



Materi Pembelajaran

- Penanganan Form Dasar
- Form Input
- Form Validasi
- File upload

Tujuan

Mahasiswa diharapkan mampu:

1. Memahami konsep form processing dan form upload
2. Mengetahui penerapan form processing dan form upload pada website dinamis

ULASAN TEORI

Pengenalan Form

Form merupakan salah satu elemen dalam HTML yang digunakan untuk menerima input dari pengunjung. Melalui form pengunjung sebuah halaman web dapat melakukan interaksi dengan web. Pengunjung dapat memasukkan data melalui elemen-elemen input seperti *textfield*, *checkbox*, *radio button*, *textarea*, *submit button*, *dropdown* dan lain-lain. Saat ini hampir seluruh website modern memerlukan *form* sebagai fitur utama. *Form* digunakan pada *form* pendaftaran, *form* login, *form* pendaftaran dan sebagainya. Sintaks dasar *form* adalah sebagai berikut:

```
<form>  
  Elemen input form  
</form>
```

Form membuat sebuah halaman website menjadi lebih interaktif dan dinamis. Dengan *form* pengguna dapat memasukkan data-datanya melalui elemen input form. Elemen-elemen input yang terdapat pada *form* antara lain:

1. *textfield*, adalah elemen yang digunakan untuk memasukkan teks dalam satu baris. Lebar karakter defaultnya adalah 20 karakter.
2. *password field*, adalah elemen yang digunakan untuk memasukkan satu baris teks dengan format *password/chiphertext*.
3. *checkbox*, adalah elemen yang dapat digunakan untuk memasukkan lebih dari satu pilihan.
4. *radio button*, adalah elemen yang digunakan untuk memasukkan pilihan (hanya dapat memilih salah satu pilih).
5. *textarea*, adalah elemen yang digunakan untuk memasukkan lebih dari satu baris teks.
6. *submit button*, adalah elemen untuk mengirimkan data dari halaman web ke server.

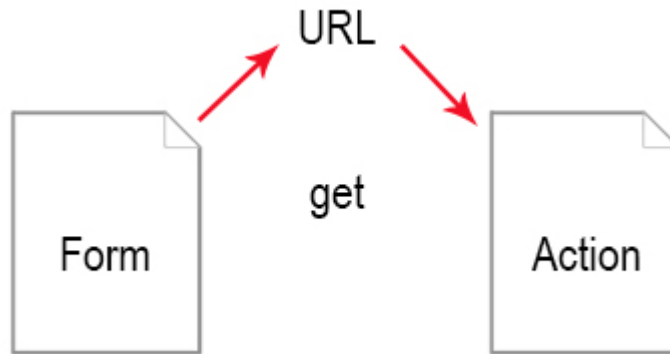
Form Handling

Untuk memproses suatu *form* digunakan 2 atribut pada elemen *form*, yaitu atribut *action* dan atribut *method*. Atribut *action* diisi dengan alamat halaman PHP yang digunakan untuk memproses data *form* tersebut. Atribut *method* akan menentukan bagaimana cara pengiriman data pada server. Terdapat 2 *method* yang dapat digunakan pada *form*, yaitu: POST dan GET. *Method* POST akan mengirimkan data atau nilai langsung ke *action* untuk ditampung, tanpa ditampilkan pada URL. Untuk menampung data atau nilai digunakan variabel `$_POST`. Gambar 1 menunjukkan ilustrasi penggunaan *method* POST:



Gambar 1. Ilustrasi penggunaan *method* POST

Berbeda dengan *method* POST, *method* GET akan menampilkan data atau nilai pada URL, kemudian ditampung oleh action. Variabel yang digunakan adalah \$_GET. Gambar 2 menunjukkan ilustrasi penggunaan *method* GET.



Gambar 2. Ilustrasi penggunaan *method* GET

Data atau nilai yang dikirimkan menggunakan *method* POST tidak terbatas, sedangkan data atau nilai yang dikirim menggunakan *method* GET tidak boleh lebih dari 2047 karakter. *Method* POST digunakan untuk mengirimkan data yang bersifat penting dan data yang tidak boleh diketahui orang lain, seperti *password* dan sebagainya.

