

Pemrograman PHP dan MySQL

Oleh:

Willy Bayuardi Suwarno, SP, MSi

willy@ipb.ac.id

Dipublikasi di <http://willy.situshijau.co.id> tanggal 7 Mei 2008

Artikel ini dapat digunakan dan disebarluaskan secara bebas, baik sebagian maupun seluruhnya, untuk tujuan non-komersial dengan syarat mencantumkan nama penulis dan sumbernya. Di luar tujuan itu, pengguna harus memperoleh izin tertulis dari penulis.

Daftar Isi

1	Pendahuluan	3
2	Informasi Sintaks Umum	3
2.1	Keluar dari HTML	3
2.2	Komentar Program	2
3	Tipe Data	2
3.1	Angka : Integer dan Floating Point	2
3.2	String	3
3.3	Array	3
3.4	Object	4
4	Variabel dan Konstanta	5
4.1	Predefined Variable	5
4.2	Operator	6
4.2.1	Operator Aritmatika	6
4.2.2	Operator <i>Bitwise</i>	6
4.2.3	Operator <i>Increment/Decrement</i>	6
4.2.4	Operator Logika	7
4.2.5	Operator <i>Assignment</i>	7
4.3	Level Operator (Operator Precedence)	8
4.4	Skup Variabel	8
4.5	Konstanta	9
5	Kontrol Aliran Program	9
5.1	if, else, elseif	9
5.2	while	10
5.3	do..while	10
5.4	for	11
5.5	foreach	11
5.6	switch	12
5.7	break dan continue	12
5.8	include dan require	13
6	Fungsi	14
7	Operasi File	15
8	MySQL	16
8.1	SQL Dasar	16
8.1.1	Ambil Data	16
8.1.2	Tambah Data	17
8.1.3	Hapus Data	17
8.1.4	Perbarui Data	17
8.2	Administrasi Database dengan phpMyAdmin	17
8.2.1	Membuka phpMyAdmin	17
8.2.2	Membuat Database	18
8.2.3	Membuat Tabel	18
8.2.4	Administrasi Tabel	19
8.2.5	Mengambil Struktur dan Isi Database	20
9	Referensi	20

1 Pendahuluan

PHP adalah bahasa pemrograman yang memungkinkan para *web developer* untuk membuat aplikasi web yang dinamis dengan cepat. PHP merupakan singkatan dari "PHP: Hypertext Preprocessor". PHP ditulis dan diperkenalkan pertama kali sekitar tahun 1994 oleh Rasmus Lerdorf melalui situsnya untuk mengetahui siapa saja yang telah mengakses ringkasan *online*-nya.

PHP merupakan salah satu bahasa *script* yang terbilang baru dan tersedia secara bebas dan masih memungkinkan untuk dikembangkan lebih lanjut. PHP dapat diintegrasikan (*embedded*) ke dalam web server, atau dapat berperan sebagai program CGI yang terpisah.

Karakteristik yang paling unggul dan paling kuat dalam PHP adalah lapisan integrasi database (*database integration layer*). Database yang didukung PHP adalah: Oracle, Adabas-D, Sybase, FilePro, mSQL, Velocis, MySQL, Informix, Solid, dBase, ODBC, Unix dbm, dan PostgreSQL. Contoh script PHP sederhana terdapat pada Listing 1.

Listing 1 : Contoh Script PHP

```
<html>
<head>
  <title>Belajar PHP</title>
</head>
<body>
  <?
    echo "Halo dari PHP!";
  ?>
</body>
</html>
```

2 Informasi Sintaks Umum

2.1 Keluar dari HTML

Listing 2 : Cara untuk Memisahkan Blok Kode PHP di dalam HTML

```
<? echo "Metode tag PHP pendek untuk keluar dari HTML<br>"; ?>
<?php echo "Keluar dengan menggunakan PHP tag penuh<br>"; ?>
<script language="php">
  echo "Beberapa editor HTML tidak menyukai pemrosesan
  instruksi<br>";
</script>
<% echo "Anda mungkin menyukai style ASP" %>
```

2.2 Komentar Program

Ada tiga macam *style* penulisan komentar program yang dapat digunakan, yaitu :

- *C style*, komentar diawali dengan tag `/*` dan diakhiri dengan `*/`, style ini digunakan untuk komentar yang lebih dari satu baris.
- *C++ style*, komentar diawali dengan tag `//` dan hanya berlaku untuk satu baris komentar, untuk baris berikutnya harus diawali tag `//` lagi.
- *Bourne Shell style*, diawali dengan tag `#` untuk satu baris komentar.

Listing 3 : Komentar Program

```
<?
echo "Halo semua!<br>"; // Ini komentar C style;
/*
Ini komentar
yang lebih dari satu baris
*/
echo "Halo lagi!<br>"; # Ini komentar shell-style
?>
```

3 Tipe Data

PHP mendukung tipe-tipe berikut : angka *floating-point*, *integer*, *string*, *array*, dan *object*. Tipe variabel ditentukan oleh konteks variabel tersebut dan tidak ditentukan secara eksplisit oleh *programmer*.

