**PERTEMUAN 7**

**PHP (*Form* dan *File*)**

1. Tujuan Pembelajaran

Pada pertemuan ini akan dijelaskan penggunaan *Form* dan *File* pada PHP. Setelah selesai materi pada pertemuan ini, mahasiswa mampu memahami konsep php (form dan file), dan mengimplementasikan konsep php ke dalam web dinamis.

1. Materi

**Pengenalan *Form***

*Form* merupakan salah satu elemen dalam HTML yang digunakan untuk menerima input dari pengunjung. Melalui *form* pengunjung sebuah halaman *web* dapat melakukan interaksi dengan *web*. Pengunjung dapat memasukan data melalui elemen-elemen input seperti *textfield*, *checkbox*, *radio button*, *textarea*, *submit button*, *dropdown* dan lain-lain. Saat ini hampir seluruh *website* modern memerlukan *form* sebagai fitur utama. *Form* digunakan pada *form* pendaftaran, *form* *login*, *form* pendaftaran dan sebagainya. Sintaks dasar *form* adalah sebagai berikut:

|  |
| --- |
| <form>  Elemen input form  </form> |

*Form* membuat sebuah halaman website menjadi lebih interaktif dan dinamis. Dengan *form* pengguna dapat memasukkan data-datanya melalui elemen input *form*. Elemen-elemen input yang terdapat pada *form* antara lain:

1. *textfield*, adalah elemen yang digunakan untuk memasukkan teks dalam satu baris. Lebar karakter defaultnya adalah 20 karakter.
2. *password field*, adalah elemen yang digunakan untuk memasukkan satu baris teks dengan format password/chiphertext.
3. *checkbox*, adalah elemen yang dapat digunakan untuk memasukkan lebih dari satu pilihan.
4. *radio button*, adalah elemen yang digunakan untuk memasukkan pilihan (hanya dapat memilih salah satu pilih.
5. *textarea*, adalah elemen yang digunakan untuk memasukkan lebih dari satu baris teks.
6. *submit button*, adalah elemen untuk mengirimkan data dari halaman web ke server.

**Praktikum Bagian 1. *Form Handling***

Untuk memroses suatu form digunakan 2 atribut pada elemen form, yaitu atribut action dan atribut method. Atribut action diisi dengan alamat halaman PHP yang digunakan untuk memroses data form tersebut. Atribut method akan menentukan bagaimana cara pengiriman data pada server. Terdapat 2 method yang dapat digunakan pada form, yaitu: POST dan GET. Method POST akan mengirimkan data atau nilai langsung ke action untuk ditampung, tanpa ditampilkan pada URL. Untuk menampung data atau nilai digunakan variabel $\_POST. Gambar 1 menunjukkan ilustrasi penggunaan method POST:

Graphical user interface, application

Description automatically generated

Gambar 1. Ilustrasi penggunaan *method* POST

Diagram

Description automatically generatedBerbeda dengan method POST, method GET akan menampilkan data atau nilai pada URL, kemudian ditampung oleh action. Variabel yang digunakan adalah $\_GET. Gambar 2 menunjukkan ilustrasi penggunaan method GET:

Gambar 2. Ilustrasi penggunaan method GET

Data atau nilai yang dikirimkan menggunakan method POST tidak terbatas, sedangkan data atau nilai yang dikirim menggunakan method GET tidak boleh lebih dari 2047 karakter. Method POST digunakan untuk mengirimkan data yang bersifat penting dan data yang tidak boleh diketahui orang lain, seperti password dan sebagainya.

