# Membangun File Manager Berbasis Web dengan PHP


**Mari bersama-sama membangun file manager berbasis web dengan PHP. File manager tersebut dapat dikembangkan menjadi semacam kontrol panel yang sering ditemui pada layanan *web hosting*.**

Andi adalah staf TI sebuah perusahaan perdagangan PT Dagang Siang Malam. Karena semakin maraknya penggunaan Internet, departeman TI perusahaan tersebut lantas berpendapat bahwa mereka harus memiliki situs dengan nama *domain* sendiri, kemudian syukur-syukur dapat menerapkan *e-commerce* suatu hari nanti. Mereka juga berpendapat bahwa memiliki situs pribadi akan memperjelas eksistensi mereka di dunia perdagangan nusantara. Andi pun melakukan hosting pada sebuah web hosting terkemuka. Untuk mengatur hosting-nya, Andi dapat menggunakan *control panel* yang diproteksi dengan *password*.

Kini, perusahaan tersebut memiliki sebuah situs pribadi dengan domain *dagang-siang-malam.com.* Pengembangan situs pun dilakukan. Perusahaan yang relatif konvensional tersebut perlahan mulai menapaki dunia Internet. Hampir semua staf penting juga diberikan fasilitas untuk memiliki homepage masing-masing. Departemen TI Dagang Siang Malam menerapkan kebijakan bahwa setiap staf yang berhak memiliki homepage akan diberikan alamat *http://www.dagang-siang-malam.com/<nama_staf>/.*

Minggu-minggu pertama, Andi pun sibuk melayani para staf yang akan melakukan *upload* file ke direktori masing-masing. Apa boleh buat. Control panel yang diberikan oleh perusahaan hosting dapat digunakan untuk mengatur situs perusahaan secara keseluruhan, dan sangat tidak mungkin untuk memberikan password control panel tersebut kepada setiap staf yang akan mengatur homepage masing-masing. Terlalu riskan, karena bisa saja situs dan data perusahaan juga ikut berubah. Andi pun semakin sibuk, karena para staf tersebut juga semakin sibuk untuk berlomba memiliki situs terindah. Setiap ketemu dengan para staf tersebut, Anda harus siap untuk meng-upload file-file mereka, menghapus file-file mereka, dan tugas lainnya yang banyak memakan waktu.

Akhirnya Andi menyerah. Sebagai staf TI, pekerjaan utama Andi adalah membuat situs perusahaan, dan bukannya mengurus isi homepage para staf di perusahaan tersebut. Andi pun berniat untuk membuat sebuah file manager berbasis web yang dapat digunakan oleh para staf untuk mengatur file mereka masing-masing. Setiap staf akan diberikan password yang berbeda-beda. Dengan demikian, diharapkan urusan upload, hapus, dan lain sebagainya dapat dilakukan masing-masing. Karena Andi cukup sibuk, maka file manager yang dibuat tidak boleh memakan waktu pembuatan terlalu lama. Perancangan pun segera dilakukan.

## Pengaturan keamanan

Andi cukup sadar bahwa urusan pengaturan file melalui web bisa sangat berbahaya. Apabila perancangan tidak dilakukan dengan cukup cermat, maka bisa-bisa staf yang satu bisa menghapus file milik staf lain, dan hal-hal tersebut dapat membuat Andi lebih sibuk lagi. Akhirnya, Andi memutuskan bahwa file manager tersebut, yang kemudian dinamakan sebagai *webfileman* harus memiliki fitur-fitur keamanan berikut ini:

- Proses *login* dan *logout* yang memadai.
- Setiap staf tidak boleh masuk ke direktori staf lain.
- Setiap staf tidak boleh masuk ke direktori lain di sistem.
- Setiap staf hanya boleh melihat file dan direktori yang mereka miliki.
- Setiap staf hanya boleh menghapus file dan direktori yang mereka miliki.
- Setiap staf hanya boleh membuat file dan direktori di bawah *home directory* mereka masing-masing.
- Ukuran file yang di-upload harus dibatasi.
- Webfileman harus mampu memastikan bahwa setiap file dan direktori yang ingin diakses oleh staf harus merupakan miliknya sendiri, dan penipuan dengan menggunakan notasi .., ../../, ../../../../dir/../file dan sebagainya harus mampu ditangani.