3.1 Angka : Integer dan Floating Point

Integer dapat dituliskan dengan notasi desimal, oktal, atau heksadesimal. Angka *floating-point* dapat dituliskan dalam notasi standar atau *scientific*.

Listing 4 : Representasi Angka

```
<?
$sint1 = 523; // desimal
$sint2 = -523; // desimal negatif
$sint3 = 01013; // representasi oktal dari 523
$sint4 = 0x20B; // representasi heksa dari 523
$float1 = 523.197; // notasi float standar
$float2 = 5.23197e2; // notasi float scientific
// Cetak semua angka
echo "$sint1<br>$sint2<br>$sint3<br>$sint4<br>$float1<br>$float2";
?>
```

3.2 String

String dalam PHP dibatasi oleh kutip ganda (“) atau kutip tunggal (‘). Perbedaannya adalah bagaimana *string* tersebut diinterpretasikan. Jika *string* dibatasi oleh kutip ganda, variabel di dalam *string* akan ditampilkan.

Listing 5 : Contoh String

```
<?
    $makanan = "eskrim";
    echo "Saya suka $makanan<br>";
    echo `Saya suka $makanan`;
?>
```

Karakter *backslash* (\) di dalam *string* dengan kutip ganda dapat digunakan untuk menuliskan karakter khusus (*escape sequences*) yang terdapat pada Tabel 1.

Tabel 1. *Escape Sequences* di dalam String dengan Petik Ganda

<i>Sequence</i>	Arti
\n	Baris baru
\r	<i>Carriage return</i>
\t	Tab horizontal
\\	Backslash
\`	Tanda kutip ganda
\\$	Tanda dollar

3.3 Array

Array dalam PHP dapat berupa *indexed array* (vektor) dan *hash tables* (*associative arrays*). PHP juga mendukung array multidimensi.

Listing 6 : Inisialisasi dan Penggunaan Array

```
<?
    // Membuat array sederhana secara eksplisit
    $a[0] = "Buku";
    $a[1] = "Pinsil";
    $a[] = "Bolpen"; // secara implisit ditempatkan pada indeks
                    // (key) 3
    $a[] = "Penghapus"; // secara implisit ditempatkan pada indeks
                    // (key) 4
    echo "$a[0], $a[1], $a[2], $a[3]<br>";

    // Membuat array asosiatif
    $tanaman["pangan"] = "padi";
    $tanaman["hias"] = "anggrek";
    $tanaman["sayuran"] = "wortel";
    echo "Contoh tanaman hias adalah {$tanaman["hias"]}<br>";
```

```
// Membuat array asosiatif spt di atas, namun caranya lebih
// sederhana
$tanaman = array("pangan" => "padi", "hias" => "anggrek",
"sayuran" => "wortel");
echo "Contoh tanaman sayuran adalah {$tanaman["sayuran"]}<br>";

// Membuat array multidimensi
$b[0][0] = "Nol Nol";
$b[0][1] = "Nol Satu";
echo "Nilai \$b[0][1] adalah {$b[0][1]}<br>";

// Membuat array asosiatif multidimensi
$kota["Jabar"][0] = "Bogor";
$kota["Jabar"][1] = "Bandung";
$kota["Jateng"][0] = "Semarang";
$kota["Jateng"][1] = "Solo";
echo "\$kota['Jabar'][1] = {$kota['Jabar'][1]}<br>";
?>
```

3.4 Object

Object adalah sebuah tipe data yang dapat berupa sebuah bilangan, variabel, atau bahkan sebuah fungsi.

Listing 7 : Penggunaan Tipe Data Object

```
<?
class tanaman
{
    function tanaman()
    {
        $this->komoditas = "apel";
    }

    function set_var($komoditas)
    {
        $this-> komoditas = $komoditas;
    }
}

$buah = new tanaman();
echo $buah -> komoditas;
$buah -> set_var("mangga");
echo "<br>".$buah -> komoditas;
?>
```

4 Variabel dan Konstanta

Variabel PHP diawali dengan tanda dolar (\$) kemudian diikuti dengan nama variabel. Semua nama variabel bersifat *case-sensitive*. Nama variabel yang valid harus diawali oleh huruf atau garis bawah yang diikuti oleh sejumlah huruf, angka, atau garis bawah.

Listing 8 : Contoh Nama Variabel

```
<?
    $variabel1 = "Printer";
    $variabel2 = "Epson";
    echo "$variabel1, $variabel2";
    $lvariabel = 123;    // nama variabel yang salah
    $_tes = "tes"; // benar, diawali oleh garis bawah
    $test_ç = "tes2";    // benar, ç adalah ASCII 128
?>
```

Variabel dapat diisi nilai maupun referensi. Ketika variabel diisi nilai, seluruh nilai dari ekspresi asli diisikan ke variabel tujuan. Setelah diisi, variabel asli bersifat independen, sehingga perubahan pada satu variabel tidak mempengaruhi yang lain.

