

# Mempresentasikan Data Teks

Teks adalah jenis data utama yang kita geluti sehari-hari. HTML adalah teks, sebagian besar isi database adalah field teks, e-mail juga teks, belum lagi kotak input atau textarea form, semuanya teks.

Dalam bab ini terdapat beberapa resep yang berhubungan dengan menampilkan data teks .

Resep 5-1: Menampilkan Teks Polos .....	66
Resep 5-2: Menampilkan Warna Tabel Selang-seling .....	69
Resep 5-3: Paging .....	70
Resep 5-4: Membuat Tabel HTML .....	77
Resep 5-5: Multikolom .....	82

## Resep 5-1: Menampilkan Teks Polos

Mencetak teks apa adanya tidak selalu sederhana jika berkaitan browser, shell, dan MySQL. Karena tiap medium kadang memiliki karakter meta tersendiri, maka harus dilakukan escaping. Hal ini kadang dilupakan oleh pemula.

### Mencetak teks apa adanya ke HTML (HTML escaping)

Misalnya Anda memiliki string `$teks` seperti di bawah dan ingin mencetak apa adanya:

```
<b>test
```

```
&quot;;
```

Maka jika Anda melakukan:

```
echo $teks;
```

yang akan tampil di browser malah:

```
test "
```

Untuk mencetak teks apa adanya Anda harus melakukan:

```
echo htmlentities($teks);
```

Hasilnya adalah:

```
<b>test &quot;;
```

Namun `htmlentities()` saja belum akan menghasilkan spasi dan newline apa adanya. Untuk mencetak spasi dan newline apa adanya, apit lagi dengan `<pre>`.

```
echo "<pre>", htmlentities($teks), "</pre>";
```

Sehingga hasilnya:

```
<b>test
```

```
&quot;;
```

**Perhatian:** Jika memiliki teks yang ingin dicetak di dalam atribut HTML seperti `<input name=sesuatu value="...">` atau di dalam elemen seperti `<textarea>...</textarea>`, maka Anda harus selalu ingat untuk menggunakan `htmlentities()`. Hal ini penting agar elemen HTML yang ada di dalam teks yang ingin dicetak tidak salah terinterpretasi sebagai bagian dari dokumen HTML.

### Mencetak teks apa adanya ke URL (URL escaping)

Misalnya ingin membentuk URL ke search engine dengan parameter atau query string yang ada di string `$keywords`.

```
echo "<a href=http://www.google.com/search?q=$keywords>Search!</a>";
```

Jika `$keywords` adalah `jerry seinfeld`, maka jika Anda cetak hasilnya:

```
<a href=http://www.google.com/search?q=jerry seinfeld>Search!</a>
```

Di mata browser, URL akan bernilai:

```
http://www.google.com/search?q=jerry
```

karena spasi tidaklah valid di dalam URL. Cara yang benar adalah:

```
echo "<a href=http://www.google.com/search?q=".  
    uri_escape($keywords).  
    ">Search!</a>";
```

sehingga jika dijalankan hasilnya adalah:

```
<a href=http://www.google.com/search?q=jerry+seinfeld>Search!</a>
```

**Perhatian:** Jika memiliki teks yang ingin dicetak sebagai bagian dari URL, misalnya di dalam link `<a href=...></a>`, maka ingat selalu untuk menggunakan `urlencode()` agar karakter-karakter tak valid di dalam teks yang ingin Anda cetak dikonversi dulu.

### Mencetak teks apa adanya ke shell (shell escaping)

Misalnya Anda ingin memindahkan file yang bernama `Billy Joel - We Didn't Start The Fire.txt` dari `/home/steven/` ke `/home/steven/lyrics/`.

```
$filename = "Billy Joel - We Didn't Start The Fire.txt";  
system("mv $filename /home/steven/lyrics/");
```

## Resep 5-1: Menampilkan Teks Polos

Maka yang terjadi adalah:

```
mv Billy Joel - We Didn't Start The Fire.txt /home/steven/lyrics/
```

Perintah ini tidak akan berhasil dieksekusi oleh shell, karena tanda minus ("-") akan diinterpretasi sebagai opsi command line. Sementara karakter kutip tunggal akan mengakibatkan ketidakseimbangan karena tidak diikuti pasangan penutupnya.