Untuk memahami form handling, ikuti langkah-langkah praktikum berikut ini:

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Buat file baru dengan nama form.html di dalam direktori praktik\_php, kemudian ketikkan kode berikut: |
| 2 | Buat file baru dengan nama postForm.php di dalam direktori praktik\_php, kemudian ketikkan kode berikut: |
| 3 | Simpan file dan jalankan kode program dengan mengetikkan  localhost/dasarWeb/praktik\_php/form.html |
| 4 | Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu! (soal no 1) |
| 5 | Buatlah file baru dengan nama form\_1.html dan salin kode program pada langkah ke-1. Modifikasi kode program dengan mengubah atribut form pada baris ke-5 menggunakan action getForm.php dan method GET, sehingga kode program menjadi seperti berikut: |
| 6 | Buat file baru dengan nama getForm.php di dalam direktori praktik\_php, kemudian ketikkan kode berikut: |
| 7 | Simpan file dan jalankan kode program dengan mengetikkan  localhost/dasarWeb/praktik\_php/form\_1.html |
| 8 | Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu! (soal no 2) |

**Praktikum Bagian 2. Validasi *Form***

Nilai yang telah diinput oleh *user* atau pengunjung *web*, tidak bisa begitu saja di simpan langsung ke dalam *database*. Karena kita tidak tahu apakah nilai tersebut telah sesuai dengan nilai yang kita kehendaki. Misalkan apakah nilai tersebut harus berupa angka, atau hanya bisa berupa huruf, atau apakah hanya bisa diinput dalam *range* tertentu saja.

Validasi pertama yang bisa dilakukan adalah memeriksa apakah elemen-elemen *form* tersebut tersedia atau tidak. Fungsi isset() digunakan untuk memvalidasi apakah elemen *form* telah didefinisikan atau telah di set sebelumnya. Fungsi isset() bernilai *true* jika sebuah variabel telah didefinisikan dan *false* jika variabel tersebut belum dibuat.

Ikuti langkah-langkah praktikum berikut ini untuk memahami fungsi isset():

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Buka browser kemudian jalankan file getForm.php dengan menggetikkan perintah localhost/dasarWeb/praktik\_php/getForm.php |
| 2 | Amati hasil yang ditampilkan |
| 3 | Buka file getForm.php yang telah dibuat sebelumnya dalam direktori praktik\_php. Modifikasi kode program dengan menambahkan fungsi isset(), sehingga kode program menjadi seperti berikut: |
| 4 | Simpan file dan jalankan kode program dengan mengetikkan  localhost/dasarWeb/praktik\_php/getForm.php |
| 5 | Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu! (soal no 3) |

**Praktikum Bagian 3. *Form Required***

Ketika mengisi *form* ada beberapa informasi yang harus ada atau wajib diisi oleh user, misalnya informasi username. Sebelum mengirim *file* atau proses submit *file*, diperlukan proses validasi untuk memeriksa apakah sebuah elemen *form* sudah terisi atau belum. Dalam PHP untuk memvalidasri apakah elemen *form* sudah terisi atau belum digunakan fungsi empty(). Fungsi empty() akan bernilai *true* jika variabel tersebut belum diisi, dan bernilai *false* jika variabel tersebut telah terisi. Dengan menggunakan stuktur IF dan fungsi empty(), kita bisa membuat logika validasi objek form mana saja yang dianggap perlu (harus diisi) dan mana yang boleh dikosongkan, kemudian menampilkan pesan error yang sesuai.

Ikuti langkah-langkah berikut untuk memahami penggunaan fungsi empty():

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Buat file form\_2.php dalam direktori praktik\_php kemudian ketikkan kode program berikut: |
| 2 | Buat file prosesForm\_2.php dalam direktori praktik\_php kemudian ketikkan kode program berikut: |
| 3 | Simpan file dan jalankan kode program dengan mengetikkan  localhost/dasarWeb/praktik\_php/form\_2.php |
| 4 | Lakukan eksperimen berikut:  a. tanpa mengisi form langsung klik tombol “kirim”  b. isi *field* nama dan kosongkan email, kemudian klik tombol “kirim”  c. isi *field* email dan kosongkan nama, kemudian klik tombol “kirim”  d. isi *field* nama dan email, kemudian klik tombol “kirim” |
| 5 | Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu! (soal no 4) |
| 6 | Jelaskan apakah kegunaan fungsi header() (soal no 5) |