Fungsi isset()

Nilai yang telah diinput oleh user atau pengunjung web, tidak bisa begitu saja di simpan langsung ke dalam database. Karena kita tidak tahu apakah nilai tersebut telah sesuai dengan nilai yang kita kehendaki. Misalkan apakah nilai tersebut harus berupa angka, atau hanya bisa berupa huruf, atau apakah hanya bisa diinput dalam range tertentu saja.

Validasi pertama yang bisa dilakukan adalah memeriksa apakah elemen-elemen form tersebut tersedia atau tidak. Fungsi `isset()` digunakan untuk memvalidasi apakah elemen form telah didefinisikan atau telah di set sebelumnya. Fungsi `isset()` bernilai `true` jika sebuah variabel telah didefinisikan dan `false` jika variabel tersebut belum dibuat.

Fungsi empty()

Ketika mengisi form ada beberapa informasi yang harus ada atau wajib diisi oleh user, misalnya informasi username. Sebelum mengirim file atau proses submit file, diperlukan proses validasi untuk memeriksa apakah sebuah elemen form sudah terisi atau belum. Dalam PHP untuk memvalidasi apakah elemen form sudah terisi atau belum digunakan fungsi `empty()`. Fungsi `empty()` akan bernilai `true` jika variabel tersebut belum diisi, dan bernilai `false` jika variabel tersebut telah terisi. Dengan menggunakan stuktur IF dan fungsi `empty()`, kita bisa membuat logika validasi objek form mana saja yang dianggap perlu (harus diisi) dan mana yang boleh dikosongkan, kemudian menampilkan pesan error yang sesuai.

Fungsi preg_match()

Validasi form selanjutnya adalah memastikan bahwa data yang diinput oleh user telah sesuai. Misalnya untuk nama harus diberi input berupa huruf bukan angka dan email harus memenuhi aturan penulisan email, yaitu "alamatemail@mail.com". Di dalam PHP terdapat fungsi yang dapat digunakan dalam manipulasi *regular expression* (regex), yaitu fungsi `preg_match()`. Fungsi `preg_match()` digunakan untuk mencocokkan pola dalam data string. Fungsi `preg_match()` akan mengembalikan data bernilai `TRUE` jika terdapat pola yang sesuai dan `FALSE` jika tidak ada pola yang sesuai. PHP memiliki fungsi built-in yang khusus digunakan untuk validasi alamat email, yaitu `filterval()`.

Penggunaan file

File dapat digunakan sebagai alternatif menyimpan data berukuran kecil yang dimasukkan user melalui form. File juga banyak digunakan untuk menyimpan data-data yang akan digunakan untuk keperluan konfigurasi sistem. PHP menyediakan fasilitas untuk mengakses file, seperti membaca, merubah, dan menyimpan data pada file.

Sebelum mengakses sebuah file, maka kita harus membuka file yang akan kita proses terlebih dahulu. PHP menyediakan fungsi *built-in* untuk melakukan pengaturan pada file salah satunya adalah untuk membuka file, yaitu `fopen()`. Sintak penggunaan `fopen()` adalah sebagai berikut:

```
<?php
    $file_handler = fopen(namaFile, mode);
?>
```

Parameter mode bisa diisi dengan kode program berikut ini:

Mode	Keterangan
r	Membuka file untuk proses pembacaan (<i>read</i>)
r+	Membuka file untuk proses pembacaan (<i>read</i>) dan penulisan (<i>write</i>)
w	Membuka file untuk proses penulisan (<i>write</i>)
w+	Membuka file untuk proses penulisan (<i>write</i>) dan pembacaan (<i>read</i>)
a	Membuka file untuk proses penambahan data (<i>append</i>)
a+	Membuka file untuk proses pembacaan (<i>read</i>) dan penambahan data (<i>append</i>)

Fungsi fread() digunakan untuk membaca data dari file yang telah dibuka. Parameter fungsi fread adalah nama file yang akan dibaca dan jumlah byte maksimum untuk dibaca. Berikut adalah sintak membaca file dengan menggunakan fread():

```
<?php
    fread($file_handler, filesize());
?>
```

Fungsi fclose() digunakan untuk menutup file. PHP menyediakan fungsi untuk menutup file, yaitu fclose(). Sintak untuk menutup file dengan fclose() adalah sebagai berikut:

```
<?php
    fclose($file_handler);
?>
```

File Upload

Proses upload adalah menyalin file dari direktori komputer lokal (*client*) ke komputer server. Banyak pemanfaatan dari metode upload ini, seperti untuk mengirim file foto dari *client* ke sistem agar bisa tampil pada halaman web. Atribut form untuk upload sedikit berbeda dengan form untuk data biasa, yang membedakan adalah adanya tambahan atribut enctype= "multipart/form-data". Tag <input> di set menggunakan atribut type = "file".