Banyak yang ambil cara mudah:

```
system("mv '$filename' /home/steven/lyrics/");
```

Tapi ini tidaklah selalu aman karena \$filename mungkin saja mengandung kutip tunggal itu sendiri. Cara yang benar:

```
system("mv \"\${filename}\" /home/steven/lyrics/");
```

Hasilnya adalah pengutipan yang benar:

```
mv 'Billy Joel - We Didn\'\'t Start The Fire.txt' /home/steven/lyrics/
```

**Perhatian:** Setiap kali ingin mengirim sebuah teks ke dalam `system()` atau `exec()`, selalu kutiplah dengan `escapeshellarg()`. Ini penting apalagi jika Anda menerima teks tersebut dari form pengunjung yang tidak dipercaya. Lupa mengutip akan mengakibatkan pengunjung yang iseng dapat mengeksekusi sembarang perintah ke shell.

### Mencetak teks apa adanya ke MySQL (MySQL escaping)

Misalnya Anda harus memasukkan string `Jum'at` ke dalam tabel MySQL menggunakan fungsi `mysql_query()`. Jika Anda melakukan seperti ini:

```
$teks = "Jum'at";
mysql_query("insert into words (word) values ('$teks')");
```

maka query akan gagal, disebabkan \$teks mengandung kutip tunggal. Cara yang benar adalah:

```
$teks = "Jum'at";
mysql_query("insert into words (word) values ('
```

## Resep 5-2: Menampilkan Warna Tabel Selang-Seling

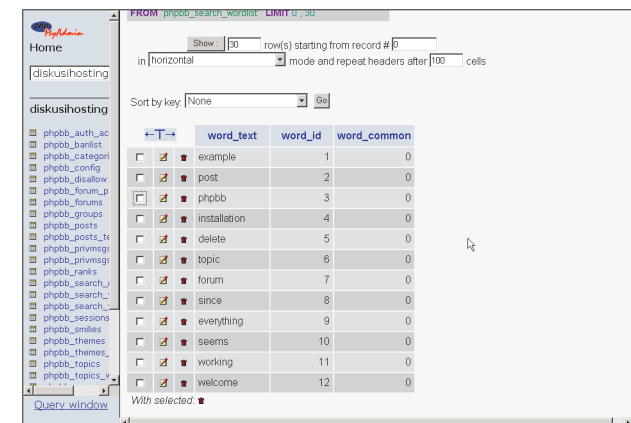
```
mysql_escape_string($teks).
"')");
```

**Perhatian:** Setiap kali ingin mengirim sebuah teks ke dalam `mysql_query()`, pergunalah `mysql_escape_string()`. Jika tidak, skrip Anda dapat rentan terhadap serangan *SQL injection*. SQL injection dapat mengakibatkan penyerang dapat memperoleh akses administrator situs Anda atau bahkan hal yang lebih parah.

Tapi perhatikan bahwa fungsi `mysql_escape_string()` belum/tidak dapat secara otomatis mengubah variabel bernilai `null` menjadi `NULL` untuk database. Anda perlu melakukan pengujian manual untuk kasus ini, seperti contoh pada resep headline Kompas di 2-2.

