

# Pengembangan Aplikasi Web

## Pertemuan Ke-4 (PHP Lanjut 2)

Noor Ifada

[noor.ifada@trunojoyo.ac.id](mailto:noor.ifada@trunojoyo.ac.id)

# Sub Pokok Bahasan

- Pemrosesan *File*

- Pemrosesan *Form*

  - *Form* HTML

  - Mengakses Data *Form* (PHP)

- Validasi *Server-side*

  - Fungsi (*Function*) untuk Validasi PHP

  - Pemrosesan *Error*

# Pemrosesan File

- *File* merupakan tempat untuk menyimpan data sekunder secara permanen di dalam suatu media penyimpan (misal *hardisk*, *flashdisk*)
- Secara umum format *file* berupa: teks atau *binary*
- Tiga tahap pengaksesan dalam skema dasar pengaksesan file:
  1. Membuka *file*
  2. Memproses *file*/manipulasi *file*
  3. Menutup *file*

# Pemrosesan File [2]

Beberapa fungsi dalam pemrosesan *file*:

- **fopen()**: membuka/mengakses *file* (fungsi ini dapat mengakses *file* dari sistem file, atau melalui HTTP atau FTP di internet)
- **fclose()**: menutup *file* digunakan
- **fgets()**: melihat isi dari *file* digunakan
- **fputs()**: memasukkan data ke dalam *file*
- **feof()**: menentukan akhir dari sebuah file (Jika sudah pada akhir file, fungsi ini akan bernilai TRUE)
- **copy()**: mengkopi suatu *file* (*file* yang akan dikopi dapat diambil melalui masukan data *form*)
- **unlink()**: menghapus *file* (bernilai TRUE jika *file* berhasil dihapus dan FALSE jika terjadi kegagalan dalam proses penghapusan)
- **rename()**: mengganti nama *file* (bernilai TRUE jika nama *file* berhasil diubah dan FALSE jika terjadi kegagalan dalam proses pengubahan)
- **file\_exists()**: memeriksa kondisi suatu *file*, apakah *file* tersebut ada atau tidak (bernilai TRUE jika *file* ada dan FALSE jika *file* tidak ada)
- **filesize()**: mengetahui besar ukuran suatu *file* digunakan (hasil bertipe *integer*, yang menyatakan ukuran *file* dalam satuan *byte*)

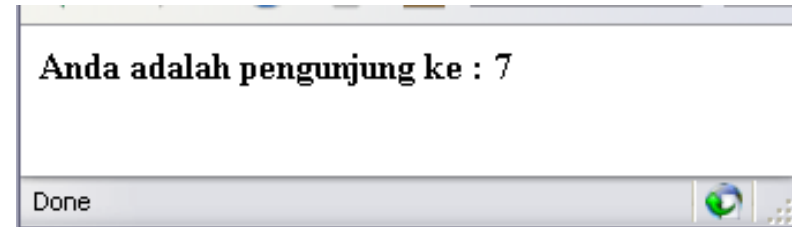
# Contoh Pemrosesan File: Membuat *Counter*

- Fungsi *counter* di sini adalah untuk menghitung berapa kali suatu halaman *web* telah dibuka oleh pengunjung
- Langkah-langkah:
  1. Membuat file teks (**counter.txt**):
    - setiap kali suatu halaman *web* ditampilkan maka terlebih dulu isi file **counter.txt** akan dibaca nilainya dan diperbaharui (ditambahkan nilainya dengan 1)
    - nilai *counter* diinisialisasi bernilai 0 (nol)
  2. Membuat *file* php (**counter.php**) untuk:
    - membaca nilai di dalam *file* teks
    - memperbaharui nilai *counter*
    - menyimpan nilai yang baru di *file*
    - menampilkan nilainya di *web browser*

# Contoh Pemrosesan File: Membuat Counter [2]

File: counter.txt

0



File: counter.php

```
<?php
    $filecounter="counter.txt"; // file yang akan diproses
    $open=fopen($filecounter,'r+'); // membuka file
    $counter=fgets($open); // melihat isi file
    fclose($open); // menutup file
    $counter++; // manipulasi isi file
    $write=fopen($filecounter,'w'); // membuka file
    fputs($write,$counter); // memasukkan data ke dalam file
    fclose($write); // menutup file
    echo "<b> Anda adalah pengunjung ke : $counter</b>";
?>
```