Ketika variabel diisi referensi, variabel baru merefer pada variabel asli. Perubahan yang dibuat pada salah satu variabel akan berpengaruh pada variabel satunya. Untuk mengisi dengan referensi, tambahkan tanda & pada nama variabel.

Listing 9 : Pengisian Variabel

```
<?
    $variabel1 = "Printer";
    $variabel2 = $variabel1; // diisi nilai
    echo "$variabel1, $variabel2<br>";
    $variabel2 = "Epson";
    echo "$variabel1, $variabel2<br>";
    $variabel3 = &$variabel1; // diisi referensi
    echo "$variabel1, $variabel3<br>";
    $variabel3 = "HP";
    echo "$variabel1, $variabel3<br>";
?>
```

4.1 Predefined Variable

PHP memiliki variabel-variabel yang telah didefinisikan sebelumnya (*predefined variable*). Nilai variabel-variabel tersebut bergantung pada sejumlah faktor, diantaranya jenis server, setting server, dan lain-lain. *Predefined variable* selengkapnya dapat dilihat dengan fungsi `phpinfo()`

4.2 Operator

4.2.1 Operator Aritmatika

Operator	Nama	Contoh	Hasil
+	Penjumlahan	$\$a + \b	Jumlah $\$a$ dan $\$b$
-	Pengurangan	$\$a - \b	Selisih $\$a$ dan $\$b$
*	Perkalian	$\$a * \b	Hasil kali $\$a$ dan $\$b$
/	Pembagian	$\$a / \b	Hasil bagi $\$a$ dan $\$b$
%	Modulus	$\$a \% \b	Sisa dari $\$a$ dibagi $\$b$

4.2.2 Operator Bitwise

Operator	Nama	Contoh	Hasil
&	And	$\$a \& \b 11 (1011 biner) & 13 (1101 biner)	9 (1001 biner)
	Or	$\$a \b 11 (1011 biner) 13 (1011 biner)	15 (1111 biner)
^	Xor	$\$a \wedge \b 11 (1011 biner) ^ 13 (1101 biner)	6 (0110 biner)
-	Not	$\sim \$a$ -11	12 (desimal)
<<	Shift Left	$\$a \ll \b 11 (1011 biner) << 2	2 (10 biner)
>>	Shift Right	$\$a \gg \b 11 (1011 biner) >> 2	44 (101100 biner)

4.2.3 Operator Increment/Decrement

Operator/Contoh	Nama	Hasil
$\$a++$	Post-increment	Operasi dilaksanakan, kemudian $\$a$ ditambah satu.
$++\$a$	Pre-increment	$\$a$ ditambah satu, kemudian operasi dilaksanakan.
$\$a--$	Post-decrement	Operasi dilaksanakan, kemudian $\$a$ dikurangi satu.
$--\$a$	Pre-decrement	$\$a$ dikurangi satu, kemudian operasi dilaksanakan.

4.2.4 Operator Logika

Operator	Nama	Contoh	Hasil
and	And	\$a and \$b	True jika \$a dan \$b keduanya bernilai true
or	Or	\$a or \$b	True jika \$a atau \$b atau keduanya bernilai true
xor	Xor	\$a xor \$b	True jika \$a atau \$b bernilai true, tetapi false jika keduanya bernilai true
!	Not	!\$a	True jika \$a bernilai tidak true
&&	And	\$a and \$b	True jika \$a dan \$b keduanya bernilai true
	Or	\$a or \$b	True jika \$a atau \$b atau keduanya bernilai true

Operator	Operasi
==	Sama dengan
!=	Tidak sama dengan
<	Lebih kecil
>	Lebih besar
<=	Lebih kecil atau sama dengan
>=	Lebih besar atau sama dengan

4.2.5 Operator Assignment

Operator	Contoh	Hasil
=	\$a = \$b	Memasukkan nilai \$b ke \$a
+=	\$a += \$b	Memasukkan nilai (\$a + \$b) ke \$a. Identik dgn \$a = \$a + \$b
-=	\$a -= \$b	Memasukkan nilai (\$a - \$b) ke \$a. Identik dgn \$a = \$a - \$b
*=	\$a *= \$b	Memasukkan nilai (\$a * \$b) ke \$a. Identik dgn \$a = \$a * \$b
/=	\$a /= \$b	Memasukkan nilai (\$a / \$b) ke \$a. Identik dgn \$a = \$a / \$b
.=	\$a .= \$b	Memasukkan nilai (\$a . \$b) ke \$a. Identik dgn \$a = \$a . \$b
%=	\$a %= \$b	Memasukkan nilai (\$a % \$b) ke \$a. Identik dgn \$a = \$a % \$b
=	\$a = \$b	Memasukkan nilai (\$a \$b) ke \$a. Identik dgn \$a = \$a \$b
&=	\$a &= \$b	Memasukkan nilai (\$a & \$b) ke \$a. Identik dgn \$a = \$a & \$b
^=	\$a ^= \$b	Memasukkan nilai (\$a ^ \$b) ke \$a. Identik dgn \$a = \$a ^ \$b
<<=	\$a <<= \$b	Memasukkan nilai (\$a << \$b) ke \$a. Identik dgn \$a = \$a << \$b
>>=	\$a >>= \$b	Memasukkan nilai (\$a >> \$b) ke \$a. Identik dgn \$a = \$a >> \$b

