

PRAKTIKUM 4

PEMROSESAN FORM

1. Tujuan Praktikum

Setelah mengikuti praktikum ini mahasiswa diharapkan dapat memahami cara untuk:

- a. Membaca isi form dengan PHP menggunakan method GET maupun POST
- b. Melakukan validasi input data dengan PHP
- c. Menampilkan isi kembali form

2. Tools

Tools yang dibutuhkan untuk melakukan praktikum ini adalah Apache web server, PHP, MySQL dan text editor.

3. Langkah-Langkah Praktikum

3.1. Praktikum 2 : Membaca isi form dengan PHP menggunakan method GET

1. Buatlah form input data mahasiswa seperti berikut menggunakan kode HTML dengan ketentuan berikut:
 - Gunakan submission method GET.
 - Pemrosesan form dilakukan pada halaman yang sama, sehingga atribut action pada elemen <form> dikosongkan.
 - Berikan nama pada setiap elemen input yang digunakan pada atribut name.
 - Elemen input yang masukannya berupa single value diberi nama berupa teks biasa, sedangkan yang masukannya dapat berupa multiple values diberi nama sebagai array.

Form Mahasiswa

Nama:

Email:

Kota/ Kabupaten:

Semarang

Jenis Kelamin:

☐ Pria

☐ Wanita

Peminatan:

☐ Coding

☐ UX Design

☐ Data Science

Submit

Reset

Berikut adalah potongan source code yang digunakan untuk membuat form seperti gambar di atas.

user_form_get.php

```
24 <div class="form-group">
25   <label for="nama">Nama:</label>
26   <input type="text" class="form-control" id="nama" name="nama" maxlength="50">
27 </div>
28 <div class="form-group">
29   <label for="email">Email:</label>
30   <input type="email" class="form-control" id="email" name="email">
31 </div>
32 <div class="form-group">
33   <label for="kota">Kota/ Kabupaten:</label>
34   <select id="kota" name="kota" class="form-control">
35     <option value="Semarang">Semarang</option>
36     <option value="Jakarta">Jakarta</option>
37     <option value="Bandung">Bandung</option>
38     <option value="Surabaya">Surabaya</option>
39   </select>
40 </div>
41 <label>Jenis Kelamin:</label>
42 <div class="form-check">
43   <label class="form-check-label">
44     <input type="radio" class="form-check-input" name="jenis_kelamin" value="pria"> Pria
45   </label>
46 </div>
47 <div class="form-check">
48   <label class="form-check-label">
49     <input type="radio" class="form-check-input" name="jenis_kelamin" value="wanita">
      Wanita
50   </label>
51 </div>
52 <br>
53 <label>Peminatan:</label>
54 <div class="form-check">
55   <label class="form-check-label">
56     <input type="checkbox" class="form-check-input" name="minat[]" value="coding"> Coding
57   </label>
58 </div>
59 <div class="form-check">
60   <label class="form-check-label">
61     <input type="checkbox" class="form-check-input" name="minat[]" value="ux_design"> UX
      Design
62   </label>
63 </div>
64 <div class="form-check">
65   <label class="form-check-label">
66     <input type="checkbox" class="form-check-input" name="minat[]" value="data_science">
      Data Science
67   </label>
68 </div>
69 <br>
70 <!-- submit, reset dan button -->
71 <button type="submit" class="btn btn-primary" name="submit" value="submit"> Submit
72 </button>
73 <button type="reset" class="btn btn-danger"> Reset</button>
</form>
```

2. Tambahkan kode PHP seperti berikut setelah kode untuk membuat form yang bertujuan untuk membaca dan menampilkan isian yang dimasukkan ke form tersebut. Karena menggunakan method GET, maka isi form diakses melalui array \$_GET.

```

75 <?php
76 if (isset($_GET["submit"])){
77     echo "<h3>Your Input:</h3>";
78     echo 'Nama = ' . $_GET['nama'] . '<br />';
79     echo 'Email = ' . $_GET['email'] . '<br />';
80     echo 'Kota = ' . $_GET['kota'] . '<br />';
81     echo 'Jenis Kelamin = ' . $_GET['jenis_kelamin'] . '<br />';
82     echo 'Minat = ' . $_GET['minat'] . '<br />';
83
84     $minat = $_GET['minat'];
85     if (!empty($minat)){
86         echo 'Peminatan yang dipilih: ';
87         foreach($minat as $minat_item){
88             echo '<br />' . $minat_item;
89         }
90     }
91 }
92 ?>

```

3. Buatlah sebuah folder bernama form di folder htdocs, lalu simpan file tersebut di folder form dengan nama user_form_get.php.
4. Jalankan file tersebut di web browser, perhatikan alamat url setelah form diisi dan di-submit.
5. Perhatikan pada field hobby, mengapa penamaan dan pengaksesannya berbeda dengan field lainnya?

