6.1. Penanganan Kesalahan (error Handling)

Ada empat jenis kesalahan dan warning pada PHP. Yaitu :
1 - Normal Function Errors
2 - Normal Warnings
4 - Parser Errors
8 - Notices (peringatan yang dapat anda abaikan, tetapi dapat meninggalkan bug pada kode anda)
Keempat error diatas dijumlahkan membentuk suatu error reporting level. Secara defaultnya adalah 7 dimana 1 + 2 + 4, atau semuanya kecuali notice. Level ini dapat diganti pada file php.ini dengan suatu error reporting directive. Dapat juga ditentukan pada Apache di file httpd.conf dengan php_error_reporting directive atau ditentukan pada runtime dengan suatu script menggunakan fungsi error_reporting() .
Semua Ekspresi PHP dapat juga dipanggil dengan awalan "@", yang mana akan mematikan reporting error untuk ekspresi tertentu. Jika suatu error terjadi dan fasilitas track_errors diaktifkan, anda dapat menemukan pesan kesalahan pada global variabel $php_errormsg.

6.2. Membuat Gambar GIF

PHP tidak hanya dapat menghasilkan output HTML, tetapi juga dapat digunakan untuk membuat file gambar GIF. Untuk melakukan hal ini, PHP harus dikompilasi dengan GD Library agar dapat berjalan dengan baik..

```
<?php
   Header("Content-type: image/gif");
   $string=implode($argv," ");
   $im = imagecreatefromgif("images/button1.gif");
   $orange = ImageColorAllocate($im, 220, 210, 60);
   $px = (imagesx($im)-7.5*strlen($string))/2;
   ImageString($im,3,$px,9,$string,$orange);
   ImageGif($im);
   ImageDestroy($im);
?>
```

Contoh diatas harus dipanggil dari suatu halaman dengan tag seperti: <img src="button.php?text">. Script button.php diatas akan menerima "text" dan menimpakannya keatas image dasar "images/button1.gif" serta mengirim image hasil tersebut. Hal ini cocok untuk mencegah pembuatan ulang tombol image setiap kali anda perlu menganti tulisan pada tombol. Dengan metode diatas dapat dihasilkan tombol-tombol secara dinamis.

6.3. HTTP authentication dengan PHP

HTTP Authentication pada PHP hanya dapat berfungsi ketika dijalankan sebagai module pada Apache. Sebagai Apache module, PHP script dapat menggunakan fungsi Header() untuk mengirim pesan "Authentication Required" ke browser klien yang akan memunculkan suatu jendela input untuk Username/Password pada browser klien. Setelah user mengisi username dan password, URL dari PHP script tersebut akan dipanggil kembali disertai dengan variabel, $PHP_AUTH_USER, $PHP_AUTH_PW dan $PHP_AUTH_TYPE yang masing-masing berupa user name, password dan authentikasi yang telah dimasukkan. Dalam hal ini hanya berupa authentikasi dasar.
Berikut ini adalah contoh script yang akan memaksakan suatu authentikasi dari klien:

```
<?php
 if(!isset($PHP_AUTH_USER)) {
   Header("WWW-Authenticate: Basic realm=\"My Realm\"");
   Header("HTTP/1.0 401 Unauthorized");
   echo "Text to send if user hits Cancel button\n";
   exit;
 } else {
   echo "Hello $PHP_AUTH_USER.<P>";
   echo "You entered $PHP_AUTH_PW as your password.<P>";
 }
?>
```