**Praktikum Bagian 4. *Form URL/e-mail***

Validasi form selanjutnya adalah memastikan bahwa data yang diinput oleh user telah sesuai. Misalnya untuk nama harus diberi input berupa huruf bukan angka dan email harus memenuhi aturan penulisan email, yaitu “alamatemail@mail.com”. Di dalam PHP terdapat fungsi yang dapat digunakan dalam manipulasi *regular expression* (regex), yaitu fungsi preg\_match(). Fungsi preg\_match() digunakan untuk mencocokan pola dalam data string. Fungsi preg\_match() akan mengembalikan data bernilai *TRUE* jika terdapat pola yang sesuai dan *FALSE* jika tidak ada pola yang sesuai. PHP memiliki fungsi built-in yang khusus digunakan untuk validasi alamat email, yaitu filterval().

Ikuti langkah-langkah berikut untuk membuat validasi fom untuk *URL/e-mail*:

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Buat file baru dengan nama form\_3.php kemudian salin kode program form\_2.php dari praktikum bagian sebelumnya |
| 2 | Modifikasi kode program sehingga kode program menjadi seperti berikut:  D:\tempsnip.png  D:\tempsnip.png |
| 3 | Buat file baru dengan nama prosesForm\_3.php kemudian salin kode program prosesForm\_2.php dari praktikum bagian sebelumnya |
| 4 | Modifikasi kode program php menambahkan fungsi preg\_match() dan filter\_var() sehingga kode program menjadi seperti berikut:  D:\tempsnip.png |
| 5 | Simpan file dan jalankan kode program dengan mengetikkan  localhost/dasarWeb/praktik\_php/form\_3.php |
| 6 | Lakukan eksperimen berikut:  a. tanpa mengisi form langsung klik tombol “kirim”  b. isi *field* nama dengan angka kemudian klik tombol “kirim”  c. isi *field* nama dengan string kemudian klik tombol “kirim”  d. isi *field* nama dan email dengan “namaku” kemudian klik tombol “kirim”  e. isi *field* nama dengan “namaku” dan email dengan “namaku@mail.com kemudian klik tombol “kirim” |
| 7 | Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu! (soal no 6) |

**Praktikum Bagian 5. Menggunakan *File***

File dapat digunakan sebagai alternatif menyimpan data berkuran kecil yang dimasukkan *user* melalui form. File juga banyak digunakan untuk menyimpan data-data yang akan digunakan untuk keperluan konfigurasi sistem. PHP menyediakan fasilitas untuk mengakses file, seperti membaca, merubah, dan menyimpan data pada file.

Sebelum mengakses sebuah file, maka kita harus membuka file yang akan kita proses terlebih dahulu. PH menyediakan fungsi built-in untuk melakukan pengaturan pada file salah satunya adalah untuk membuka file, yaitu *fopen()*. Sintak penggunaan *fopen()* adalah sebagai berikut:

|  |
| --- |
| <?php  $file\_handler = fopen(namaFile, mode);  ?> |

Parameter mode bisa diisi dengan kode program berikut ini:

|  |  |
| --- | --- |
| **Mode** | **Keterangan** |
| r | Membuka file untuk proses pembacaan (*read*) |
| r+ | Membuka file untuk proses pembacaan (*read*) dan penulisan (*write*) |
| w | Membuka file untuk proses penulisan (*write*) |
| w+ | Membuka file untuk proses penulisan (*write*) dan pembacaan (*read*) |
| a | Membuka file untuk proses penambahan data (*append*) |
| a+ | Membuka file untuk proses pembacaan (*read*) dan penambahan data (*append*) |

