

Jakarta Selatan') '');

Sehingga script akhir menjadi sebagai berikut :

```
1.<?php
2. $dsn = 'mysql:host=localhost;dbname=test';
3. $username = 'opan';
4. $password = 'opansan';
5.
6. try {
7.   $dbh = new PDO($dsn, $username, $password);
8.   echo "Database connected";
9. }
10. catch (PDOException $e)
11. {
12.   echo $e->getMessage();
13. }
14.
15. // Insert Database
16. $query=$dbh->exec ("insert into mahasiswa (nim,nama,alamat) values
17. ('1211500100','Budiman','Jl.Mawar No.14 Jakarta Selatan')");
18.
19. if ($query)
20. {
21.   echo "Berhasil di insert";
22. }
23. else
24. {
25.   echo "Tidak berhasil di insert";
26. }
27. ?>
```

Select

Untuk menampilkan isi database, anda menggunakan perintah 'select' pada query. Query lengkap nya adalah sbb :

```
<?php
1.$dsn = 'mysql:host=localhost;dbname=test';
2.$username = 'opan';
3.$password = 'opansan';
4.
5. try {
6.   $dbh = new PDO($dsn, $username, $password);
7.   echo "Database connected <br/>";
8. }
9. catch (PDOException $e)
10. {
11.   echo $e->getMessage();
12. }
13.
14. // Select database //
15. $querynya="select * from mahasiswa";
16.
17. foreach ($dbh->query($querynya) as $row)
18. {
19.   echo "NIM :".$row['nim']."<br/>";
20.   echo "Nama :".$row['nama']."<br/>";
```

```
21. echo "Alamat :".$row['alamat']."<br/>";
22. }
23. ?>
```

BAB XI

Membuat aplikasi CRUD sederhana

CRUD (Create, Read, Update dan Delete) sering digunakan dalam banyak aplikasi, termasuk juga di dalam pemrograman web. Create, yakni melakukan input data ke dalam tabel di sebuah database, Read, yakni membaca data pada tabel, Update, melakukan edit data dan Delete, menghapus data.

Untuk dapat menjalankan aplikasi sederhana ini, anda perlu menginstalasi Apache, PHP dan Mysql. Cara mudah untuk menginstall aplikasi-aplikasi tersebut adalah dengan mendownload dan menginstall Xampp, yang dapat anda download di www.apachefriends.org. Setelah melakukan instalasi, script aplikasi CRUD ini dapat anda taruh di direktori `c:/xampp/htdocs` jika anda menggunakan Sistem Operasi Windows atau di direktori `/opt/lampp/htdocs`, jika anda menggunakan sistem operasi Linux.

IX.1.Contoh Aplikasi

Pada Contoh Aplikasi kali ini, kita akan membuat aplikasi CRUD sederhana, yakni, melakukan entri sebuah bilangan (n), yang bilangan tersebut akan melakukan pencetakan karakter “*” sebanyak n kali dan memunculkan keterangan, yakni. Jika nilai lebih besar atau sama dengan 100, Keterangan = “Besar” , jika nilai lebih besar atau sama dengan 50 dan lebih kecil dari 100, maka keterangan = “Sedang” dan selain itu, keterangan = “Kecil”. Banyak karakter yang dicetak dan keterangan disimpan di dalam sebuah tabel.

Langkah-langkah:

1. Membuat form entri.

Form entri hanya memuat sebuah Text Field dan Button.



Gambar XI.1 : Form Entri

Pada form entri tersebut, terdapat menu Entri dan Lihat. Menu Entri dan Lihat selalu ada di setiap halaman aplikasi, oleh karena itu, kita perlu memisahkan file menu dan dipanggil di setiap file lain, dengan menggunakan include.

2. Membuat proses untuk pengolahan data

Setelah data n di entri, maka data dipergunakan untuk mencetak karakter * sebanyak n kali dan dicari keterangan dengan menggunakan aturan yang ditentukan. Pencetakan karakter menggunakan looping dan pencarian keterangan dengan condition.

