



Nama	: Rizqi Rohmatul Huda
Kelas	: 2G – TI
No.Absen	: 26
NIM	: 2141720264

Topik

Pengenalan konsep dasar php

Tujuan

Mahasiswa diharapkan mampu:

1. memahami konsep php (*form* dan *file*)
2. mengimplementasikan konsep php ke dalam web dinamis

Pendahuluan

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.

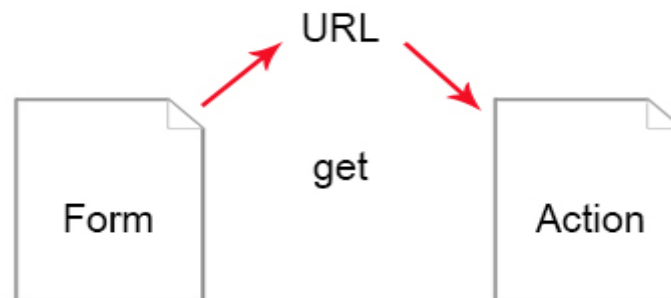
Praktikum Bagian 1. *Form Handling*

Untuk memproses suatu *form* digunakan 2 atribut pada elemen *form*, yaitu atribut *action* dan atribut *method*. Atribut *action* diisi dengan alamat halaman PHP yang digunakan untuk memproses 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:



Gambar 1. Ilustrasi penggunaan *method* POST

Berbeda 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:

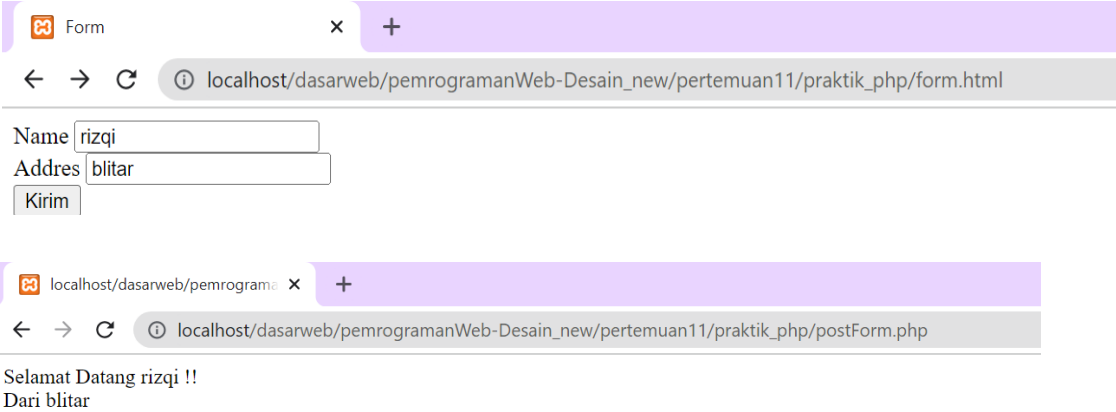


Gambar 2. Ilustrasi penggunaan *method* GET

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 <code>form.html</code> di dalam direktori <code>praktik_php</code> , kemudian ketikkan kode berikut:

	<pre> 1 <html> 2 <head> 3 </head> 4 <body> 5 <form action="postForm.php" method="post"> 6 <label>Name</label> 7 <input type="text" name="myname" placeholder="Masukkan nama"></br> 8 <label>Address</label> 9 <input type="text" name="myaddress" placeholder="Masukkan alamat"></br> 10 <input type="submit"> 11 </form> 12 </body> 13 </html> </pre>
2	<p>Buat file baru dengan nama postForm.php di dalam direktori praktik_php, kemudian ketikkan kode berikut:</p> <pre> 1 <html> 2 <head> 3 </head> 4 <body> 5 Selamat Datang <?php echo \$_POST["myname"]; ?> !!
 6 Dari <?php echo \$_POST["myaddress"]; ?> 7 </body> 8 </html> </pre>
3	<p>Simpan file dan jalankan kode program dengan mengetikkan localhost/dasarWeb/praktik_php/form.html</p>
4	<p>Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu! (soal no 1)</p>  <p>Jawab : method POST akan akan mengirimkan data dari form ke action untuk ditampung tanpa ditampilkan pada URL lalu untuk menampung data atau nilai digunakan variabel \$_POST.</p>
5	<p>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:</p>


