

## FORM

1. TUJUAN:
  - a. Memahami dan menguasai penggunaan form pada PHP
2. MATERI:
  - 2.1 FORM**

Form biasanya digunakan untuk mengumpulkan informasi dari user, sehingga memungkinkan suatu web server untuk menerima informasi dari pemakai melalui sejumlah elemen yang disebut kontrol. Kontrol ini bisa berupa suatu textbox, checkbox, radio button, push button, list menu dan lainnya. Standart penulisan form :

```
<form method="post" action=url" enctype="multipart/form-
data" >
..
.
</form>
```

Atribut yang ada dalam form:

Atribut	Keterangan							
action	Menunjukkan lokasi URL kemana form tersebut di kirim							
method	Metode HTTP yang digunakan ketika mengirimkan data form (default:GET) method="post" : data dikirimkan secara terpisah method="get" : data dikirimkan ke server dengan cara meletakkan data pada bagian akhir url yang ditunjuk							
name	Nama spesifik yang digunakan sebagai identitas dari form (biasanya digunakan pada DOM html)							
enctype	Penulisan kode dari data yang dikirim. Nilai dari enctype: <table><tr><td>application/x-www-form-urlencoded</td><td>default nilai dari enctype</td></tr><tr><td>multipart/form-data</td><td>digunakan untuk mengupload sebuah file</td></tr><tr><td>text/plain</td><td>Mengirim data tanpa adanya encoding</td></tr></table>		application/x-www-form-urlencoded	default nilai dari enctype	multipart/form-data	digunakan untuk mengupload sebuah file	text/plain	Mengirim data tanpa adanya encoding
application/x-www-form-urlencoded	default nilai dari enctype							
multipart/form-data	digunakan untuk mengupload sebuah file							
text/plain	Mengirim data tanpa adanya encoding							
novalidate	ketika ditampilkan akan menentukan bahwa form-data (input) boleh untuk tidak divalidasi ketika dimasukkan							
target	Menunjukkan dimana respon dari form ditampilkan, misalnya window, tab atau dalam frame Value dari target bisa diisi: <table><tr><td>_blank</td><td>Membuka link di jendela/tab baru</td></tr><tr><td>_self</td><td>Default. Membuka link di frame yang sama seperti yang di klik</td></tr></table>		_blank	Membuka link di jendela/tab baru	_self	Default. Membuka link di frame yang sama seperti yang di klik		
_blank	Membuka link di jendela/tab baru							
_self	Default. Membuka link di frame yang sama seperti yang di klik							

	<code>_parent</code>	Membuka link di frame induk
	<code>_top</code>	Membuka link dalam body penuh di satu jendela
	<code>frameName</code>	Membuka link sesuai nama frame nya

Tipe-tipe inputan yang ada didalam form antara lain:

Tag	keterangan
<code>&lt;input&gt;</code>	Digunakan untuk menetapkan sebuah input kontrol
<code>&lt;textarea&gt;</code>	Digunakan untuk menetapkan multiple input control (textarea)
<code>&lt;label&gt;</code>	Digunakan untuk memberi label untuk sebuah elemen input
<code>&lt;fieldset&gt;</code>	Grup dari elemen-elemen yang berelasi dalam sebuah form
<code>&lt;legend&gt;</code>	Digunakan untuk memberikan judul
<code>&lt;select&gt;</code>	Digunakan untuk membuat dropdown list
<code>&lt;optgroup&gt;</code>	Digunakan untuk membuat opsi yang berelasi dalam sebuah dropdown list
<code>&lt;option&gt;</code>	Digunakan untuk membuat opsi pada dropdown list
<code>&lt;button&gt;</code>	Digunakan untuk membuat button/ tombol yang dapat diklik
<code>&lt;datalist&gt;</code>	Digunakan untuk menentukan daftar opsi yang telah ditentukan oleh control input
<code>&lt;keygen&gt;</code>	Digunakan untuk membangkitkan kunci (pada form)
<code>&lt;output&gt;</code>	Digunakan untuk hasil dari kalkulasi