3. Membuat Tampilan data

Tampilan data menggunakan tabel, berisi data-data yang sudah disimpan di dalam database. Tampilan data memiliki opsi edit dan delete. Edit, data yang terpilih dapat di edit dan Delete, data yang terpilih dapat



No	Nilai	KETERANGAN	Opsi
1	40	*****	Edit Hapus
2	40	*****	Edit Hapus
3	20	*****	Edit Hapus
4	5	*****	Edit Hapus
5	7	*****	Edit Hapus
6	2	**	Edit Hapus
7	2	**	Edit Hapus
8	2	**	Edit Hapus
9	2	**	Edit Hapus
10	2	**	Edit Hapus
11	2	**	Edit Hapus
12	2	**	Edit Hapus
13	2	**	Edit Hapus
14	2	**	Edit Hapus
15	4	****	Edit Hapus
16	2	**	Edit Hapus
17	2	**	Edit Hapus

didelete.

Gambar XI.2 : Tampil Data

4. Informasi Aksi

Setiap Entri dan Delete yang sukses, diinformasikan kepada user. Halaman informasi ini perlu juga kita buat.



Gambar XI.3 : Informasi Aksi

Bab XII

Validasi dan Verifikasi

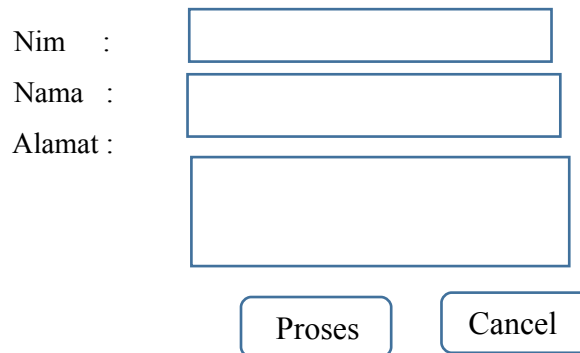
XII.1. Validasi

Di dalam sebuah form, kita perlu melakukan validasi terhadap isian form, seperti sebuah field yang tidak boleh kosong dan harus terisi atau input e-mail yang harus sesuai format e-mail. Validasi di dalam Ilmu Komputer adalah meyakinkan data dimasukkan ke dalam aplikasi agar sesuai format dan input kriteria lainnya (en.wikipedia.org).

Validasi di dalam web dapat menggunakan php dan javascript. Untuk saat ini, kita mencoba melakukan validasi dengan menggunakan JavaScript dan PHP, yaitu validasi inputan, yang harus di isi pada field nim pada form inpuan data mahasiswa.

Penggunaan JavaScript sebagai validasi pada pemrograman web di Client memiliki keuntungan, yakni, validasi lebih user interaktif dibanding dengan PHP, namun PHP lebih aman, karena scriptnya tidak dapat dilihat dan tidak dapat dibatalkan atau di By Pass dengan menggunakan opsi “disabled” di browser, seperti halnya JavaScript.

XII.2 Validasi di form dengan menggunakan JavaScript.



Nim :

Nama :

Alamat :

Gambar XII.1. Form Entri

Pada form di atas, kita buat validasi dengan mewajibkan user untuk mengisi NIM. Untuk itu, kita insert kode Java Script di dalam html. Nama file nya adalah : form_mhs_validasi_php.php

```
1.<!DOCTYPE html>
2.<html>
3. <head>
4.   <title>Form Mahasiswa</title>
5.   <script>
6.     function cek()
7.     {
8.       var nim=myform.vnim.value;
9.       if (nim.length==0)
10.      {
11.        alert ("NIM tidak boleh kosong");
12.        return false;
13.      }
14.    }
15.  </script>
16.</head>
17.<body>
18.  <form name="myform" method="post" onSubmit="return cek()" >
20.    <table>
21.      <caption>Form Mahasiswa</caption>
```

```

22.     <tr>
23.         <td>NIM</td><td>:</td><td><input type="text" name="vnim" size="20" /></td>
24.     </tr>
25.     <tr>
26.         <td>Nama</td><td>:</td><td><input type="text" name="vnama" size="20" /></td>
27.     </tr>
28.     <tr>
29.         <td>Alamat</td><td>:</td><td><textarea rows="3" cols="40"
30.                                     name="valamat"></textarea></td>
31.     </tr>
32.     <tr>
33.         <td colspan="3"><input type="submit" value="Proses" /><input type="button"
34.         value="Cancel" /></td>
35.     </tr>
36. </table>
37. </form>
38. </body>
39. </html>

```

Pada Script baris 5 – 15, kita insert function Java Script untuk validasi.

```

5. <script>
6.     function cek()
7.     {
8.         var nim=myform.vnim.value;
9.         if (nim.length==0)
10.        {
11.            alert ("NIM tidak boleh kosong");
12.            return false;
13.        }
14.    }
15. </script>

```

Pada script HTML, kita insert event **onSubmit**, yang berfungsi untuk menjalankan function ketika form di submit. Saat field NIM tidak diisi dan kita tekan tombol, maka akan keluar pop up dengan kalimat : “NIM tidak boleh kosong”.