	<pre> 1 <html> 2 <head> 3 </head> 4 <body> 5 <form action="getForm.php" method="get"> 6 <label>Name</label> 7 <input type="text" name="myname" placeholder="Masukkan nama"></br> 8 <label>Address</label> 9 <input type="text" name="myaddress" placeholder="Masukkan alamat"></br> 10 <input type="submit"> 11 </form> 12 </body> 13 </html> </pre>
6	<p>Buat file baru dengan nama <code>getForm.php</code> di dalam direktori <code>praktik_php</code>, kemudian ketikkan kode berikut:</p> <pre> 1 <html> 2 <head> 3 </head> 4 <body> 5 Selamat Datang <?php echo \$_GET["myname"]; ?> !!
 6 Dari <?php echo \$_GET["myaddress"]; ?> 7 </body> 8 </html> </pre>
7	<p>Simpan file dan jalankan kode program dengan mengetikkan <code>localhost/dasarWeb/praktik_php/form_1.html</code></p>
8	<p>Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu! (soal no 2)</p>  <p>Jawab : method GET akan menampilkan data atau nilai pada URL, kemudian ditampung oleh action.</p>

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 <code>getForm.php</code> dengan mengetikkan perintah <code>localhost/dasarWeb/praktik_php/getForm.php</code>
2	Amati hasil yang ditampilkan
3	<p>Buka file <code>getForm.php</code> yang telah dibuat sebelumnya dalam direktori <code>praktik_php</code>. Modifikasi kode program dengan menambahkan fungsi <code>isset()</code>, sehingga kode program menjadi seperti berikut:</p> <pre> 1 <html> 2 <head> 3 </head> 4 <body> 5 <?php 6 if(isset(\$_GET["myname"]) AND isset(\$_GET["myaddress"])) 7 { 8 echo "Selamat Datang " . \$_GET["myname"] . "!!
"; 9 echo "Dari " . \$_GET["myaddress"]; 10 } 11 else{ 12 echo "Anda harus mengakses halaman ini dari form_1.html"; 13 } 14 <?> 15 </body> 16 </html> </pre>
3	Simpan file dan jalankan kode program dengan mengetikkan <code>localhost/dasarWeb/praktik_php/getForm.php</code>
4	<p>Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu! (soal no 3)</p>  <p>Anda harus mengakses halaman ini dari form_1.html</p> <p>Jawab : hasil akan meminta untuk mengakses halaman form_1. Setelah mengunjungi form_1 hasil akan menampilkan nama dan alamat penginput, jadi fungsi <code>isset</code> akan mengecek apakah nilai bernilai <code>true</code> atau <code>false</code>, jika <code>true</code> akan menampilkan statement <code>true</code>, yaitu hasil akan menampilkan nama dan alamat penginput</p>

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 memvalidasi 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()`:

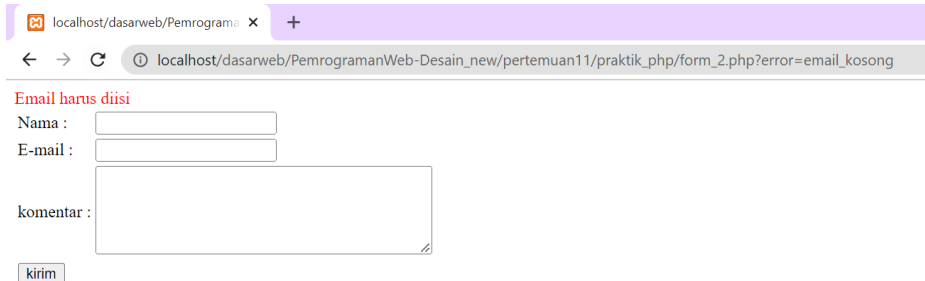
Langkah	Keterangan
1	<p>Buat file <code>form_2.php</code> dalam direktori <code>praktik_php</code> kemudian ketikkan kode program berikut:</p> <pre> 1 <!DOCTYPE html> 2 <html> 3 <head> 4 <style> 5 .error {color: #FF0000;} 6 </style> 7 </head> 8 <body> 9 <?php 10 //ambil nilai variabel error 11 if (isset(\$_GET['error'])) { 12 \$error=\$_GET['error']; 13 } 14 else { 15 \$error=""; 16 } 17 18 //siapkan pesan kesalahan 19 \$pesan=""; 20 if (\$error=="variabel_belum_diset") { 21 \$pesan="Anda harus mengakses halaman ini dari form_2.php"; 22 } 23 else if (\$error=="nama_kosong") { 24 \$pesan="Nama harus diisi"; 25 } 26 else if (\$error=="email_kosong") { 27 \$pesan="Email harus diisi"; 28 } 29 30 //Siapkan isian form jika terjadi kesalahan 31 if (isset(\$_GET['nama']) AND isset(\$_GET['email']) AND isset(\$_GET['komentar'])) { 32 \$nama=\$_GET['nama']; 33 \$email=\$_GET['email']; 34 \$komentar=\$_GET['komentar']; 35 } 36 else 37 { 38 \$nama=""; 39 \$email=""; 40 \$komentar=""; 41 } 42 43 <?> 44 45 <?php echo \$pesan;?> 46 47 <table> 48 <form method="get" action="prosesForm_2.php"> 49 <tr> 50 <td> Nama: </td> 51 <td> <input type="text" name="nama" value="<?php echo \$nama;?>"> </pre>