4.3 Level Operator (Operator Precedence)

Asosiativitas	Operator
kiri	,
kiri	or
kiri	xor
kiri	and
kanan	print
kiri	= += -= *= /= .= %= &= = ^= -= <<= >>=
kiri	?:
kiri	
kiri	&&
kiri	
kiri	^
kiri	&
non-asosiatif	== != ===
non-asosiatif	<<= >>=
kiri	<< >>
kiri	+ - .
kiri	* / %
kanan	! - ++ -- (int) (double) (string) (array) (object) @
kanan	[
non-asosiatif	new

Listing 10 : Operator Precedence

```
<?
$x = 10 - 6 / 2;
echo "\$x = $x<br>";
// hasilnya 7, bukan 2, karena level * lebih tinggi dari /
$y = 27 / 9 / 3;
echo "\$y = $y";
// hasilnya 1, bukan 9, karena operator / memiliki
// asosiativitas kiri (dikerjakan dari kiri ke kanan)
?>
```

4.4 Skup Variabel

Secara umum, semua variabel PHP memiliki skup tunggal. Skup ini mencakup *file* yang di-include dan di-require. Di dalam fungsi yang didefinisikan pengguna (*user-defined functions*), yang berlaku adalah skup lokal fungsi. Variabel global harus dideklarasikan global di dalam fungsi jika akan digunakan di dalam fungsi. PHP juga mendukung variabel statis di dalam fungsi.

Listing 11 : Skup Variabel

```

<?
$global1 = "Ini sebuah contoh";
include "contoh11_inc.php";
function cetak()
{
    echo "$global1<br>";
}
cetak();
function cetak2()
{
    global $global1;
    echo "$global1<br>";
}
cetak2();
function fungsistatis()
{
    static $nilai = 0;
    echo "$nilai<br>";
    $nilai++;
}
fungsistatis();
fungsistatis();
?>

--- Isi file, contoh11_inc.php ---
<?
    echo "$global1<br>";
?>

```

4.5 Konstanta

Untuk mendefinisikan konstanta baru, gunakan fungsi define().

Listing 12 : Konstanta

```

<?
define("konst_string", "Belajar PHP");
define("konst_angka", "1");
echo "Kita telah mendefinisikan" . konst_angka .
"konstanta.<br>";
echo "Nilainya adalah `" . konst_string . "`<br>";
?>

```

5 Kontrol Aliran Program**5.1 if, else, elseif**

if mengontrol aliran program melalui pencabangan pada ekspresi kondisional.

Listing 13 : Contoh penggunaan if, else, dan elseif

```

<?
  if (1 < 2)
    echo "Ini akan dicetak.<br>";
  else
    echo "Ini tidak akan dicetak<br>";
  $nilai = 2;
  if ($nilai == 1)
  {
    // Gunakan kurung kurawal untuk membatasi blok statement
    // Hanya diperlukan untuk blok statement ganda (> 1)
    echo "\$nilai == 1<br>";
  }
  elseif ($nilai == 2)
  {
    echo "\$nilai == 2<br>";
  }
  elseif($nilai == 3)
  {
    echo "\$nilai == 3<br>";
  }
  else
  {
    echo "\$nilai bukan 1, 2 atau 3<br>";
  }
?>

```

5.2 while

While adalah tipe pengulangan yang paling sederhana di PHP.

5.3 do..while

Meskipun serupa dengan pengulangan while, pengulangan do..while memeriksa kondisi pengulangan setelah eksekusi pertama dari *statement* pengulangan. Hal ini memastikan bahwa *statement* pengulangan akan dieksekusi paling sedikit satu kali.

Listing 14 : Contoh Penggunaan while dan do..while

```

<?
  echo "Menghitung dari kecil ke besar menggunakan
  <b>while</b>.<br>";
  $indeks = 0;
  // cetak angka dari 0 sampai 9
  while ($indeks < 10)
  {
    echo "$indeks<br>";
    $indeks++;
  }

```

```

echo "Menghitung dari besar ke kecil menggunakan
<b>do..while</b>. <br>";
// cetak angka dari 10 sampai 1
do
{
    print "$indeks<br>";
    $indeks--;
} while ($indeks > 0);
?>

```

5.4 for

Pengulangan for adalah *statement* pengulangan yang paling kompleks di PHP. Sintaksnya adalah sebagai berikut :

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

Ekspresi pertama (expr1) dievaluasi sekali pada awal pengulangan. Ekspresi kedua (expr2) dievaluasi pada tiap awal iterasi. Jika nilainya true, pengulangan berlanjut, dan *statement*(s) dieksekusi. Pada akhir tiap iterasi, ekspresi ketiga (expr3) dievaluasi. Salah satu dari ketiga ekspresi tersebut dapat dikosongkan.