XII.3 Validasi pada form dengan menggunakan PHP

Kita akan mencoba membuat validasi pada form di Gambar XII.1, yakni dengan mem validasi NIM agar harus melakukan inputan dengan tidak melanjutkan proses pengolahan data dan memberikan pesan, agar field NIM diisi, jika kosong.

Script lengkapnya adalah sebagai berikut :

```

1.<!DOCTYPE html>
2.<html>
3. <head>
4.     <title>Form Mahasiswa</title>
5. </head>
6. <body>
7.     <form name="myform" method="post"
8.         action="<?php echo $_SERVER['PHP_SELF'];?>">
9.     <table>
10.         <caption>Form Mahasiswa</caption>
11.     <tr>
12.         <td>NIM</td><td>:</td><td><input type="text" name="vnim" size="20" /></td>

```

```

13. </tr>
14. <tr>
15. <td>Nama</td><td>:</td><td><input type="text" name="vnama" size="20" /></td>
16. </tr>
17. <tr>
18. <td>Alamat</td><td>:</td><td><textarea rows="3" cols="40"
19. name="valamat"></textarea></td>
20. </tr>
21. <tr>
22. <td colspan="3"><input type="submit" value="Proses" /><input type="button"
23. value="Cancel" /></td>
24. </tr>
25. </table>
26. </form>
27. </body>
28. <?php
29. if (isset ($_POST["vnim"]))
30. {
31.     $xnim=$_POST["vnim"];
32.     if (strlen($xnim)==0)
33.     {
34.         echo "NIM tidak boleh kosong";
35.     }
36.     else
37.     {
38.         echo "Proses Lanjut";
39.         if (isset ($_POST["vnama"]))
40.         {
41.             $xnama=$_POST["vnama"];
42.         }
43.         if (isset ($_POST["valamat"]))
44.         {
45.             $xalamat=$_POST["valamat"];
46.         }
47.     }
48. }
49. }
50. ?>
51. </html>

```

Pada script no.29 s/d 50, kita menuliskan kode PHP untuk melakukan validasi, yakni dengan menghitung jumlah karakter pada field NIM, jika tidak diisi atau jika panjang karakter field NIM adalah 0, maka muncul pesan : “NIM tidak boleh kosong” dan jika NIM ada isinya, maka proses aplikasi lanjut, dengan dicetak pesan : ”Proses Lanjut”.

XII.4 Verifikasi

Data verification is a process in which different types of data are checked for accuracy and inconsistency after data migration is done (https://en.wikipedia.org/wiki/Data_verification).

Berdasarkan definisi tersebut, di dalam input data, verifikasi adalah memastikan bahwa data yang diinput adalah memang data yang benar, seperti inputan pada form login, yang meminta username dan password. Setelah form menerima inputan username dan password, maka form memproses data tersebut untuk dibandingkan, data inputan username dan password dengan data username dan password di database. Jika kedua data sama, maka user dapat masuk ke aplikasi.

Berikut adalah contoh aplikasi verifikasi Login, dengan verifikasi menggunakan PHP. Sehubungan dalam hal ini, membandingkan data inputan dengan data di database, maka kita tidak dapat menggunakan

JavaScript, karena JavaScript tidak dapat mengakses langsung database, kecuali JavaScript “versi” lain, yakni node.js.