	<pre> 52 <tr> 53 <td> E-mail: </td> 54 <td> <input type="text" name="email" value="<?php echo \$email;?>"> </td> 55 </tr> 56 <tr> 57 <td> Komentar: </td> 58 <td> <textarea name="komentar" rows="5" cols="40"><?php echo \$komentar;?> 59 </textarea></td> 60 </tr> 61 <tr> 62 <td><input type="submit" name="kirim" value="Kirim"> </td> 63 </tr> 64 </form> 65 </table> 66 </form> 67 </body> 68 </html> </pre>
2	<p>Buat file prosesForm_2.php dalam direktori praktik_php kemudian ketikkan kode program berikut:</p> <pre> 1 <?php 2 if (isset(\$_GET['nama']) AND isset(\$_GET['email'])) { 3 \$nama=\$_GET['nama']; 4 \$email=\$_GET['email']; 5 \$komentar=\$_GET['komentar']; 6 \$isi_form="&nama=\$nama&email=\$email&komentar=\$komentar"; 7 } 8 else{ 9 header("Location:form_2.php?error=variabel_belum_diset"); 10 } 11 12 if(empty(\$nama)){ 13 header("Location:form_2.php?error=nama_kosong".\$isi_form); 14 } 15 else if(empty(\$email)){ 16 header("Location:form_2.php?error=email_kosong".\$isi_form); 17 } 18 else{ 19 echo "Nama: \$nama
 Email: \$email
 Komentar: \$komentar"; 20 } 21 ?> </pre>
3	<p>Simpan file dan jalankan kode program dengan mengetikkan localhost/dasarWeb/praktik_php/form_2.php</p>
4	<p>Lakukan eksperimen berikut:</p> <ol style="list-style-type: none"> tanpa mengisi form langsung klik tombol “kirim” isi <i>field</i> nama dan kosongkan email, kemudian klik tombol “kirim” isi <i>field</i> email dan kosongkan nama, kemudian klik tombol “kirim” isi <i>field</i> nama dan email, kemudian klik tombol “kirim”
5	<p>Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu! (soal no 4)</p> <ol style="list-style-type: none"> tanpa mengisi form langsung klik tombol “kirim” 

Terdapat peringatan bahwa pada bagian nama harus diisi terlebih dahulu karena terdapat pengecekan kondisi sebagai berikut:

```
} else if ($error == "nama_kosong") {  
    $pesan = "Nama harus diisi";  
}
```

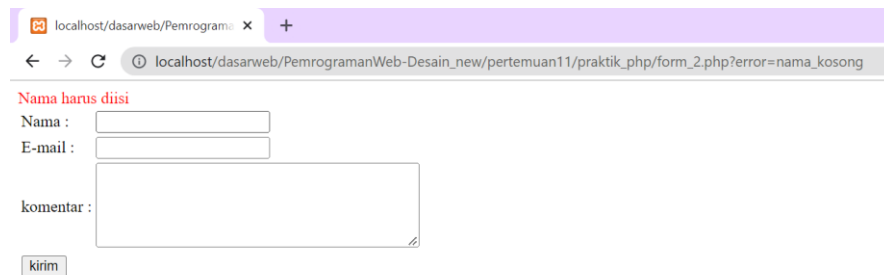
- b. isi *field* nama dan kosongkan email, kemudian klik tombol “kirim”



Tidak dapat mengirim form, Email harus diisi terlebih dahulu sesuai dengan kondisi sintaks di bawah ini.

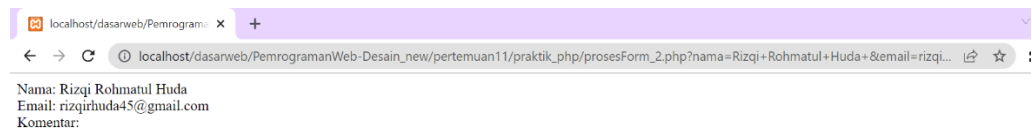
```
} else if ($error == "email_kosong") {  
    $pesan = "Email harus diisi";  
}
```

- c. isi *field* email dan kosongkan nama, kemudian klik tombol “kirim”



Hasil akan sama pada point (a) nama harus tetap diisi.

- d. isi *field* nama dan email, kemudian klik tombol “kirim”



Form akan terkirim dan akan menampilkan nama, email, dan komentar dari penginput, meskipun kolom komentar diisi atau tidak diisi maka form akan tetap terkirim.

6

Jelaskan apakah kegunaan fungsi `header()` (soal no 5)

Jawab : fungsi `header()` mengirimkan header HTTP mentah ke klien.

