terkini. Misalnya, Anda bisa menampilkan isi *database* ke halaman Web.
- Pada prinsipnya, PHP mempunyai fungsi yang sama dengan skrip-skrip seperti ASP (*Active Server Page*), Cold Fusion, ataupun Perl.
- Dapat diterapkan pada *web server* seperti Apache, PWS (*Personal Web Server*), IIS (*Internet Information Server*), dan Xitami.

Skrip PHP

- Skrip PHP berkedudukan sebagai tag dalam bahasa HTML. Sebagaimana diketahui, HTML (*HyperText Markup Language*) adalah bahasa standar untuk membuat halaman-halaman Web. Sebagai contoh, berikut adalah kode HTML (disimpan dengan ekstensi .htm atau .html):

```
<HTML>
<HEAD>
<TITLE>Latihan Pertama</TITLE>
</HEAD>
<BODY>
  Welcome Di SMA Negeri 3 Yogyakarta.<BR>
  Selamat bergabung dengan komunitas PHP <BR>
  SMA Negeri 3 Yogyakarta
</BODY>
</HTML>
```

- Bila dijalankan melalui *browser*, kode tersebut membentuk tampilan seperti berikut:



- Adapun kode berikut adalah contoh kode PHP yang berada di dalam kode HTML:

	Skrip : selamat.php
---	----------------------------

```

<HTML>
<HEAD>
<TITLE>Latihan Pertama</TITLE>
</HEAD>
<BODY>

    Selamat Belajar PHP.<BR>

    <?php
        printf("Tgl. Sekarang: %s ", Date("d F Y"));
    ?>

</BODY>
</HTML>

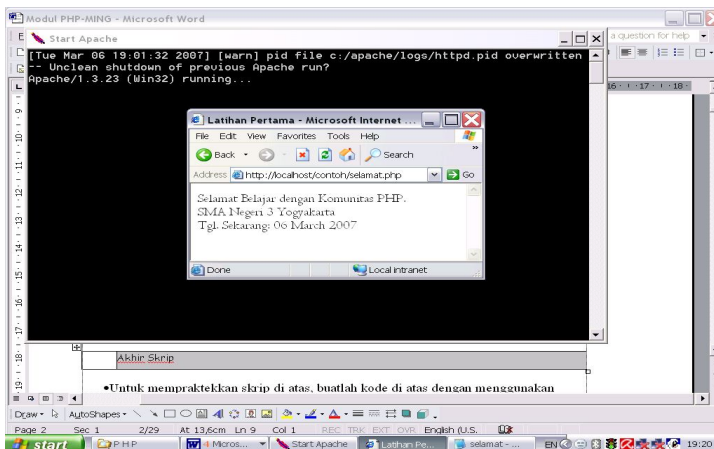
```

	Akhir Skrip
--	--------------------

- Untuk mempraktekkan skrip di atas, buatlah kode di atas dengan menggunakan editor teks yang tersedia pada Linux. Simpan dengan nama **selamat.php**, diletakkan pada direktori **public_html**.
- Setelah itu, jalankan *browser* dan ketikkan

localhost/~labkom/selamat.php

- Hasilnya sbb:



Dasar PHP

Pernyataan

- Pernyataan adalah satu kesatuan perintah yang dapat dijalankan oleh PHP dengan ciri diakhiri tanda titik-koma (;)
- Contoh :

```
printf("Tgl. Sekarang: %s ", Date("d F Y"));
```

Variabel

- Variabel digunakan untuk menyimpan nilai dan nilainya bisa diubah sewaktu-waktu.
- Nama variabel diawali dengan \$ diikuti dengan sebuah huruf dan berikutnya bisa berupa huruf, angka, dan garis bawah (_).
- Huruf kecil dan huruf kapital dibedakan
- Contoh pemberian nilai:

```
$nama = "padmanaba";

<html>
<head>
<title>Test Penyisipan PHP Pada HTML</title>
</head>
<body>
Pemimpin Republik Indonesia <br>
<?php
// Berikut ini adalah inisiasi beberapa variabel
$Presiden = "Ir. Sukarno";
$Presidennya = "Jend. Suharto";
$Presidenku = "Prof. Habibie";
$Presiden4 = "Kyai Gus Dur";
$Presiden5 = "Megawati";
$6Presiden = "Jend. SBY";
$_Presiden7 = "Jend. Yudhoyono";

?>
<b>Inilah Presiden Republik Indonesia.<br>
<?php
echo "Presiden RI :<br> $Presiden, <br>
$Presidennya,<br> $Presidenku,<br> $Presiden4,<br> $
Presiden5,<br> $6Presiden,<br> $_Presiden7.<br> </b>";
?>
</body>
</html>
```

Komentar

- Komentar berguna sebagai pemberi keterangan dalam program dan akan diabaikan oleh PHP
- Komentar diawali dengan # dan berlaku dari karakter tsb hingga akhir baris
- Contoh:

```
print "Tes"; # Menampilkan tes
```

Tipe Data

- Tipe data dapat berupa integer (bilangan bulat, ± 2 milyar), double (bilangan real), dan string (deretan karakter).

Fungsi untuk Konversi Data

- **intval(\$variabel)** atau **(integer) \$variabel** menghasilkan nilai bulat dari isi \$variabel
- **doubleval(\$variabel)** atau **(double) \$variabel** menghasilkan nilai real dari isi \$variabel
- **strval(\$variabel)** atau **(string) \$variabel** menghasilkan nilai real dari isi \$variabel

- Contoh:

```
intval("32.5 derajat") → 32
doubleval("32.5 derajat") → 32.5
```

Konstanta String

- Ditulis dalam tanda petik ganda
- Contoh:


```
"Hai"
```
- Terdapat sejumlah karakter yang menggunakan penulisan secara khusus, yaitu didahului dengan simbol *backslash* (\). Misalnya, \n berarti *newline* (karakter pindah baris).

Tabel Karakter-karakter yang diawali dengan tanda \.

