

# Mempresentasikan Data Gambar

Sebuah gambar bermakna seribu kata. Untungnya, di PHP Anda tak perlu menulis seribu kata atau baris kode untuk membuat grafik atau gambar. Library GD adalah salah satu library yang telah dikemas bersama distribusi PHP sejak lama. GD bisa langsung dipakai membuat bentuk-bentuk geometri sederhana, mengubah ukuran gambar, memotong atau menyusun gambar, dan lain sebagainya.

Resep-resep di bab ini kebanyakan hanya mengandalkan GD. Tetapi ada juga yang membahas library lain seperti **JpGraph** dan **NetPBM** untuk melakukan tugas-tugas yang lebih khusus atau high level.

<b>Resep 6-1: Membuat Grafik Garis .....</b>	<b>86</b>
<b>Resep 6-2: Membuat Grafik Batang .....</b>	<b>88</b>
<b>Resep 6-3: Membuat Grafik Kue .....</b>	<b>92</b>
<b>Resep 6-4: Membuat Grafik dengan JpGraph .....</b>	<b>95</b>
<b>Resep 6-5: Membuat Thumbnail .....</b>	<b>97</b>
<b>Resep 6-6: Membuat Ikon (.ICO) .....</b>	<b>99</b>
<b>Resep 6-7: Membuat Barcode .....</b>	<b>102</b>

## Resep 6-1: Membuat Grafik Garis

```

1|<?
|
|
3|//
4|// grafik-garis.php
5|//
|
7|function grafik_garis($width, $height, $data) {
8| // $data dalam format: array( array("data1",100),
  array("data2",75), ...)
|
10| $im = imagecreate($width, $height);
11| $background_color = imagecolorallocate($im, 255, 255, 255);
12| $axis_color = $text_color = imagecolorallocate($im, 0, 0,
  0);
13| $item_color = imagecolorallocate($im, 128, 0, 0);
|
15| // diagram grafik hanya 95% ukuran gambar, agar sumbu bisa
  terlihat
16| $chart_width = 0.95*$width;
17| $chart_height = 0.95*$height;
|
19| // N = jumlah data
20| $N = count($data);
|
22| // tentukan rentang nilai terbesar dan terkecil
23| $max = 0; $min = 0;
24| foreach ($data as $datum) {
25|     if ($max < $datum[1]) $max = $datum[1];
26|     if ($min > $datum[1]) $min = $datum[1];
27| }
28| if ($max == $min) $max = $min+1;
|
30| // faktor adalah: 1.0 nilai data = berapa pixel?
31| $factx = $chart_width / ($N+1);
32| $facty = $chart_height / ($max-$min);
|
34| // posisi pixel titik (0,0)
35| $x0 = ($width-$chart_width)/2;

```

```

36| $y0 = $height - ($height-$chart_height)/2 -
37|     (0-$min)/($max-$min)*$chart_height;
|
39| // gambar sumbu x
40| imagesetthickness($im, 1);
41| imageline($im, $x0, $y0, $x0+$chart_width, $y0,
  $axis_color);
|
43| // gambar sumbu y
44| imageline($im, $x0, ($height-$chart_height)/2,
45|     $x0, $height-($height-$chart_height)/2,
46|     $axis_color);
|
48| // gambar data
49| imagesetthickness($im, 3);
50| $xprev = null; $yprev = null;
51| for ($i=0; $i < $N; $i++) {
52|     $legend = $data[$i][0];
53|     $value = $data[$i][1];
|
55|     $x = $x0 + $factx * ($i+1);
56|     $y = $y0 - $facty * $value;
|
58|     if (!is_null($xprev))
59|         imageline($im, $xprev, $yprev, $x, $y, $item_color);
|
61|     $xprev = $x; $yprev = $y;
62| }
|
64| // kembalikan grafik, siap dioutput dengan imagepng(),
  imagegif(), dsb.
65| return $im;
|
67|}
|
69|// mari cetak sebuah grafik
70|$data = array(
71|     array("season1", 205),
72|     array("season2", 350),
73|     array("season3", 550),
74|     array("season4", 850),

```

```

75| array("season5", 870),
76| array("season6", 810),
77| array("season7", 795),
78| array("season8", 745),
79| array("season9", 1030));
80|
81| header("Content-Type: image/png");
82| imagepng(grafik_garis(640, 480, $data));
83|
84| ?>

