

Pengaturcaraan Web dengan PHP

Advance PHP

Kandungan Bengkel

1. Hari pertama
 - a. Session Handling
 - b. Submit form to DB
2. Hari kedua
 - a. Output from DB and reporting
 - b. PHP Framework with CodeIgniter

Sebelum kita mula...

1. Pastikan anda sedia dengan editor teks
 - a. notepad++
 - b. netbeans
2. Mulakan server MySQL
3. Larikan Server Web
4. Pastikan anda dapat akses kepada localhost
5. Mudah jika guna WAMP / XAMP

Hari 1 Sessi Pagi

Session

mengurus session dengan PHP

Pengendalian SESSION

Session menyimpan data pengguna

- dapat digunakan oleh banyak banyak laman web
- biasanya, session kekal sehingga pengguna menutup pelayar web
- session akan menyimpan data seorang pengguna dan dapat digunakan oleh kesemua laman dalam satu aplikasi

Pengendalian Session

- Data session disimpan di dalam server
- user-key disimpan di dalam komputer pengguna.
 - user-key ini akan diakses apabila user mengakses kepada laman lain
 - jika user-key ini sama dengan key di server. Data session di server akan dicapai.
 - Jika tidak sama, session baru akan dimulakan
- Jika aplikasi perlukan storan kekal, session boleh disimpan di dalam pangkalan data.

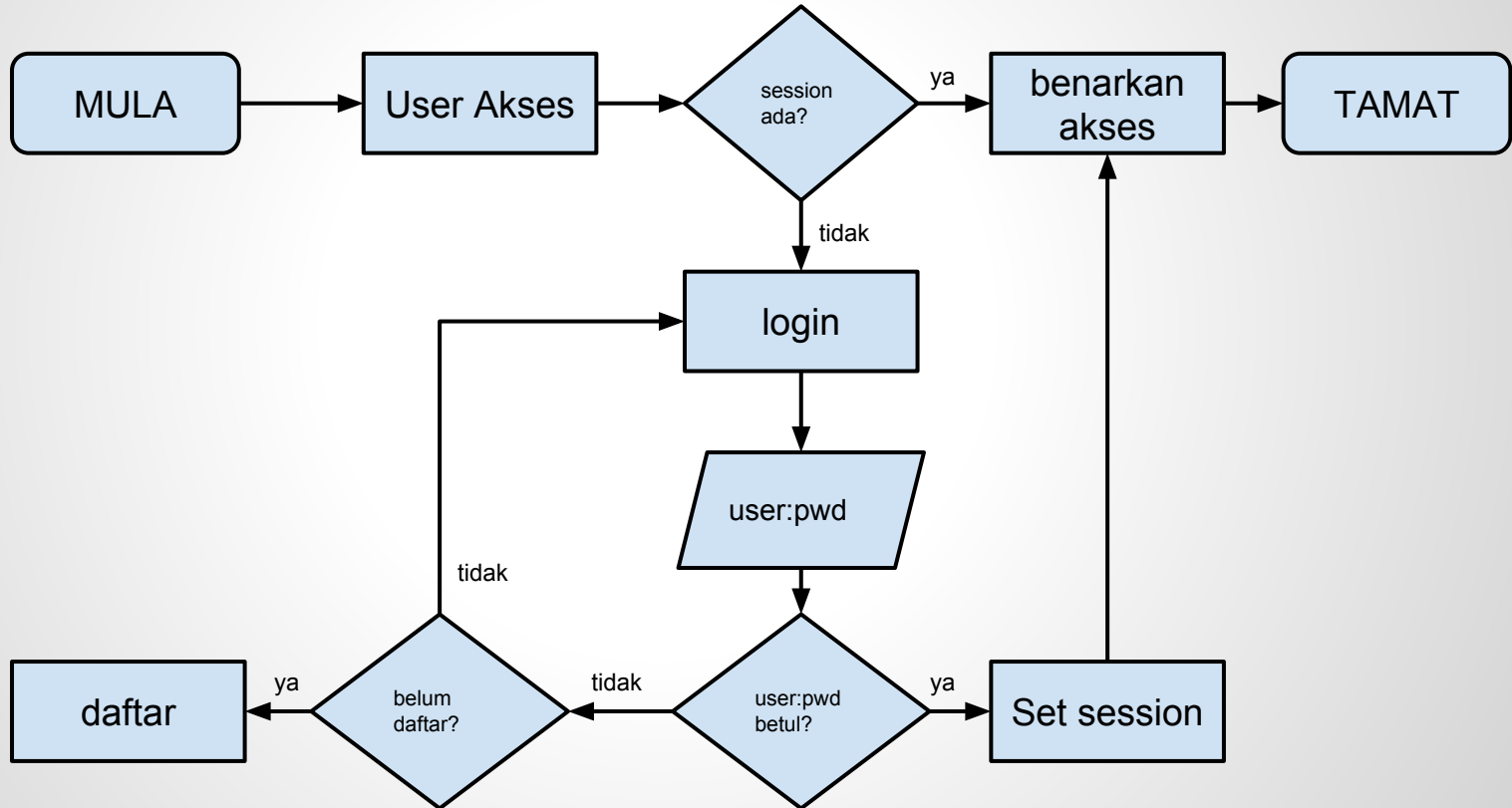
Contoh penggunaan session

1. User login
2. Kart pembelian pelanggan
3. User custom page setting

User Login dengan Session

1. Session untuk user disimpan selepas user isikan username & katalaluan yang betul.
2. Session diperiksa pada setiap laman web yang diakses
3. Session dimusnahkan apabila user logout.

User Login dengan Session



Kita perlukan...

1. Laman index.php
 - a. laman ini akan memeriksa session user
2. Laman dengan borang login, login_form.php
3. Sambungan kepada database mysql
4. database dengan table pengguna
5. login.php dengan SQL statement.