Fungsi fread() digunakan untuk membaca data dari file yang telah dibuka. Parameter fungsi fread adalah nama file yang akan dibaca dan jumlah byte maksimum untuk dibaca. Berikut adalah sintak membaca file dengan menggunakan fread():

|  |
| --- |
| <?php  fread($file\_handler, filesize());  ?> |

Fungsi fclose() digunakan untuk menutup file. PHP menyediakan fungsi untuk menutup file, yaitu fclose(). Sintak untuk menutup file dengan fclose() adalah sebagai berikut:

|  |
| --- |
| <?php  fclose($file\_handler);  ?> |

Ikuti langkah-langkah berikut untuk memahami cara menggunakan file:

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Buat file baru dengan nama testFile.txt di dalam direktori praktik\_php, kemudian ketikkan kalimat berikut: |
| 2 | Buat file baru dengan nama file.php di dalam direktori praktik\_php, kemudian ketikkan kode berikut: |
| 3 | Simpan file dan jalankan kode program dengan mengetikkan  localhost/dasarWeb/praktik\_php/file.php |
| 4 | Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu! (soal no 7) |
| 5 | Modifikasi kode program pada langkah ke-2 dengan mengubah parameter fungsi fread() pada baris ke-8 menjadi filesize($myFile), sehingga kode program menjadi seperti berikut: |
| 6 | Simpan file dan jalankan kode program dengan mengetikkan  localhost/dasarWeb/praktik\_php/file.php |
| 7 | Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu! (soal no 8) |
| 8 | Buat file baru dengan nama file\_1.php dan salin kode program pada langkah ke-5. Modifikasi kode program dengan mengubah fungsi fread() pada baris ke-8 dengan fungsi fgets($fh), sehingga kode program menjadi seperti berikut: |
| 9 | Simpan file dan jalankan kode program dengan mengetikkan  localhost/dasarWeb/praktik\_php/file\_1.php |
| 10 | Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu! (soal no 9) |

**Praktikum Bagian 6. *File Upload***

Proses *upload* adalah menyalin file dari direktori komputer lokal (*client)* ke komputer *server*. Banyak pemanfaatan dari metode *upload* ini, seperti untuk mengirim file foto dari *client* ke sistem agar bisa tampil pada halaman web.

Atribut form untuk *upload* sedikit berbeda dengan form untuk data biasa, yang membedakan adalah adanya tambahan atribut enctype= “multipart/form-data”. Tag <input> di set menggunakan atribut type = “file”.

Untuk menerima file digunakan $\_FILES[nama\_field][atribut], berikut adalah isi atributnya:

* *tmp\_name* = berisi *path* *temporary* dimana file hasil proses *upload* disimpan di *server*
* *name* = berisi *path* asli dari file yang di*upload* oleh *user*.

Inti dari proses upload pada PHP adalah berada pada penerapan fungsi built-in yaitu:

* move\_uploaded\_file() atau copy(), yang berfungsi memindahkan dari *folder* *temporary* ke *folder* tujuan *upload*

Ikuti langkah-langkah berikut untuk memahami cara membuat form untuk upload file:

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Buat direktori baru dengan nama “uploads” di dalam direktori praktik\_php |
| 2 | Buat file baru dengan nama formUpload.html di dalam direktori praktik\_php, kemudian ketikkan kode berikut: |
| 3 | Buat file baru dengan nama fileUpload.php di dalam direktori praktik\_php, kemudian ketikkan kode berikut: |
| 4 | Simpan file dan jalankan kode program dengan mengetikkan  localhost/dasarWeb/praktik\_php/formUpload.html |
| 5 | Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu! (soal no 10) |
| 6 | Tunjukkan hasil tangkapan layar isi dari direktori uploads (soal no 11) |

Referensi :

1. Nixon, Robin. (2018). Learning PHP, MySQL, JavaScript, CSS & HTML: A Step-by-step Guide to Creating Dynamic Websites, 5th Edition. O’Reilly Media, Inc.
2. Forbes, Alan. (2012). The Joy of PHP: A Beginners’s Guide to Programming Interactive Web Applications with PHP and MySQL, 5th Edition. Plum Island Publishing