# Pemrosesan *Form*

## Client/Browser

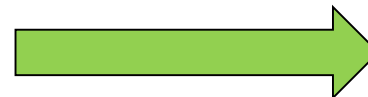
MyForm.html

```
<html>
  <form action="processData.php"
        method="POST">
    ...
  </form>
</html>
```



message

**Submit**



HTTP Post



## Web Server



processData.php

```
<?php
...
?>
```

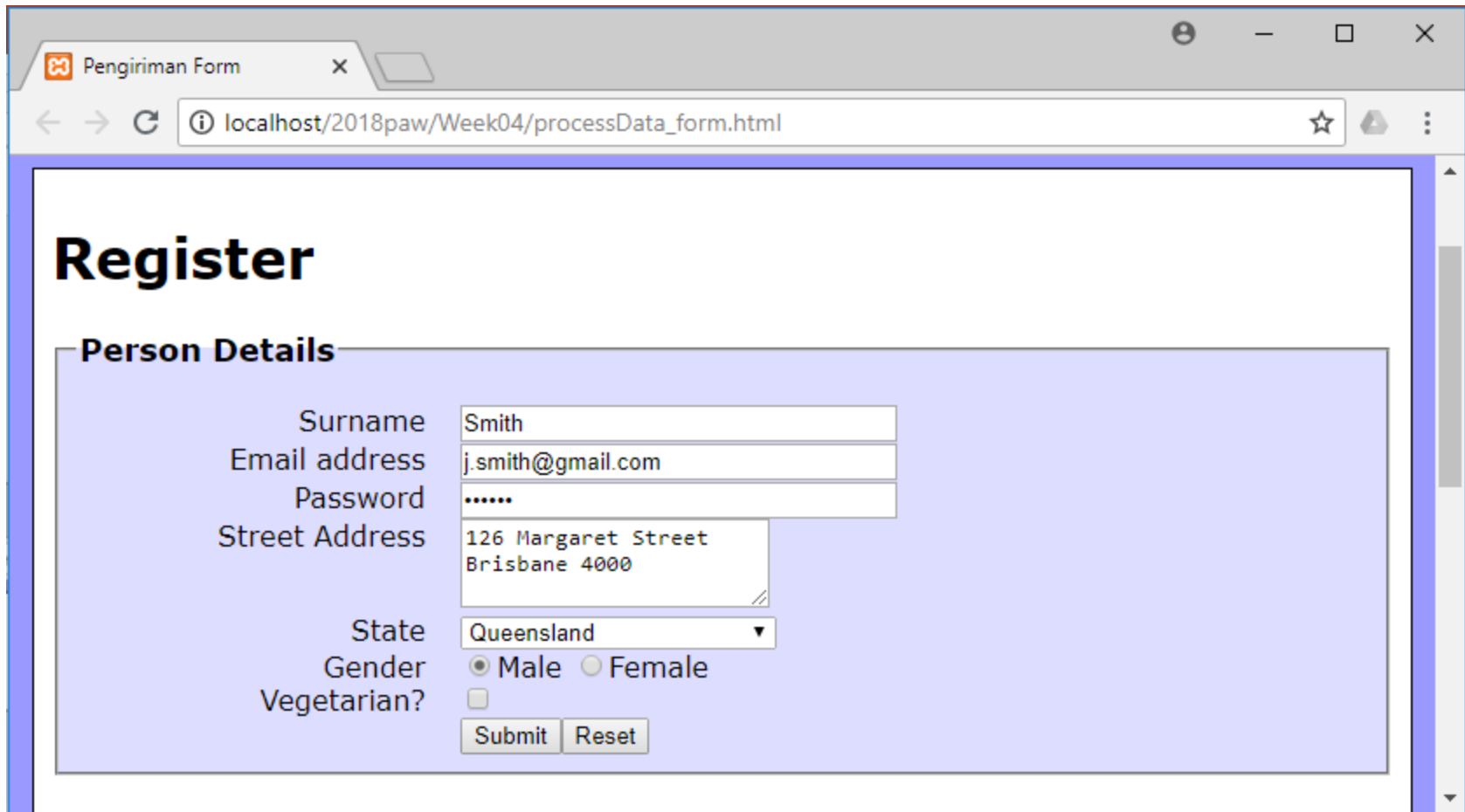
**NOTE:** pemrosesan server-side berdasarkan atribut `name`, bukan atribut `id` (atribut `id` hanya digunakan pada pemrosesan client-side)