Pada pengembangan selanjutnya, Andi juga berpendapat bahwa *logging* untuk setiap operasi harus dilakukan. Apabila suatu hari terdapat penyalahgunaan fasilitas ini, maka logging dapat digunakan.

## Persiapan database

Langkah pertama yang dilakukan adalah membuat sebuah database dengan spefisikasi sebagai berikut:

| Variabel | Nilai |
|---|---|
| Nama database | Webfileman |
| User dengan hak penuh | Webfileman, password webfileman |

Andi pun memberikan perintah berikut ini untuk membuat database tersebut:

```
CREATE DATABASE webfileman;

GRANT ALL PRIVILEGES ON webfileman.* TO webfileman@localhost IDENTIFIED BY 'webfileman';

FLUSH PRIVILEGES;
```

Setelah itu, Andi membuat sebuah tabel, yang sementara digunakan untuk menyimpan pasangan nama dan password staf. Tabel tersebut dinamakan sebagai webfileman_auth, dengan spefisikasi field sebagai berikut:

| Field | Keterangan |
|---|---|
| Username | Varchar (15), menyimpan nama staf |
| Passwd | Varchar (15), menyimpan password staf. Untuk saat ini, password tidak dienkrip. |

Berikut ini adalah perintah SQLnya:

```
CREATE TABLE webfileman_auth(
    username varchar(15),
    passwd varchar(15));
```

Selanjutnya, Andi mulai memasukkan data staf yang berhak memiliki homepage pribadi:

| Username | Password |
|---|---|
| jengkolman | jengkolman123 |
| drupadi | drupadi123 |
| sendi | sendi123 |

Berikut ini adalah perintah SQLnya:

```
INSERT INTO webfileman_auth VALUES
('jengkolman','jengkolman123');
INSERT INTO webfileman_auth VALUES
('drupadi','drupadi123');
INSERT INTO webfileman_auth VALUES
('sendi','sendi123');
```

Dengan demikian, urusan database pun selesai. Kini Andi akan melanjutkan ke tahap pembuatan direktori yang akan berfungsi sebagai home directory bagi masing-masing staf.

## Pembuatan direktori

Karena alamat staf berupa *http://www.dagang-siang-malam.com/<nama_staf>/*, maka direktori untuk masing-masing staf perlu disiapkan. Direktori-direktori tersebut di buat di dalam direktori utama tempat menyimpan file-file *www.dagang-siang-malam.com.*

Program mkdir dapat digunakan sebagai berikut untuk membuat direktori-direktori tersebut:

```
mkdir jengkolman
mkdir drupadi
mkdir sendi
```

## Persiapan permission

Andi menyadari bahwa file manager berbasis web yang akan dibuat memungkinkan para staf untuk membuat file dan direktori baru, kemudian menghapus file dan direktori tersebut. Tentunya, kondisi demikian akan bisa terjadi kalau direktori para staf tersebut dapat ditulisi oleh user yang menjalankan web server.

Pada perusahaan hosting di mana PT. Dagang Siang Malam meletakkan websitenya, pengaturan permission tersebut telah diatur, sehingga webfileman dapat dijalankan dengan mudah.

Namun, karena Andi membuat webfileman ini di dalam *desktop*-nya yang berisikan Red Hat Linux 9, maka beberapa pengaturan perlu dilakukan terlebih dahulu.