## Resep 5-2: Menampilkan Warna Tabel Selang-seling

Sebuah tabel yang memiliki beberapa baris data biasanya oleh desainer web diberi gradasi warna selang-seling yang membantu mata kita membedakan satu baris dari



Gambar 5-1. Tabel Berwarna Selang-seling

baris lainnya (Gambar 5-1). Untuk melakukan hal ini sebetulnya sederhana saja, tapi di milis ternyata kadang ada yang mengalami kesulitan melakukannya.

```
1|<?
```

```

3|//
4|//  tabel-selang-seling.php
5|//
6|
7|$data = array(
8|    array("1", "steven", "steven@masterwebnet.com"),
9|    array("2", "jseinfeld", "jerry@seinfeld.com"),
10|    array("3", "jason", "jason@alexander.com"),
11|    array("4", "mrichards", "michael@richards.com"),
12|    array("5", "jldreyfus", "julia@louis-dreyfus.com"),
13|    array("6", "larry", "larry@david.com")
14|);
15|
16|echo "<table border=1 cellpadding=5>";
17|echo "<tr bgcolor=#ccccff>";
18|    <td>no</td>
19|    <td>user</td>
20|    <td>email</td>
21|</tr>";
22|for ($i=0; $i<count($data); $i++) {
23|    $row = $data[$i];
24|    echo "<tr bgcolor=#".($i % 2 ? "f0f0f0" : "cccccc").">";
25|        <td>$row[0]</td>
26|        <td>$row[1]</td>
27|        <td>$row[2]</td>
28|    </tr>";
29|}
30|echo "</table>";
31|
32|?>

```

Kuncinya adalah pada operator modulus, %, di baris 24. Baris ini mengatakan, jika nomor baris ganjil ( $i \% 2 == 1$ ) gunakan warna #f0f0f0; jika genap gunakan warna #cccccc. Hasilnya di Gambar 5-2.

## Resep 5-3: Paging

Nama julukan lain untuk resep ini adalah batching, previous-next, tampilan perhalaman. Masalah menampilkan banyak data yang umumnya dari database ke dalam beberapa halaman sering sekali muncul di milis.



no	user	email
1	steven	steven@masterwebnet.com
2	jseinfeld	jerry@seinfeld.com
3	jason	jason@alexander.com
4	mrichards	michael@richards.com
5	jldreyfus	julia@louis-dreyfus.com
6	larry	larry@david.com

Gambar 5-2. Contoh Paging.

Sebuah tabel, view, atau hasil lengkap sebuah query SELECT bisa saja terdiri dari ratusan atau ribuan record. Namun rata-rata database sudah memiliki cara untuk mengambil sebagian saja dari banyak record ini dengan sintaksnya masing-masing. Misalnya, di MySQL terdapat klausa LIMIT:

```

SELECT ... LIMIT count
SELECT ... LIMIT offset, count

```

Sebagai contoh, LIMIT 10 artinya hanya ambil 10 hasil pertama. LIMIT 30,10 artinya ambil hasil ke-31 sampai 40.

Di PostgreSQL juga terdapat klausa LIMIT, meski sintaksnya sedikit berbeda:

```

SELECT ... LIMIT count
SELECT ... LIMIT count OFFSET offset

```

Misalnya, pada contoh LIMIT 10 artinya hanya ambil 10 hasil pertama, LIMIT 10 OFFSET 30 artinya ambil hasil ke-31 s.d. 40. Sintaks PostgreSQL bagi kebanyakan orang lebih jelas daripada MySQL. Di Firebird, sintaksnya sebagai berikut:

```

SELECT FIRST count ...
SELECT FIRST count SKIP offset ...

```

Pada contoh ini, FIRST 10 artinya hanya mengambil 10 hasil pertama, FIRST 10 SKIP 30 artinya ambil hasil ke-31 hingga 40. Sintaks Firebird merupakan yang paling bagus karena sangat jelas. Di MS SQL Server pun terdapat SELECT TOP count ... yang dapat dikombinasikan dengan subquery untuk memperoleh fungsionalitas offset). Di Oracle dan DB2 terdapat fungsionalitas yang sama, meskipun sintaksnya berbeda. Sayang memang semuanya berbeda sintaks, tapi