Penulisan	Karakter yang dimaksud
\"	Petik ganda
\\	<i>Backslash</i>
\\$	Tanda dolar
\n	<i>Newline</i> (pindah baris)
\r	<i>Carriage return</i>
\t	Tab
\x00 s/d \xFF	Karakter heksadesimal

- Sebagai contoh, untuk menampilkan tulisan :

```
"Selamat Belajar PHP"
```

pada *browser*, Anda perlu menuliskannya menjadi seperti berikut:

```
print("\Selamat Belajar PHP\");
```

- Untuk menuliskan \$nama pada browser, Anda perlu menuliskannya menjadi

```
print ("\$nama");
```

Operator

- Berguna untuk membentuk ekspresi, misalnya untuk melakukan penjumlahan
- Operator aritmetika:

Tabel Operator-operator aritmetika.

Operator	Kegunaan	Prioritas
+	Penjumlahan	Ketiga
-	Pengurangan	Ketiga
*	Perkalian	Kedua
/	Pembagian	Kedua
%	Sisa pembagian	Kedua
++	Penaikan	Pertama
--	Penurunan	Pertama

- Operator aritmetika juga dapat dikenakan pada string. Contoh:

1 + "2 buah" → 3

- Operator perbandingan berguna untuk melakukan perbandingan.

Tabel Daftar operator perbandingan.

Operator	Makna
==	Sama dengan
<	Kurang dari
>	Lebih dari
<=	Kurang dari atau sama dengan
>=	Lebih dari atau sama dengan
!=	Tidak sama dengan
<>	Tidak sama dengan

- Hasil perbandingan akan berupa 1 bila operasi perbandingan bermakna benar atau 0 jika operasi perbandingan bermakna salah.
- Operator logika biasa digunakan untuk menggabungkan kondisi berganda dan menghasilkan sebuah ekspresi yang bernilai benar (nilai 1) atau salah (nilai 0). Yang tergolong sebagai operator kelompok ini adalah:

- and atau &&
- or atau ||
- xor
- !

Contoh:

```
$kar >= 'A' and $kar <= 'Z'
```

menghasilkan benar kalau \$kar berisi huruf kapital.

Pernyataan if

- Berguna untuk pengambilan keputusan. Bentuk:

```
if (ekspresi)
{
    pernyataan_1
}
else
{
    pernyataan_2
}
```

pernyataan_1 dijalankan kalau ekspresi bernilai benar
pernyataan_2 dijalankan kalau ekspresi bernilai salah

- Contoh skrip (simpan dengan nama **hariini.php**):

	Skrip : hariini.php
---	----------------------------

<HTML>

```

<HEAD>
<TITLE>Latihan Menentukan Nama Hari</TITLE>
</HEAD>

Hari ini:
<?php
    $nama_hari = date("l");
    if ($nama_hari == "Sunday")
        print("Minggu");
    elseif ($nama_hari == "Monday")
        print("Senin");
    elseif ($nama_hari == "Tuesday")
        print("Selasa");
    elseif ($nama_hari == "Wednesday")
        print("Rabu");
    elseif ($nama_hari == "Thursday")
        print("Kamis");
    elseif ($nama_hari == "Friday")
        print("Jumat");
    else
        print("Sabtu");

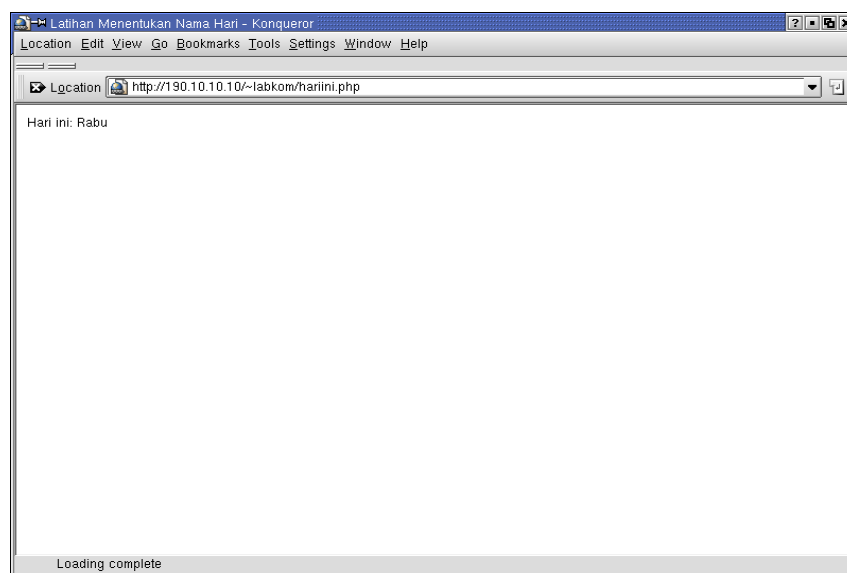
?>

</BODY>
</HTML>

```

	Akhir Skrip
--	-------------

Catatan:
Argumen date berupa huruf l bukan angka satu.



ⓄBerbagai argumen date:

Tabel Kode yang bisa digunakan pada string format *date*.

Argumen	Nilai Balik
a	"am" atau "pm"
A	"AM" atau "PM"

Argumen	Nilai Balik
d	2 digit yang menyatakan tanggal sekarang (Dari "01" hingga "31")
D	Nama hari yang terdiri atas tiga huruf (Misalnya "Sun" dan "Fri")
F	Nama bulan sekarang (Misalnya "January")
g	Jam berformat 12 jam (Dari "1" sampai dengan "12")
G	Jam berformat 24 jam (Dari "0" sampai dengan "23")
h	2 digit yang menyatakan jam berformat 12 jam (Dari "01" sampai dengan "12")
H	2 digit yang menyatakan jam berformat 24 jam (Dari "00" sampai dengan "23")
i	2 digit menit sekarang (Dari "00" sampai dengan "23")
I	"1" kalau <i>Daylight Saving Time</i> "0" kalau tidak
j	Tanggal sekarang ("1" sampai dengan "31")
l (Huruf L kecil)	Nama hari sekarang, dengan format panjang (Misalnya, "Friday")
L	"0" jika tahun sekarang bukan tahun kabisat "1" jika tahun sekarang adalah tahun kabisat
m	Kode bulan sekarang ("01" sampai dengan "12")
M	Nama bulan, terdiri atas tiga huruf. Misalnya, "Jan"
n	Kode bulan sekarang tanpa nol di depan (dari "1" sampai dengan "12")
r	Tanggal dengan format RFC 822.
s	Detik sekarang (Dari "00" sampai dengan "59")
S	Akhiran "th" atau "nd"
t	Jumlah hari pada bulan sekarang ("28" sampai dengan "31")
T	Pengaturan Timezone pada mesin. Misalnya, "MDT"
w	Kode hari dalam angka: "0" = Minggu "1" = Senin, dst.
Y	Tahun dalam bentuk 4 digit. Misalnya, "2001"
y	Tahun dalam bentuk dua digit (tanpa abad)
z	Tanggal dalam tahun (Dari "0" sampai dengan "365")

Latihan:

Kembangkan skrip hariini.php sehingga bisa menampilkan tulisan semacam berikut:

Hari ini hari Rabu, tanggal 19 Juni 2001.