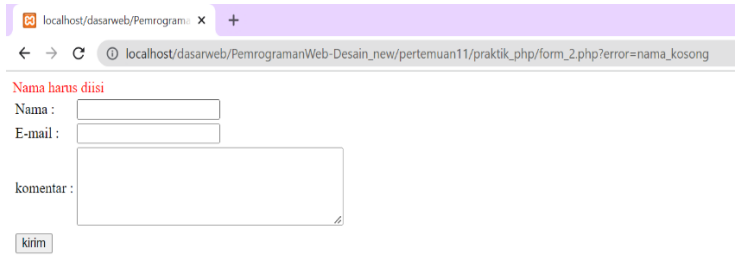


Praktikum Bagian 4. Form URL/e-mail

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 mencocokkan 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 `filter_var()`.

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

Langkah	Keterangan
1	Buat file baru dengan nama <code>form_3.php</code> kemudian salin kode program <code>form_2.php</code> dari praktikum bagian sebelumnya
2	Modifikasi kode program sehingga kode program menjadi seperti berikut: <pre>1 <?DOCTYPE html> 2 <html> 3 <head> 4 <style> 5 .error {color: #FF0000;} 6 </style> 7 </head> 8 <body> 9 <?php 10 //ambil nilai variabel error 11 if (isset(\$_GET['error'])){ 12 \$error=\$_GET['error']; 13 } 14 else{ 15 \$error=""; 16 } 17 18 //siapkan pesan kesalahan 19 \$pesan=""; 20 if (\$error=="variabel_belum_diset"){ 21 \$pesan="Anda harus mengakses halaman ini dari form_3.php"; 22 } 23 else if (\$error=="nama_kosong"){ 24 \$pesan="Nama harus diisi"; 25 } 26 else if (\$error=="nama_invalid"){ 27 \$pesan="Nama diisi huruf dan spasi"; 28 } 29 else if (\$error=="email_kosong"){ 30 \$pesan="Email harus diisi"; 31 } 32 if (\$error=="email_invalid"){ 33 \$pesan="Email tidak sesuai"; 34 }</pre>

	<pre> 35 36 //Siapkan isian form jika terjadi kesalahan 37 if (isset(\$_GET['nama']) AND isset(\$_GET['email']) AND isset(\$_GET['komentar'])) { 38 \$nama=\$_GET['nama']; 39 \$email=\$_GET['email']; 40 \$komentar=\$_GET['komentar']; 41 } 42 else 43 { 44 \$nama=""; 45 \$email=""; 46 \$komentar=""; 47 } 48 ?> 49 50 <?php echo \$pesan;?> 51 52 <table> 53 <form method="get" action="prosesForm 3.php"> 54 <tr> 55 <td> Nama: </td> 56 <td> <input type="text" name="nama" value="<?php echo \$nama;?>"> 57 </tr> 58 <tr> 59 <td> E-mail: </td> 60 <td> <input type="text" name="email" value="<?php echo \$email;?>"> </td> 61 </tr> 62 <tr> 63 <td> Komentar: </td> 64 <td> <textarea name="komentar" rows="5" cols="40"><?php echo \$komentar;?> 65 </textarea></td> 66 </tr> 67 <tr> 68 <td><input type="submit" name=" kirim" value="Kirim"> </td> 69 </tr> 70 </form> 71 </table> 72 </form> 73 </body> 74 </html> </pre>
3	<p>Buat file baru dengan nama prosesForm_3.php kemudian salin kode program prosesForm_2.php dari praktikum bagian sebelumnya</p>
4	<p>Modifikasi kode program php menambahkan fungsi preg_match() dan filter_var() sehingga kode program menjadi seperti berikut:</p> <pre> 1 <?php 2 if (isset(\$_GET['nama']) AND isset(\$_GET['email'])) { 3 \$nama=\$_GET['nama']; 4 \$email=\$_GET['email']; 5 \$komentar=\$_GET['komentar']; 6 \$isi_form="&nama=\$nama&email=\$email&komentar=\$komentar"; 7 } 8 else { 9 header("Location:form_3.php?error=variabel_belum_diset"); 10 } 11 12 if(empty(\$nama)) { 13 header("Location:form_3.php?error=nama_kosong".\$isi_form); 14 } 15 else if(!preg_match("/^[a-zA-Z]*\$/",\$nama)) { 16 header("Location:form_3.php?error=nama_invalid".\$isi_form); 17 } 18 else if(empty(\$email)) { 19 header("Location:form_3.php?error=email_kosong".\$isi_form); 20 } 21 else 22 if(!filter_var(\$email, FILTER_VALIDATE_EMAIL)){ 23 header("Location:form_3.php?error=email_invalid".\$isi_form); 24 } 25 else { 26 echo "Nama: \$nama
 Email: \$email
 Komentar: \$komentar"; 27 } 28 ?> </pre>