Untuk menerima file digunakan \$_FILES[nama_field][atribut], berikut adalah isi atributnya:

- tmp_name = berisi path temporary dimana file hasil proses upload disimpan di server
- name = berisi path asli dari file yang diupload oleh user.

Inti dari proses upload pada PHP adalah berada pada penerapan fungsi built-in yaitu:

- move_uploaded_file() atau copy(), yang berfungsi memindahkan dari folder temporary ke folder tujuan upload

LANGKAH-LANGKAH PRAKTIKUM

Praktikum 1: Form Handling

Langkah	Keterangan
1	<p>Buat file baru dengan nama form.html di dalam folder minggu7, kemudian ketikkan kode berikut:</p> <pre> 1 <html> 2 <head> 3 </head> 4 <body> 5 <form action="postForm.php" method="post"> 6 <label>Username: </label> 7 <input type="text" name="username" placeholder="username">
 8 <label>E-mail: </label> 9 <input type="text" name="email" placeholder="e-mail">

 10 <input type="submit"> 11 </form> 12 </body> 13 </html> </pre>
2	<p>Buat file baru dengan nama postForm.php di dalam folder minggu7, kemudian ketikkan kode berikut:</p> <pre> 1 <html> 2 <head> 3 </head> 4 <body> 5 Welcome <?php echo \$_POST["username"]; ?> !!
 6 Your email address is: <?php echo \$_POST["email"]; ?> 7 </body> 8 </html> </pre>
3	<p>Simpan file dan jalankan kode program dengan mengetikkan localhost/minggu7/form.html</p>
4	<p>Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu! (soal no 1)</p>
5	<p>Buatlah file baru dengan nama form_1.html dan salin kode program pada langkah ke-1. Modifikasi kode program dengan mengubah atribut form pada baris ke-5 menggunakan action getForm.php dan method GET, sehingga kode program menjadi seperti berikut:</p> <pre> 1 <html> 2 <head> 3 </head> 4 <body> 5 <form action="getForm.php" method="get"> 6 <label>Username: </label> 7 <input type="text" name="username" placeholder="username">
 8 <label>E-mail: </label> 9 <input type="text" name="email" placeholder="e-mail">

 10 <input type="submit"> 11 </form> 12 </body> 13 </html> </pre>

6	<p>Buat file baru dengan nama getForm.php di dalam folder minggu7, kemudian ketikkan kode berikut:</p> <pre> 1 <html> 2 <head> 3 </head> 4 <body> 5 Welcome <?php echo \$_GET["username"]; ?>
 6 Your email address is: <?php echo \$_GET["email"]; ?> 7 </body> 8 </html> </pre>
7	<p>Simpan file dan jalankan kode program dengan mengetikkan localhost/minggu7/form_1.html</p>
8	<p>Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu! (soal no 2)</p>

Praktikum 2: Validasi Form

Langkah	Keterangan
1	Buka browser kemudian jalankan file getForm.php dengan mengetikkan perintah localhost/minggu7/getForm.php
2	Amati hasil yang ditampilkan
3	<p>Buka file getForm.php yang telah dibuat sebelumnya dalam folder minggu7. Modifikasi kode program dengan menambahkan fungsi isset(), sehingga kode program menjadi seperti berikut:</p> <pre> 1 <html> 2 <head> 3 </head> 4 <body> 5 <?php 6 if(isset(\$_GET["username"]) AND isset(\$_GET["email"])){ 7 echo "Welcome " . \$_GET["username"] . "!
"; 8 echo "Your email address is: " . \$_GET["email"]; 9 } 10 else{ 11 echo "Maaf, Anda harus mengakses halaman ini dari form_1.html"; 12 } 13 ?> 14 </body> 15 </html> </pre>
3	Simpan file dan jalankan kode program dengan mengetikkan localhost/minggu7/getForm.php
4	Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu! (soal no 3)