memang LIMIT/SKIP tidak ditetapkan di standar SQL. Dengan bekal pengetahuan ini, mudah saja untuk menampilkan data per halaman. Contoh di bawah adalah untuk MySQL:

```

1|<?
2|
3|//
4|// paging.php
5|//
6|
7|$DB_HOST = "localhost";
8|$DB_DB   = "primes";
9|$DB_USER = "primes";
10|$DB_PASS = "2345711";
11|
12|$PAGE_DEFAULT = 1;
13|$PAGESIZE_DEFAULT = 25;
14|$PAGESIZE_LOWER_LIMIT = 10;
15|$PAGESIZE_UPPER_LIMIT = 100;
16|
17|include "terbilang.php";
18|
19|echo "<h1>Demo Paging</h1>";
20|
21|mysql_connect($DB_HOST, $DB_USER, $DB_PASS)
22|  or die("Can't connect to MySQL!");
23|mysql_select_db($DB_DB);
24|
25|// parameter untuk mengatur paging: $page = nomor halaman
26|// halaman pertama = 1. halaman terakhir = -1.
27|if (!isset($page)) $page = $PAGE_DEFAULT;
28|if ($page < 1 && $page != -1) $page = 1;
29|
30|// parameter untuk mengatur paging: $pagesize = ukuran
   halaman batas
31|// ukuran terbesar dan terkecil diatur oleh
   $PAGESIZE_LOWER_LIMIT dan
32|// $PAGESIZE_UPPER_LIMIT.
33|if (!isset($pagesize)) $pagesize = $PAGESIZE_DEFAULT;
34|if ($pagesize < $PAGESIZE_LOWER_LIMIT)
35|  $pagesize = $PAGESIZE_LOWER_LIMIT;

```

```

36|if ($pagesize > $PAGESIZE_UPPER_LIMIT)
37|  $pagesize = $PAGESIZE_UPPER_LIMIT;
38|
39|// kita perlu mengetahui jumlah record dulu jika menginginkan
   halaman
40|// terakhir
41|$res = mysql_query("SELECT COUNT(*) FROM primes");
42|$total_rows = mysql_fetch_row($res);
43|
44|// konversi dari $page & $pagesize menjadi $count & $offset
   untuk klausa
45|// LIMIT MySQL
46|if ($page == -1) {
47|  $count = $pagesize;
48|  $offset = $total_rows - $pagesize;
49|} else {
50|  $count = $pagesize;
51|  $offset = ($page-1) * $pagesize;
52|}
53|
54|// buat bar navigasi prev, next, first, last, pages...
55|$last_page = ceil($total_rows/$pagesize); // hlm
   terakhir = jml hlm
56|
57|$adjacent_pages_links = ($page > 6 ? "... " : "");
58|for ($i=$page-5; $i<$page; $i++) {
59|  if ($i < 1) continue;
60|  $adjacent_pages_links .= " ".
61|    "<a href=$PHP_SELF?page=$i>$i</a>";
62|}
63|$adjacent_pages_links .= " <b>$page</b>";
64|for ($i=$page+1; $i<($page+6); $i++) {
65|  if ($i > $last_page) break;
66|  $adjacent_pages_links .= " ".
67|    "<a href=$PHP_SELF?page=$i>$i</a>";
68|}
69|$adjacent_pages_links .= ($page+5 < $last_page ? " ... " :
   "");
70|
71|$navigasi = "<table bgcolor=ccccff border=0 width=100%><tr>".

```

```

73| // link ke halaman pertama (jika bukan di halaman pertama)
74| "<td>".($page == 1 ? "First" :
75|     "<a href=$PHP_SELF?page=1>First</a>")."</td>".
76|
77| // link ke halaman sebelumnya (jika bukan di halaman
78| pertama)
79| "<td>".($page == 1 ? "Prev" :
80|     "<a href=$PHP_SELF?page=".($page-1).">Prev</
81|     a>")."</td>".
82|
83|
84| // link ke halaman sebelumnya (jika bukan di halaman
85| terakhir)
86| "<td align=right>".($page == $last_page ? "Next" :
87|     "<a href=$PHP_SELF?page=".($page+1).">Next</
88|     a>")."</td>".
89|
90| // link ke halaman terakhir (jika bukan di halaman
91| terakhir)
92| "<td align=right>".($page == $last_page ? "Last" :
93|     "<a href=$PHP_SELF?page=$last_page>Last</a>")."</
94|     td>".
95|
96| "</tr></table>";
97|
98| echo $navigasi; // atas
99|
100| // sekarang baru ambil dan tampilkan data
101| echo "<table width=100% border=1 cellpadding=5>
102|     <tr bgcolor=#cccccc><td>Urutan</td>
103|     <td>Angka</td>
104|     <td>Terbilang</td>
105|     </tr>";
106| $res = mysql_query("SELECT * FROM primes LIMIT
107|     $offset,$count");
108| $i = $offset+1;
109| while ($row = mysql_fetch_array($res)) {
110|     echo "<tr bgcolor=#",($i %2 ? "fff0f0":"fff8f8"),">
111|         <td>$i</td>