5	<p>Simpan file dan jalankan kode program dengan mengetikkan <code>localhost/dasarWeb/praktik_php/form_3.php</code></p>
6	<p>Lakukan eksperimen berikut:</p> <ol style="list-style-type: none"> tanpa mengisi form langsung klik tombol “ kirim ” isi <i>field</i> nama dengan angka kemudian klik tombol “ kirim ” isi <i>field</i> nama dengan string kemudian klik tombol “ kirim ” isi <i>field</i> nama dan email dengan “namaku” kemudian klik tombol “ kirim ” isi <i>field</i> nama dengan “namaku” dan email dengan “namaku@mail.com” kemudian klik tombol “ kirim ”
7	<p>Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu! (soal no 6)</p> <p>a. tanpa mengisi form langsung klik tombol “ kirim ”</p>  <p>Tidak dapat mengirim form, nama harus diisi terlebih dahulu</p> <p>b. isi <i>field</i> nama dengan angka kemudian klik tombol “ kirim ”</p>  <p>Nama harus diisi dengan huruf, selain karakter huruf form tidak akan terkirim.</p> <p>c. isi <i>field</i> nama dengan string kemudian klik tombol “ kirim ”</p>  <p>Tidak mengirim email harus diisi terlebih dahulu</p> <p>d. isi <i>field</i> nama dan email dengan “namaku” kemudian klik tombol “ kirim ”</p>

Email tidak sesuai

Nama :

E-mail :

komentar :

Penulisan email berbeda dengan nama, email harus memenuhi aturan penulisan email, yaitu “alamatemail@mail.com”.

e. isi *field* nama dengan “namaku” dan email dengan “namaku@mail.com kemudian klik tombol “kirim”

localhost/dasarweb/Pemrograma x +

localhost/dasarweb/PemrogramanWeb-Desain_new/pertemuan11/praktik_php/form_2.php

Nama :

E-mail :

komentar :

localhost/dasarweb/Pemrograma x +

localhost/dasarweb/PemrogramanWeb-Desain_new/pertemuan11/praktik_php/prosesForm_2.php?nama=nama+ku&email=namaku%40gmail.c...

Nama: nama ku
Email: namaku@gmail.com
Komentar:

Karena semua kondisi bernilai true maka form akan terkirim dan menampilkan nama serta alamat email dari penginput.

Praktikum Bagian 5. Menggunakan File

File dapat digunakan sebagai alternatif menyimpan data berukuran 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. PHP 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 (<i>read</i>)
r+	Membuka file untuk proses pembacaan (<i>read</i>) dan penulisan (<i>write</i>)
w	Membuka file untuk proses penulisan (<i>write</i>)
w+	Membuka file untuk proses penulisan (<i>write</i>) dan pembacaan (<i>read</i>)
a	Membuka file untuk proses penambahan data (<i>append</i>)
a+	Membuka file untuk proses pembacaan (<i>read</i>) dan penambahan data (<i>append</i>)

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());
?>
```

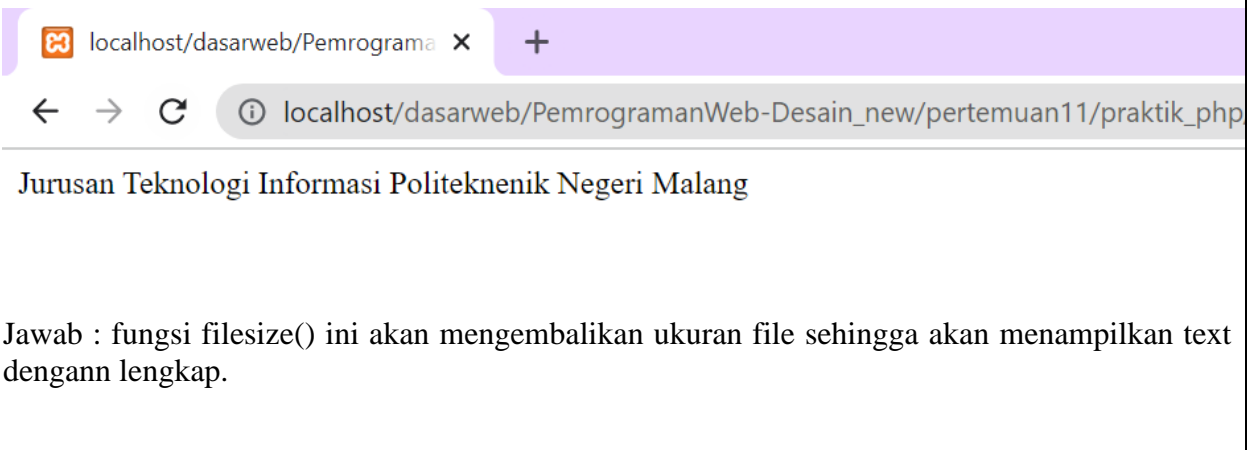
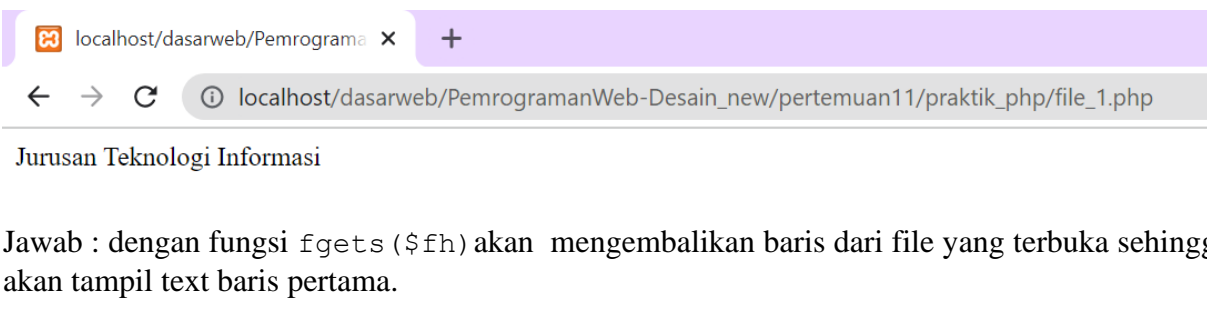
Fungsi `fclose()` digunakan untuk menutup file. PHP menyediakan fungsi untuk menutup file, yaitu `fclose()`. Sintak untuk menutup file dengan `fclose()` adalah sebagai berikut:

```
<?php
    fclose($file_handler);
?>
```

Ikuti langkah-langkah berikut untuk memahami cara menggunakan file:

Langkah	Keterangan
1	Buat file baru dengan nama <code>testFile.txt</code> di dalam direktori <code>praktik_php</code> , kemudian ketikkan kalimat berikut: <pre>1 Jurusan Teknologi Informasi 2 Politeknik Negeri Malang</pre>
2	Buat file baru dengan nama <code>file.php</code> di dalam direktori <code>praktik_php</code> , kemudian ketikkan kode berikut:

	<pre> 1 <html> 2 <head> 3 </head> 4 <body> 5 <?php 6 \$myFile = "testFile.txt"; 7 \$fh = fopen(\$myFile, 'r'); 8 \$theData = fread(\$fh, 5); 9 fclose(\$fh); 10 echo \$theData; 11 ?> 12 </body> 13 </html> </pre>
3	<p>Simpan file dan jalankan kode program dengan mengetikkan localhost/dasarWeb/praktik_php/file.php</p>
4	<p>Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu! (soal no 7)</p>  <p>Jawab : 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. Pada sintaks tersebut parameter diisi dengan jumlah 5 sehingga hanya akan terbaca “jurus”.</p> <pre>\$theData = fread(\$fh, 5);</pre>
5	<p>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:</p> <pre> 1 <html> 2 <head> 3 </head> 4 <body> 5 <?php 6 \$myFile = "testFile.txt"; 7 \$fh = fopen(\$myFile, 'r'); 8 \$theData = fread(\$fh, filesize(\$myFile)); 9 fclose(\$fh); 10 echo \$theData; 11 ?> 12 </body> 13 </html> </pre>
6	<p>Simpan file dan jalankan kode program dengan mengetikkan localhost/dasarWeb/praktik_php/file.php</p>
7	<p>Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu! (soal no 8)</p>

	 <p>Jurusan Teknologi Informasi Politeknik Negeri Malang</p> <p>Jawab : fungsi filesize() ini akan mengembalikan ukuran file sehingga akan menampilkan text dengann lengkap.</p>
8	<p>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:</p> <pre> 1 <html> 2 <head> 3 </head> 4 <body> 5 <?php 6 \$myFile = "testFile.txt"; 7 \$fh = fopen(\$myFile, 'r'); 8 \$theData = fgets(\$fh); 9 fclose(\$fh); 10 echo \$theData; 11 <?> 12 </body> 13 </html> </pre>
9	<p>Simpan file dan jalankan kode program dengan mengetikkan localhost/dasarWeb/praktik_php/file_1.php</p>
10	<p>Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu! (soal no 9)</p>  <p>Jurusan Teknologi Informasi</p> <p>Jawab : dengan fungsi fgets(\$fh) akan mengembalikan baris dari file yang terbuka sehingga akan tampil text baris pertama.</p>

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:

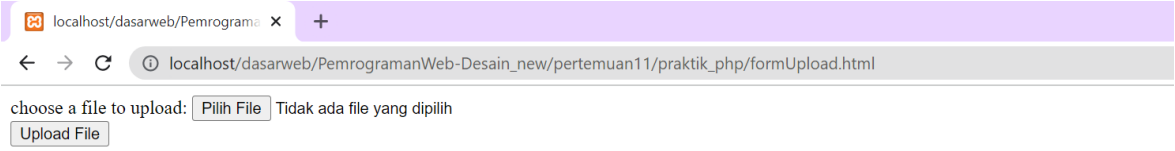
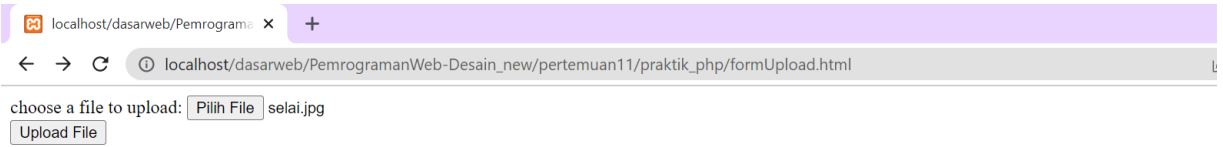
- `tmp_name` = berisi path temporary dimana file hasil proses upload disimpan di server
- `name` = berisi path asli dari file yang diupload oleh user.

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

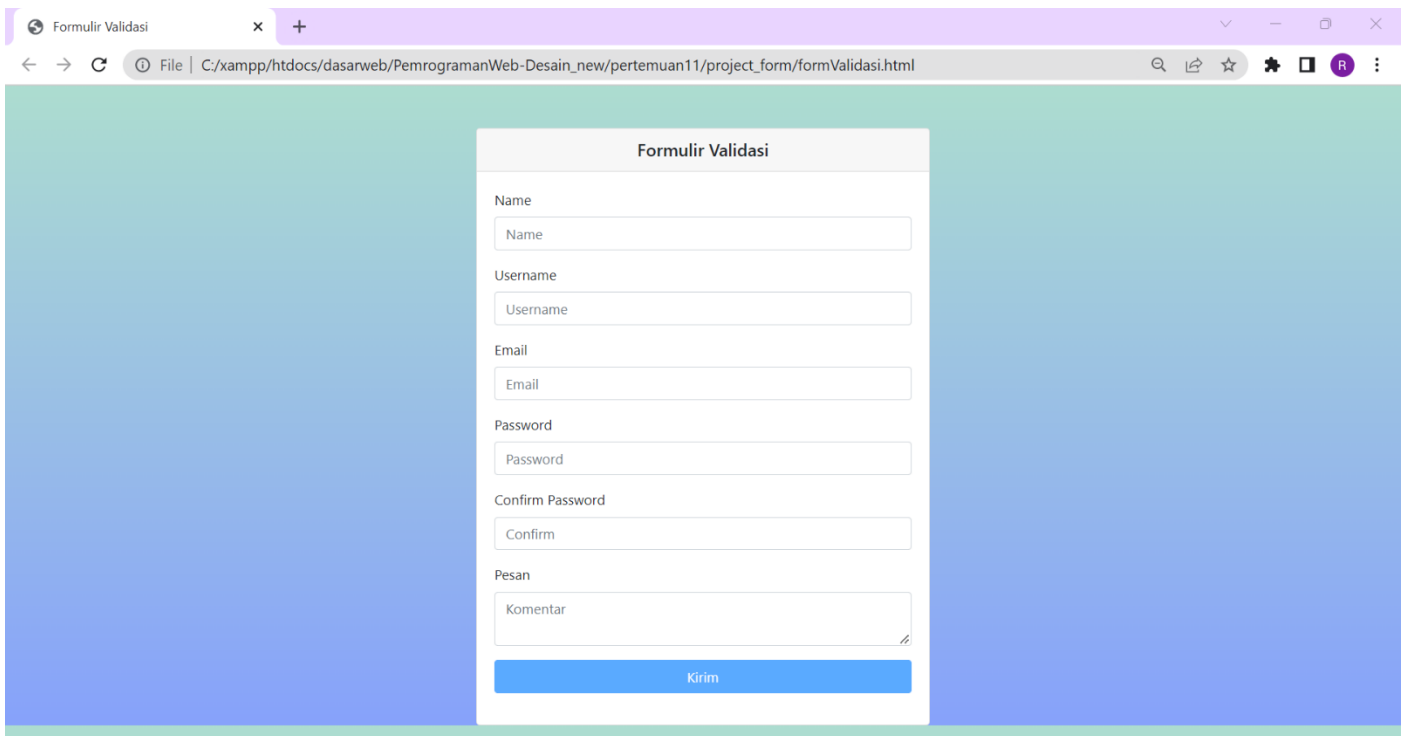
Ikuti langkah-langkah berikut untuk memahami cara membuat form untuk upload file:

Langkah	Keterangan
1	Buat direktori baru dengan nama "uploads" di dalam direktori praktik_php
2	<p>Buat file baru dengan nama <code>formUpload.html</code> di dalam direktori praktik_php, kemudian ketikkan kode berikut:</p> <pre> 1 <html> 2 <head> 3 </head> 4 <body> 5 <form enctype="multipart/form-data" action="fileUpload.php" method="POST"> 6 choose a file to upload: 7 <input name="uploadedfile" type="file">
 8 <input type="submit" value="Upload File"> 9 </form> 10 </body> 11 </html> </pre>
3	<p>Buat file baru dengan nama <code>fileUpload.php</code> di dalam direktori praktik_php, kemudian ketikkan kode berikut:</p> <pre> 1 <?php 2 \$target_path ="uploads/"; 3 4 \$target_path =\$target_path . basename(5 \$_FILES['uploadedfile']['name']); 6 7 if(move_uploaded_file(\$_FILES['uploadedfile']['tmp_name'],\$target_path)){ 8 echo "The file " . basename(\$_FILES['uploadedfile']['name'])."has been uploaded"; 9 } else{ 10 echo "There was an error uploading the file, please try again!"; 11 } 12 > </pre>
4	Simpan file dan jalankan kode program dengan mengetikkan <code>localhost/dasarWeb/praktik_php/formUpload.html</code>
5	Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu! (soal no 10)

	 <p>Jawab : 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.</p>
6	<p>Tunjukkan hasil tangkapan layar isi dari direktori uploads (soal no 11)</p> 

Tugas Pertemuan 11 – Membuat Form

Gambar tampilan form pada browser



The screenshot shows a web browser window with the address bar displaying the file path: C:\xampp\htdocs\dasarweb\PemrogramanWeb-Desain_new\pertemuan11\project_form\formValidasi.html. The page content features a white form titled "Formulir Validasi" centered on a blue gradient background. The form includes the following fields: "Name" (text input), "Username" (text input), "Email" (text input), "Password" (text input), "Confirm Password" (text input), and "Pesan" (text area with a placeholder "Komentar"). A blue button labeled "Kirim" is positioned at the bottom of the form.

Code html

