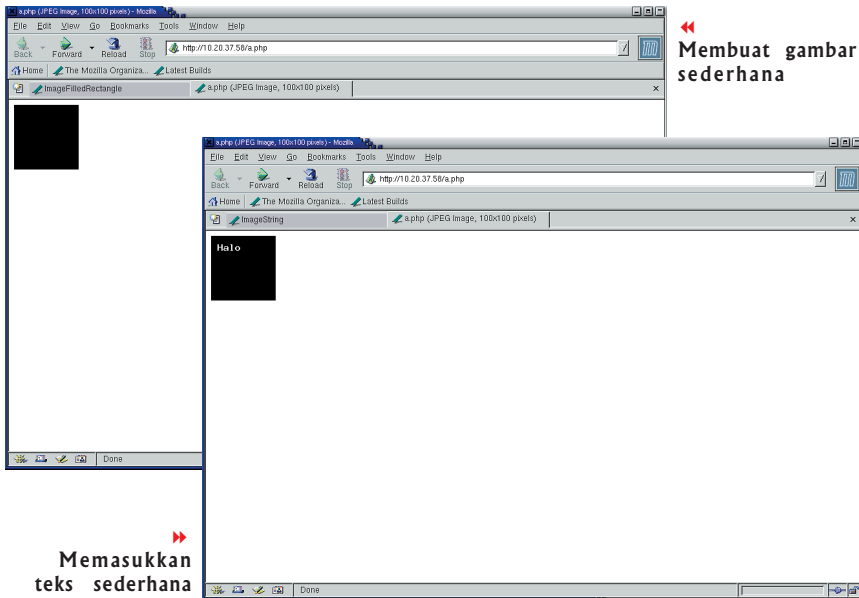


Membuat Gambar dengan PHP



« **Membuat gambar sederhana**

» **Memasukkan teks sederhana**

Tidak mesti menggunakan GIMP untuk membuat gambar sederhana, PHP pun bisa.

Beberapa hari raya telah berlalu, orang-orang merayakan dan bergembira. Tentunya tidak lupa untuk saling berkirim ucapan. Penjual kartu ucapan pun marak ditemui di mana-mana. Warung Internet pun meramai bersamaan dengan datangnya orang-orang yang ingin berkirim kartu ucapan elektronik lewat jasa-jasa yang ditawarkan lewat Internet. Ponsel semakin sering berbunyi menandakan SMS yang masuk dan berbagai kesibukan lainnya. Orang-orang merayakan dan bergembira, sambil tidak lupa untuk menularkannya untuk yang lain.

Kartu elektronik, jika kita perhatikan, pada umumnya berupa HTML, baik dinamis ataupun statis, yang akan menguraikan diri menjadi beberapa file apabila kartu ucapan Anda memiliki gambar.

Sekarang, bagaimana kalau kita membuat kartu ucapan dalam bentuk gambar? Dengan gambar, sang penerima dapat menyimpannya dalam satu file saja dan bisa membukanya kapan saja dengan berbagai pembuka gambar yang tersedia. Akan tetapi, kita tidak akan membuatnya dengan GIMP ataupun pengolah gambar lainnya,

kita akan membuatnya secara dinamis dengan PHP, dan bisa diakses lewat internet. Tertarik?

PHP, seperti yang telah diketahui, telah menjadi bahasa pemrograman di web yang sukses dan sangat banyak digunakan di berbagai website yang ada. Untuk membuat gambar, PHP Anda harus dilengkapi GD (<http://www.boutell.com>).

Membuat gambar sederhana

Perjalanan kita akan dimulai dengan membuat sebuah gambar sederhana berlatarbelakang warna hitam dengan ukuran sebesar 100x100pixel. Berbagai format gambar seperti JPEG, GIF, dan PNG didukung, tetapi artikel ini hanya akan membahas pembuatan gambar dengan format JPEG.