Listing 15 : Contoh Penggunaan for

```

<?
// cetak angka dari 0 sampai 9
for ($indeks = 0; $indeks < 10; $indeks++)
{
    echo "$indeks<br>";
}
/*
$indeks sekarang bernilai 10. Contoh di bawah ini
menunjukkan bahwa salah satu dari ketiga ekspresi
dapat dikosongkan.
*/
for (; $indeks > 0; $indeks--)
{
    echo "$indeks<br>";
}
?>

```

5.5 foreach

Statement foreach adalah mekanisme yang baik untuk mengiterasi *array*. PHP mendukung dua sintaks :

```
foreach (ekspresi_array as nama_variabel_nilai) statement
```

```
foreach (ekspresi_array as nama_variabel_kunci => nama_variabel_nilai)
statement
```

Listing 16 : Contoh Penggunaan foreach

```

<?
$warna = array( "Merah", "Hijau", "Biru" );
foreach ($warna as $nilai)
{
    echo "Nilai warna saat ini adalah $nilai<br>";
}
$warna = array ( "Merah" => "#FF0000",
    "Hijau" => "#00FF00",
    "Biru" => "#0000FF" );
foreach ($warna as $kunci => $nilai)
{
    echo "Nilai heksa dari warna $kunci adalah $nilai<br>";
}
?>

```

5.6 switch

Statement switch menyederhanakan evaluasi banyak kondisi. *Statement* ini sering digunakan untuk menggantikan *statement* if..elseif yang kompleks. Dalam PHP, *string* dapat digunakan sebagai ekspresi switch.

Listing 17 : Contoh Penggunaan switch

```

<?
$indeks = 17;
switch ($indeks)
{
    case 0:
        echo "Nol<br>";
        break;
    case 1:
        echo "Satu<br>";
        break;
    case 2:
        echo "Dua<br>";
        break;
    default:
        echo "Bukan nol, satu, atau dua<br>";
        break;
}
?>

```

5.7 break dan continue

Statement break mengakhiri eksekusi struktur kontrol pada saat itu (*statement* pengulangan atau switch). *Statement* continue hanya digunakan dalam pengulangan. *Statement* continue menyebabkan PHP melewati sisa pengulangan yang ada dan menuju ke awal iterasi selanjutnya.

Listing 18 : Contoh Penggunaan break dan continue

```

<?
/*
periksa apakah ada nilai yang lebih tinggi
dari nilai maksimum sekarang
*/
$angka = array (4, 5, 15, 12, 7, 3, 20, 11, 31);
$maksimum_sekarang = 17;
foreach ($angka as $nilai)
{
    if ($nilai > $maksimum_sekarang)
    {
        $maksimum_sekarang = $nilai;
        break;
    }
}
echo "Nilai maksimum sekarang adalah $maksimum_sekarang<br>";
// cetak angka ganjil dari 0 sampai 20
$indeks = 0;
for ($indeks = 0; $indeks < 20; $indeks++)
{
    if (($indeks % 2) == 0)
        continue;
    echo "$indeks<br>";
}
?>

```

5.8 include dan require

PHP menyediakan dua mekanisme untuk memasukkan *file* eksternal, yakni `include()` dan `require()`. *Statement* `include()` adalah fungsi PHP reguler, sedangkan `require()` adalah konstruksi bahasa khusus dan memiliki sejumlah pembatasan dalam penggunaannya.

Fungsi `include()` dievaluasi setiap kali ditemukan dan dapat berada dalam pengulangan atau *statement* kondisional. Ini berarti bahwa *file* dapat dimasukkan secara kondisional atau kelompok *file* dapat dimasukkan dengan menggunakan sebuah pengulangan. Fungsi `include()` juga membolehkan *file* target menghasilkan nilai kembali (*return value*) dan menyimpan nilai tersebut dalam sebuah variabel. Pemrosesan *file* dalam sebuah *statement* `include()` diakhiri jika ditemukan *statement* `return`.

Statement `require()` berbeda dari *statement* `include()` dimana *statement* `require()` tidak *subject* pada struktur kontrol apapun. Hal ini berarti *file* tidak dapat dimasukkan secara kondisional dengan menggunakan `require()`. *Statement* akan dieksekusi satu kali jika muncul di dalam sebuah pengulangan atau bahkan jika muncul di dalam sebuah *statement* kondisional yang hasil evaluasinya `false`. Perbedaan lainnya adalah sebuah *file* dalam *statement* `require()` tidak akan mengembalikan suatu nilai.