```

Resep di atas adalah kerangka untuk membuat grafik garis. Grafik garis adalah yang paling sederhana, Anda cukup membuat image lalu menggambar-gambar garis dengan fungsi `imageline()`. Ada beberapa hal yang bisa dilakukan resep ini. Pertama, mengatur skala otomatis sesuai rentang nilai (dilakukan oleh baris 22-28) agar semua data dapat terlihat di grafik. Kedua, warna dan ketebalan sumbu dan garis grafik dibedakan dengan menggunakan fungsi `setimagethickness()` di baris 40 dan 49. Namun fungsi ini hanya ada di library GD 2.0 ke atas. Yang belum ada masih banyak, misalnya judul grafik, skala pada sumbu, grid, label item data. Kita akan menambahkan fasilitas ini di resep-resep berikutnya.

Jika fungsi-fungsi GD tidak ada yang dikenali oleh PHP, maka kemungkinan harus memuat dulu extension gd PHP di awal skrip dengan cara sebagai berikut:

```
if (!extension_loaded("gd")) dl("gd.so");
```

Ganti `gd.so` dengan `php_gd2.dll` jika Anda berada di Windows. Atau, preload extension gd dari `php.ini`.

## Resep 6-2: Membuat Grafik Batang

```

1|<?
2|
3|//
4|// grafik-batang.php
5|//
6|

```

```

7|function grafik_batang($width, $height, $data, $judul) {
8|
9|    $im = imagecreate($width, $height);
10|    $background_color = imagecolorallocate($im, 255, 255, 255);
11|    $axis_color = $text_color = imagecolorallocate($im, 0, 0, 0);
12|    $item_colors = array(
13|        imagecolorallocate($im, 128, 0, 0),
14|        imagecolorallocate($im, 0, 128, 0),
15|        imagecolorallocate($im, 0, 0, 128),
16|        imagecolorallocate($im, 128, 0, 128),
17|        imagecolorallocate($im, 128, 128, 0),
18|        imagecolorallocate($im, 0, 128, 128),
19|        imagecolorallocate($im, 0, 64, 0),
20|        imagecolorallocate($im, 64, 64, 64));
21|
22|    $chart_width = 0.85*$width;
23|    $chart_height = 0.85*$height;
24|
25|    // N = jumlah data
26|    $N = count($data);
27|
28|    // tentukan rentang nilai terbesar dan terkecil
29|    $max = 0; $min = 0;
30|    foreach ($data as $datum) {
31|        if ($max < $datum[1]) $max = $datum[1];
32|        if ($min > $datum[1]) $min = $datum[1];
33|    }
34|    if ($max == $min) $max = $min+1;
35|
36|    // faktor adalah: 1.0 nilai data = berapa pixel?
37|    $factx = $chart_width / ($N+1);
38|    $facty = $chart_height / ($max-$min);
39|
40|    // posisi pixel titik (0,0)
41|    $x0 = ($width-$chart_width)/2;
42|    $y0 = $height - ($height-$chart_height)/2 -
43|        (0-$min)/($max-$min)*$chart_height;
44|
45|    // ketebalan tiap batang
46|    $bar_width = $factx / 1.5;

```

```

48| // gambar sumbu y
49| imageline($im, $x0, $y0, $x0+$chart_width, $y0,
    $axis_color);
51| // gambar sumbu y
52| imageline($im, $x0, ($height-$chart_height)/2,
53|           $x0, $height-($height-$chart_height)/2,
54|           $axis_color);
56| // gambar data
57| $iwarna = 0;
58| $labels = array();
59| for ($i=0; $i < $N; $i++) {
60|     $legend = $data[$i][0];
61|     $value = $data[$i][1];
63|     $x1 = $x0 + $factx * ($i+1) - $bar_width/2;
64|     $y1 = $value > 0 ? $y0-1 : $y0+1;
66|     $x2 = $x1 + $bar_width;
67|     $y2 = $y0 - $facty * $value;
69|     if ($y2 < $y1) { $tmp = $y1; $y1 = $y2; $y2 = $tmp; }
71|     imagefilledrectangle($im, $x1, $y1, $x2, $y2,
72|                          $item_colors[$iwarna]);
74|     $labels[] = array($x1, ($value > 0 ? $y1-3-10 :
    $y1+3+10),
75|                      "$legend ($value)");
77|     // pilih warna berikutnya
78|     $iwarna++;
79|     if ($iwarna == count($item_colors)) $iwarna = 0;
80| }
82| // gambar label
83| for ($i=0; $i < $N; $i++) {
84|     imagestring($im, 1,
85|                $labels[$i][0], $labels[$i][1],