--	--

Praktikum 3: Form Required

Langkah	Keterangan
1	<p>Buat file form_2.php dalam folder minggu7 kemudian ketikkan kode program berikut:</p> <pre> 1 <!DOCTYPE html> 2 <html> 3 <head> 4 <style> 5 .error {color: #FF0000;} 6 </style> 7 </head> 8 <body> 9 <?php 10 //ambil nilai variabel error 11 if (isset(\$_GET['error'])){ 12 \$error=\$_GET['error']; 13 } 14 else{ 15 \$error=""; 16 } 17 18 //siapkan pesan kesalahan 19 \$pesan=""; 20 if (\$error=="variabel_belum_diset"){ 21 \$pesan="Anda harus mengakses halaman ini dari form_2.php"; 22 } 23 else if (\$error=="nama_kosong"){ 24 \$pesan="Nama harus diisi"; 25 } 26 else if (\$error=="email_kosong"){ 27 \$pesan="Email harus diisi"; 28 } 29 30 //Siapkan isian form jika terjadi kesalahan 31 if (isset(\$_GET['nama']) AND isset(\$_GET['email']) AND isset(\$_GET['komentar'])){ 32 \$nama=\$_GET['nama']; 33 \$email=\$_GET['email']; 34 \$komentar=\$_GET['komentar']; 35 } 36 else 37 { 38 \$nama=""; 39 \$email=""; 40 \$komentar=""; 41 } 42 43 <?> 44 45 <?php echo \$pesan;?> 46 47 <table> 48 <form method="get" action="prosesForm_2.php"> 49 <tr> 50 <td> Nama: </td> 51 <td> <input type="text" name="nama" value="<?php echo \$nama;?>"> </pre>

	<pre> 52 <tr> 53 <td> E-mail: </td> 54 <td> <input type="text" name="email" value="<?php echo \$email;?>"> </td> 55 </tr> 56 <tr> 57 <td> Komentar: </td> 58 <td> <textarea name="komentar" rows="5" cols="40"><?php echo \$komentar;?> 59 </textarea></td> 60 </tr> 61 <tr> 62 <td><input type="submit" name="kirim" value="Kirim"> </td> 63 </tr> 64 </form> 65 </table> 66 </form> 67 </body> 68 </html> </pre>
2	<p>Buat file prosesForm_2.php dalam folder minggu7 kemudian ketikkan kode program berikut:</p> <pre> 1 <?php 2 if (isset(\$_GET['nama']) AND isset(\$_GET['email'])) { 3 \$nama=\$_GET['nama']; 4 \$email=\$_GET['email']; 5 \$komentar=\$_GET['komentar']; 6 \$isi_form."&nama=\$nama&email=\$email&komentar=\$komentar"; 7 } 8 else{ 9 header("Location:form_2.php?error=variabel_belum_diset"); 10 } 11 12 if(empty(\$nama)){ 13 header("Location:form_2.php?error=nama_kosong".\$isi_form); 14 } 15 else if(empty(\$email)){ 16 header("Location:form_2.php?error=email_kosong".\$isi_form); 17 } 18 else{ 19 echo "Nama: \$nama
 Email: \$email
 Komentar: \$komentar"; 20 } 21 ?> </pre>
3	<p>Simpan file dan jalankan kode program dengan mengetikkan localhost/minggu7/form_2.php</p>
4	<p>Lakukan eksperimen berikut:</p> <ol style="list-style-type: none"> tanpa mengisi form langsung klik tombol “kirim” isi <i>field</i> nama dengan “violet” kemudian klik tombol “kirim” isi <i>field</i> nama dan email dengan “violet” kemudian klik tombol “kirim”
5	<p>Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu! (soal no 4)</p>
6	<p>Jelaskan apakah kegunaan fungsi header() (soal no 5)</p>