```

```

107|         <td>$row[number]</td>
108|         <td>".terbilang($row['number']),"</td>
109|         </tr>";
110|     $i++;
111| }
112| echo "</table>";
113|
114| echo $navigasi; // bawah
115|
116| ?>

```

Contoh data resep ini menggunakan daftar bilangan prima. Berikut ini fungsi PHP untuk mengetes apakah sebuah bilangan adalah bilangan prima.

```

1|<?
2|
3|//
4|// prima.php
5|//
6|
7|// mengembalikan true jika prima, false jika bukan
8|function is_prime($n) {
9|    if ($n == 2) return true; // 2 adalah prima
10|    if ($n == 3) return true; // 3 juga
11|    if ($n < 2) return false; // < 2 bukan
12|
13|    $f = floor(sqrt($n)); // batas faktor pembagi
14|    for ($i=2; $i<=$f; $i++) {
15|        if ($n % $i == 0) return false;
16|    }
17|
18|    return true;
19|}
20|
21|?>

```

Kemudian saya buat tabel MySQL sebagai berikut:

```

create table primes(
    number int unsigned not null primary key
);

```

Kemudian mengisi tabelnya dengan angka prima antara 2 sampai 10000:

```
<?
include "prima.php";

$DB_HOST = "localhost";
$DB_DB   = "primes";
$DB_USER = "primes";
$DB_PASS = "235711";

mysql_connect($DB_HOST, $DB_USER, $DB_PASS)
or die("Can't connect to MySQL!");
mysql_select_db($DB_DB);

for ($i=2; $i<10000; $i++) {
    if (is_prime($i))
        mysql_query("INSERT IGNORE INTO primes VALUES ($i)");
}

?>
```

Hasilnya adalah 1229 angka prima:

```
mysql> select count(*) from primes;
+-----+
| count(*) |
+-----+
|      1229 |
+-----+

1 row in set (0.00 sec)
```

#### Pembahasan

Inti pekerjaan “paging” adalah mengonversi parameter nomor halaman menjadi *count* dan *offset* yang akan kita masukkan ke klausa LIMIT SQL. Kode ini ada di baris 25-28. Selain menerima parameter nomor halaman, bisa saja menerima parameter *count* dan *offset* langsung. Tapi bermain nomor halaman lebih sederhana dalam mengkalkulasi link *Previous*, *Next* dan sebagainya. Sebagian besar kompleksitas kode ada di 57-92, yaitu dalam membuat bar navigasi. Di dalam bar ini terdapat link ke halaman pertama (*First*), halaman sebelumnya (*Prev*), halaman

berikutnya (*Next*), dan halaman terakhir (*Last*). Link “dimatikan” (tidak mengandung `<a href=...>`). Untuk mengambil datanya sendiri mudah saja (baris 102), karena *count* dan *offset* telah dikalkulasi. Hasil program di Gambar 5-3.