Source code nya adalah sebagai berikut :

```
1.<!DOCTYPE html>
2.<html>
3.<head>
4. <title>Form Login</title>
5.</head>
6.<body>
7. <table align="center">
8.   <form method="POST" action="<?php echo $_SERVER['PHP_SELF'];?>">
9.     <caption>Login</caption>
10.    <tr>
11.      <td>Username</td><td>:</td>
12.      <td><input type="text" name="vusername" size="15" /></td>
13.    </tr>
14.    <tr>
15.      <td>Password</td><td>:</td>
16.      <td><input type="password" name="vpassword" size="15" /></td>
17.    </tr>
18.    <tr>
19.      <td colspan="3"><input type="submit" value="Proses" />
20.      <input type="button" value="Batal" /></td>
21.    </tr>
22.  </form>
23.</table>
24.<?php
25. $dsn='mysql:host=localhost;dbname=test';
26. $username='root';
27. $password='opansan';
28.
29. try {
30.     $dbh=new PDO($dsn,$username,$password);
31. }
32. catch (PDOException $e)
33. {
34.     echo $e->getMessage();
35. }
36.
37. if (isset($_POST["vusername"]))
38. {
39.     $username=$_POST["vusername"];
40. }
41. else
42. {
43.     $username="";
44. }
45.
46. if (isset($_POST["vpassword"]))
47. {
48.     $password=$_POST["vpassword"];
49.     $zpassword=md5($password);
50. }
51. else
52. {
```



```

53.         $xpassword="";
54.         $zpassword="";
55.     }
56.
57.     $querynya="select count(*) from user where username='$username' and
58.         password='$zpassword'";
59.     $jumlah=$dbh->query($querynya)->fetchColumn();
60.     echo "<br/>";
61.     if ($jumlah>0)
62.     {
63.         echo "<center>Lanjut</center>";
64.     }
65.     else
66.     {
67.         echo "<center>Gagal</center>";
68.     }
69. ?>
70. </body>
71. </html>

```

Kita membutuhkan tabel “user” dengan struktur sebagai berikut :

No	Nama field	Tipe Data
1	username	varchar (30)
2	password	Varchar (100)

Alur program di atas, adalah, jika username dan password yang dimasukkan ada di salah satu record pada field username dan password di atas, maka tercetak kata “Lanjut”. Sebaliknya, jika tidak ada datanya, maka tercetak kata “Gagal”. Password yang dibandingkan adalah password yang sudah dengan format enkripsi md5.

Source code dapat anda peroleh pada Git di url, dengan keterangan awal “Bab 12” :
https://github.com/sofwanbl/modul_web_programming

Bab XIII

Pencarian

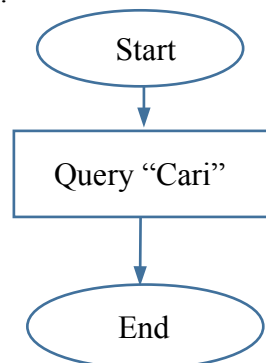
Pencarian atau Searching adalah proses menemukan sebuah nilai yang diberikan di dalam sebuah list nilai [1]. Di dalam pencarian data di database, nilai yang diberikan, diinput user pada sebuah text field, lalu dikirimkan nilainya untuk menjadi nilai untuk syntax “where” di dalam SQL. Contoh, pada program penyimpanan nilai di Bab XI (Membuat aplikasi CRUD sederhana), dicari nilai: 15, yang disimpan didalam sebuah variabel, misalkan “\$xnilai”. Lalu, pada SQL, nilai 15 dimasukkan menjadi nilai yang dicari di field “nilai” pada tabel “contoh_program”. SQL nya akan menjadi seperti ini : “select * from contoh_program where nilai =’\$xnilai’”, \$xnilai = 15.

Form yang kita gunakan untuk implementasi Searching adalah form Lihat Data pada program CRUD di Bab XI. Pertama-tama, kita akan buat text field searching dan tombol “Cari” nya. Setelah dimodifikasi, maka tampilan akan seperti ini :

No	Nilai	KARAKTER	KETERANGAN	Opsi
1	15	=====	Kecil	Edit Hapus
2	4	=====	Kecil	Edit Hapus
3	0	=====	Kecil	Edit Hapus

Gambar XIII.1 Tabel Pencarian

Alur programnya adalah sebagai berikut :