Pernyataan while

- Pernyataan **while** merupakan salah satu pernyataan yang berguna untuk melakukan suatu pengulangan.
- Sebagai contoh, Anda bisa menampilkan bilangan 1 sampai dengan 25 cukup dengan menggunakan kode yang pendek.
- Bentuk pernyataan ini:

```
while (ekspresi)
{
    pernyataan_pernyataan
}
```

Pernyataan **while** akan memeriksa nilai *ekspresi* terlebih dulu. Jika bernilai benar, maka pernyataan-pernyataan yang terdapat dalam {} akan dijalankan dan

kemudian *ekspresi* dievaluasi lagi. Proses ini diulang terus-menerus sampai *ekspresi* bernilai salah.

•Contoh:

```
$i = 0;
while ($i < 4)
{
    print("$i\n");
    $i++;
}
```

Latihan:

Buatlah skrip PHP yang dapat menampilkan bilangan genap antara 2 sampai dengan 20. Gunakan `while`.

Pernyataan for

- Pernyataan **for** juga berguna untuk menangani pengulangan proses.
- Pernyataan ini mempunyai bentuk sebagai berikut:

```
for (ekspr1; eksp2; eksp3)
{
    pernyataan_pernyataan
}
```

•Contoh:

```
for ($i = 0; $i < 4; $i++)
{
    print("$i\n");
}
```

identik dengan contoh **while** di depan.

Pernyataan break

- break** berguna untuk keluar dari suatu proses pengulangan.

•Contoh:

```
for ($i = 1; $i <= 25; $i++)
{
    print("$i <BR>\n");

    if ($i == 10)
        break;
}
```

membuat pengulangan dihentikan ketika \$i bernilai 10.

Pernyataan exit

- Berguna untuk mengakhiri eksekusi program.
- Contoh:


```

for($i = 1; $i <= 25; $i++)
{
    print("$i <BR>\n");

    if ($i == 10)
        exit;
}

print("Selesai <BR>\n");

```

Pada kode di atas, eksekusi dihentikan ketika \$i bernilai 10.

Kode HTML

- Dasar skrip PHP adalah kode HTML.
- Beberapa tag HTML akan dibahas secara singkat.

<H1>, <H2>, <H3>, <H4>, <H5>, dan <H6>

Tag ini berguna untuk membuat judul pada tubuh dokumen HTML. Semakin kecil angka yang digunakan maka ukuran judul semakin besar.

Contoh pemakaian:

```
<H1>Perl</H1>
```

Tag ini berguna untuk menebalkan teks.

Contoh:

```
<B>Tebal</B>
```

<I>

Tag <I> berguna untuk membuat teks dimiringkan.

Contoh:

```
<I>Miring</I>
```

<CENTER>

Membuat teks ditampilkan di tengah-tengah jendela *Web browser*.

Contoh:

```
<CENTER>Teks ini ditengahkan</CENTER>
```

<SUB>

Tag ini berguna untuk membuat teks dalam bentuk subskrip (x_1). Contoh:

X₁

<SUP>

Tag ini berguna untuk membuat teks dalam bentuk superskrip (x^2). Contoh:

X¹

Tag ini berguna untuk menentukan font tulisan beserta atributnya.

Contoh:

Perl

akan membuat tulisan Perl ditampilkan berukuran 5 titik lebih besar daripada ukuran normal.

Perl

akan membuat tulisan Perl ditampilkan dengan warna hijau. Angka 00FF00 secara berturut-turut (masing-masing berupa 2 digit) menyatakan komponen merah, hijau, dan biru.

**
**

Tag ini berguna untuk membuat *newline* (pindah baris). Bentuk:

tanpa disertai </BR>.

<HR>

Tag ini berguna untuk membuat garis horisontal sepanjang layar. Bentuk:

<HR>

tanpa disertai </HR>.

<A>

Tag ini menandai *hyperlink* (teks bergaris bawah yang kalau diklik akan membuat halaman Web lain ditampilkan).

Contoh:


```
<A HREF=http://localhost/cgi-bin/tes.cgi>tes.cgi</A>
```

akan membuat <http://localhost/cgi-bin/tes.cgi> dipanggil kalau tes.cgi diklik.

<PRE>

Berguna untuk membuat teks yang berada dalam <PRE> dan </PRE>, termasuk baris kosong dan spasi putih (spasi dan tab), akan ditampilkan apa adanya.

Contoh:

	Skrip : tag.php
---	------------------------

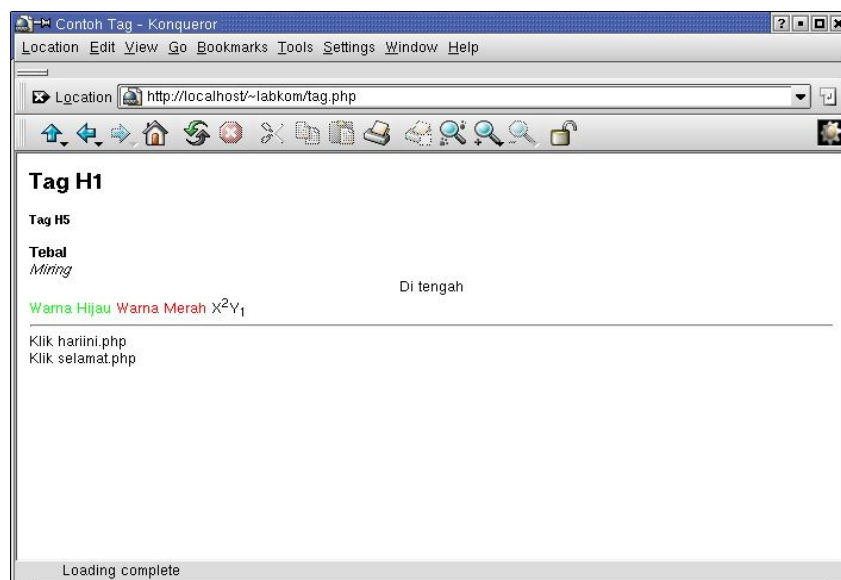
```

<HTML>
<HEAD>
<TITLE>Contoh Tag</TITLE>
</HEAD>
<BODY>

<H1>Tag H1</H1>
<H5>Tag H5</H5>
<B>Tebal </B> <BR> <I>Mi ri ng</I >
<CENTER>Di tengah</CENTER>
<FONT COLOR="#00FF00">Warna Hi j au</FONT>
<FONT COLOR="#FF0000">Warna Merah</FONT>
X<SUP>2</SUP>Y<SUB>1</SUB>
<HR>
Kl i k <AHREF=hari i ni . php>hari i ni . php</A>
<BR>
Kl i k <AHREF=sel amat . php>sel amat . php</A>
</BODY>
</HTML>