Langkah pertama adalah dengan menentukan jenis MIME yang akan dikirimkan ke *web browser*. Karena kita akan membuat gambar dengan format JPEG, maka jenis MIME yang dikirimkan adalah *image/jpeg*. Berikut ini adalah caranya:

```
header("Content-type: image/jpeg");
```

Langkah kedua adalah menentukan ukuran gambar, lebar, dan tinggi. Sebaiknya, kita juga menentukan kualitas kompresi gambar, mengingat JPEG adalah format file terkompresi. Kompresi yang digunakan di format JPEG adalah kompresi *lossy*, yang akan mengurangi kualitas gambar. Berikut ini adalah caranya:

```
$quality = 100;
```

```
$width = 100;
```

```
$height = 100;
```

Langkah ketiga adalah memanggil fungsi *ImageCreate* untuk membuat gambar. Fungsi tersebut akan mengembalikan satu nilai berupa pengenalan gambar. Berikut ini adalah caranya:

```
$image = imagecreate($width, $height);
```

Langkah keempat adalah mengalokasikan warna latar belakang. Untuk itu, digunakan fungsi *ImageColorAllocate*. Kita akan memberikan warna hitam dan untuk itu, kita akan mengisi nilai 0 ke setiap komponen RGB. Berikut ini adalah caranya:

```
$r = 0;
$g = 0;
$b = 0;
$back = imagecolorallocate($image, $r, $g, $b);
```

Langkah kelima adalah menampilkan image tersebut ke web browser dan sekaligus menghapus sumber daya memori yang digunakan. Berikut ini adalah caranya:

```
$imgfilename = "";
ImageJpeg($image, $imgfilename, $quality);
ImageDestroy ($image);
```

Arahkan web browser Anda ke *script* PHP tersebut dan Anda akan mendapatkan suatu gambar kotak hitam berukuran 100x100pixel.

Memasukkan teks

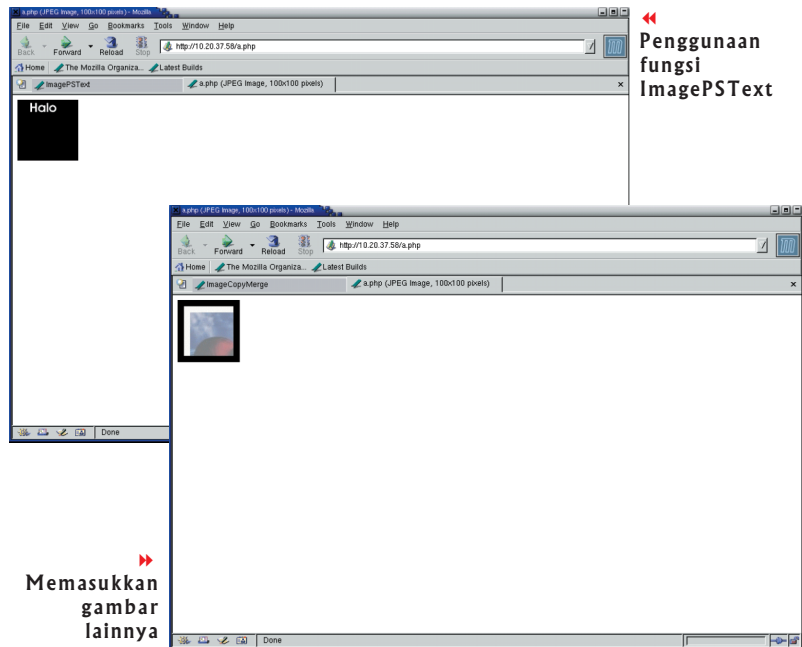
Sebuah gambar hitam berukuran 100x100 tentulah tidak akan menarik, apalagi jika digunakan sebagai kartu ucapan. Untuk itu, memasukkan sebuah teks pada umumnya diperlukan. Dan untuk itu, masih dengan sebuah kotak hitam, kita akan memasukkan sebaris teks ke dalamnya.