## 2.2 Contoh Penggunaan Form

### 2.2.1 `<input>`

Input digunakan untuk menetapkan sebuah input control. Jenis-jenis input yang dapat digunakan dalam input ini antara lain:

a. Input type text

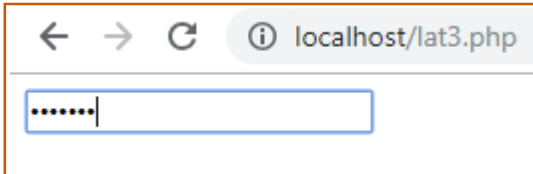
Contoh :

```
<input type="text" name="nama" value="" placeholder="Ketikkan Nama">
```

The screenshot shows a web browser window with the address bar displaying 'localhost/lat3.php'. Below the address bar, there is a text input field with the placeholder text 'Ketikkan Nama'.

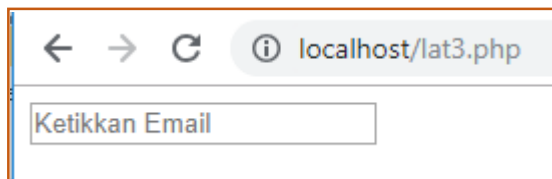
## b. input type password

```
<input type="password" name="password" value="" placeholder="Ketikkan Password">
```

A screenshot of a web browser window with the address bar showing 'localhost/lat3.php'. Below the address bar is a password input field with a blue border and a placeholder text 'Ketikkan Password'. The field contains several dots, indicating it is a password field.

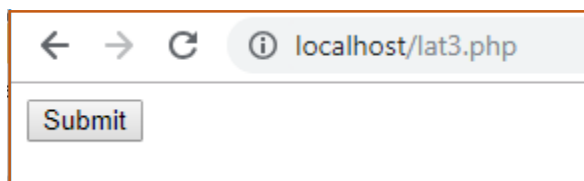
## c. input type email

```
<input type="email" name="email" value="" placeholder="Ketikkan Email">
```

A screenshot of a web browser window with the address bar showing 'localhost/lat3.php'. Below the address bar is an email input field with a blue border and a placeholder text 'Ketikkan Email'.

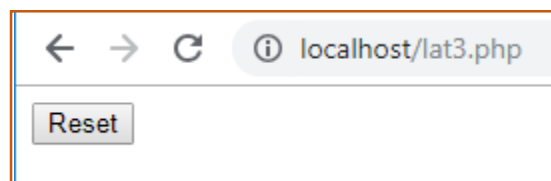
## d. input type submit

```
<input type="submit" name="submit" value="Submit">
```

A screenshot of a web browser window with the address bar showing 'localhost/lat3.php'. Below the address bar is a submit button with a grey border and the text 'Submit'.

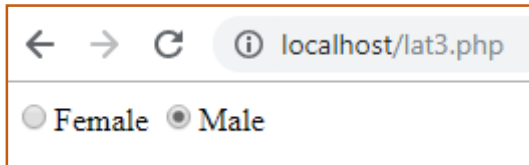
## e. input type reset

```
<input type="reset" name="reset" value="Reset">
```

A screenshot of a web browser window with the address bar showing 'localhost/lat3.php'. Below the address bar is a reset button with a grey border and the text 'Reset'.