6 Fungsi

PHP mendukung pembuatan fungsi yang didefinisikan oleh pengguna (*user-defined functions*). PHP mendukung fitur-fitur berikut: nama variabel fungsi, jumlah variabel argumen, argumen *default*, argumen yang dikirimkan berdasarkan nilai, dan argumen yang dikirimkan berdasarkan referensi.

Secara *default*, argumen dikirimkan berdasarkan nilai. Untuk mengirimkan nilai berdasarkan referensi, awali nama variabel dengan tanda &. Ketika menggunakan argumen *default*, semua argumen *default* harus berada di kanan semua argumen *non-default*. Fungsi `func_num_arg()`, `func_get_arg()`, dan `func_get_args()` tersedia untuk mendapatkan informasi tentang argumen yang dikirimkan. Contoh di bawah ini memperlihatkan penggunaan umum dari fungsi PHP.

Listing 19 : Contoh Fungsi yang Didefinisikan Pengguna

```
<?
// sebuah fungsi sederhana
function tambah ($a, $b)
{
    return $a + $b;
}

// dikirimkan berdasarkan referensi
function tambahstring (&$stringawal, $stringtambahan)
{
    // karena dikirimkan berdasarkan referensi, nilai
    // $stringawal akan berubah diluar skop fungsi ini
    $stringawal .= $stringtambahan;
}

// nilai default
/*
Fungsi ini dapat dipanggil dengan :
cetaktag("href", "text");
atau
cetaktag("href", "text", "target");
*/
function cetaktag($href, $text, $targ = "")
{
    if ($targ == "")
    {
        echo "a href=\"\$href\">$text</a>";
    }
    else
    {
        echo "a href=\"\$href\" target=\"\$targ\">$text</a>";
    }
}

// daftar argumen
function cetaksemua()
{
    $jumarg = func_num_args();
    for ($indeks = 0; $indeks < $jumarg; $indeks++)
```

```

    {
        $nilaiarg = func_get_arg($indeks);
        echo "Argumen $indeks: $nilaiarg<br>";
    }
}

echo ( "tambah(3, 5) : " . tambah(3,5) . "<br>");

$kalimat = "Saya sedang";
tambahstring($kalimat, "belajar PHP");
echo "$kalimat<br>";

cetaktag("contoh10.php", "Tampilkan Contoh 10 lagi");
echo "<br>";
cetaktag("contoh10.php", "Tampilkan Contoh 10 lagi, dalam
window baru", "_blank");
echo "<br>";

echo "Panggil cetaksemua(1,2,3,4,5) :<br>";
cetaksemua(1,2,3,4,5);
?>

```

7 Operasi File

PHP menyediakan fungsi `fopen()` untuk membuka *file*. Sintaksnya sebagai berikut :

`fopen(namafile, mode)`

Mode menunjukkan operasi yang akan dilakukan pada *file* tersebut. Beberapa mode yang ada dapat dilihat pada Tabel 2.

Tabel 2. Mode Operasi File

Mode	Keterangan
r	Membuka <i>file</i> hanya untuk dibaca; pointer ditempatkan di awal <i>file</i> .
r+	Membuka <i>file</i> untuk dibaca dan ditulis; pointer ditempatkan di awal <i>file</i> .
w	Membuka <i>file</i> hanya untuk ditulis; pointer diletakkan di awal <i>file</i> dan ukuran <i>file</i> dibuat nol. Jika <i>file</i> tidak ada, <i>file</i> akan dibuat.
w+	Membuka <i>file</i> untuk dibaca dan ditulis; pointer diletakkan di awal <i>file</i> dan ukuran <i>file</i> dibuat nol. Jika <i>file</i> tidak ada, <i>file</i> akan dibuat.
a	Membuka <i>file</i> hanya untuk ditulis; pointer diletakkan di akhir <i>file</i> . Jika <i>file</i> tidak ada, <i>file</i> akan dibuat.
a+	Membuka <i>file</i> untuk dibaca dan ditulis, pointer diletakkan di akhir <i>file</i> . Jika <i>file</i> tidak ada, <i>file</i> akan dibuat.

Listing 20 : Hit Counter

```

<?
  $file="counter.txt";
  $fp=fopen($file,"r");
  $jumlah=fgets($fp,255);
  fclose($fp);
  $fw=fopen($file,"w");
  $hit = $hit +1;
  flock($fw,1);
  fputs($fw,$hit);
  fclose($fw);
  echo "Hit Counter : $hit";
?>

```

8 MySQL

MySQL adalah suatu sistem manajemen database. Suatu database adalah suatu koleksi data terstruktur. Data tersebut dapat berupa apa saja, dari *list* sederhana sampai sebuah galeri gambar. Untuk menambah, mengakses, dan memproses data yang tersimpan dalam sebuah database, dibutuhkan suatu sistem manajemen database seperti halnya MySQL. Sejak komputer menjadi suatu alat yang digunakan untuk menanggulangi data dalam ukuran besar, manajemen database memegang peranan utama dalam perhitungan, sebagai utilitas tunggal maupun sebagai bagian dari aplikasi lain.