```

```

86|                $labels[$i][2],
87|                $text_color);
88| }
90| // cetak judul di bawah wilayah grafik
91| imagestring($im, 4, $x0+3, $height-($height-$chart_height)/
    2 + 3,
92|            $judul, $text_color);
94| // kembalikan grafik, siap dioutput dengan imagepng(),
    imagegif(), dsb.
95| return $im;
97|}
99|// mari cetak sebuah grafik
100|$data = array(
101|    array("season1", 205),
102|    array("season2", 350),
103|    array("season3", 550),
104|    array("season4", 850),
105|    array("season5", 870),
106|    array("season6", 810),
107|    array("season7", 795),
108|    array("season8", 745),
109|    array("season9", 1030));
111|header("Content-Type: image/png");
112|imagepng(grafik_batang(640, 480, $data,
113|                      "Popularitas Acara Seinfeld Per
    Musim"));
115|?>

```

Resep ini beranjak dari resep sebelumnya. Kode menggambar sumbu, menghitung skala, semuanya sama. Yang membedakan hanya penggunaan `imagefilledrectangle()`. Contoh output bisa dilihat di Gambar 6-1.

Perhatikan baris 63-69 di mana kita menghitung koordinat pojok tiap batang. Di baris 64 kita menghitung `$y1` sebagai `$y0-1` (satu pixel di atas sumbu x jika nilai data positif/batang di atas sumbu) atau `$y0+1` (satu pixel di bawah sumbu jika nilai

data negatif/batang di bawah sumbu x). Tujuannya agar garis sumbu tidak kita timpa dengan grafik batang, karena sumbu kita gambar sebelum batang (alternatifnya. Meskipun demikian, bisa saja Anda gambar sumbu setelah menggambar batang.

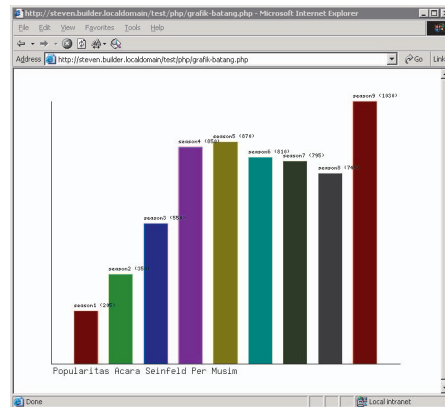
Baris 69 penting untuk memastikan \$y2 lebih dari \$y1. Jika \$y2 lebih kecil dari \$y1, maka kode di baris ini menukar nilainya sehingga \$y2 tetap yang lebih besar.

Ini diperlukan karena `imagefilledrectangle()` mewajibkan x2 dan y2 lebih besar dari x1 dan y1. Jika tidak, batang tidak akan nampak di gambar.

Baris 78-79 adalah kode untuk menentukan warna setiap batang. Kita mendeklarasikan beberapa warna di variabel `$item_colors`. Jika jumlah item di data lebih banyak, maka kode di baris ini akan mendaur ulang warna dari pertama.

Kode untuk menggambar teks label setiap item dikeluarkan dari loop menggambar batang, supaya tidak ada teks yang tertimpa oleh batang di sebelah kanannya. Baris 91-92 mencetak judul grafik di bawah wilayah chart. Karena font-nya dipilih yang agak besar, maka area chart diperkecil 85% ukuran gambar (baris 22-23). Jika ingin membuat batang lebih tipis atau tebal, sesuaikan isi pada baris 46.

GD mendukung arsiran atau pola isian. Di PHP, kita dapat menggunakan fungsi `imagestitle()` untuk memilih sebuah gambar kecil yang dapat dipakai sebagai pola isian. Di resep ini kita membedakan tiap item hanya dengan warna dan semua arsiran sama-sama solid, sehingga tidak membutuhkan fungsi tersebut.



Gambar 6-1. Hasil Grafik Batang.