Salah satu cara untuk memasukkan teks yang termudah adalah dengan cara memanggil fungsi *ImageString* yang akan menulis teks horizontal dan fungsi *ImageStringUp* yang akan menulis teks vertikal. Berikut ini adalah cara penggunaan kedua fungsi tersebut:

```
$white = imagecolorallocate($image,255, 255, 255);
ImageString($image,1, 10, 10, "Halo", $white);
```

Baris pertama dimaksudkan untuk mengalokasikan suatu warna putih. Parameter kedua, ketiga, dan keempat dari fungsi *ImageColorAllocate* yang masing-masing bernilai 255 dimaksudkan sebagai nilai setiap komponen RGB untuk warna putih.

Baris kedua adalah cara pemanggilan fungsi *ImageString*. Parameter pertama adalah variabel untuk gambar yang dihasilkan dari fungsi *ImageCreate*. Parameter kedua adalah jenis *font* yang digunakan. Apabila bernilai 1, 2, 3, 4, atau 5, maka font internal akan digunakan. Parameter ketiga dan keempat adalah koordinat x,y untuk teks, dengan acuan koordinat x,y untuk kiri atas adalah 0,0. Parameter kelima adalah teks yang



akan dimasukkan dan parameter terakhir adalah parameter warna.

Jika kita mengarahkan web browser kita ke script yang memuat kode-kode tersebut, kita akan menjumpai sebuah kotak hitam 100x100pixel yang kini telah dilengkapi teks warna putih bertuliskan Halo, dengan ukuran yang cukup kecil. Ganti-gantilah parameter font internal pada fungsi *ImageString* untuk mendapatkan ukuran font yang berbeda.

Cara yang kita telah bahas tersebut adalah cara yang sederhana. Untuk memasukkan teks, kita juga bisa menggunakan fungsi *ImagePSText* yang akan menggambar teks di atas gambar menggunakan font *PostScript Type1*. Untuk itu, Anda harus memiliki font tersebut. Sebaiknya, *copy*-kan salah satu font dari direktori */usr/share/fonts/type1/gsfonts/* (sesuai dengan lokasi yang ditetapkan oleh distro Anda) ke direktori yang sama dengan script Anda. Sebagai contoh, kita akan menggunakan font dengan nama file *a010015l.pfb*.

Untuk mencoba, berikut ini adalah langkah-langkahnya:

```
$font = ImagePsLoadFont("a010015l.pfb");
ImagePSText($image, "Halo", $font, 20, $white,
$back, 20,20);
ImagePSFreeFont($font);
```

Baris pertama adalah pemanggilan fungsi untuk membuka suatu font. Baris kedua adalah inti yang akan kita bahas. Sementara baris ketiga adalah pemanggilan fungsi untuk membebaskan sumber

daya memori font yang telah dibuka dengan fungsi *ImagePsLoadFont*.

Fungsi *ImagePSText* adalah fungsi yang sangat kompleks. Setidaknya, fungsi ini dapat menerima sampai 12 parameter! Tetapi umumnya, untuk menulis sebuah teks horizontal, kita memberikan 8 parameter seperti contoh pada baris kedua tersebut.

```
array imagepstext ( int image, string text, int font, int  
size, int foreground, int background, int x, int y [, int  
space [, int tightness [, float angle [, int  
antialias_steps]]]])
```

Parameter pertama adalah variabel gambar yang dihasilkan dari fungsi *ImageCreate*. Parameter kedua adalah teks yang ingin ditulis. Parameter ketiga adalah variabel font yang dihasilkan dari fungsi *ImagePsLoadFont*. Parameter keempat adalah ukuran teks dalam satuan pixel. Parameter kelima adalah *foreground*, warna teks yang akan digambar. Parameter keenam adalah *background*, warna latar belakang, suatu warna di mana teks akan bekerja dengan dipengaruhi antialias.

Dalam hal ini, tidak ada pixel dengan warna latar belakang yang benar-benar digambarkan, sehingga latar belakang gambar tidak harus berupa warna monoton. Selanjutnya, parameter ketujuh dan kedelapan adalah koordinat x,y untuk posisi teks. Akan tetapi, koordinat ini akan mengacu kepada pixel kiri bawah karakter pertama. Hal ini tentunya berbeda dengan koordinat x,y pada fungsi *ImageString* yang mengacu kepada pixel kiri atas teks.