```
pertemuan11 > project_form > formValidasi.html > html
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Formulir Validasi</title>
5      <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet"></link>
6      <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
7      <script src="https://cdnjs.cloudflare.com/ajax/libs/1000hz-bootstrap-validator/0.11.9/validator.min.js"></script>
8      <link rel="stylesheet" href="styleIndex.css">
9
10 </head>
11 <body>
12     <div class="container mt-5">
13         <div class="card">
14             <h5 class="card-header text-center">Formulir Validasi</h5>
15             <div class="card-body">
16                 <form role="form" data-toggle="validator">
17                     <div class="form-group">
18                         <label>Name</label>
19                         <input type="text" class="form-control" data-error="You must have a name." id="inputName" placeholder="Name" required>
20
21                     <div class="help-block with-errors"></div>
22                 </div>
23                 <div class="form-group">
24                     <label>Username</label>
25                     <input type="text" class="form-control" name="username" maxLength="10" minLength="3"
26                         pattern="^[a-zA-Z0-9_-]*$" id="inputUsername" placeholder="Username" required>
27
28                     <div class="help-block with-errors"></div>
29
```

```

30 </div>
31 <div class="form-group">
32   <label>Email</label>
33   <input type="email" class="form-control" id="inputEmail" placeholder="Email" required>
34
35   <div class="help-block with-errors"></div>
36 </div>
37
38 <div class="form-group">
39   <label>Password</label>
40   <div class="form-group">
41     <input type="password" data-minlength="4" class="form-control" id="inputPassword"
42       data-error="Have atleast 4 characters" placeholder="Password" required />
43
44     <div class="help-block with-errors"></div>
45   </div>
46 </div>
47 <div class="form-group">
48   <label>Confirm Password</label>
49   <div class="form-group">
50     <input type="password" class="form-control" id="inputConfirmPassword"
51       data-match="#inputPassword" data-match-error="Password don't match"
52       placeholder="Confirm" required />
53
54     <div class="help-block with-errors"></div>
55   </div>
56 </div>
57
58 <div class="form-group">
59   <label>Pesan</label>
60   <textarea class="form-control" data-error="Please enter message." id="inputMessage"
61     placeholder="Komentar" required=""></textarea>
62
63   <div class="help-block with-errors"></div>
64 </div>
65
66 <div class="form-group">
67   <button type="submit" class="btn btn-primary btn-block">Kirim</button>
68 </div>
69 </form>
70 </div>
71 </div>
72 </div>
73 </body>
74 </html>

```

Code css

```

pertemuan11 > project_form > styleIndex.css > ...
1 body {
2   background: #6F38C5;
3   background: -webkit-linear-gradient(to top, #87A2FB, #ADD8D0);
4   background: linear-gradient(to top, #87A2FB, #ADD8D0);
5 }
6
7 .container {
8   max-width: 550px;
9 }
10
11 .has-error label,
12 .has-error input,
13 .has-error textarea {
14   color: red;
15   border-color: red;
16 }
17
18 .list-unstyled li {
19   font-size: 13px;
20   padding: 4px 0 0;
21   color: red;
22 }
23

```

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