Script lengkapnya adalah sebagai berikut:

```
1.<?php
2.include ("hubung_db.php");
3.include ("header.php");
4.??>
5. <form method="post" action="<?php $_SERVER['PHP_SELF'];?>" >
6. Cari : <input type="text" name="vcari" size="5" /><input type="submit" value="Cari" />
7. </form>
8. <br/>
9. <table border='1' class="tabel_tampil">
```

```

10. <th>No</th><th>Nilai</th><th>KARAKTER</th><th>KETERANGAN</th><th>Opsi</th>
11. <?php
12.     if (isset($_POST["vcari"]))
13.     {
14.         $xcari=$_POST["vcari"];
15.     }
16.     else
17.     {
18.         $xcari="";
19.     }
20.
21.     if (strlen($xcari)>0)
22.     {
23.         $katawhere=" where nilai='$xcari'";
24.     }
25.     else
26.     {
27.         $katawhere="";
28.     }
29. $query="select * from contoh_program". $katawhere;
30. $result=$dbh->query($query);
31. $no=1;
32. foreach($result as $row)
33.     {
34.         echo "<tr>";
35.         echo "<td>$no</td><td>". $row['nilai']. "</td><td>". $row['karakter']. "</td>";
36.         echo "<td>". $row['keterangan']. "</td>";
37.         echo "<td><a href='edit.php?id=".$row['id']."'>Edit</a> |";
38.         echo "<a href='proses_hapus.php?id=".$row['id']."'>Hapus</a></td>";
39.         echo "</tr>";
40.         $no++;
41.     }
42. ?>
43.</table>
44.</form>
45.</div>
46. <?php include ("footer.php"); ?>

```

Penjelasan :

Pada source code 12- 29, adalah script untuk melakukan pencarian. Diawali dengan mengambil nilai “Cari” dari text field “Cari”.

```

12. if (isset($_POST["vcari"]))
13.     {
14.         $xcari=$_POST["vcari"];
15.     }
16.     else
17.     {
18.         $xcari="";
19.     }

```

Lalu memberi query Where, jika karakter “cari” diisi.

```

20. if (strlen($xcari)>0)
22.     {
23.         $katawhere=" where nilai='$xcari'";
24.     }

```

```
25. else
26. {
27.     $katawhere="";
28. }
```

Dan menambahkan query where ke dalam query pencarian.

```
29. $query="select * from contoh_program". $katawhere;
```

File lengkap beserta SQL create table nya, dapat anda peroleh di link :
<https://goo.gl/BaPLhp>

Bab XIV

Pagination

Saat tampilan data sudah banyak, maka halaman webnya akan semakin panjang menampilkan data dan program akan bekerja keras untuk menampilkan data. User pun akan melihat tampilan data dengan menscroll ke bawah layar dengan cukup lama.

Hal ini dapat di atasi dengan membatasi jumlah tampilan data yang tampil pada setiap halamannya misalkan . Hal ini dikenal dengan nama **Pagination** atau **Paging**. Konsep Pagination, yaitu menampilkan data untuk jumlah tertentu pada setiap halaman dengan menggunakan limit di query. Kita akan menggunakan tampilan pada program CRUD yang sudah pernah kita buat sebelumnya. Hasil akhir dari halaman yang sudah diberi pagination adalah seperti di bawah ini :



Gambar XIV.1 : Pagination

Pada gambar di atas, kita memberikan link perpindahan halaman di bawah tabel, yang saat ini, jumlah total data di atas adalah 4 buah dan per halamannya, kita buat tampil per 2 record. Pembuatan session ini mengambil referensi dari halaman “Membuat Paging dengan PHP dan MySQL [2]”.

Alur program pagination adalah berikut :

1. Menentukan jumlah record yang muncul disetiap halaman. Pada contoh tersebut adalah, 2.
2. Menentukan nilai mulai halaman, dengan cara, jika halaman pertama yang tampil, maka nilai mulai adalah 0 dan jika halaman kedua atau selanjutnya yang tampil, maka nilai mulai adalah halaman - 1 x 2 . 2 adalah jumlah record per halaman yang muncul.
3. Menghitung jumlah total record. Contoh di atas, jumlah total record = 4.
4. Menghitung jumlah total record dibagi jumlah record per halaman. Contoh di atas, 4/2. Kita menggunakan fungsi ceil() untuk pembulatan ke atas, sehingga hasilnya adalah 2. Nilai 2 ini kita pergunakan untuk mencetak jumlah halaman di bawah tabel, karena ada 2, maka kita mencetak nilai 1 dan 2. Nilai ini merupakan link yang dapat diklik.