Langkah	Keterangan
1	Buat file baru dengan nama form_3.php kemudian salin kode program form_2.php dari praktikum bagian sebelumnya
2	<p>Modifikasi kode program sehingga kode program menjadi seperti berikut:</p> <pre> 1 <!DOCTYPE html> 2 <html> 3 <head> 4 <style> 5 .error {color: #FF0000;} 6 </style> 7 </head> 8 <body> 9 <?php 10 //ambil nilai variabel error 11 if (isset(\$_GET['error'])) { 12 \$error=\$_GET['error']; 13 } 14 else { 15 \$error=""; 16 } 17 18 //siapkan pesan kesalahan 19 \$pesan=""; 20 if (\$error=="variabel_belum_diset") { 21 \$pesan="Anda harus mengakses halaman ini dari form_3.php"; 22 } 23 else if (\$error=="nama_kosong") { 24 \$pesan="Nama harus diisi"; 25 } 26 else if (\$error=="nama_invalid") { 27 \$pesan="Nama diisi huruf dan spasi"; 28 } 29 else if (\$error=="email_kosong") { 30 \$pesan="Email harus diisi"; 31 } 32 if (\$error=="email_invalid") { 33 \$pesan="Email tidak sesuai"; 34 } 35 36 //Siapkan isian form jika terjadi kesalahan 37 if (isset(\$_GET['nama']) AND isset(\$_GET['email']) AND isset(\$_GET['komentar'])) { 38 \$nama=\$_GET['nama']; 39 \$email=\$_GET['email']; 40 \$komentar=\$_GET['komentar']; 41 } 42 else { 43 { 44 \$nama=""; 45 \$email=""; 46 \$komentar=""; 47 } 48 } 49 50 <?php echo \$pesan;?> 51 52 <table> 53 <form method="get" action="prosesForm 3.php"> 54 <tr> 55 <td> Nama: </td> 56 <td> <input type="text" name="nama" value="<?php echo \$nama;?>"> 57 </tr> 58 <tr> 59 <td> E-mail: </td> 60 <td> <input type="text" name="email" value="<?php echo \$email;?>"> </td> 61 </tr> </pre>

	<pre> 62 <tr> 63 <td> Komentar: </td> 64 <td> <textarea name="komentar" rows="5" cols="40"><?php echo \$komentar;?> 65 </textarea></td> 66 </tr> 67 <tr> 68 <td><input type="submit" name=" kirim" value="Kirim"> </td> 69 </tr> 70 </form> 71 </table> 72 </form> 73 </body> 74 </html> </pre>
3	Buat file baru dengan nama prosesForm_3.php kemudian salin kode program prosesForm_2.php dari praktikum bagian sebelumnya
4	<p>Modifikasi kode program php menambahkan fungsi preg_match() dan filter_var() sehingga kode program menjadi seperti berikut:</p> <pre> 1 <?php 2 if (isset(\$_GET['nama']) AND isset(\$_GET['email'])) { 3 \$nama=\$_GET['nama']; 4 \$email=\$_GET['email']; 5 \$komentar=\$_GET['komentar']; 6 \$isi_form="&nama=\$nama&email=\$email&komentar=\$komentar"; 7 } 8 else { 9 header("Location:form_3.php?error=variabel_belum_diset"); 10 } 11 12 if(empty(\$nama)){ 13 header("Location:form_3.php?error=nama_kosong".\$isi_form); 14 } 15 else if(!preg_match("/^[a-zA-Z]*\$/",\$nama)) { 16 header("Location:form_3.php?error=nama_invalid".\$isi_form); 17 } 18 else if(empty(\$email)){ 19 header("Location:form_3.php?error=email_kosong".\$isi_form); 20 } 21 else 22 if(!filter_var(\$email, FILTER_VALIDATE_EMAIL)){ 23 header("Location:form_3.php?error=email_invalid".\$isi_form); 24 } 25 else { 26 echo "Nama: \$nama
 Email: \$email
 Komentar: \$komentar"; 27 } 28 ?> </pre>
5	Simpan file dan jalankan kode program dengan mengetikkan localhost/minggu7/form_3.php
6	<p>Lakukan eksperimen berikut:</p> <ol style="list-style-type: none"> tanpa mengisi form langsung klik tombol “kirim” isi <i>field</i> nama dengan “123” kemudian klik tombol “kirim” isi <i>field</i> nama dengan “violet” kemudian klik tombol “kirim” isi <i>field</i> nama dan email dengan “violet” kemudian klik tombol “kirim” isi <i>field</i> nama dengan “violet” dan email dengan “violet@mail.com kemudian klik tombol “kirim”
7	Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu! (soal no 6)