## Resep 6-3: Membuat Grafik Kue

Grafik lingkaran atau grafik kue ("pie chart") kadang-kadang digunakan untuk memvisualisasikan proporsi. Di grafik lingkaran, nilai data negatif tidak diperbolehkan. Untuk membuat grafik lingkaran sederhana tidaklah sulit dengan GD, yang Anda perlukan hanyalah sedikit dasar geometri mengenai konsep derajat. Demi kesederhanaan, resep ini tidak menggambar label item.

```

1|<?
2|
3|//
4|// grafik-kue.php
5|//
6|
7|function grafik_kue($width, $height, $data, $judul) {
8|
9|    $im = imagecreate($width, $height);
10|    $background_color = imagecolorallocate($im, 255, 255, 255);
11|    $text_color = imagecolorallocate($im, 0, 0, 0);
12|    $item_colors = array(
13|        imagecolorallocate($im, 128, 0, 0),
14|        imagecolorallocate($im, 0, 128, 0),
15|        imagecolorallocate($im, 0, 0, 128),
16|        imagecolorallocate($im, 128, 0, 128),
17|        imagecolorallocate($im, 128, 128, 0),
18|        imagecolorallocate($im, 0, 128, 128),
19|        imagecolorallocate($im, 0, 64, 0),
20|        imagecolorallocate($im, 64, 64, 64));
21|
22|    $N = count($data);
23|
24|    if (!$N) die("Harus ada data!");
25|
26|    $sum = 0;
27|    foreach ($data as $datum) {
28|        if ($datum[1] < 0) die("Nilai tidak boleh negatif!");
29|        $sum += $datum[1];
30|    }
31|
32|    if (!$sum) die("Harus ada nilai positif pada data!");
33|
34|    // faktor: 1 pada nilai data = berapa derajat?
35|    $factor = 360.0 / $sum;
36|
37|    // titik pusat kue
38|    $xc = $width/2;
39|    $yc = $height/2;
40|
41|    // lebar dan tinggi kue

```

```

42| $pie_width = 0.60 * $width;
43| $pie_height = 0.60 * $height;
44|
45| // gambar data
46| $iwarna = 0; $s = 0;
47| for ($i=0; $i < $N; $i++) {
48|     $legend = $data[$i][0];
49|     $value = $data[$i][1];
50|
51|     $e = $s + $value*$factor;
52|
53|     imagefilledarc($im, $xc, $yc, $pie_width, $pie_height,
54|                   $s, $e,
55|                   $item_colors[$iwarna],
56|                   IMG_ARC_PIE | IMG_ARC_EDGED);
57|
58|     $s = $e;
59|
60|     // pilih warna berikutnya
61|     $iwarna++;
62|     if ($iwarna == count($item_colors)) $iwarna = 0;
63| }
64|
65| // cetak judul di pojok kiri bawah
66| imagestring($im, 4,
67|            10, $height-20,
68|            $judul, $text_color);
69|
70| return $im;
71|
72|}
73|
74|// mari cetak sebuah grafik
75|$data = array(
76| array("jerry", 1000000),
77| array("kramer", 200000),
78| array("elaine", 200000),
79| array("george", 200000),
80| array("estelle", 25000),
81| array("frank", 25000),
82| array("newman", 25000));

```

```

84| header("Content-Type: image/png");
85| imagepng(grafik_kue(640, 480, $data,
86|                      "Proporsi Pendapatan Para Pemain"));
87|
88| ?>

```

## Resep 6-4: Membuat Grafik dengan JpGraph

Untuk aplikasi “betulan”, resep-resep sebelumnya terlalu sederhana. Biasanya dalam membuat grafik yang informatif dan cantik, perlu ada opsi untuk skala manual selain otomatis. Perlu juga ada label dan garis-garis pada skala. Kadang-kadang juga diinginkan ada grid untuk membantu melihat nilai item grafik. Lalu bagaimana jika ingin membuat grafik pie? Atau grafik 3 D? Bagaimana dengan legend, beberapa set data sekaligus, dua atau lebih tipe grafik dalam satu gambar, dan sebagainya?