## f. input type radio

```
<input type="radio" name="gender" value="F">Female  
<input type="radio" name="gender" value="M" checked="">Male
```

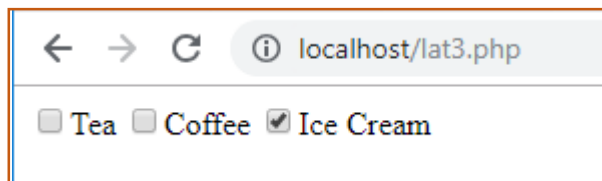


← → ↻ ⓘ localhost/lat3.php

☐ Female ☒ Male

g. input type checkbox

```
<input type="checkbox" name="bavage" value="Tea">Tea
<input type="checkbox" name="bavage" value="Coffee">Coffee
<input type="checkbox" name="bavage" value="Icecream" checked="">Ice Cream
```

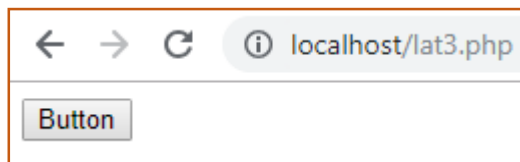


← → ↻ ⓘ localhost/lat3.php

☐ Tea ☐ Coffee ☒ Ice Cream

h. input type button:

```
<input type="button" name="button" value="Button">
```

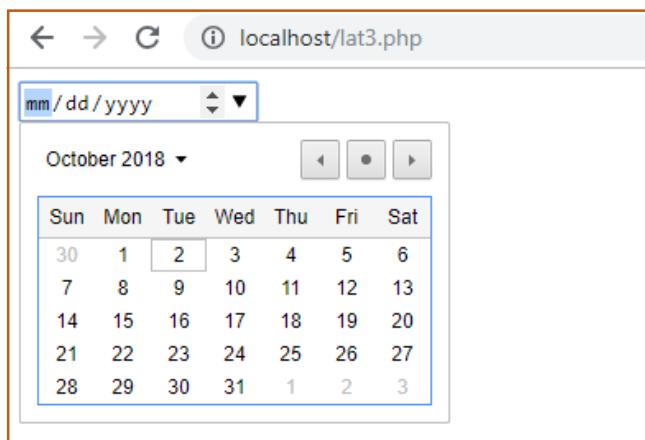


← → ↻ ⓘ localhost/lat3.php

Button

i. input type date:

```
<input type="date" name="tanggallahir" value="">
```



← → ↻ ⓘ localhost/lat3.php

mm/dd/yyyy

October 2018

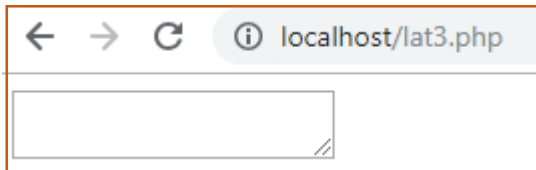
Sun	Mon	Tue	Wed	Thu	Fri	Sat
30	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3

Masih ada beberapa type input lainnya, antara lain:

datetime-local, week, month, number, time, range, search, url dll. Silakan dicoba sendiri.

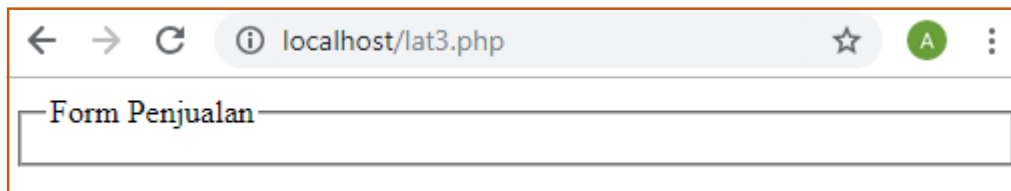
### 2.2.2 <textarea>

```
<textarea name="alamat"> </textarea>
```



### 2.2.3 <fieldset>,<legend>

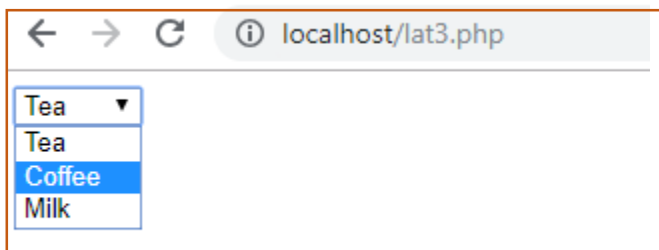
```
<fieldset>
  <legend>Form Penjualan</legend>
</fieldset>
```