Praktikum 5: Menggunakan File

Langkah	Keterangan
1	<p>Buat file baru dengan nama testFile.txt di dalam folder minggu7, kemudian ketikkan kalimat berikut:</p> <pre> 1 Jurusan Teknologi Informasi 2 Politeknik Negeri Malang </pre>
2	<p>Buat file baru dengan nama file.php di dalam folder minggu7, kemudian ketikkan kode berikut:</p> <pre> 1 <html> 2 <head> 3 </head> 4 <body> 5 <?php 6 \$myFile = "testFile.txt"; 7 \$fh = fopen(\$myFile, 'r'); 8 \$theData = fread(\$fh, 5); 9 fclose(\$fh); 10 echo \$theData; 11 ?> 12 </body> 13 </html> </pre>
3	<p>Simpan file dan jalankan kode program dengan mengetikkan localhost/minggu7/file.php</p>
4	<p>Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu! (soal no 7)</p>
5	<p>Modifikasi kode program pada langkah ke-2 dengan mengubah parameter fungsi fread() pada baris ke-8 menjadi filesize(\$myFile), sehingga kode program menjadi seperti berikut:</p> <pre> 1 <html> 2 <head> 3 </head> 4 <body> 5 <?php 6 \$myFile = "testFile.txt"; 7 \$fh = fopen(\$myFile, 'r'); 8 \$theData = fread(\$fh, filesize(\$myFile)); 9 fclose(\$fh); 10 echo \$theData; 11 ?> 12 </body> 13 </html> </pre>
6	<p>Simpan file dan jalankan kode program dengan mengetikkan localhost/minggu7/file.php</p>
7	<p>Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu! (soal no 8)</p>

8	<p>Buat file baru dengan nama file_1.php dan salin kode program pada langkah ke-5. Modifikasi kode program dengan mengubah fungsi fread() pada baris ke-8 dengan fungsi fgets(\$fh), sehingga kode program menjadi seperti berikut:</p> <pre> 1 <html> 2 <head> 3 </head> 4 <body> 5 <?php 6 \$myFile = "testFile.txt"; 7 \$fh = fopen(\$myFile, 'r'); 8 \$theData = fgets(\$fh); 9 fclose(\$fh); 10 echo \$theData; 11 <?> 12 </body> 13 </html> </pre>
9	<p>Simpan file dan jalankan kode program dengan mengetikkan localhost/minggu7/file_1.php</p>
10	<p>Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu! (soal no 9)</p>

Praktikum 6: File Upload

Langkah	Keterangan
1	Buat direktori baru dengan nama "uploads" di dalam folder minggu7
2	<p>Buat file baru dengan nama formUpload.html di dalam folder minggu7, kemudian ketikkan kode berikut:</p> <pre> 1 <html> 2 <head> 3 </head> 4 <body> 5 <form enctype="multipart/form-data" action="fileUpload.php" method="POST"> 6 choose a file to upload: 7 <input name="uploadedfile" type="file">
 8 <input type="submit" value="Upload File"> 9 </form> 10 </body> 11 </html> </pre>
3	Buat file baru dengan nama fileUpload.php di dalam folder minggu7, kemudian ketikkan kode berikut:

	<pre> 1 <?php 2 \$target_path ="uploads/"; 3 4 \$target_path =\$target_path . basename(5 \$_FILES['uploadedfile']['name']); 6 7 if(move_uploaded_file(\$_FILES['uploadedfile']['tmp_name'],\$target_path)){ 8 echo "The file " . basename(\$_FILES['uploadedfile']['name'])."has been uploaded"; 9 } else{ 10 echo "There was an error uploading the file, please try again!"; 11 } 12 ?> </pre>
4	Simpan file dan jalankan kode program dengan mengetikkan localhost/minggu7/formUpload.html
5	Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu! (soal no 10)
6	Tunjukkan hasil tangkapan layar isi dari direktori uploads (soal no 11)

Referensi :

- 1) Nixon, Robin. (2018). *Learning PHP, MySQL & JavaScript With jQuery, CSS & HTML5* (5th ed.). California: O'Reilly Media, Inc.
- 2) Forbes, Alan. (2012). *The Joy of PHP: A Beginners's Guide to Programming Interactive Web Applications with PHP and MySQL* (5th ed.). Plum Island Publishing.