Selanjutnya, parameter kesembilan adalah parameter *space*, yang akan mengubah nilai default spasi di dalam teks. Nilai ini bisa berupa bilangan positif ataupun negatif dan akan ditambahkan ke nilai *default*. Parameter kesepuluh adalah *tightness*, yang mengizinkan untuk mengubah jumlah *white space* di antara karakter. Sama seperti *space*, nilai ini bisa berupa bilangan positif ataupun negatif dan akan ditambahkan ke nilai default. Parameter kesebelas adalah sudut penggambaran teks, dalam satuan derajat.

Dengan mengubah parameter ini, Anda dapat menentukan orientasi penggambaran teks. Parameter terakhir berhubungan dengan antialias. Gunakan nilai 4 sampai 16. Nilai yang lebih tinggi direkomendasikan untuk teks dengan ukuran lebih kecil dari 20. Dengan ukuran yang lebih besar, gunakan nilai 4.

Memasukkan gambar lain

Orang-orang bijak mengatakan, "Sebuah gambar bisa mewakili banyak tulisan". Jika Anda setuju,

tentunya Anda tidak akan puas hanya dengan memasukkan teks ke dalam gambar yang kita buat. Akan lebih baik, jika gambar yang kita buat bisa juga mengandung gambar lain.

Berikut ini kita akan membuat gambar yang bisa mengandung gambar lainnya, dengan syarat, gambar lainnya tersebut telah tersedia di file sistem. Sama seperti pembuatan gambar, kita juga bisa menggunakan berbagai format gambar yang ada. Akan tetapi, kita tetap akan menggunakan gambar dengan format JPEG, sehingga gambar yang harus dimasukkan ke dalam gambar yang kita buat harus berada di dalam format JPEG.

Langkah-langkahnya adalah:

```
$image2 = ImageCreateFromJPEG("./meikarke1.jpg");  
ImageCopyMerge( $image, $image2, 10, 10, 0, 0, 80,  
80, 50);
```

Baris pertama berfungsi untuk mengambil suatu gambar dari file sistem. Dalam hal ini, kita mengambil gambar dengan format JPEG. Selanjutnya, pada baris kedua, kita akan menggabungkannya dengan gambar yang sedang dibuat. Fungsi *ImageCopyMerge* akan meng-copy dan menggabungkan gambar.

Parameter pertama pada fungsi *ImageCopyMerge* adalah gambar tujuan, dalam hal ini gambar yang sedang kita buat. Parameter kedua adalah gambar yang kita ambil dari file sistem. Parameter ketiga dan keempat adalah koordinat x,y tujuan peletakan gambar. Parameter kelima dan keenam adalah koordinat x,y sumber pada gambar yang ingin dimasukkan.

Parameter ketujuh dan kedelapan adalah ukuran lebar dan tinggi gambar di file sistem yang akan diambil. Parameter kesembilan, parameter terakhir akan berfungsi sebagai penentu intensitas peng-copy-an dan penggabungan gambar. Apabila nilai 0 diberikan, maka gambar yang diambil tidak akan terlihat. Apabila nilai 100 diberikan, maka gambar yang diambil akan terlihat paling jelas.

Setelah gambar selesai dibuat, jangan lupa untuk membebaskan sumber daya yang digunakan oleh kedua gambar dengan fungsi *ImageDestroy*.

Demikianlah pengenalan kita dengan pembuatan gambar dengan PHP. Tentunya, kita hanya membahas kulitnya saja. Masih banyak fungsi lain yang bisa digunakan untuk menghasilkan gambar yang semakin menarik. Selanjutnya, masing-masing dari kita pun bisa membuat layanan kartu elektronik dalam bentuk gambar. 🐧

Noprianto (noprianto@infolinux.co.id)