### 2.2.3 <select> & <option>

Digunakan untuk membuat dropdown

```
<select name="baverage">
  <option value="tea">Tea</option>
  <option value="coffee">Coffee</option>
  <option value="milk">Milk</option>
</select>
```



## 2.3 Method pada Form

Sebuah data dari form dapat di parsing/di submit melalui 2 method, yaitu POST dan GET. Kedua method tersebut sebenarnya memiliki tujuan yang sama, perbedaannya terletak pada cara pengiriman data.

Perbedaan Method POST dan GET:

POST	GET
<ul style="list-style-type: none"> <li>✓ Menyembunyikan variable pada URL</li> <li>✓ Lebih aman</li> <li>✓ Tidak memiliki batasan panjang string</li> <li>✓ Menggunakan \$_GET untuk mengambil value dari variable</li> <li>✓ Bisa digunakan untuk mengirim file</li> </ul>	<ul style="list-style-type: none"> <li>✓ Menampilkan variable pada URL</li> <li>✓ Kurang aman</li> <li>✓ Batasan String s/d 2047 karakter</li> <li>✓ Menggunakan \$_POST untuk mengambil value dari variable</li> <li>✓ Tidak dapat digunakan untuk mengirim File, hanya bisa kirim teks</li> </ul>
<ul style="list-style-type: none"> <li>✓ Data tidak dapat disimpan pada history browser</li> <li>✓ Data tidak dapat dibookmark</li> <li>✓ Karena dianggap sebagai data sensitive, maka ketika kita merefresh browser akan muncul konfirmasi pengiriman ulang data</li> </ul>	<ul style="list-style-type: none"> <li>✓ Simpel, dan data mudah diedit, tinggal diganti value pada url nya</li> <li>✓ Halaman dapat di bookmark dan disimpan pada history browser sehingga mudah untuk diakses kembali</li> <li>✓ Dapat kembali ke halaman sebelumnya dengan mudah. Dengan meng-klik tombol back pada browser</li> <li>✓ Dapat di refresh dengan mudah</li> <li>✓ Dapat dishare/didistribusikan</li> </ul>

### 2.3.1 Method GET

Pada method **GET**, Form mengirimkan informasi dengan cara menambahkan data pada url dalam bentuk string Query yang terdiri dari nama\_data=nilai\_data secara berpasangan. String ini mengandung nama dan nilai data, yang di-pisah dengan ampersand &, sedangkan jika data mengandung spasi maka akan di pisah dengan tanda plus + atau % 20. Data tersebut digabung dengan url utama yang dipisahkan dengan tanda ?.

Sebelum dikirim, terlebih dahulu data diproses sehingga memenuhi standar format URL. URL hanya boleh memuat **huruf** (besar dan kecil), **angka**, dan beberapa karakter lain dalam ASCII Character Set seperti (".\_~), karakter di luar itu akan diubah ke format tertentu yang diawali tanda % kemudian diikuti dengan 2 digit hexadecimal. Contoh:

karakter	URL Encoded
?	%3F
@	%40
=	%3D

Angka pada kolom (URL Encoded) merupakan nilai hexadecimal dari character ASCII,

### 2.3.2 Method POST

Pada protokol HTTP, metode POST dapat dikirim baik melalui query string maupun body, seperti pada GET, data yang dikirim melalui query string akan ditampilkan pada URL dan sedangkan yang dikirim melalui body tidak terlihat oleh user.

Pada PHP, data POST yang dikirim melalui query string disimpan pada variabel `$_GET` (seperti metode GET) sedangkan yang dikirim melalui body disimpan pada variabel `$_POST`.

Sama seperti `$_GET`, variabel `$_POST` juga berbentuk associative array dan bersifat global yang artinya dapat diakses dimana saja, selain itu juga dapat dilakukan manipulasi sebagaimana variabel array lainnya.