Yang pertama, user yang menjalankan Apache pada Red Hat Linux 9 adalah user apache. Saat ini, program webfileman dimiliki oleh user Andi. Andi pun memberikan sebagian kepemilikan direktori webfileman dan seluruh isinya kepda user apache dengan mengubah grup pemilik. Berikut adalah perintah yang dijalankan:

```
chgrp apache -R webfileman
```

Yang kedua, user apache harus memiliki kemampuan menulisi direktori webfileman tersebut. Dengan demikian, Andi memberikan hak tulis kepada user apache terhadap direktori webfileman tersebut:

```
chmod 775 -R webfileman
```

Untuk saat ini, Andi sangat menyadari bahwa pemberikan permission seperti ini cukuplah rentan. Namun, sekali lagi, Andi harus segera menyelesaikan program ini, karena berbagai tugas utama sudah mengantri untuk dikerjakan.

Sampai saat ini, seluruh isi direktori webfileman kini dapat ditulisi oleh user apache. Pengaturan file sistem pun selesai. Andi siap untuk masuk ke langkah berikutnya.

## Proses login dan logout

Andi ingin membuat webfileman ini sesederhana mungkin. Dalam artian, tidak ada pesan atau tulisan yang tidak perlu di layar. Proses login yang berhasil akan memperlihatkan file dan direktori mereka. Proses login yang gagal akan segera kembali ke halaman login. Begitu juga dengan proses logout. Setelah para staf melakukan operasi logout, maka halaman login kembali ditampilkan. Untuk proses login dan logout, Andi menggunakan *session*.

Pertama-tama, staf akan melihat halaman login.html yang berfungsi sebagai tempat memasukkan *username* dan *password*. Halaman ini jugalah yang akan ditampilkan apabila proses login gagal. Proses otentikasi akan dilakukan oleh halaman login.php. Apabila otentikasi berhasil, maka staf kan dibawa ke halaman staff.php.

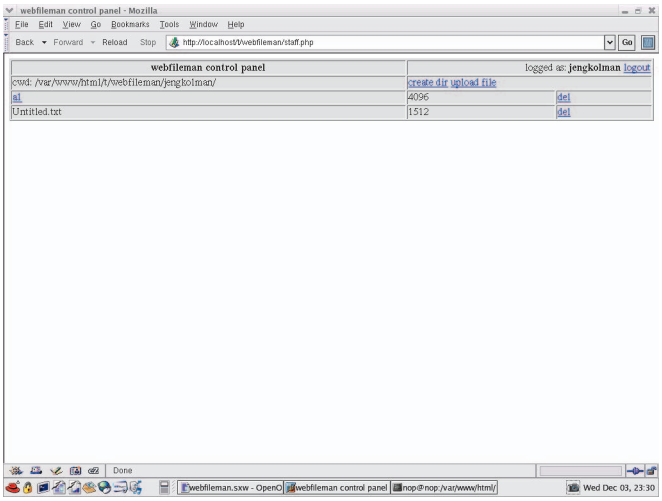
Apabila terdapat para staf yang nakal dan mencoba mengunjungi staff.php tanpa melakukan otentikasi, maka secara otomatis para staf tersebut akan menjumpai halaman login.html. File staff.php berfungsi sebagai halaman utama tempat para staf dapat melihat file dan direktori mereka serta melakukan berbagai operasi terhadap file dan direktori miliknya. File staff.php juga menampilkan lokasi direktori aktif dan menyediakan *link* untuk melakukan logout. Di halaman yang sama, staf akan menjumpai link untuk meng-upload file dan membuat direktori. Operasi sebenarnya untuk setiap file dan direktori tidak akan dilakukan oleh staff.php. File fs.php lah yang bertanggung jawab untuk hal tersebut.

Ketika staf melakukan klik pada link untuk logout, file logout.php akan dipanggil, dan segala session yang ada akan dibersihkan. Halaman login.html kembali ditampilkan.