```

	Akhir Skrip
--	-------------



Penanganan Form

- Form HTML memungkinkan pemakai memasukkan suatu masukan dari *keyboard* dan masukan ini dapat ditangkap oleh skrip CGI. Tag yang digunakan berupa <FORM>. Di dalam tag ini terdapat atribut ACTION yang menentukan berkas yang akan dijalankan sekiranya tombol SUBMIT diklik. Selain itu juga terdapat atribut METHOD yang menentukan metode untuk menyampaikan string query ke Web server.

- GET

Pada metode ini, string *query* dijadikan satu dalam URL, sehingga ketika skrip dijalankan pada URL akan muncul semacam berikut:


```
http://localhost/cgi-bin/login.php?pemakai=akadir&password=abc45678
```

- POST

Pada metode ini, data masukan diletakkan ke dalam *standard input* sehingga tidak terlihat pada URL.

- Tag <INPUT> dalam FORM digunakan untuk menentukan elemen masukan. Jika TYPE diisi dengan "TEXT" maka pemakai dapat mengisikan suatu teks ke dalam kotak teks. Atribut NAME digunakan untuk menentukan variabel yang dipakai untuk menyimpan masukan. Atribut SIZE menentukan panjang maksimal teks yang dapat dimasukkan pemakai.
- INPUT TYPE="PASSWORD" menyatakan bahwa masukan berupa *password* dan dengan sendirinya setiap karakter yang diketik pemakai akan ditampilkan dengan tanda bintang (*).
- INPUT TYPE="SUBMIT" digunakan untuk menampilkan tombol. Atribut VALUE dipakai untuk menentukan label pada tombol. Jika tombol ini diklik maka URL yang disebutkan pada ACTION-nya FORM akan dieksekusi.
- INPUT TYPE="RESET" digunakan untuk mereset form
- INPUT TYPE="CHECKBOX" digunakan untuk menangani kotak cek
- INPUT TYPE="RADIO" digunakan untuk membentuk tombol radio
- Tag <SELECT> digunakan untuk mengimplementasikan kotak kombo

Sebuah contoh form untuk menyimulasikan proses login:

	Skrip : login.php
---	--------------------------

```
<HTML>
<HEAD>
<TITLE>Form Login</TITLE>
</HEAD>
<BODY>
<H1>Form Login</H1>
<FORM ACTION="akses.php"
      METHOD="POST" >
<PRE>
Nama pemakai : <INPUT TYPE="TEXT"
                NAME="pemakai "
                SIZE=8>
Password      : <INPUT TYPE="PASSWORD"
                NAME="password"
                SIZE=8>
</PRE>
```

```
<INPUT TYPE="SUBMIT" VALUE="Login">
</FORM>
</BODY>
</HTML>
```

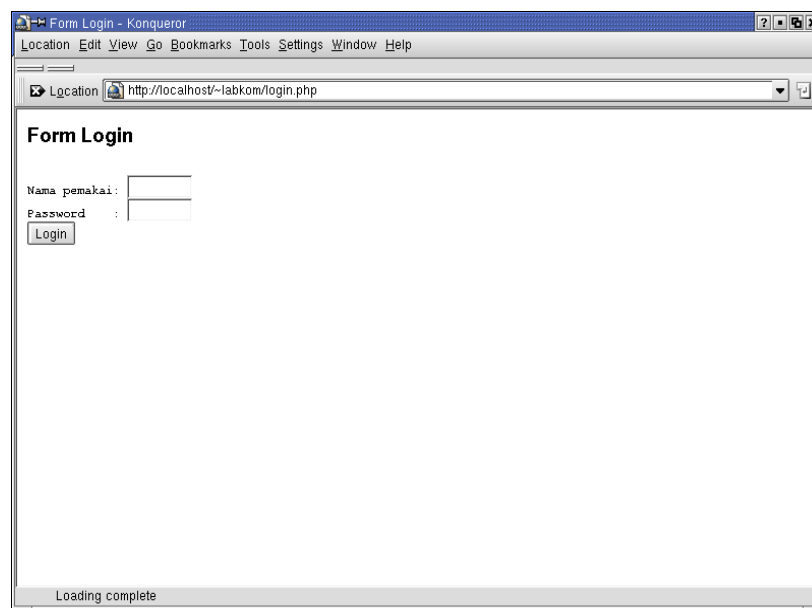
Akhir Skrip

- Jika Anda mengklik tombol Login, akan muncul kesalahan karena akses.php belum didefinisikan.
- Berikut adalah kode skrip skses.php

Skrip : **akses.php**

```
<HTML>
<HEAD>
<TITLE>Form Akses</TITLE>
</HEAD>
<BODY>
```