## 2.4 Menangkap data dari form HTML dengan PHP

Setelah data dikirim oleh form HTML, maka data tersebut siap diolah.

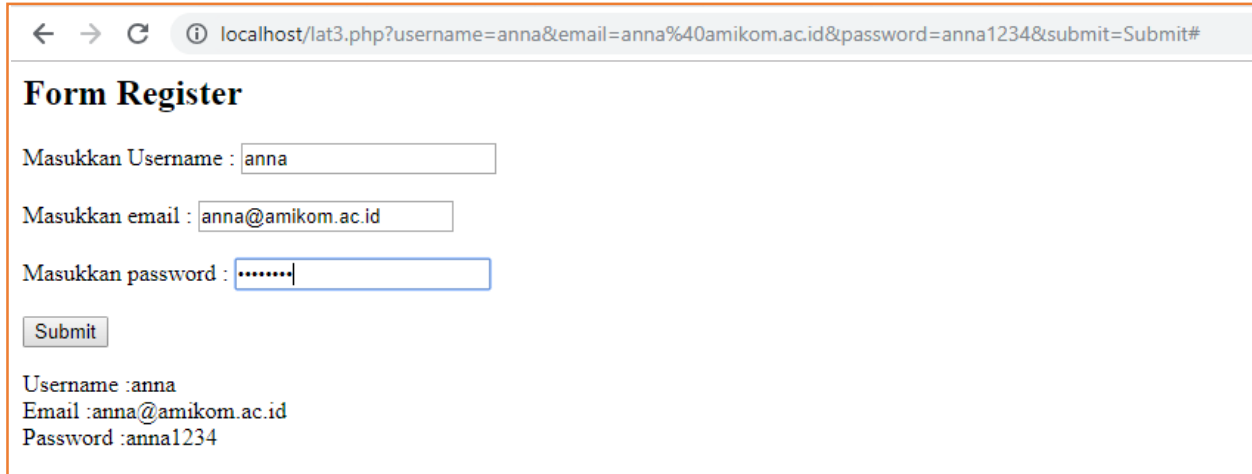
### 2.4.1 Menangkap data GET

Untuk menangkap data yang dikirim dengan method GET, gunakan variable `$_GET`. Variable tersebut berbentuk array dengan index berupa nilai atribut name pada elemen input.

Contoh:

```
<form action="#" method="GET">
  <h2>Form Register</h2>
  Masukkan Username : <input type="text" name="username" value=""><br><br>
  Masukkan email : <input type="email" name="email" value=""><br><br>
  Masukkan password : <input type="password" name="password" value=""><br><br>
  <input type="submit" value="Submit" name="submit">
</form>
<?php
  if($_GET){
    echo "Username :".$_GET['username']."<br>";
    echo "Email :".$_GET['email']."<br>";
    echo "Password :".$_GET['password']."<br>";
  }
?>
```

Output:



← → ↻ ⓘ localhost/lat3.php?username=anna&email=anna%40amikom.ac.id&password=anna1234&submit=Submit#

### Form Register

Masukkan Username :

Masukkan email :

Masukkan password :

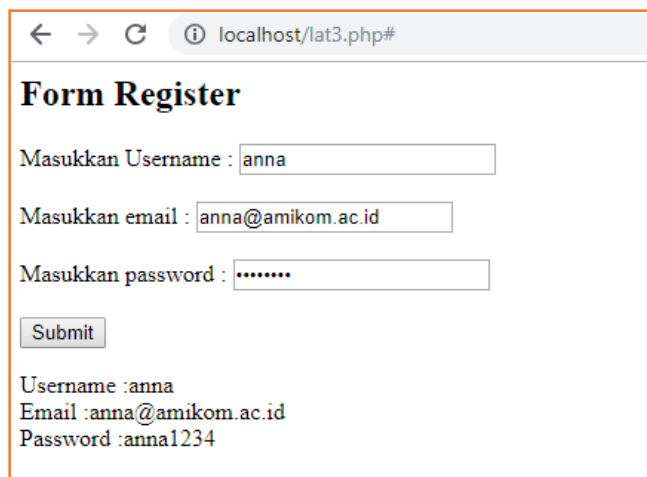
Username :anna  
Email :anna@amikom.ac.id  
Password :anna1234