Bisa dilihat bahwa GD terlalu rendah level-nya untuk kebutuhan seperti ini. Karena itulah orang membuat library di atas GD khusus untuk diagram grafik. Salah satu yang paling terkenal dan lengkap adalah **JpGraph**—sayang, tidak gratis untuk tujuan komersial. Beberapa fitur JpGraph, di antaranya mendukung TrueType Font, baik di Windows maupun Unix serta perataan teks di kiri, kanan atau center.

Untuk menginstalasi JpGraph, ambil tarball dari <http://www.aditus.nu/jpgraph/> atau dari CD buku ini, lalu ekstrak ke direktori yang diinginkan. Biasanya library disimpan di subdirektori /lib di bawah \$DOCUMENT\_ROOT. JpGraph membutuhkan library GD versi 2.x ke atas.

Berikut ini dua resep pendek untuk membuat grafik. Bandingkan dengan ketiga resep sebelumnya, betapa mudah dan singkatnya jika menggunakan JpGraph.

```

<?php
// grafik batang dengan JpGraph

include ("DOCUMENT_ROOT/lib/jpgraph/jpgraph.php");
include ("DOCUMENT_ROOT/lib/jpgraph/jpgraph_bar.php");

```

```

$datay=array(12,8,19,3,10,5);

$graph = new Graph(640,480,"auto");
$graph->SetScale("textlin");
$graph->img->SetMargin(40,30,20,40);

$bplot = new BarPlot($datay);
$bplot->SetFillColor('orange');

$bplot->value->Show();
$bplot->value->SetFormat('%d');
$bplot->value->SetFont(FF_FONT1,FS_BOLD);

$bplot->SetValuePos('center');
$graph->Add($bplot);

$graph->title->Set("Centered values for bars");
$graph->xaxis->title->Set("X");
$graph->yaxis->title->Set("Y");
$graph->title->SetFont(FF_FONT1,FS_BOLD);
$graph->yaxis->title->SetFont(FF_FONT1,FS_BOLD);
$graph->xaxis->title->SetFont(FF_FONT1,FS_BOLD);

$graph->Stroke();
?>

```

Grafik kue 3-D:

```

<?php
include ("DOCUMENT_ROOT/lib/jpgraph/jpgraph.php");
include ("DOCUMENT_ROOT/lib/jpgraph/jpgraph_pie.php");
include ("DOCUMENT_ROOT/lib/jpgraph/jpgraph_pie3d.php");

$data = array(20,27,45,75,90);

$graph = new PieGraph(350,200,"auto");
$graph->SetShadow();

$graph->title->Set("Grafik Kue 3D");
$graph->title->SetFont(FF_VERDANA,FS_BOLD,18);
$graph->title->SetColor("darkgreen");

```

```

$graph->legend->Pos(0.1,0.2);

$p1 = new PiePlot3d($data);
$p1->SetTheme("sand");
$p1->SetCenter(0.4);
$p1->SetAngle(30);
$p1->value->SetFont(FF_ARIAL,FS_NORMAL,12);
$p1->SetLegends(array(
    "Jan","Feb","Mar","Apr","May","Jun",
    "Jul","Aug","Sep","Oct"));
$graph->Add($p1);
$graph->Stroke();
?>

```

## Resep 6-5: Membuat Thumbnail

```

1|<?
|
3|//
4|// thumbnail.php
5|//
|
7|// Berikan sebuah file atau URL, maka fungsi ini akan
  |mengambil gambar dan
8|// mengembalikan gambar yang telah diperkecil.
|
10|function thumbnail($f, $width, $height) {
11|    $fp = fopen($f, "rb"); $content = "";
12|    while (1) {
13|        $chunk = fread($fp, 64*1024);
14|        $content .= $chunk;
15|        if (!strlen($chunk)) break;
16|    }
17|    fclose($fp);
|
19|    $im = imagecreatefromstring($content);
20|    $sizes = getimagesize($f);
|
22|    // hitung ukuran