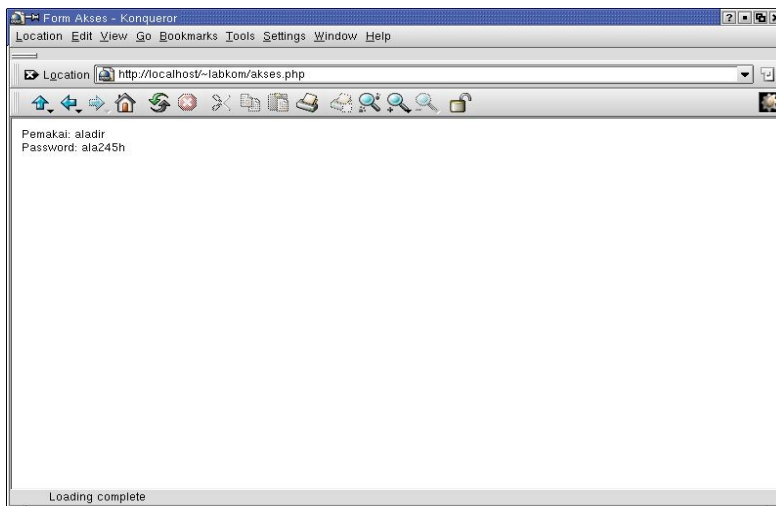
```
<?php
print("Pemakai : $pemakai \n<BR>");
print("Password: $password\n<BR>");
```



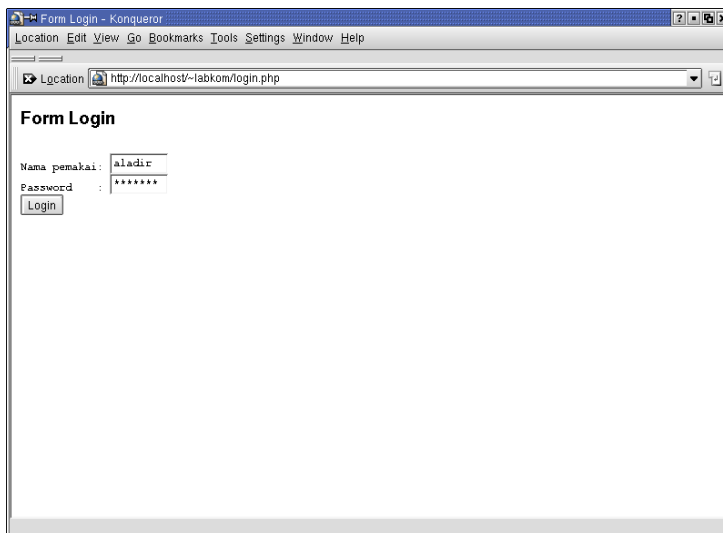
?>

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

Akhir Skrip



Contoh untuk membuat form pemasukan data pegawai.



Skrip : entripeg.php

```
<HTML>
<HEAD>
<TITLE>Entri Pegawai</TITLE>
</HEAD>
<BODY>
<H1>Form Entri Pegawai</H1>
<FORM ACTION="savepeg.php"
      METHOD="POST">

<PRE>
NIP          : <INPUT TYPE="TEXT" NAME="nip"      SIZE="5">
Nama         : <INPUT TYPE="TEXT" NAME="nama"    SIZE="25">
Alamat      : <INPUT TYPE="TEXT" NAME="alamat"  SIZE="35">
Kota        : <INPUT TYPE="TEXT" NAME="kota"    SIZE="15">
Tanggal Lahir: <SELECT NAME="tanggal">
                <OPTION VALUE="0" SELECTED >Tanggal

                <?php
                    // bentuk pilihan tanggal 1 s/d 31
                    for ($i = 1; $i < 32; $i++)
                        print("<OPTION VALUE=$i >$i");
                ?>
```

```

        </OPTION>
    </SELECT>

    <SELECT NAME="bulan">
        <OPTION VALUE="0" SELECTED >Bulan
        <OPTION VALUE="1">Januari
        <OPTION VALUE="2">Pebruari
        <OPTION VALUE="3">Maret
        <OPTION VALUE="4">April
        <OPTION VALUE="5">Mei
        <OPTION VALUE="6">Juni
        <OPTION VALUE="7">Juli
        <OPTION VALUE="8">Agustus
        <OPTION VALUE="9">September
        <OPTION VALUE="10">Oktober
        <OPTION VALUE="11">Nopember
        <OPTION VALUE="12">Desember
    </OPTION>
</SELECT>

    <SELECT name="tahun">
        <OPTION VALUE="0" SELECTED >Tahun

        <?php
            // Bentuk tahun sebanyak 120 tahun
            // dimulai dari tahun sekarang
            $sekarang = (integer) date("Y");
            for ($i = $sekarang; $i > ($sekarang - 120); $i--)
                print("<OPTION VALUE=$i >$i<BR>");
        ?>

        </OPTION>
    </SELECT>

<BR>
Jenis Kelamin: <INPUT TYPE="RADIO" NAME="sex" VALUE="P" CHECKED >Pria
               <INPUT TYPE="RADIO" NAME="sex" VALUE="W" >Wanita

<BR>
<BR>
Bagian      :   <SELECT NAME="bagian">
                 <OPTION VALUE="0" SELECTED >Pilih Bagian
                 <OPTION VALUE="1">Akunting
                 <OPTION VALUE="2">Personalia
                 <OPTION VALUE="3">Pemasaran
                 <OPTION VALUE="4">Produksi
                 <OPTION VALUE="5">EDP
                 </OPTION>
            </SELECT>

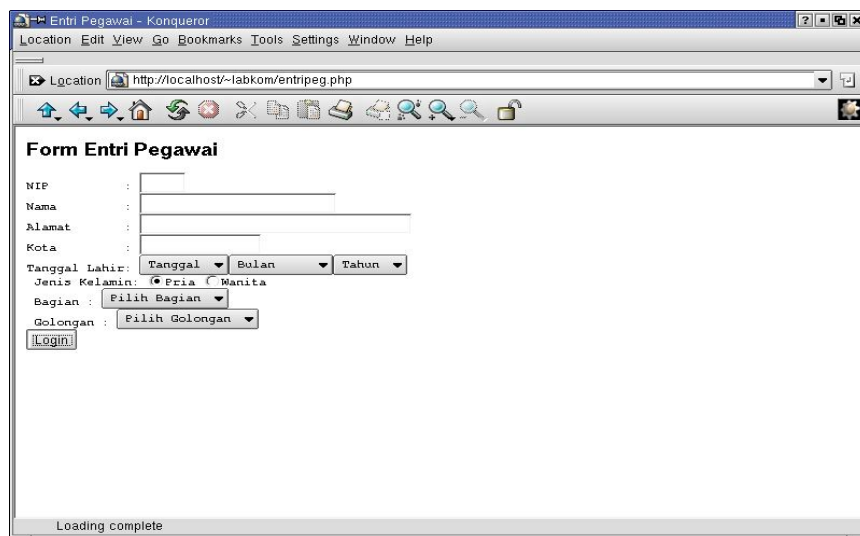
<BR>
<BR>
Golongan    :   <SELECT NAME="golangan">
                 <OPTION VALUE="0" SELECTED >Pilih Golongan
                 <OPTION VALUE="1A">1A
                 <OPTION VALUE="1B">1B
            <OPTION VALUE="2A">2A
                 <OPTION VALUE="2B">2B
                 <OPTION VALUE="3A">3A
                 <OPTION VALUE="3B">3B
                 <OPTION VALUE="4A">4A
                 <OPTION VALUE="4B">4B
                 <OPTION VALUE="5A">5A
                 <OPTION VALUE="5B">5B
                 </OPTION>
            </SELECT>

</PRE>