MySQL adalah suatu sistem manajemen database relasional. Suatu database relasional menyimpan data dalam tabel yang terpisah. Hal ini menambah kecepatan dan fleksibilitas. Tabel-tabel tersebut terhubung oleh suatu relasi terdefinisi yang memungkinkan memperoleh kombinasi data dari beberapa tabel dalam suatu permintaan. SQL (*Structured Query Language*) adalah bahasa standar yang digunakan untuk mengakses database.

MySQL adalah perangkat lunak *open source*. *Open source* berarti dapat digunakan dan dimodifikasi oleh siapa saja. Semua orang dapat men-*download* MySQL dari internet dan menggunakannya secara gratis.

Untuk administrasi database, seperti pembuatan database, pembuatan tabel, dan sebagainya, dapat digunakan aplikasi berbasis web seperti phpMyAdmin.

8.1 SQL Dasar

8.1.1 Ambil Data

```
SELECT * FROM namatabel WHERE kondisi ORDER BY namafield
```

8.1.2 Tambah Data

```
INSERT INTO namatabel (field_1, field_2, ..., field_n) VALUES
(nilai_1, nilai_2, ..., nilai_n) WHERE kondisi
```

8.1.3 Hapus Data

```
DELETE FROM namatabel WHERE kondisi
```

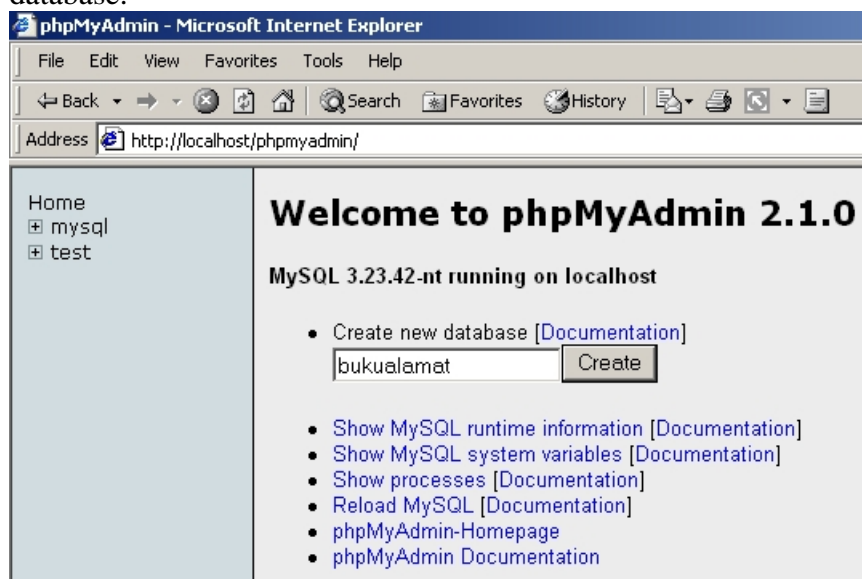
8.1.4 Perbarui Data

```
UPDATE namatabel SET kolom_1=nilai_1, kolom_2=nilai_2,
kolom_n=nilai_n WHERE kondisi
```