3.2. Praktikum 2 : Membaca isi form dengan PHP menggunakan method POST

1. Ubah file user_form_get.php, ganti nilai atribut method pada elemen <form> dengan POST.

```

21 <form method="POST" autocomplete="on" action="">

```

2. Lalu, ubahlah kode PHP yang digunakan untuk membaca isi form dari \$_GET menjadi \$_POST, seperti berikut:

```

74 <?php
75 if (isset($_POST["submit"])){
76     echo "<h3>Your Input:</h3>";
77     echo 'Nama = ' . $_POST['nama'] . '<br />';
78     echo 'Email = ' . $_POST['email'] . '<br />';
79     echo 'Kota = ' . $_POST['kota'] . '<br />';
80     echo 'Jenis Kelamin = ' . $_POST['jenis_kelamin'] . '<br />';
81     $minat = $_POST['minat'];
82     if (!empty($minat)){
83         echo 'Peminatan yang dipilih: ';
84         foreach($minat as $minat_item){
85             echo '<br />' . $minat_item;
86         }
87     }
88 }
89 ?>

```

3. Simpan file tersebut dengan nama baru, yaitu user_form_post.php.
4. Jalankan file tersebut di web browser, perhatikan alamat url setelah form diisi dan di-submit.
5. Apa perbedaan antara method GET pada file user_form_get.php dan method POST pada file user_form_post.php?
6. Manakah yang lebih baik, method GET atau POST? Uraikan jawaban Anda.

3.3. Praktikum 3 : Validasi form dengan PHP

Form pada file user_form_get.php maupun user_form_post.php belum ditambahkan penanganan validasi. Sebelumnya, tambahkan field alamat menggunakan elemen <textarea> setelah field email pada file user_form_post.php, sehingga tampilannya menjadi seperti berikut:

Form Mahasiswa

Nama:

Email:

Alamat:

Kota/ Kabupaten:

Semarang

1. Tambahkan kode PHP berikut untuk menambahkan aturan validasi dan letakkan di bagian atas, sebelum kode untuk membuat form.

```

20 <body>
21 <?php
22 if (isset($_POST['submit'])) {
23     //validasi nama: tidak boleh kosong, hanya dapat berisi huruf dan spasi
24     $nama = test_input($_POST['nama']);
25     if (empty($nama)) {
26         $error_nama = "Nama harus diisi";
27     } elseif (!preg_match("/^[a-zA-Z ]*$/", $nama)) {
28         $error_nama = "Nama hanya dapat berisi huruf dan spasi";
29     }
30     //validasi email: tidak boleh kosong, format harus benar
31     $email = test_input($_POST['email']);
32     if ($email == '') {
33         $error_email = "Email harus diisi";
34     } elseif (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
35         $error_email = "Format email tidak benar";
36     }
37     //validasi alamat: tidak boleh kosong
38     $alamat = test_input($_POST['alamat']);
39     if ($alamat == '') {
40         $error_alamat = "Alamat harus diisi";
41     }
42     //validasi jenis kelamin: tidak boleh kosong
43     if ($jenis_kelamin == '') {
44         $error_jenis_kelamin = "Jenis kelamin harus diisi";
45     }
46     //validasi kota: tidak boleh kosong
47     $kota = $_POST['kota'];
48     if ($kota == '' || $kota == 'kota') {
49         $error_kota = "Kota harus diisi";
50     }
51     //validasi minat: tidak boleh kosong
52     $minat = $_POST['minat'];
53     if (empty($minat)) {
54         $error_minat = "Peminatan harus dipilih";
55     }

```

```

58 function test_input($data) {
59     $data = trim($data);
60     $data = stripslashes($data);
61     $data = htmlspecialchars($data);
62     return $data;
63 }
64 ?>

```

2. Tambahkan kode berikut pada setiap elemen input form untuk menampilkan pesan error. Berikut adalah contoh pada field nama dan email, tambahkan pula untuk field lainnya dengan cara yang sama.

```

70 <div class="form-group">
71     <label for="nama">Nama:</label>
72     <input type="text" class="form-control" id="nama" name="nama" maxlength="50">
73     <div class="error"><?php if(isset($error_nama)) echo $error_nama;?></div>
74 </div>
75 <div class="form-group">
76     <label for="email">Email:</label>
77     <input type="email" class="form-control" id="email" name="email">
78     <div class="error"><?php if(isset($error_email)) echo $error_email;?></div>
79 </div>

```

3. Simpan file tersebut dengan nama baru user_form_post1.php.
4. Jalankan file tersebut di web browser, kosongkan sebagian field, lalu submit, perhatikan apakah pesan error sudah ditampilkan dengan benar? Perhatikan apakah field yang sudah terisi sebelumnya dapat ditampilkan kembali? Mengapa?
5. Apakah peran fungsi test_input(\$data) pada proses validasi tersebut? Mengapa fungsi tersebut hanya dikenakan pada isian nama, email dan alamat, namun tidak dikenakan pada isian jenis kelamin, kota dan peminatan?
6. Bagaimanakah aturan validasi yang digunakan untuk mengecek agar isian tidak boleh kosong?