## Resep 5-4: Membuat Tabel HTML

Setiap programmer PHP tentu mengenal elemen HTML `<table>` dkk (`<td>`, `<tr>`). Tabel telah menjadi sesuatu yang sentral dalam HTML, bukan hanya untuk

Demo Paging		
First	Prev	Next Last
...	13 14 15 16 17 18 19 20 21 22 23 ...	
Urutan	Angka	Terbilang
426	2957	dua ribu sembilan ratus lima puluh tujuh
427	2963	dua ribu sembilan ratus enam puluh tiga
428	2969	dua ribu sembilan ratus enam puluh sembilan
429	2971	dua ribu sembilan ratus tujuh puluh satu
430	2999	dua ribu sembilan ratus sembilan puluh sembilan
431	3001	tiga ribu satu
432	3011	tiga ribu sebelas
433	3019	tiga ribu sembilan belas
434	3023	tiga ribu dua puluh tiga
435	3037	tiga ribu tiga puluh tujuh
436	3041	tiga ribu empat puluh satu
437	3049	tiga ribu empat puluh sembilan
438	3061	tiga ribu enam puluh satu
439	3067	tiga ribu enam puluh tujuh
440	3079	tiga ribu tujuh puluh sembilan
441	3083	tiga ribu delapan puluh tiga
442	3089	tiga ribu delapan puluh sembilan
443	3109	tiga ribu seratus sembilan
444	3119	tiga ribu seratus sembilan belas
445	3121	tiga ribu seratus dua puluh satu
446	3137	tiga ribu seratus tiga puluh tujuh
447	3163	tiga ribu seratus enam puluh tiga
448	3167	tiga ribu seratus enam puluh tujuh
449	3169	tiga ribu seratus enam puluh sembilan
450	3181	tiga ribu seratus delapan puluh satu
First	Prev	Next Last
...	13 14 15 16 17 18 19 20 21 22 23 ...	

Gambar 5-3. Tabel HTML

mempresentasikan data tabular tapi juga untuk mengatur layout. Meskipun seharusnya untuk layout cara yang benar saat ini adalah dengan menggunakan CSS positioning. Tapi tidak semua programmer suka mencetak tag-tag tabel secara manual dengan `echo "<table ..."`. Selain cukup rumit, jika sudah ada nested table atau sel-sel yang bergabung, cara manual ini juga menyita waktu.

Jika Anda menggunakan sistem template, pada kasus-kasus tertentu kita bisa menghindari membuat tabel secara programatis melainkan menggunakan editor WYSIWYG dalam membuat tabel. Caranya, tabel dibuat dengan editor GUI, lalu setelah tabel jadi barulah template ini di-include dari skrip PHP. Tapi pada kasus-kasus lain kita tetap harus menghasilkan tabel secara programatis. Misalnya jumlah

field tidak diketahui, warna tiap sel dapat berubah-ubah sesuai isi data, ada sel yang bergabung-gabung bergantung isi data, dan lain sebagainya. Untuk ini, di PEAR terdapat paket `HTML_Table` yang dapat membantu mempermudah generasi tabel. Berikut ini beberapa contoh:

```

1|<?
|
3|//
4|// html_table-1.php
5|//
|
7|require_once 'HTML/Table.php';
|
9|$table = new HTML_Table('border=1 cellpadding=5');
|
11|for ($x=0; $x<=10; $x++) {
12|    for ($y=0; $y<=10; $y++) {
13|        $table->setCellContents($y, $x, "{$x}x{$y}=" . ($x*$y));
14|    }
15|}
|
17|echo $table->toHtml();
|
19|?>

```

Hasilnya seperti di Gambar 5-4. Seperti dapat Anda lihat, kemudahan utama kelas `HTML_Table` adalah kita dapat memperlakukan tabel HTML seperti sebuah spreadsheet, memberi alamat tiap sel (baris, kolom) secara acak, dan mengisi sel yang tanpa harus berurutan. Dengan kelas ini, Anda tak perlu kuatir salah