#### 2.4.2 Menangkap data POST

Untuk menangkap data yang dikirim dengan method POST , gunakan variable `$_POST`. Variable tersebut berbentuk array dengan index berupa nilai atribut name pada elemen input.

```
<form action="#" method="POST">
  <h2>Form Register</h2>
  Masukkan Username : <input type="text" name="username" value=""><br><br>
  Masukkan email : <input type="email" name="email" value=""><br><br>
  Masukkan password : <input type="password" name="password" value=""><br><br>
  <input type="submit" value="Submit" name="submit">
</form>
<?php
  if($_POST){
    echo "Username :".$_POST['username']."<br>";
    echo "Email :".$_POST['email']."<br>";
    echo "Password :".$_POST['password']."<br>";
  }
?>
```

Output:



← → ↻ ⓘ localhost/lat3.php#

### Form Register

Masukkan Username :

Masukkan email :

Masukkan password :

Username :anna  
Email :anna@amikom.ac.id  
Password :anna1234



### 2.4.2 Menangkap data dengan \$\_REQUEST

PHP menyediakan variabel global bernama \$\_REQUEST, variabel ini merupakan gabungan dari tiga variabel yaitu: \$\_GET, \$\_POST dan \$\_COOKIE. Karakteristik variabel ini sama dengan \$\_GET dan \$\_POST yaitu berbentuk associative array yang bersifat global dan dapat kita manipulasi seperti variabel array lainnya.

Dalam pembentukan variabel \$\_REQUEST, variabel yang menjadi prioritas adalah \$\_POST, sehingga jika antara \$\_GET dan \$\_POST terdapat key yang sama, maka yang digunakan adalah data pada variabel \$\_POST.

CONTOH:

```
<form action="?action=edit" method="POST">
  <h2>Form Register</h2>
  Masukkan Username : <input type="text" name="username" value=""><br><br>
  Masukkan email : <input type="email" name="email" value=""><br><br>
  Masukkan password : <input type="password" name="password" value=""><br><br>
  <input type="hidden" name="action" value="insert">
  <input type="submit" value="Submit" name="submit">
</form>
<?php
  if($_POST){
    print_r($_GET);
    echo"<BR>";echo"<BR>";
    print_r($_POST);
    echo"<BR>";echo"<BR>";
    print_r($_REQUEST);
  }
?>
```

← → ↻ ⓘ localhost/lat3.php?action=edit

## Form Register

Masukkan Username :

Masukkan email :

Masukkan password :

Array ( [action] => edit )

Array ( [username] => anna [email] => anna@amikom.ac.id [password] => anna123 [action] => insert [submit] => Submit )

Array ( [action] => insert [username] => anna [email] => anna@amikom.ac.id [password] => anna123 [submit] => Submit )

## 2.5 Fungsi ISSET

Fungsi isset digunakan untuk menyatakan variable sudah di set atau belum. Jika variable sudah diset, maka variable akan mengembalikan nilai True, begitu pula sebaliknya. Jika variable belum di set maka akan mengembalikan nilai false.