```

```
<INPUT TYPE="SUBMIT" VALUE="Login">
</FORM>
</BODY>
</HTML>
```

Akhir Skrip



- Skrip savepeg.php yang dijalankan oleh entripeg.php dapat dilihat di bawah ini:

Skrip : savepeg.php

```
<?php
    $ok = TRUE;
    if (empty($nip))
    {
        print("NIP perlu diisi \n<BR>");
        $ok = FALSE;
    }

    if ($tanggal == "0")
    {
        print("Tanggal perlu diisi \n<BR>");
        $ok = FALSE;
    }

    if (! $ok)
    {
        print("Tekan tombol BACK u/ ");
        print("kembali ke form pengisian\n<BR>");
        exit;
    }
?>

<HTML>
<HEAD>
<TITLE>Simpan Data Pegawai /TITLE>
</HEAD>
<BODY>

<H1>Simulasi Penyimpanan Data Pegawai </H1>

<PRE>
<?php
```



```

print("NIP           : $nip\n<BR>");
print("Nama          : $nama\n<BR>");
print("Alamat         : $alamat\n<BR>");
print("Kota           : $kota\n<BR>");
print("Tanggal Lahir  : $tanggal /$bulan/$tahun\n<BR>");
print("Jenis Kelamin  : ");

if ($kelamin == "P")
    print("Pria\n<BR>");
else
    print("Wanita\n<BR>");

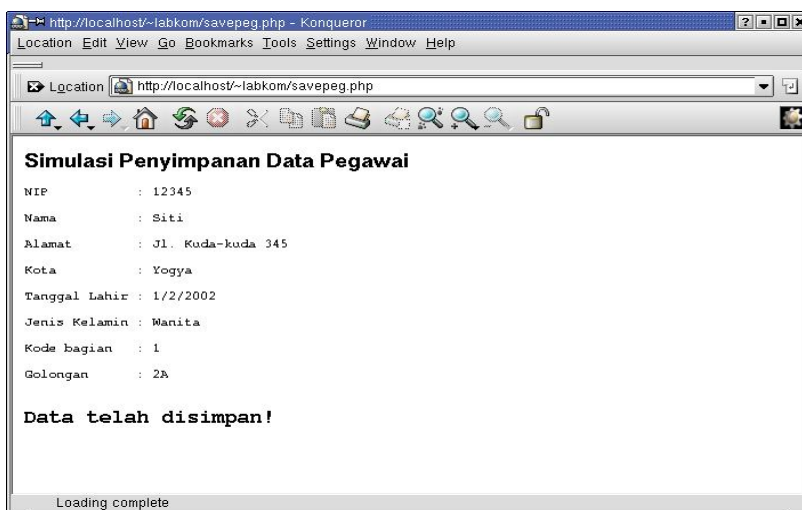
print("Kode bagian   : $bagian\n<BR>");
print("Golongan      : $golongan\n<BR>");

print("<H2>Data telah disimpan! </H2>");
?>

</PRE>
</BODY>
</HTML>

```

Akhir Skrip



Array

- array adalah suatu wadah yang dapat menampung sejumlah nilai.
- Contoh pendefinsian array:

```

$kota = array("Yogya",
             "Solo",
             "Bandung",
             "Bogor");

```

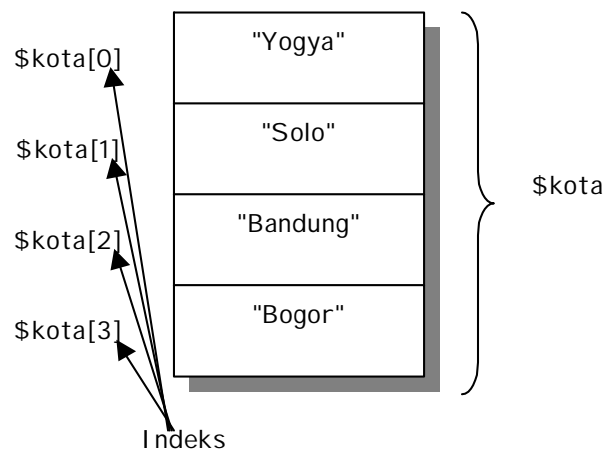
- Alternatif yang lain, Anda bisa menggunakan sederetan pernyataan berikut:

```

$kota[0] = "Yogya";
$kota[1] = "Solo";
$kota[2] = "Bandung";
$kota[3] = "Bogor";

```

- Angka yang diletakkan di dalam tanda [] biasa disebut **kunci** atau **indeks**. PHP,



secara bawaan, menggunakan indeks dimulai dengan nol.

- Indeks semua elemen array dan nilainya dapat diakses melalui foreach. Bentuk **foreach**:

```
foreach(ekspresi_array as $indeks => $nilai)
    pernyataan
```

Pada bentuk ini, indeks elemen *array* akan diberikan ke variabel **kunci** dan nilainya diberikan ke variabel **nilai**.