Program lengkapnya adalah sebagai berikut :

```
1.<?php
2.include ("hubung_db.php");
3.include ("header.php");
4.?>
5. <form method="post" action="<?php $_SERVER['PHP_SELF'];?>" >
6. Cari : <input type="text" name="vcari" size="5" /><input type="submit" value="Cari" />
7. </form>
8. <br/>
9. <table border='1' class="tabel_tampil">
10. <th>No</th><th>Nilai</th><th>KARAKTER</th><th>KETERANGAN</th><th>Ops</th>
11. <?php
```

```

12.
13. if (isset($_POST["vcari"]))
14. {
15.     $xcari=$_POST["vcari"];
16. }
17. else
18. {
19.     $xcari="";
20. }
21.
22. if (strlen($xcari)>0)
23. {
24.     $katawhere=" where nilai='$xcari'";
25. }
26. else
27. {
28.     $katawhere="";
29. }
30.
31. // Cari Total //
32. $halaman=2;
33. if (isset($_GET['halaman']))
34. {
35.     $shal=(int)$_GET['halaman'];
36. }
37. else
38. {
39.     $shal=1;
40. }
41.
42. if ($shal>1)
43. {
44.     $mulai=($shal-1)*$halaman;
45. }
46. else
47. {
48.     $mulai=0;
49. }
50.
51. $query_total="Select count(*) from contoh_program". $katawhere;
52. $query_totalnya=$dbh->query($query_total);
53. $total=$query_totalnya->fetchColumn();
54. $halamannya=ceil($total/$halaman);
55. $query="select * from contoh_program". $katawhere ." limit $mulai, $halaman";
56. $result=$dbh->query($query);
57.
58. $no=$mulai+1;
59. foreach($result as $row)
60. {
61.     echo "<tr>";
62.     echo "<td>$no</td><td>". $row['nilai']. "</td><td>". $row['karakter']. "</td>";
63.     echo "<td>". $row['keterangan']. "</td>";
64.     echo "<td><a href='edit.php?id=" . $row['id'] . "'>Edit</a> | <a href='proses_hapus.php?";
65.     id=" . $row['id'] . "'>Hapus</a></td>";
66.     echo "</tr>";
67.     $no++;
68. }

```

```
69.     ?>
70. </table>
71. </form>
72. <br/>
73. <center>
74. <?php
75. for ($i=1;$i<=$shalamannya;$i++)
76.     { ?>
77.         <a href="?halaman=<?php echo $i;?>"><?php echo $i;?></a>
78. <?php
79.     }
80. ?>
81. </center>
82. <br/>
83.</div>
84.<?php include ("footer.php"); ?>
```

Source code pada Bab ini tidak tersedia dan akan tersedia secara penuh setelah digabungkan dengan topik : session, yang merupakan topik pada Bab selanjutnya, Bab XV.

Bab XV

Session dan program CRUD

Pada bab XV ini, kita akan membahas mengenai Session di dalam PHP. Session adalah cara sederhana untuk menyimpan data untuk user individu dengan menggunakan nama session. Nama session dikirim ke dalam browser menggunakan session cookies. ID session ini juga dapat diambil. Untuk menggunakan session, pertama kali kita mengaktifkan session dengan perintah: `session_start()`. Selanjutnya kita dapat mengirim session dengan perintah `$_SESSION['nama_session']`. Session dapat dihapus dengan menggunakan perintah: `unset(session)`.

Contoh :

```
1.<?php
2. session_start();
3. $_SESSION['nama']="Opan";
4. echo $_SESSION['nama'];
5. unset($_SESSION['nama']);
6. echo $_SESSION['nama'];
7. ?>
```

Pada source code di atas, kita membuat session 'nama', mencetak session tersebut dan kemudian menghilangkan session tersebut, lalu dicetak kembali. Output program di atas, di komputer saya adalah sebagai berikut :

Opan

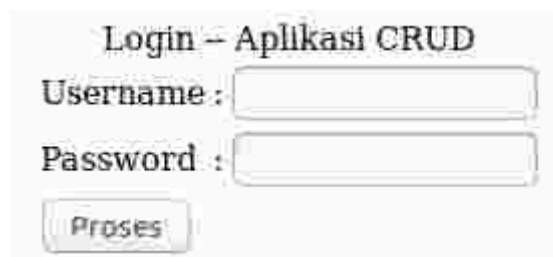
Notice: Undefined index: nama in /var/www/html/contoh_session.php on line 6

Kata "Opan", adalah nilai session "nama". Lalu session tersebut dihilangkan dan menjadi error, ketika dicetak kembali.