# Pemrosesan Form: Form HTML

```
16 <legend>Person Details</legend>
17 <form name="myForm" action="processData.php" method="POST">
18   <div class="field">
19     <label for="surname">Surname</label>
20     <input type="text" name="surname" size="31"/>
21   </div>
22   <div class="field">
23     <label for="email">Email address</label>
24     <input type="text" name="email" size="31"/>
25   </div>
26   <div class="field">
27     <label for="passwd">Password</label>
28     <input type="password" name="passwd" size="31"/>
29   </div>
30   <div class="field">
31     <label for="address">Street Address</label>
32     <textarea name="address" rows="3" cols="22"></textarea>
33   </div>
34   <div class="field">
35     <label for="state">State</label>
36     <select name="state">
37       <option value="-1" selected>[choose your state]</option>
38       <option value="1">Australian Capital Territory</option>
39       <option value="2">Queensland</option>
40       <option value="3">Victoria</option>
41       <option value="4">New South Wales</option>
42       <option value="5">Tasmania</option>
43       <option value="6">South Australia</option>
44       <option value="7">Western Australia</option>
45       <option value="8">Nothern Territory</option>
46     </select>
47   </div>
48   <div class="field">
49     <input type="hidden" name="country" value="Australia">
50   </div>
51   <div class="field">
52     <label for="sex">Gender</label>
53     <input type="radio" name="sex" value="male"/>Male
54     <input type="radio" name="sex" value="female"/>Female
55   </div>
56   <div class="field">
57     <label for="vegetarian">Vegetarian?</label>
58     <input type="checkbox" name="vegetarian" />
59   </div>
60   <div class="field">
61     <label>&nbsp;</label>
62     <input type="submit" value="Submit" name="submit" /><input type="reset" value="Reset" name="reset" />
63   </div>
64 </form>
65 </fieldset>
```



# Pemrosesan *Form*: *Form* HTML [2]



The screenshot shows a web browser window with the title 'Pengiriman Form'. The address bar displays 'localhost/2018paw/Week04/processData\_form.html'. The main content area features a 'Register' form with a light blue background. The form is titled 'Person Details' and contains several input fields: 'Surname' (Smith), 'Email address' (j.smith@gmail.com), 'Password' (masked with dots), 'Street Address' (126 Margaret Street, Brisbane 4000), 'State' (Queensland), 'Gender' (Male selected), and 'Vegetarian?' (checkbox). There are 'Submit' and 'Reset' buttons at the bottom.

**Register**

**Person Details**

Surname: Smith

Email address: j.smith@gmail.com

Password: .....

Street Address: 126 Margaret Street  
Brisbane 4000

State: Queensland

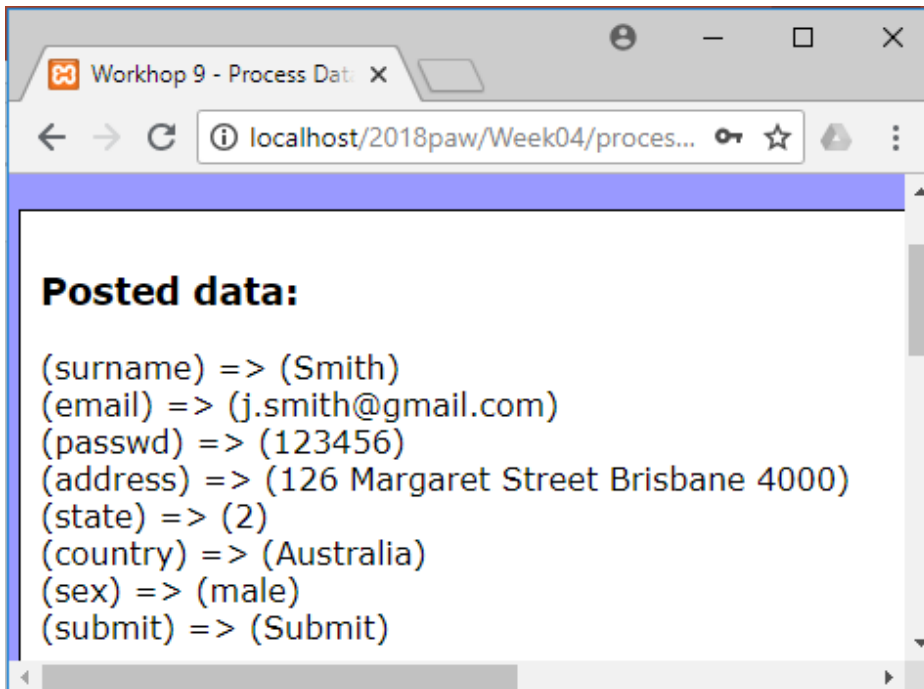
Gender: ☒ Male ☐ Female

Vegetarian?: ☐

Submit Reset

# Pemrosesan *Form*: Akses Data *Form* (PHP)

```
14 <h3>Posted data: </h3>
15 <?php
16 foreach ($_POST as $key => $value)
17     echo "($key) => ($value)<br/>";
18
19 // $surname = $_POST['surname'];
20 ?>
```



**NOTE:** Cek dokumen praktikum (bagian I)!

# Validasi Server-Side

- Jangan hanya mengandalkan validasi *client-side*
- Karena:
  1. *Client* mungkin tidak mengaktifkan JavaScript pada *web browser* yang digunakan
  2. *Hacker* dapat menuliskan program untuk mengirimkan pesan HTTP request ke halaman PHP