```

```

23| if ($width == 0) {
24|     $width = $sizes[0] * ($height/$sizes[1]);
25| } elseif ($height == 0) {
26|     $height = $sizes[1] * ($width/$sizes[0]);
27| }
28|
29| $im2 = imagecreate($width, $height);
30| imagecopyresized($im2, $im, 0, 0, 0, 0,
31|                 $width, $height, $sizes[0], $sizes[1]);
32| return $im2;
33|}
34|
35| if (!isset($which)) $which = 0;
36|
37| if ($which == 1) {
38|
39|     header("Content-Type: image/png");
40|     imagepng( thumbnail("arkadia.jpg", 50, 0) );
41|
42| } elseif ($which == 2) {
43|
44|     header("Content-Type: image/png");
45|     imagepng( thumbnail("arkadia.jpg", 0, 120) );
46|
47| } elseif ($which == 3) {
48|
49|     header("Content-Type: image/png");
50|     imagepng( thumbnail("arkadia.jpg", 120, 50) );
51|
52| } else {
53|
54|     echo "
55|         <h2>Beberapa variasi parameter thumbnail</h2>
56|
57|         <p>Gambar asli</p>
58|         <img src=arkadia.jpg>
59|
60|         <p>Lebar menjadi 50px, tinggi proporsional</p>
61|         <img src=$PHP_SELF?which=1>
62|
63|         <p>Tinggi menjadi 120px, lebar proporsional</p>

```

```

64|     <img src=$PHP_SELF?which=2>
65|
66|     <p>Lebar menjadi 120px, tinggi menjadi 50px</p>
67|     <img src=$PHP_SELF?which=3>
68|
69| ";
70|
71|}
72|
73|?>

```

Membuat thumbnail sangat sederhana yaitu dengan `imagecopyresized()` atau `imagecopyresampled()`. Pada resep di atas, fungsi `thumbnail()` memberikan kemudahan dalam menyebutkan ukuran. Jika `$width` diset 0 dan `$height` diset sebuah nilai, maka `$width` akan dihitung secara proporsional mengikuti `$height`.

Sebaliknya, jika `$height` diset 0 dan `$width` diset ke sebuah angka tertentu, maka tinggi gambar akan mengikuti lebar secara proporsional. Dan jika `$height` maupun `$width` diset, maka kedua ukuran akan dipatuhi, dengan konsekuensi hasil thumbnail dapat menjadi tidak proporsional.

Perhatikan penggunaan fungsi `imagecreatefromstring()`. Dibandingkan `imagecreatefromgif()`, `imagecreatefromjpeg()`, dsb., fungsi `imagecreatefromstring()` dapat mendeteksi sendiri isi gambar dalam format file apa. Sementara keluaran hasil thumbnail di resep selalu dalam format PNG. Selain dengan GD, thumbnail juga dapat dibuat dengan ekstensi atau library exif.

## Resep 6-6: Membuat Ikon (.ICO)

File .ICO adalah format gambar yang dipergunakan di Windows untuk ikon-ikon kecil. File .ICO adalah format tersendiri yang berbeda dengan Windows BMP, karena sebetulnya dapat memuat lebih dari satu ikon dalam satu file .ICO.

Sejak Internet Explorer memperkenalkan dan mempopulerkan konsep favorite icon dengan meletakkan file di lokasi toplevel situs, `/favicon.ico`. Maka file ini nanti bisa dibaca oleh IE untuk membuat setiap situs di daftar bookmark dapat memiliki ikonnya sendiri yang unik. Dan kadang-kadang programmer atau pemilik situs yang menggunakan Unix atau Linux pun perlu menghasilkan file ini. Untuk membuat .ICO di Windows bisa digunakan program bawaan **Paint** atau shareware-shareware yang bisa diperoleh di situs seperti Tucows.



Sayangnya, library GD belum mendukung format .ICO. Untunglah, ada library seperti NetPBM yang bisa mengonversi ke dan dari .ICO. Source code NetPBM dan biner untuk Windows disediakan di CD buku ini, atau bisa diambil sendiri dari <http://netpbm.sourceforge.net>. Di Windows, ekstraklah seluruh file .zip yang ada ke sebuah direktori. Misalnya C:\gnuwin32, sehingga seluruh file-file .EXE dari semua .zip berada di c:\gnuwin32\bin\, librari di c:\gnuwin32\lib\, dan seterusnya.