8.2 Administrasi Database dengan phpMyAdmin

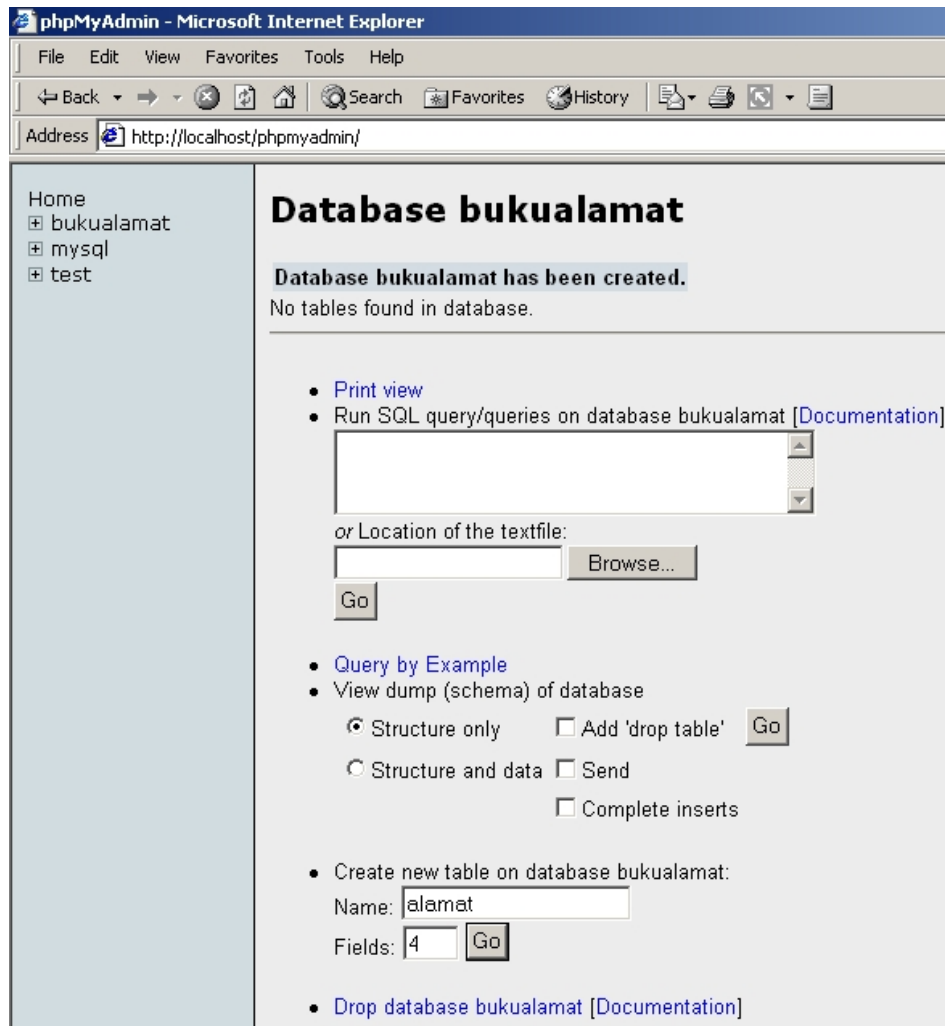
8.2.1 Membuka phpMyAdmin

Jika Anda telah menginstall phpTriad, phpMyAdmin sudah terinstall di direktori apache/htdocs/phpMyAdmin. Jalankan Apache (Start-Program-phpTriad-Start Apache) dan MySQL (Start-Program-phpTriad-Start MySQL), kemudian buka Internet Explorer. Pada kolom address, tuliskan : `http://127.0.0.1/phpMyAdmin`, lalu “Enter”. Tampilan muka phpMyAdmin akan muncul seperti gambar pada halaman berikut. Frame sebelah kiri memperlihatkan daftar database dan tabel yang ada, dan frame sebelah kanan menampilkan halaman untuk membuat database.



8.2.2 Membuat Database

Tuliskan nama database yang ingin Anda buat pada kolom “Create New Database”, dan klik tombol “Create”. Setelah itu akan muncul tampilan seperti gambar di bawah ini.



8.2.3 Membuat Tabel

Lihat Gambar di halaman 16. Pada bagian “Create new table on database xxx”, tuliskan nama tabel pada kolom “Name” dan jumlah fields pada kolom “Fields” dan klik “Go”. Formulir seperti gambar di bawah ini akan tampil. Tuliskan nama field, type, length/set, attributes, null, default, extra pada kolom yang disediakan. Jika suatu field ingin dijadikan Primary Key, beri tanda (v) pada kolom “Primary”. Setelah selesai, klik “Save” untuk menyimpan tabel.

Database bukualamat - table alamat

Field	Type	Length/Set	Attributes	Null	Default	Extra	Primary	Index	Unique
id	INT		UNSIGNED	not null		auto_increment	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
nama	VARCHAR	30		not null			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
alamat	VARCHAR	50		not null			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
telepon	VARCHAR	15		not null			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Table comments:

[\[Documentation\]](#)

8.2.4 Administrasi Tabel

Database bukualamat

table	Action	Records
alamat	Browse Select Insert Properties Drop Empty	0

- [Print view](#)
- Run SQL query/queries on database bukualamat [\[Documentation\]](#):

 or Location of the textfile:
- [Query by Example](#)
- View dump (schema) of database
 - Structure only Add 'drop table'
 - Structure and data Send Complete inserts
- Create new table on database bukualamat:

Name:

Fields:
- [Drop database bukualamat \[Documentation\]](#)

Pada frame di sebelah kiri, jika Anda mengklik tanda (+) di samping nama database maka nama tabel-tabel yang ada di dalam database itu akan muncul di bawahnya. Di frame sebelah kanan, Anda dapat memilih “Browse” untuk melihat isi tabel, “Select” untuk menampilkan isi tabel berdasarkan kriteria tertentu, “Insert” untuk memasukkan data ke dalam tabel, “Properties” untuk menambah/menghapus/mengubah field, “Drop” untuk menghapus tabel, dan “Empty” untuk mengosongkan tabel.

8.2.5 Mengambil Struktur dan Isi Database

Lihat gambar di halaman 17. Pada bagian “View dump (schema) of database”, pilih “Structure and data” untuk mengambil informasi tentang struktur dan data dari tabel-tabel yang telah dibuat. Beri tanda (v) pada “Send” dan klik “Go” untuk menyimpannya ke dalam file.

9 Referensi

Schwendiman, B. 2001. PHP4 Developer’s Guide. The McGraw-Hill Companies, Inc. USA. 775 p.