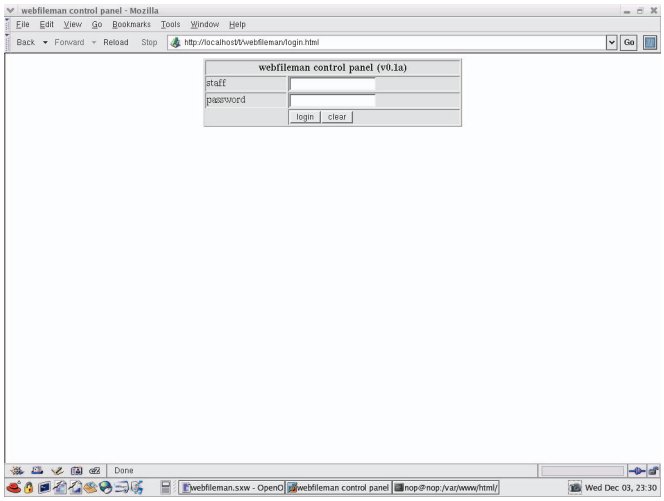
## Halaman utama: staff.php

File staff.php adalah file utama dari program webfileman ini. File ini bertanggung jawab untuk menampilkan berbagai informasi dan link yang berhubungan dengan direktori aktif user. File ini juga menyediakan link untuk logout. Kemudian, file ini juga bertanggung jawab untuk memberikan operasi tertentu kepada file dan direktori.

Sebuah file tidak dapat diklik di webfileman ini. Sebaliknya, direktori dapat diklik, dan apabila hal tersebut dilakukan, staf akan masuk ke dalam direktori tersebut. Sebuah link baru untuk kembali ke direktori orang tua pun segera ditampilkan oleh staff.php. Apabila sebuah direktori masing kosong, maka sebuah link untuk




Halaman login Webfileman.




Daftar file dan direktori.

menghapus direktori akan ditampilkan. Link untuk menghapus file akan selalu ditampilkan.

Pada setiap direktori, staff.php juga akan menampilkan link untuk meng-upload dan membuat direktori baru.

Apabila staf melakukan operasi klik pada sebuah direktori, maka staf tersebut akan dibawa masuk ke dalam direktori tersebut. Akan tetapi, file staff.php tidak bertanggung jawab untuk melakukan hal tersebut. Demikian juga ketika staf melakukan klik pada link untuk menghapus file. Staff.php juga tidak bertanggung jawab untuk menghapus file tersebut. Apa yang dilakukan oleh staff.php hanyalah menampilkan segala sesuatu yang boleh ditampilkan. Segala operasi file sistem menjadi tanggung jawab fs.php.

## Penanggung jawab operasi file sistem: fs.php

Setiap operasi file sistem akan ditangani oleh file ini. File ini akan menerima maksimal tiga parameter yang akan menentukan cara kerjanya. Yang pertama adalah *type* dari file dan direktori. Type dapat berupa file atau dir.

Yang kedua adalah *task*. Parameter task berisikan tugas yang akan dilakukan terhadap file atau direktori tersebut. Task yang mungkin untuk file adalah *delete* dan *upload*. Sementara, task yang mungkin untuk direktori adalah cd (*change directory*, harap diperhatikan, operasi ini tidak benar-benar masuk ke suatu direktori, melainkan hanya mengganti variabel penanda direktori aktif), *up* (kembali ke direktori orang tua, hanya mengubah penanda direktori aktif), create (membuat direktori) dan *del* (menghapus direktori kosong). Sebagai salah satu cara mencegah tindakan yang tidak bertanggung jawab, *task up* tidak akan berlaku untuk *home directory* para staf atau pun segala direktori di luar home directory staf tersebut.

Yang ketiga adalah filename apabila nilai parameter pertama adalah file. Parameter ini akan bernilai dirname apabila nilai parameter pertama adalah dir. Tidak semua task memerlukan parameter ini. Task up pada type dir misalnya, tidak membutuhkan parameter ini.

Katakanlah seorang staf memiliki file file1 dan direktori dir1. Link untuk menghapus file ini akan mengacu kepada fs.php?type=file&task=del&filename=file1. Sementara itu, apabila direktori dir1 masing kosong, maka link untuk menghapus direktori ini akan mengacu kepada fs.php?type=dir&task=del&dirname=dir1. Dan link untuk masuk ke dalam direktori ini akan mengacu kepada fs.php?type=dir&task=cd&dirname=dir1.