Untuk mengonversikan GIF atau PNG ke .ICO, pertama-tama konversikan dulu ke format .PNM, baru ke .ICO. Program-program giftopnm, pngtopnm, dan ppmtowinicon berada di c:\gnuwin32\bin.

```
$ giftopnm <inputfile.gif> | ppmtowinicon -output favicon.ico
```

```
$ pngtopnm <inputfile.png> | ppmtowinicon -output favicon.ico
```

Berikut ini sebuah resep skrip PHP yang menerima upload file .GIF atau .PNG lalu mengubahnya menjadi .ICO yang siap di-download.

```
1|<?
2|
3|//
4|// make-ico.php
5|//
6|
7|// lokasi program seperti giftopnm, dsb. di Windows, pakai
8|// "C:\\gnuwin32\\bin"
9|
10|$PROG_DIR = "/usr/bin";
11|
12|session_register('icon');
13|
14|if (isset($submit)) {
15|
16|    $tmpname = $_FILES['file1']['tmp_name'];
17|    $name = $_FILES['file1']['name'];
18|
19|    if (preg_match("/\\.gif/i", $name)) {
20|        $command = "$PROG_DIR/giftopnm $tmpname | ppmtowinicon";
21|        $icon = '$command';
22|        echo "<a href=$PHP_SELF/favicon.ico>Download icon</a>";
```

```
23|    } elseif (preg_match("/\\.png/i", $name)) {
24|        $command = "$PROG_DIR/pngtopnm $tmpname | ppmtowinicon";
25|        $icon = '$command';
26|        echo "<a href=$PHP_SELF/favicon.ico>Download icon</a>";
27|    } else {
28|        die("Harap masukkan file .GIF atau .PNG saja!");
29|    }
30|
31|} elseif (isset($PATH_INFO)) {
32|
33|    header("Content-Type: application/x-something");
34|    header("Content-Length: ".strlen($icon));
35|    echo $icon;
36|
37|} else {
38|
39|    echo '
40|
41|<h1>Menghasilkan .ICO dari .GIF/.PNG</h1>
42|<form enctype="multipart/form-data" method="POST">
43|<p>File input (GIF atau PNG):<br>
44|    <input name="file1" type="file">
45|<p><input type="submit" name="submit" value="Kirim">
46|</form>
47|
48|';
49|
50|}
51|
52|?>
```

Pada resep di atas, isi file .ICO yang telah digenerasi disimpan dalam variabel sesi, dengan tujuan semata-mata agar tidak perlu pusing menyimpan secara eksplisit ke dalam file. Pada baris 33, saya membuat Content-Type “asal” yang tidak dikenal, sebagai salah satu trik untuk memaksa browser IE menampilkan kotak dialog dan bukannya menampilkan langsung di dalam layout window.

Agar dapat dipakai sebagai ikon favorite atau untuk shortcut, spesifikasi ukurannya harus tertentu yaitu 16x16 pixel atau 32x32 atau 64x64. Dengan GD, tentunya mudah saja mengkonversi sebuah file .GIF/.PNG yang berukuran lebih besar menjadi berukuran untuk ikon. Silahkan dicoba sebagai latihan.

## Resep 6-7: Membuat Barcode

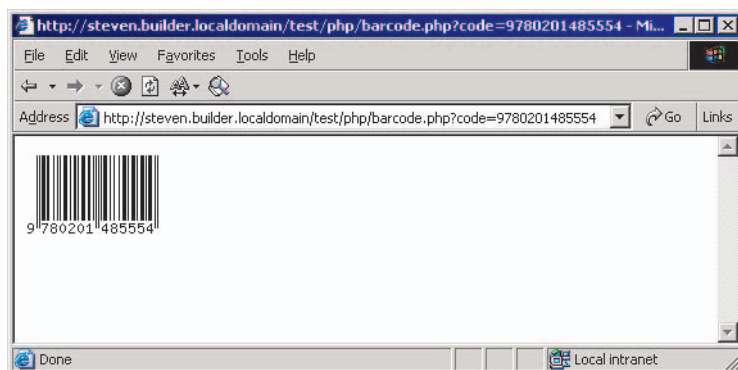
```
<?
require_once "Image/Barcode.php";

if (!isset($code)) die("Masukkan kode EAN");

Image_Barcode::draw($code, "ean13", "png");

?>
```

Image\_Barcode adalah paket PEAR. Argumen pertama adalah angka yang ingin digambar, argumen kedua adalah “ean13” atau “int25”, dan argumen ketiga dapat berupa “gif” atau “png”. Contoh keluaran bisa dilihat di Gambar 6-2.



Gambar 6-2. Barcode yang Menggunakan Standar EAN.