Contoh penggunaan fungsi isset untuk mengecek tombol submit :

```
<form action="#" method="POST">
  <h2>Form Register</h2>
  Masukkan Username : <input type="text" name="username" value=""><br><br>
  Masukkan email : <input type="email" name="email" value=""><br><br>
  Masukkan password : <input type="password" name="password" value=""><br><br>
  Masukkan Program :
  <input type="radio" name="program" value="reguler">Reguler
  <input type="radio" name="program" value="premium">Premium<br><br>
  <input type="submit" value="Submit" name="submit">
  <input type="submit" value="Click Me" name="clickme">
  <input type="reset" value="Reset" name="reset">
</form>
<?php
  if(isset($_POST['submit'])){
    echo "Username :".$_POST['username']."<br>";
    echo "Email :".$_POST['email']."<br>";
    echo "Password :".$_POST['password']."<br>";
    echo "program :".$_POST['program']."<br>";
  }
?>
```

← → ↻ ⓘ localhost/lat3.php#

## Form Register

Masukkan Username :

Masukkan email :

Masukkan password :

Masukkan Program : ☐ Reguler ☒ Premium

Username :anna  
Email :anna@amikom.ac.id  
Password :anna1234  
program :premium

## 2.6 Upload File

Untuk menangkap variable file yang diupload ke server, kita bisa menggunakan `$_FILES`. Variable ini isi nya adalah array yang menampung data-data tentang file nya. Di sana ada beberapa indeks:

- name adalah nama file yang di-upload;
- type adalah jenis file yang di-upload;
- tmp\_name adalah nama file yang berada di dalam direktori temporer server;
- error menyatakan apakah ada error atau tidak;
- size adalah ukuran file-nya.

Lokasinya akan disimpan di direktori temporer atau direktori sementara di server. Untuk memindahkan file ini ke dalam direktori aplikasi kita, maka kita membutuhkan sebuah fungsi bernama: **move\_uploaded\_file()**.

Contoh:

```
lat3.php x
<form action="#" method="POST" enctype="multipart/form-data">
  <h2>Form Upload</h2>
  Pilih File : <input type="file" name="berkas" value=""><br><br>
  <input type="submit" value="Submit" name="submit">
  <input type="reset" value="Reset" name="reset">
</form>
<?php
  if(isset($_POST['submit'])){
    echo "<pre>";
    print_r($_FILES);
    echo "</pre>";
  }
?>
```



The screenshot shows a web browser window with the address bar displaying 'localhost/lat3.php#'. The page title is 'Form Upload'. Below the title, there is a text input field labeled 'Pilih File :', a 'Choose File' button, and the filename 'mawar1.jpg'. Below this, there are 'Submit' and 'Reset' buttons. The output of the script is displayed as a PHP array:

```
Array
(
    [berkas] => Array
        (
            [name] => mawar1.jpg
            [type] => image/jpeg
            [tmp_name] => C:\xampp\tmp\php63E1.tmp
            [error] => 0
            [size] => 9547
        )
)
```

Untuk langkah berikutnya, silahkan lakukan pengembangan sendiri.

Seperti:

- Menyimpan data gambar di Database.
- Melakukan filter terhadap data yang di-upload.
- Melakukan pengolahan citra atau *image processing* dengan PHP

Contoh:

Berkas file yang akan di upload, akan disimpan di folder **uploads** di documentroot dari server, maka langkah pertama adalah buat folder **uploads** di documentroot.

```
lat3.php x
<form action="#" method="POST" enctype="multipart/form-data">
  <h2>Form Upload</h2>
  Pilih File : <input type="file" name="berkas" value=""><br><br>
  <input type="submit" value="Submit" name="submit">
  <input type="reset" value="Reset" name="reset">
</form>
<?php
  if(isset($_POST['submit'])){
    echo "<pre>";
    print_r($_FILES);
    echo "</pre>";
    // ambil data file
    $namaFile = $_FILES['berkas']['name'];
    $namaSementara = $_FILES['berkas']['tmp_name'];

    // tentukan lokasi file akan dipindahkan
    $dirUpload = "uploads/";

    // pindahkan file
    $terupload = move_uploaded_file($namaSementara, $dirUpload.$namaFile);

    if ($terupload) {
      echo "Upload berhasil!<br/>";
      echo "Link: <a href='".$dirUpload.$namaFile."'>".$namaFile."</a>";
    } else {
      echo "Upload Gagal!";
    }
  }
?>
```