Hal tersebut disadari oleh Andi sangatlah rentan. Andi bukanlah seseorang yang mengenal betul keamanan sistem, tapi Andi sangat sadar bahwa staf jengkolman, teman dekatnya yang kebetulan juga berhak untuk memiliki homepage, adalah rekan kerja yang sedikit nakal.

Apabila jengkolman sukses melakukan authentikasi dan kemudian mengganti alamat URL menjadi fs.php?type=file&task=del&filename=../drupadi/file1 misalnya, maka file1 milik drupadi bisa saja terhapus. Atau jengkolman bisa saja mengarahkan URL menjadi fs.php?type=dir&task=cd&dirname=../drupadi/, dan file-file drupadi akan terlihat semuanya.

Kelemahan tersebut dapat ditangani dengan hanya mengambil basename dari dirname dan filename, sehingga hanya nama file dan direktori yang akan diambil. Dengan demikian, ../../drupadi/file1 akan menjadi file1. Namun, karena memiliki sedikit waktu luang, cara tersebut tidak digunakan oleh Andi. Sebagai gantinya, Andi mendaftar semua file dan direktori milik staf yang sedang aktif ketika fs.php dipanggil, dan setiap operasi terhadap file dan direktori hanya bisa dilakukan terhadap file milik staf tersebut. Semua file dan direktori tersebut akan dimasukkan ke dalam sebuah *array*.

File fs.php adalah program yang cukup mulia. Tidak ada satu pun tulisan yang dicetak oleh file ini. Karena, setelah semua tugas selesai, halaman aktif segera dipindahkan ke staff.php.

## Pembuatan direktori

Pembuatan direktori ditangani oleh file createdir.php. Apabila staf mengakses file ini tanpa melakukan authentikasi terlebih dahulu, maka halaman login.html kembali akan ditampilkan. Sederhananya, nama direktori akan dilewatkan ke fs.php dengan parameter type=dir&task=create&dirname=<nama_direktori>.

## Upload file

Karena keterbatasan waktu, Andi hanya membatasi staf untuk mengupload satu file dalam satu waktu. Artinya, apabila staf ingin meng-upload file kedua, maka upload.php harus dipanggil kembali. Atau link untuk upload file harus diklik sekali lagi.

Sama seperti createdir.php dan halaman lainnya, upload.php akan membawa staf kembali ke login.html apabila proses otentikasi belum dilakukan oleh staf tersebut.

## Variabel-variabel session

Apabila proses login berhasil, maka beberapa variabel session berikut ini akan diatur:

| Variabel | Deskripsi |
|---|---|
| $login | Berisikan nama staf yang login |
| $homedir | Berisikan home directory user, yaitu lokasi direktori aktif ditambah dengan nama staf. $homedir = getcwd() . "/" . $login . "/" |
| $rpath | Berisikan path yang sedang aktif. Pada awalnya, path aktif ini sama dengan $homedir. Selanjutnya, bisa dimodifikasi, terutama oleh fs.php dalam operasi cd dan up. |

## Source code dan penjelasan

### login.html (tanpa penjelasan):

```
<html>
<head>
<title>
webfileman control panel
</title>
</head>
<body>
<form action=login.php method=post>
<table border=1 valign=center align=
center width=40% bgcolor=#dddddd>
<tr>
<td colspan=2 align=center>
<b>webfileman control panel (v0.1a)</b>
</td>
</tr>
<tr>
<td>staff</td><td><input type=text
width=15 name=staff></td>
</tr>
<tr>
<td>password</td><td><input type
=password width=15 name=passwd>
</td>
</tr>
<tr>
<td></td>
<td><input type=submit value=login>
<input type=reset value=clear></td>
</tr>
</table>
</form>
</body>
</html>
```