Membuat Fungsi

- Fungsi adalah blok kode yang ditujukan untuk melaksanakan suatu tugas tertentu. Dengan menciptakan fungsi, yang dibuat sekali, fungsi dapat dipanggil berkali-kali di bagian manapun dalam skrip.
- Bentuk fungsi:

```
function nama_fungsi(parameter)
{
    blok kode
}
```

- Pernyataan digunakan untuk menentukan nilai balik fungsi.
- Contoh:

```
function kuadrat($bilangan)
{
    $tmp = $bilangan * $bilangan;
    return $tmp;
}
```

- Berdasarkan definisi di atas, dapat dilakukan pemanggilan seperti berikut:

```
$a = kuadrat(5);
```

- Jika parameter hendak diubah oleh fungsi, parameter dalam definisi harus diawali dengan & tepat sebelum tanda \$. Contoh:

```
function tukar(&$x, &$y)
{
    $tmp = $x;
    $x    = $y;
    $y    = $tmp;
}
```

Fungsi Validasi Tanggal

- Fungsi **checkdate** berguna untuk memeriksa keabsahan suatu tanggal.
- Bentuk pemanggilannya:

checkdate(*bulan, tanggal, tahun*)

Semua argumen bertipe integer. Adapun hasilnya berupa:

- TRUE kalau tanggal yang menjadi argumennya adalah tanggal yang valid
- FALSE kalau tanggal yang menjadi argumennya adalah tanggal yang tidak valid (misalnya tanggal 31 Februari 2001).

Latihan:

Terapkan validasi tanggal pada skrip savepeg.php.

➤ **Fungsi-Fungsi mysql**

▪ **Fungsi mysql_connect**

- Fungsi **mysql_connect** digunakan untuk membuat hubungan ke *database server* MySQL yang terdapat pada suatu *host*. Bentuk pemanggilannya :

mysql_connect (*host, nama_pemakai, password*)

- host* menyatakan nama *host* tempat *MySQL* berada,
- pemakai* adalah nama pemakai yang digunakan untuk melakukan hubungan,
- password* adalah *password* milik pemakai.
- Fungsi ini menghasilkan nilai balik berupa nilai bertipe integer yang menyatakan pengenalan hubungan dan digunakan sebagai pengenalan pada berbagai fungsi yang berawalan **mysql_**. Bila terjadi sesuatu kesalahan, nilai balik akan berupa FALSE.

Contoh :

```
$id_mysql = mysql_connect(
    "localhost", "php", "penguin");
```

▪ **Fungsi mysql_close**

- Fungsi ini berguna untuk menutup hubungan ke *database server* MySQL. Bentuk pemanggilannya:

mysql_close (*pengenal_hubungan*)

- *pengenal_hubungan* menyatakan pengenal yang diperoleh dari pemanggilan fungsi **mysql_connect**.
- Contoh untuk menguji apakah hubungan ke MySQL dapat dibentuk atau tidak.

 Skrip : **tesmysql.php**

```
<HTML>
<HEAD>
<TITLE>Tes MySQL</TITLE>
</HEAD>
<BODY>

<?php
    $pemakai = "php";
    $password = "penguin";
    $id_mysql = mysql_connect("localhost",
                            $pemakai,
                            $password);

    if (! $id_mysql)
        die("Database server MySQL tak dapat diakses");

    mysql_close($id_mysql);

    print("Sukses");
?>

</BODY>
</HTML>
```

 Akhir Skrip

▪ Fungsi **mysql_select_db**

- Fungsi ini berguna untuk memilih *database*, seperti kalau Anda menggunakan perintah USE pada program klien mysql.
- Format pemanggilannya :

mysql_select_db (*database*, *pengenal_hubungan*)

- *pengenal_hubungan* menyatakan pengenal yang diperoleh dari pemanggilan fungsi **mysql_connect** dan argumen *database* adalah nama *database* yang dipilih.
- Contoh :

```
mysql_select_db("hrd", $id_mysql);
```

▪ Fungsi **mysql_query**

- Fungsi **mysql_query** berguna untuk mengeksekusi permintaan terhadap sebuah tabel atau sejumlah tabel.
- Bentuk pemanggilannya:

`mysql_query` (*permintaan*, *pengenal_hubungan*)

- *pengenal_hubungan* menyatakan pengenal yang diperoleh dari pemanggilan fungsi `mysql_connect` dan argumen *permintaan* berupa pernyataan SQL untuk mengakses tabel. Tabel-tabel yang dapat diakses adalah tabel milik *database* yang disebutkan melalui `mysql_select_db`.

▪ Fungsi `mysql_db_query`

- Fungsi ini berguna untuk menjalankan suatu permintaan terhadap suatu *database*.
- Bentuk pemanggilannya:

`mysql_db_query` (*database*, *permintaan*, *pengenal_hubungan*)

- *pengenal_hubungan* menyatakan pengenal yang diperoleh dari pemanggilan fungsi `mysql_connect` dan argumen *database* adalah nama *database* yang dipilih.
- Adapun, *permintaan* berupa pernyataan SQL.
- Perlu diketahui, argumen ketiga bisa tidak ditulis. Bila tidak ditulis, maka dengan sendirinya akan dihubungkan ke hubungan MySQL yang sudah terbentuk atau akan melakukan hubungan seperti kalau Anda memanggil `mysql_connect`.
- Fungsi ini memberikan nilai balik berupa suatu angka bulat yang menyatakan pengenal hasil, yang diperlukan pada berbagai fungsi berawalan `mysql_` lainnya. Hasil FALSE menyatakan bahwa terdapat sesuatu kesalahan.
- Sebagai contoh, dua pernyataan berikut

```
mysql_select_db("hrd", $id_mysql)
mysql_query("SELECT * FROM pribadi", $id_mysql);
```

dapat diganti dengan

```
mysql_query_db("hrd", "SELECT * FROM pribadi",
$id_mysql);
```

▪ Fungsi `mysql_num_rows`

- Fungsi ini berguna untuk memperoleh jumlah baris dari suatu hasil permintaan (*query*) yang menggunakan SELECT.
- Bentuk pemanggilannya:

`mysql_num_rows` (*pengenal_hasil*)

- *pengenal_hasil* menyatakan nilai yang diperoleh dari pemanggilan fungsi seperti `mysql_db_query` atau `mysql_query`.

▪ Fungsi `mysql_affected_rows`

- Fungsi ini berguna untuk memperoleh jumlah baris yang dikenai operasi INSERT, DELETE, dan UPDATE. Bentuk pemanggilannya:

`mysql_affected_rows` ([*pengenal_hubungan*])

- Dalam hal ini, *pengenal_hubungan* menyatakan nilai yang diperoleh dari pemanggilan fungsi `mysql_connect`. Argumen ini bersifat opsional.