Sebagai penganti dari hanya mencetak $PHP_AUTH_USER dan $PHP_AUTH_PW pada kode diatas, anda dapat mengantinya menjadi memeriksa kebenaran username dan password yang dimasukkan dengan mengirimnya ke suatu query terhadap database, atau mencari data user tersebut dalam suatu dbm file.
Untuk mencegah seseorang melakukan tindakan seakan-akan password telah ter-authentikasi dengan suatu mekanisme luar, maka variabel PHP_AUTH tidak akan diset jika suatu autentikasi luar dilakukan terhadap halaman tersebut. Dalam hal ini, variabel $REMOTE_USER dapat digunakan untuk mengindentifikasi apakah authentikasi dilakukan user secara eksternal.
Contoh. HTTP Authentication example forcing a new name/password

```
<?php
 function authenticate() {
   Header( "WWW-authenticate: basic realm='Test Authentication System'");
   Header( "HTTP/1.0 401 Unauthorized");
   echo "You must enter a valid login ID and password to access this resource\n";
   exit;
 }
 if(!isset($PHP_AUTH_USER) || ($SeenBefore == 1 &&
!strcmp($OldAuth, $PHP_AUTH_USER)) )
{
   authenticate();
 }
 else {
   echo "Welcome: $PHP_AUTH_USER<BR>";
   echo "Old: $OldAuth";
   echo "<FORM ACTION=\"$PHP_SELF\" METHOD=POST>\n";
   echo "<INPUT TYPE=HIDDEN NAME=\"SeenBefore\" VALUE=\"1\">\n";
   echo "<INPUT TYPE=HIDDEN NAME=\"OldAuth\" VALUE=\"$PHP_AUTH_USER\">\n";
   echo "<INPUT TYPE=Submit VALUE=\"Re Authenticate\">\n";
   echo "</FORM>\n";
}
?>
```

Catatan, hal tersebut tidak dapat mencegah akses langsung ke halaman yang tidak membutuhkan authentikasi.
Juga perlu dicatat bahwa hal ini tidak akan berfungsi pada Microsoft's IIS server dan CGI versi PHP karena merupakan keterbatasan dari IIS.

6.4. Mendukung HTTP cookie

PHP secara transparan mendukung HTTP cookies. Cookies adalah suatu mekanisme untuk menyimpan data pada remote browser dan selanjutnya digunakan untuk mengenali pengunjung. Anda dapat membuat cookies dengan menggunakan fungsi setcookie(). Cookies adalah bagian dari HTTP header, maka fungsi SetCookie() harus dipanggil sebelum output lainnya dikirim ke browser. Hal ini memiliki batasan yang sama seperti fungsi Header().
Setiap cookies yang dikirim kepada anda dari klien secara otomatis diubah ke variabel PHP sama seperti metode GET dan POST. Jika anda ingin memasukkan banyak nilai pada suatu cookie tunggal, tambahkan saja [] pada nama cookie. Untuk jelasnya lihat pada fugnsi setcookie().

6.5. Mendukung upload File

PHP dapat menerima upload file yang dilakukan dari browser yang sesuai dengan RFC-1867 (termasuk Netscape Navigator 3, dan seterusnya keatas , Microsoft Internet Explorer 3 dengan patch dari Microsoft, dan seterusnya). Kemampuan ini memungkinkan anda untuk melakukan upload teks file maupun binari. Dengan melakukan authentikasi dan fungsi-fungsi manipulasi file, anda memiliki kendali penuh untuk mengatur siapa saja yang boleh melakukan upload dan apa yang akan dilakukan setelah file tersebut selesai diupload.

Berikut ini adalah kode HTML untuk membuat suatu form upload file:

```
<FORM ENCTYPE="multipart/form-data" ACTION="_URL_" METHOD=POST>
  <INPUT TYPE="hidden" name="MAX_FILE_SIZE" value="1000">
  Send this file: <INPUT NAME="userfile" TYPE="file">
  <INPUT TYPE="submit" VALUE="Send File">
</FORM>
```

URL harus menunjuk ke suatu file php. Field hidden MAX_FILE_SIZE harus mengawali field file input yang merupakan nilai dari ukuran maksimal file yang diperbolehkan. Nilai ini dinyatakan dalam byte. Pada file tujuan ini, variabel berikut akan didefinisikan sesaat setelah upload berhasil dilakukan:
$userfile - nama file sementara dari file yang berhasil diupload dan disimpan pada server.
$userfile_name - nama file asli pada sistem pengirim.
$userfile_size - ukuran file upload dalam byte.
$userfile_type - type mime dari file jika browser menyediakan informasi ini. Suatu contoh misalnya "image/gif".