### login.php:

```
<?
    mysql_connect('localhost','webfileman',
    'webfileman') or
            die('unable to connect to
            database server');

    mysql_select_db('webfileman');

    $q = "select passwd from webfileman
    _auth where username=\"$staff\";";
    $r = mysql_query($q);
    $l = mysql_fetch_array($r);
    $correct_passwd = $l[0];

    if (mysql_affected_rows() == 1 &&
    $passwd == $correct_passwd)
    {
        if (!session_is_registered('login'))
        {
            session_register('login');
            session_register('homedir');
            session_register('rpath');
            $login = $staff;
            $homedir = getcwd() . "/" .
            $login . "/";
            $rpath = $homedir;
            header("location: staff.php");
            }
        }
        else
        {
            header('location: login.html');
        }
?>
```

**Penjelasan:**

Pertama-tama, koneksi ke database akan dilakukan. Pencocokan username dan password pun dilakukan. Apabila gagal, maka halaman login.html akan ditampilkan. Apabila berhasil, maka beberapa variabel session akan diatur.

### logout.php

```
<?
        session_start();
        session_unset();
        session_destroy();
        header('location: login.html');
?>
```

**Penjelasan:**

File ini bertugas untuk membersihkan semua variabel session dan membawa staf ke halaman login.

### staff.php:

```
<?

    function filecount($path)
    {
        $fcount = 0;
        if ($handle = opendir($path))
        {
            while (false !== ( $file =
            readdir($handle)))
            {
                if ($file != "." && $file !=
                "..")
                {
                    $fcount++;
                }
            }
        }
            return $fcount;
    };


    session_start();
    if (!session_is_registered('login'))
    {
        header('location: login.html');
    }

?>


<html>
<head>
<title>
webfileman control panel
```

```
</title>
</head>
<body>
<table bgcolor=#dddddd width=100%
border=1>
<tr>
<td align=center><b>webfileman
control panel</b></td>
<td align=right colspan=2>logged as:
<b><? echo $login ?></b> <a
href=logout.php>logout</a></td>
</tr>
<tr>
<td>cwd: <? echo $rpath;?></td>
<td colspan=2> <a href=createdir.
php>create dir</a>
<a href=upload.php>upload file</a>
<?
   if ($rpath != $homedir)
   {
      echo "<a href=fs.php?type=dir&
      task=up>up</a>";
   }
?>
</td>
</tr>

<?
if ($handle = opendir($rpath))
{
   while (false !== ( $file = readdir
   ($handle)))
   {
      if ($file != "." && $file != "..")
      {
         echo "<tr>";
         if (is_file($rpath . $file))
         {
            echo "<td>$file</td>";
            echo "<td>";
            echo filesize($rpath . $file);
            echo "</td>";
            echo "<td>";
            echo "<a href=fs.php?type=
            file&task=del&filename=
            $file>del</a>";
         }else
         if (is_dir($rpath . $file))
         {
            echo "<td><a href=fs.php?
            type=dir&task=cd&dirname
            =$file>$file</a></td>";
            echo "<td>";
            echo filesize($rpath . $file);
            if (filecount($rpath . $file)
            == 0)
            {
               echo "</td>";
               echo "<td>";
               echo "<a href=fs.php?
               type=dir&task=
               del&dirname=$file>del</
               a>";
            }
         }
         echo "</td>";
         echo "</tr>";
      }
   }
}

closedir($handle);
?>
</table>
</body>
</html>
```

**Penjelasan:**

Ini adalah file yang terpanjang. Hampir semua tampilan adalah urusan file ini. Pada awal file, kita membuat fungsi yang berguna untuk mengembalikan jumlah file di dalam satu direktori. Apabila jumlahnya nol, maka direktori tersebut dapat dihapus.

Kemudian, direktori akan dibuka dan semua file dan direktorinya akan ditampilkan, beserta ukuran dan operasi yang bisa dilakukan terhadap file dan direktori tersebut. Lokasi direktori bergantung kepada variabel $rpath.