Programming

coding : www.github.com/amree/kelas-php

Sambungan kepada mysql

- asingkan fail yang mengandungi username dan password
- fail dbconfig.ini boleh dikawal akses secara khusus.

```
dbconfig.ini
DatabaseName = "sistem123"
Hostname = "localhost"
Username = "test"
Password = "password123"
```

```
$config = parse_ini_file("dbconfig.ini");

mysql_connect($config["Hostname"], $config["Username"], $config["Password"]) or die(mysql_error());

mysql_select_db($config["DatabaseName"]) or die(mysql_error());
```

Jika banyak Database

```
dbconfig.ini
[DatabaseA]
Hostname = 'identiti.politech.my'
DatabaseName = 'identiti'
Username= 'user1'
Password= 'abcpassworddef'
[DatabaseB]
Hostname = 'courses.politech.my'
DatabaseName = 'courses'
Username= 'user2'
Password= '123password456'
```

```
$config = parse_ini_file("dbconfig.ini");

$dba = mysql_connect(
$config["DatabaseA"]["Hostname"],
$config["DatabaseA"]["Username"],
$config["DatabaseA"]["Password"]);

$dbb = mysql_connect(
$config["DatabaseB"]["Hostname"],
$config["DatabaseB"]["Username"],
$config["DatabaseB"]["Password"], true);

mysql_select_db(
$config["DatabaseA"]["DatabaseName"], $dba);

mysql_select_db(
$config["DatabaseB"]["DatabaseName"], $dbb);
```

Cipta DB di MySQL

- Cipta DB baru dengan nama 'bengkel'
- Gunakan phpmyadmin atau mysql console
- gunakan SQL dibawah;

Database users

```
CREATE TABLE `users` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `email` varchar(255) DEFAULT NULL,  
  `pass` varchar(255) DEFAULT NULL,  
  `name` varchar(255) DEFAULT NULL,  
  `photo` longblob,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB
```

Hari 1 Sessi Petang

Submit ke Database

Simpan data ke MySQL DB dengan PHP

Sessi ini kita perlu

- sambungan kepada MySQL, database dan table 'users'
- laman web dengan borang pendaftaran users
- Submit borang kepada new.php
- SQL statement INSERT INTO users ...
- Paparkan status pendaftaran.

Sambungan kepada MySQL

- menggunakan cara yang sama dengan kaedah sesi session tadi.
- Fail-fail konfigurasi seeloknya diletakkan diluar folder web (`‘/www/includes/namasistem/failconf.ini.php’`)
- gunakan kaedah `require` dan `require_once`
- apa beza `require`, `require_once`, `include_once` dan `include`?

Borang pendaftaran user

- Gunakan coding HTML5 yang standard.
- Lebih mudah menggunakan mana-mana css framework untuk styling.
- Asingkan sebaik-baiknya kod php dan kod HTML.

```
<!DOCTYPE html>
<html><head>
<meta charset="UTF-8">
<title>Title of the document</title>
</head>
<body>
  <form action="" method="post">
    <label>Username</label>
    <input type="text" name="username" maxlength="60">
    <label>password</label>
    <input type="password" name="pass" maxlength="10">
    <input type="submit" name="submit" value="Register"
  >
  </form>
</body></html>
```

new.php

- PHP membaca data daripada function `$_POST` atau `$_GET`
- Guna `print_r()` untuk tengok data yang dihantar
- Hantar data ke page yang sama atau page yang lain
- Guna `mysql_escape_string` untuk mengelakkan SQL Injection
- Guna `mysql_affected_rows()` untuk periksa data yang dimasukkan
- Kumpul error dalam array untuk dipaparkan

edit.php

- Data asal diambil menggunakan `$_GET['id']`
- Pastikan data lama wujud terlebih dahulu
- Kemaskini data tersebut berdasarkan `$_GET['id']` yang dihantar
- Guna form yang sama dari `new.php`. Tapi kali ni masukkan nilai yang asal dalam field
- Sembunyikan field yang tidak perlu menggunakan `if`
- Buat satu field hidden 'id' untuk simpan 'id' data (juga akan dihantar bersama field)

upload.php

- Terdapat dua cara simpan fail yang diupload:
 - Guna kolum BLOB MySQL
 - Simpan fail secara fizikal dan simpan metadata dalam MySQL
- Fail yang diupload akan disimpan dalam direktori sementara
- Dari situ, boleh pilih di antara dua cara di atas
- Form mesti diset kepada enctype="multipart/form-data"
- Boleh buat validation berdasarkan jenis gambar, saiz gambar dan sebagainya
- Maklumat fail yang diupload disimpan dalam \$_FILE

upload.php

- Untuk membuat paparan, boleh guna teknik inline image menggunakan format base64 (convert data gambar ke dalam format ini). Cara ini lebih mudah dan senang.

Kemaskini data

- Untuk update data yang sedia ada dalam table, perlu cari dahulu data dari table
- carian perlukan input dari form
- hasil carian biasanya akan lebih dari 1.
 - oleh itu, perlu ada logik (loop) untuk paparkan hasil carian
- Hasil dari carian perlu diletakkan dalam form supaya boleh dikemaskini

Kemaskini data

- gunakan SQL query update untuk update data
- \$result adalah samada true / false

Buang Data

- Perlukan confirmation apabila user tekan delete untuk buang data.

```
<a href="delete.page?id=1"
```

```
onclick="return confirm('Are you sure you want to delete?')">
```

```
Delete</a>
```

- Pastikan hanya satu data sahaja dibuang dengan menambah **LIMIT 1** pada hujung SQL statement

```
DELETE FROM `users` WHERE id=1 LIMIT 1;
```