0x0=0	1x0=0	2x0=0	3x0=0	4x0=0	5x0=0	6x0=0	7x0=0	8x0=0	9x0=0	10x0=0
0x1=0	1x1=1	2x1=2	3x1=3	4x1=4	5x1=5	6x1=6	7x1=7	8x1=8	9x1=9	10x1=10
0x2=0	1x2=2	2x2=4	3x2=6	4x2=8	5x2=10	6x2=12	7x2=14	8x2=16	9x2=18	10x2=20
0x3=0	1x3=3	2x3=6	3x3=9	4x3=12	5x3=15	6x3=18	7x3=21	8x3=24	9x3=27	10x3=30
0x4=0	1x4=4	2x4=8	3x4=12	4x4=16	5x4=20	6x4=24	7x4=28	8x4=32	9x4=36	10x4=40
0x5=0	1x5=5	2x5=10	3x5=15	4x5=20	5x5=25	6x5=30	7x5=35	8x5=40	9x5=45	10x5=50
0x6=0	1x6=6	2x6=12	3x6=18	4x6=24	5x6=30	6x6=36	7x6=42	8x6=48	9x6=54	10x6=60
0x7=0	1x7=7	2x7=14	3x7=21	4x7=28	5x7=35	6x7=42	7x7=49	8x7=56	9x7=63	10x7=70
0x8=0	1x8=8	2x8=16	3x8=24	4x8=32	5x8=40	6x8=48	7x8=56	8x8=64	9x8=72	10x8=80
0x9=0	1x9=9	2x9=18	3x9=27	4x9=36	5x9=45	6x9=54	7x9=63	8x9=72	9x9=81	10x9=90
0x10=0	1x10=10	2x10=20	3x10=30	4x10=40	5x10=50	6x10=60	7x10=70	8x10=80	9x10=90	10x10=100

Gambar 5-4. Tabel HTML untuk Menampilkan Spreadsheet.

mencetak jumlah `<td>` sehingga tabel bergeser, kekurangan tag penutup `</tr>` atau `</table>` sehingga tabel berantakan. Ingat, baris dan kolom dimulai dari 0.

```

1|<?php
|
3|//
4|// websafecolor.php
5|//
|
7|require_once 'HTML/Table.php';
|
9|$table = new HTML_Table('width=100%');
10|$table->setCaption('<b>Websafe color</b>');
11|$i = $j = 0;
12|for ($R = 0; $R <= 255; $R += 51) {
13|    for ($G = 0; $G <= 255; $G += 51) {
14|        for ($B = 0; $B <= 255; $B += 51) {
15|            $table->setCellAttributes(
16|                $i,
17|                $j,
18|                'bgcolor=#'.sprintf('%02X%02X%02X', $R, $G, $B).
19|                ' title=#'.sprintf('%02X%02X%02X', $R, $G, $B));
20|            $j++;
21|        }
22|    }
23|    $i++;
24|    $j = 0;
25|}
26|echo $table->toHtml();
|
28|?>

```

Contoh di atas adalah modifikasi sedikit dari contoh default yang disertakan di paket `HTML_Table`. Contoh tersebut memperlihatkan bahwa selain mengeset konten, kita juga dapat mengeset atribut. Pisahkan setiap atribut dengan spasi. Hasilnya seperti pada Gambar 5-5. Tentu saja, contoh ini akan nampak jauh lebih indah pada monitor berwarna daripada di halaman buku ini.