# Validasi Server-Side [2]

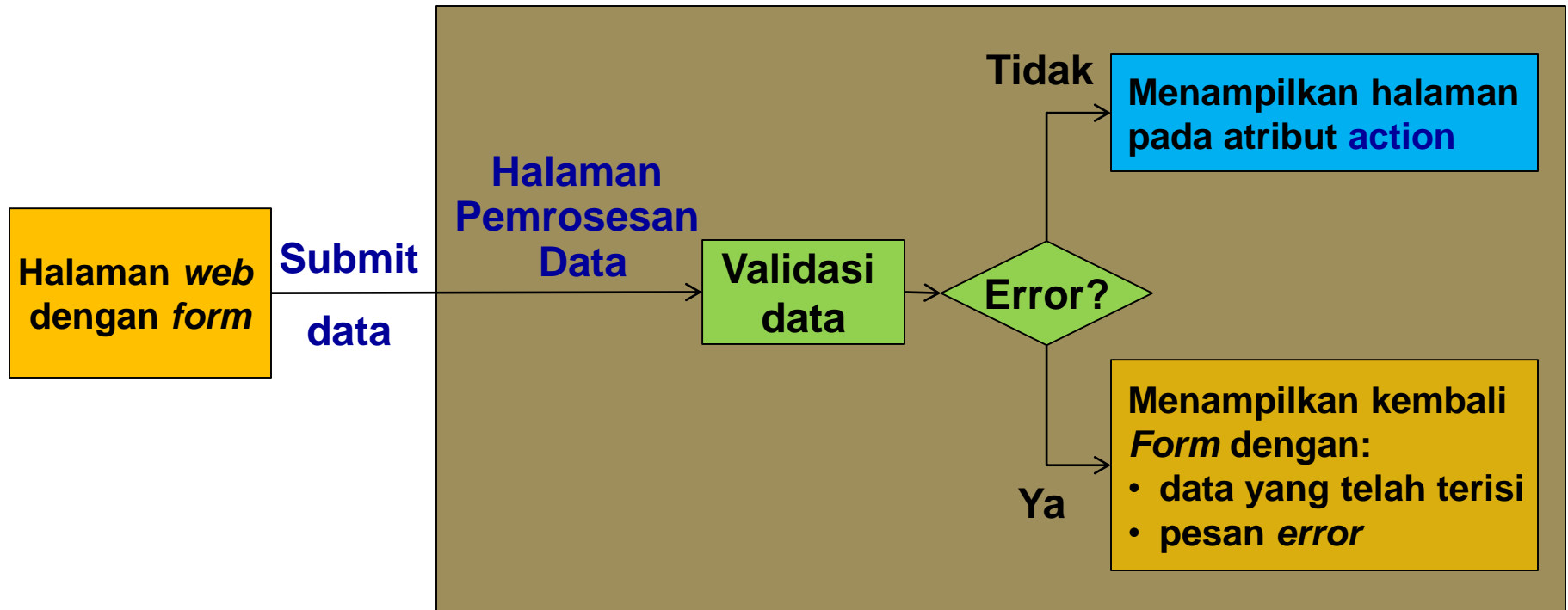
```
basicValidation.php
4  <?php
5      require_once 'validate.inc';
6
7      $errors = array();
8      validatePattern($errors, $_POST, 'name', '/^[a-zA-Z]+$/' );
9      validatePattern($errors, $_POST, 'phone', '/^((\d{4}[ ]?\d{4})|(04\d{2}[ ]?\d{3}[ ]?\d{3}))$/');
10     validateDate($errors, $_POST, 'dob');
11     // ...
12
13     if (count($errors) == 0)
14     {
15         // Data validates so continue with processing
16         // ...
17     }
18     else
19     {
20         // Data doesn't validate, so display error message
21         // ...
22     }
23  ?>
```

```
11  function validatePattern(&$errors, $field_list, $field_name, $pattern)
12  {
13      if (!isset($field_list[$field_name]) || $field_list[$field_name] == '')
14          $errors[$field_name] = 'Required';
15      else if (!preg_match($pattern, $field_list[$field_name]))
16          $errors[$field_name] = 'Invalid';
17  }
```

# Fungsi (*Function*) untuk Validasi PHP

- *Regular expressions*: pregmatch
- *String functions*: trim, strtolower, strtoupper
- *Filter functions*:
  - filter\_var, filter\_input
  - FILTER\_VALIDATE\_EMAIL,
  - FILTER\_VALIDATE\_URL,
  - FILTER\_VALIDATE\_FLOAT
  - FILTER\_VALIDATE\_IP,
  - ...
- *Type testing functions*: is\_float, is\_int, is\_numeric, is\_string
- *Date functions*: checkdate

# Pemrosesan *Error* (Validasi)



**NOTE:** Cek dokumen praktikum (bagian II & III)!