Perlu dicatat bahwa bagian "$userfile" dari variabel diatas adalah nama yang berasal dari INPUT field dari TYPE=file pada upload form. Pada contoh form upload diatas, kita pilih namanya adalah "userfile".
File secara default file upload akan ditempatkan pada suatu temporary directory di server. Hal ini dapat diganti dengan mengubah variabel lingkungan TMPDIR dimana PHP dijalankan dengan fungsi PutEnv().
PHP script harus menentukan apa yang harus dilakukan terhadap hasil upload. Sebagai contoh anda dapat menggunakan variabel $file_size untuk memeriksa ukuran dari file dan menghapusnya jika terlalu besar atau terlalu kecil. Demikian juga $file_type untuk jenis file. Anda harus menghapus file tersebut dari directory temporary atau memindahkannya ke tempat yang sesuai.
File tersebut akan otomatis dihapus dari directory temporary pada akhir dari request jika file tersebut tidak dipindahkan atau diganti nama.
Perlu dicatat bahwa CERN httpd tidak mendukung fasilitas upload file.

6.6. Mendukung Penggunaan Remote File

PHP dapat digunakan untuk mengelola file yang ada di Web server, dan kemudian menampilkannya di browser.
Berikut adalah contoh bagaimana menampilkan judul dari sebuah halaman web yang ada disuatu Webserver
Contoh. Getting the title of a remote page

```
<?php
  $file = fopen("http://www.php.net/", "r");
  if (!$file) {
    echo "<p>Unable to open remote file.\n";
    exit;
  }
  while (!feof($file)) {
    $line = fgets($file, 1024);
    /* This only works if the title and its tags are on one line. */
    if (eregi("<title>(.*)</title>", $line, $out)) {
      $title = $out[1];
      break;
    }
  }
  fclose($file);
?>
```

Selain dapat menampilkan, anda juga dapat menulis kedalam file dengan menggunakan FTP selama anda terhubung ke jaringan dengan user dan akses yang diperbolehkan dan file yang akan ditulis harus belum ada.
Contoh. Storing data on a remote server

```
<?php
  $file = fopen("ftp://ftp.php.net/incoming/outputfile", "w");
  if (!$file) {
    echo "<p>Unable to open remote file for writing.\n";
    exit;
  }
  /* Write the data here. */
  fputs($file, "$HTTP_USER_AGENT\n");
  fclose($file);
?>
```

6.7. Penanganan Koneksi

Berikut ini diaplikasikan pada PHP version 3.0.7 dan seterusnya
Secara internal pada PHP suatu status koneksi dapat diatur. Ada 3 status yang mungkin:
0 - NORMAL
1 - ABORTED
2 - TIMEOUT

Status NORMAL aktif ketika PHP script berjalan dengan normal.
Jika remote klien melakukan pemutusan dengan klik pada tombol stop, maka status flag ABORT di aktifkan

Jika batas waktu yang ditentukan terlewati (lihat set_time_limit()) maka status flag TIMEOUT diaktifkan.
Anda dapat memutuskan apakah pemutusan diakibatkan oleh pemakai atau karena time out, dan kadang-kadang suatu script harus diselesaikan walaupun pemakai melakukan pemutusan. Secara defaultnya script akan dihentikan kalau pemakai melakukan pemutusan. Perlakukan ini dapat ditentukan dengan direktif ignore_user_abort pada file php.ini atau dapat juga ditentukan pada direktif php_ignore_user_abort di file Apache .conf atau dengan fungsi ignore_user_abort().
Script anda dapat juga dihentikan dengan suatu built-in script timer. Time out default adalah 30 detik, hal ini dapat diganti pada max_execution_time php.ini directive atau the corresponding php_max_execution_time Apache .conf directive atau dengan fungsi set_time_limit(). Ketika waktu tercapai script akan dibatalkan dan klien didisconnect, jika fungsi shutdown ada maka akan diaktifkan. Pada fungsi shutdown anda dapat memeriksa penyebab fungsi tersebut diaktifkan dengan memanggil fungsi connection_timeout(). Kalau dikembalikan true berarti diaktifkan karena timeout.