Contoh terakhir di bawah memperlihatkan bagaimana menangani `COLSPAN` dan `ROWSPAN`. Hasilnya seperti tampak di Gambar 5-6. Tanpa kelas seperti `HTML_Table`, bisa dibayangkan cukup repotnya menyusun tabel jadwal seperti ini.



```

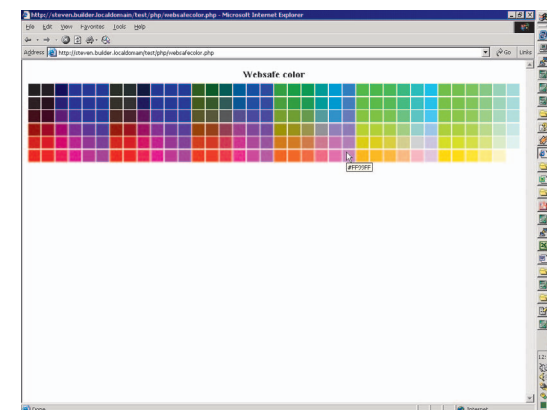
1|<?
2|
3|//
4|// html_table-1.php
5|//
6|
7|require_once 'HTML/Table.php';
8|
9|$table = new HTML_Table('width=100% border=0 '.
10|                        'cellpadding=5 cellspacing=3');
11|
12|// buat header hari
13|$hari = array("Senin","Selasa","Rabu","Kamis","Jumat",
14|              "Sabtu","Minggu");
15|for ($i=0; $i<count($hari); $i++) {
16|    $table->setCellContents (0, $i+1, $hari[$i]);
17|    $table->setCellAttributes(0, $i+1, "align=middle
18|                                     bgcolor=#ffcccc");
19|}
20|
21|// buat header jam
22|for ($i=0; $i<24; $i++) {
23|    $j = $i+1; if ($j==24) $j=0;
24|    $table->setCellContents ($i+1,0, sprintf("%02d:00-
25|                                           %02d:00", $i, $j));
26|    $table->setCellAttributes($i+1,0, "align=right
27|                                     bgcolor=#ffcccc");
28|}
29|
30|// Anton bekerja shift 1 hari Senin, Selasa, Rabu, Kamis,
31|    Jumat
32|// shift 1 = 07:00 - 15:00 = 8 jam
33|for ($day=1; $day<=5; $day++) {
34|    $table->setCellContents (7+1,$day, 'Anton');
35|    $table->setCellAttributes(7+1,$day, 'rowspan=8
36|                                     bgcolor=#ccffcc');
37|}
38|
39|// Bobby bekerja shift 2 hari Senin, Selasa, Rabu
40|// shift 2 = 15:00 - 23:00 = 8 jam
41|for ($day=1; $day<=3; $day++) {

```

```

36| $table->setCellContents (15+1,$day, 'Bobby');
37| $table->setCellAttributes(15+1,$day, 'rowspan=8
38|   bgcolor=#ccccff');
39|}
40|
41|// Chandra bekerja shift 2 hari Kamis, Jumat
42|for ($day=4; $day<=5; $day++) {
43|    $table->setCellContents (15+1,$day, 'Chandra');
44|    $table->setCellAttributes(15+1,$day, 'rowspan=8
45|   bgcolor=#cccccc');
46|}
47|
48|// Danu giliran Sabtu dan Minggu, hanya 08:00 s.d. 14:00 = 6
49|    jam.
50|for ($day=6; $day<=7; $day++) {
51|    $table->setCellContents (8+1, $day, 'Danu');
52|    $table->setCellAttributes(8+1, $day, 'rowspan=6
53|   bgcolor=#ffffcc');
54|}
55|}
56|
57|echo $table->toHtml();
58|
59|?>

```



Gambar 5-5. Tabel HTML Hasil Modifikasi Default Paket Tabel.



## Resep 5-5: Multikolom

```
13| preg_match("/^([^\,]+),(\d+)/", $line, $m);
14| $data[$m[1]] = $m[2];
15|}
|
17|// urutkan data per nama kota
18|ksort($data);
|
20|// tampilkan dalam tabel 20 baris secara vertikal
21|// (isi ke bawah dulu, baru bawah)
|
23|$table = new HTML_Table('width=100% border=0');
24|$row = 0; $col = 0;
25|foreach ($data as $k => $v) {
26|    $table->setCellContents ($row, $col*3 , $k);
27|    $table->setCellAttributes($row, $col*3 , "width=20%");
28|    $table->setCellContents ($row, $col*3+1, $v);
29|    $table->setCellAttributes($row, $col*3+1, "");
30|    $table->setCellContents ($row, $col*3+2, "&nbsp;");
31|    $table->setCellAttributes($row, $col*3+2, "width=5%");
32|    $row++;
33|    if ($row == 20) { $row = 0; $col++; }
34|}
|
36|echo $table->toHtml();
|
38|?>
```