Session sering digunakan untuk Login dan Logout. Pada Login, biasanya kita diminta untuk memasukkan username dan password, yang jika username dan password tersebut valid, maka kita boleh memasuki halaman aplikasi. Session menyimpan login ini dan akan dihapus atau didestroy ketika user melakukan logout. Keberlakuan session memiliki waktu tertentu, yang jika waktu tersebut sudah habis, maka session secara otomatis akan dihapus.

Manfaat session di dalam Login adalah, ketika kita keluar program tanpa melakukan logout atau menghapus session, maka ketika masuk kembali ke program, maka akan bisa, tidak ditanyai username dan password di halaman Login. Dan ketika kita sudah melakukan Logout, maka kita tidak bisa masuk ke halaman aplikasi tanpa melakukan Login, termasuk mem by pass URL aplikasi.

Untuk menerapkan session di dalam aplikasi CRUD yang pernah kita buat sebelumnya, kita tambahkan halaman login, seperti di bawah ini :



Login - Aplikasi CRUD

Username :

Password :

Gambar XV.1 Login

Source code Login tersebut adalah sebagai berikut :

```
1.<!DOCTYPE html>
2.<html>
3.<table id="tabel_entri" align="center">
4.<form method="post" action="proses.php">
5.<caption>Login -- Aplikasi CRUD</caption>
6.<tr>
7.<td>Username</td><td>:</td><td><input type="text" name="vusername" size="10" /></td>
8.</tr>
9.<tr>
10.<td>Password</td><td>:</td>
11.<td><input type="password" name="vpassword" size="10" /></td>
12.</tr>
13.<tr>
14.<td colspan="3"><input type="submit" value="Proses" /></td>
15.</tr>
16.</form>
17.</table>
18.</div>
19.<?php include("footer.php"); ?>
```

Setelah Login terbentuk, maka kita dapat menggunakan session untuk menyimpan nilai saat Login dan menghapus atau destroy session saat Logout. Pada awal file, kita perlu mengecek apakah user sudah login atau belum, untuk menghindari bypass ke dalam aplikasi.

Session dimulai dengan perintah `session_start()`. Untuk mengecek apakah session sudah dimulai atau belum, kita perlu mengecek dengan perintah :

```
if(session_status() == PHP_SESSION_NONE)
```

Jika session belum dimulai, kita perlu memberikan perintah `session_start()`.

Source code lengkap aplikasi CRUD dengan Login dan Logout dengan session, dapat anda peroleh dengan mendownload file : contoh_aplikasi_pdo_full.zip di : http://kuliah.sofwan.net/?page_id=264 atau di https://github.com/sofwanbl/modul_web_programming

Penutup

Demikian lah tutorial pemrograman web dengan menggunakan PHP dan MySQL. Untuk penjelasan mengenai PHP lebih lanjut, anda dapat melihat dokumentasi dari situs resmi PHP, yakni : www.php.net, dan untuk mengenal Mysql, anda dapat melihat ke www.mysql.com. Untuk mengetahui lebih lanjut mengenai MariaDB, silahkan anda lihat ke www.mariadb.org.

Referensi

- [1] PDO Functions, <http://www.nusphere.com/kb/phpmanual/ref.pdo.htm>
- [2] Muzanni, "Membuat Paging dengan PHP dan MySQL",
<https://www.malasngoding.com/membuat-paging-dengan-php-dan-mysql/>, waktu akses : 18 Desember 2018
- [3] "Object Oriented PHP for Beginners", <https://code.tutsplus.com/tutorials/object-oriented-php-for-beginners>—net-12762
- [4] "php website", php.net
- [5] How does PHP work ? . <https://www.quora.com/How-does-PHP-work>, waktu akses : 09 Maret 2019

Biografi Penulis

Akhmad Sofwan - Menyelesaikan pendidikan Sarjana di Teknik Informatika Universitas Budi Luhur ,tahun 2001 dan Magister Ilmu Komputer, Fakultas Ilmu Komputer ,Universitas Indonesia, tahun 2017. Penulis saat ini adalah seorang Freelance Software Developer. Selain itu, penulis juga mengajar mata kuliah ilmu komputer di Perguruan Tinggi Raharja, Tangerang dan di Program Vokasi, Universitas Indonesia. Penulis juga aktif di Pusat Kajian Biostatistik dan Informatika Kesehatan (PKBIK) Fakultas Kesehatan Masyarakat, Universitas Indonesia sebagai Software Developer dan Peneliti. Bidang penelitian beliau adalah Machine

Learning.