Baris kode yang panjang sebenarnya hanyalah menentukan tipe file dan direktori, kemudian menyiapkan operasi yang bisa dilakukan terhadap file dan direktori tersebut.

## fs.php:

```
<?


   session_start();
   if (!session_is_registered('login'))
   {
      header('location: login.html');
   };

   $myfiles = split("[[:space:]]",
   shell_exec("find $homedir"));

   if ($type == "dir")
   {
      if ($task == "cd")
      {
         if (in_array($rpath . $dirname,
         $myfiles))
         $rpath = $rpath . $dirname . "/";
      }
      if ($task == "up")
      {
         $temp = $rpath . $dirname;
         $temp = substr($temp,0,
         strlen($temp) - 1);
         if (in_array($temp . $dirname,
         $myfiles))
         if ($rpath != $homedir &&
         strlen($rpath) > strlen($homedir))
         {
            $rpath_a = explode('/',
            $rpath);
            array_splice($rpath_a,
            count($rpath_a) - 2);
            $rpath = implode ('/',
            $rpath_a);
            $rpath = $rpath . '/';
         }
      }

      if ($task == "create")
      {
         if ($dirname != "")
         mkdir($rpath . $dirname , 0755);
      }
      if ($task == "del")
      {
         if (in_array($rpath . $dirname,
         $myfiles))
         {
         rmdir($rpath . $dirname);
         }
      }
   }
   else
   if ($type == "file")
   {
      if ($task == "del")
      {
         if (in_array($rpath . $filename,
         $myfiles))
         {
            unlink($rpath . $filename);
         }
```

```
        }
        if ($task == "upload")
        {
            $uploadfile = $rpath . $filename_
            name;
            move_uploaded_file($filename,
            $uploadfile);
        }
    }

    header('location: staff.php');
?>
```

**Penjelasan:**
Untuk mendaftar semua file dan direktori milik seorang staf, kita menggunakan bantuan program *find* yang dijalankan dengan shell_exec(). Setiap operasi yang berbahaya akan diantisipasi terlebih dahulu. Antisipasi di sini tidak menjamin keamanan 100%.

### createdir.php (tanpa penjelasan):

```
<?
        session_start();
        if (!session_is_registered('login'))
        {
                header('location: login.html');
        }
?>


<html>
<head>
<title>
webfileman control panel
</title>
</head>
<body>
<form action=fs.php?type=
dir&task=create method=post>
<table bgcolor=#dddddd width=50%>
<tr>
<td>directory name</td>
<td>
<input type=text name=dirname
size=15>
<input type=submit value=create>
<input type=button value=cancel
onClick="location.href='staff.php';">
</td>
</tr>
</table>
</form>
</body>
</html>
```

### upload.php (tanpa penjelasan):

```
<?
        session_start();
        if (!session_is_registered('login'))
        {
                header('location: login.html');
        }
?>


<html>
<head>
<title>
webfileman control panel
</title>
</head>
<body>
<form enctype=multipart/form-data
action=fs.php?type=file&task=upload
method=post>
<input type=hidden name=
MAX_FILE_SIZE value=1048576>
<table bgcolor=#dddddd width=50%>
<tr>
<td>choose file (max size 1M)</td>
<td>
<input type=file name=filename
size=15>
<input type=submit value=upload>
<input type=button value=cancel
onClick="location.href='staff.php';">
</td>
</tr>
</table>
</form>
</body>
</html>
```

Andi sangat menyadari bahwa webfileman ini masih memiliki banyak kekurangan. Mulai dari tidak terdapatnya fasiltas untuk mengubah nama/memindahkan file, operasi *copy/cut/paste*, upload banyak file sekaligus, pembuatan file langsung, dan lain sebagainya. Untuk masalah keamanan sendiri, Andi merasa masih harus melakukan berbagai pengujian dan perbaikan. Ada *developer* yang tertarik untuk membantu Andi?

**Noprianto** *(noprianto@infolinux.co.id)*