3.4. Praktikum 4 : Menampilkan kembali isi form

Pada file form_customer2.php jika terjadi kesalahan maka isian form akan kembali kosong lagi. Untuk menampilkannya kembali tambahkan kode berikut pada setiap elemen input yang bersesuaian. Perhatikan bahwa cara yang digunakan berbeda untuk elemen input yang berbeda.

Field nama (perhatikan kode yang diblok warna abu-abu):

```

69 <div class="form-group">
70     <label for="nama">Nama:</label>
71     <input type="text" class="form-control" id="nama" name="nama" maxlength="50" value="
72     <?php if(isset($nama)) {echo $nama;} ?>">
73     <div class="error"><?php if(isset($error_nama)) echo $error_nama;?></div>
74 </div>

```

Field alamat:

```

79 <div class="form-group">
80     <label for="alamat">Alamat:</label>
81     <textarea class="form-control" id="alamat" rows="3" name="alamat">
82     <?php if(isset($alamat)) {echo $alamat;} ?></textarea>
83     <div class="error"><?php if(isset($error_alamat)) echo $error_alamat;?></div>
84 </div>

```

Field kota:

```

87 <select id="kota" name="kota" class="form-control">
88     <option value="Semarang" <?php if (isset($kota) && $kota=="Semarang") echo
89     'selected="true"';?>>Semarang</option>
90     <option value="Jakarta" <?php if (isset($kota) && $kota=="Jakarta") echo
91     'selected="true"';?>>Jakarta</option>

```

Lengkapi untuk option yang lainnya.

Field jenis kelamin:

```
97 <div class="form-check">
98 <label class="form-check-label">
99 <input type="radio" class="form-check-input" name="jenis_kelamin" value="pria"
    <?php if (isset($jenis_kelamin) && $jenis_kelamin=="pria") echo "checked";?>Pria
100 </label>
```

Lengkapi untuk opsi radio button jenis kelamin yang lainnya.

Field peminatan:

```
115 <div class="form-check">
116 <label class="form-check-label">
117 <input type="checkbox" class="form-check-input" name="minat[]" value="ux_design"
    <?php if (isset($minat) && in_array('ux_design',$minat)) echo 'checked';?>UX Design
118 </label>
119 </div>
```

Lengkapi untuk pilihan checkbox minat yang lainnya.

Setelah semua selesai, simpan dengan nama user_form_post2.php. Jalankan pada browser dan amatai perbedaanya dengan user_form_post1.php. Periksa apakah isi form yang sebelumnya sudah ditampilkan dengan benar.

Latihan:

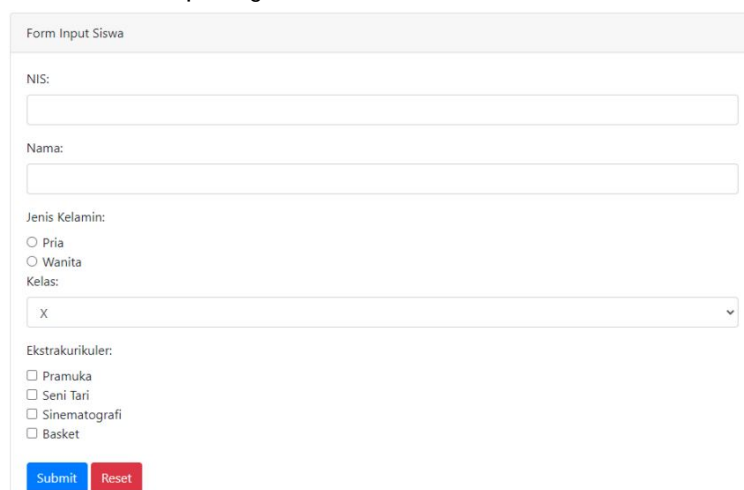
Selesaikanlah file 4 file yang langkah-langkahnya telah diuraikan pada bagian sebelumnya, yaitu:

- user_form_get.php
- user_form_post.php
- user_form_post1.php
- user_form_post2.php

Simpan keempat file tersebut dalam folder form.

4. Tugas

Buatlah form seperti gambar berikut:



Aturan validasi:

- Semua field harus diisi.
- NIS terdiri atas 10 karakter dan hanya boleh berisi angka 0..9.
- Jika siswa kelas X atau XI, maka program menampilkan pilihan ekstrakurikuler. Siswa wajib memilih kegiatan ekstrakurikuler yang diminati, minimal 1 maksimal 3. Jika kelas XII siswa tidak boleh mengikuti kegiatan ekstrakurikuler, sehingga program tidak perlu menampilkan kegiatan ekstrakurikuler.

===== Selamat Berlatih, Semoga Sukses =====