Pada permintaan yang mengandung DELETE tetapi tanpa klausa WHERE, penghapusan akan dilakukan pada semua baris. Namun, fungsi `mysql_affected_rows` hanya akan memberikan nilai balik berupa nol.

▪ Fungsi `mysql_num_fields`

- Fungsi ini berguna untuk memperoleh jumlah kolom pada suatu hasil permintaan. Bentuk pemanggilannya:

```
mysql_num_fields (pengenal_hasil)
```

- Dalam hal ini, *pengenal_hasil* menyatakan nilai yang diperoleh dari pemanggilan fungsi seperti `mysql_db_query` atau `mysql_query`.

▪ Fungsi `mysql_fetch_row`

- Fungsi ini menghasilkan suatu *array* yang berisi seluruh kolom dari sebuah baris pada suatu himpunan hasil. Bentuk pemanggilannya:

```
mysql_fetch_row (pengenal_hasil)
```

- Dalam hal ini, *pengenal_hasil* menyatakan nilai yang diperoleh dari pemanggilan fungsi seperti `mysql_db_query` atau `mysql_query`.
- Fungsi ini mempunyai sifat membaca baris berikutnya dalam suatu himpunan hasil. Bila tak ada baris lagi, fungsi ini akan memberikan nilai balik berupa nilai FALSE. Perlu diketahui, indeks pada *array* hasil dimulai dari nol.
- Contoh berikut memberikan gambaran pengaksesan data pada tabel pribadi:



Skrip : `query.php`

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

<?php
    $pemakai = "php";
    $password = "penguin";
    $id_mysql = mysql_connect("localhost",
                            $pemakai,
                            $password);

    if (! $id_mysql)
        die("Database MySQL tak dapat dibuka");

    if (! mysql_select_db("hrd", $id_mysql))
        die("Database hrd tidak bisa dipilih");

    $query = "SELECT * FROM pribadi";
    $hasil = mysql_query($query, $id_mysql);
    if (! $hasil)
        die("Permintaan gagal dilaksanakan");

    print("<H1>Data Pegawai</H1>\n<BR>");
```

```

print("<TABLE BORDER=1>\n<BR>");

print("<TR>\n");
print("<TH>NIP</TH>");
print("<TH>NAMA</TH>");
print("<TH>ALAMAT</TH>");
print("<TH>KOTA</TH>");
print("<TH>TANGGAL LAHIR</TH>");
print("<TH>JENIS KELAMIN</TH>");
print("</TR>\n");

while ( $baris = mysql_fetch_row($hasil) )
{
    print("<TR>\n");
    print("<TD>$baris[0]</TD>");
    print("<TD>$baris[1]</TD>");
    print("<TD>$baris[2]</TD>");
    print("<TD>$baris[3]</TD>");
    print("<TD>$baris[4]</TD>");
    print("<TD>$baris[5]</TD>");
    print("</TR>\n");
}

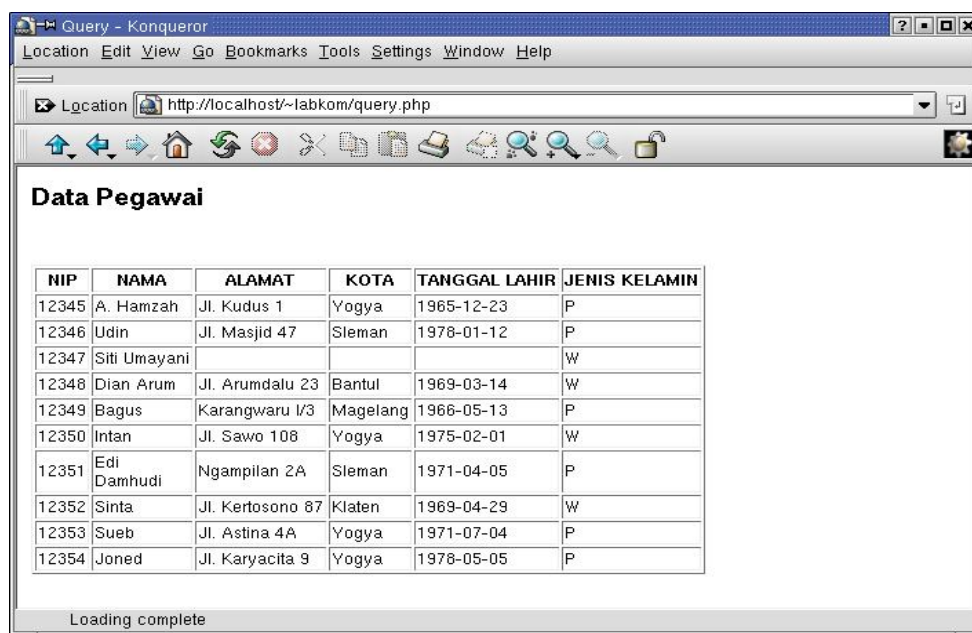
print("</TABLE>\n<BR>");

mysql_close($id_mysql);
?>

</BODY>
</HTML>

```

Akhir Skrip



The screenshot shows a web browser window titled "Query - Konqueror" with the address bar displaying "http://localhost/~labkom/query.php". The main content area shows a table titled "Data Pegawai" with the following data:

NIP	NAMA	ALAMAT	KOTA	TANGGAL LAHIR	JENIS KELAMIN
12345	A. Hamzah	Jl. Kudus 1	Yogya	1965-12-23	P
12346	Udin	Jl. Masjid 47	Sleman	1978-01-12	P
12347	Siti Umayani				W
12348	Dian Arum	Jl. Arumdalu 23	Bantul	1969-03-14	W
12349	Bagus	Karangwaru I/3	Magelang	1966-05-13	P
12350	Intan	Jl. Sawo 108	Yogya	1975-02-01	W
12351	Edi Damhudi	Ngampilan 2A	Sleman	1971-04-05	P
12352	Sinta	Jl. Kertosono 87	Klaten	1969-04-29	W
12353	Sueb	Jl. Astina 4A	Yogya	1971-07-04	P
12354	Joned	Jl. Karyacita 9	Yogya	1978-05-05	P

The status bar at the bottom of the browser window indicates "Loading complete".