

**LAPORAN HASIL PRAKTIKUM
PEMROGRAMAN WEB DAN MOBILE I**



NAMA : M. ADE SHOFY
NIM : 193010503006
KELAS : A
MODUL : II (FORM HANDLING)

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PALANGKA RAYA
2020**

BAB I

TUJUAN DAN LANDASAN TEORI

1.1. TUJUAN

- 1.1.1.** Mahasiswa mampu membuat handling yang mampu mengola data dari form HTML.
- 1.1.2.** Mahasiswa mampu membuat batasan-batasan untuk menangani inputan dari form HTML.

1.2. LANDASAN TEORI

Form biasanya digunakan untuk mengumpulkan data dari pengunjung web kita. Mulai dari form untuk login, form kontak, form untuk pendaftaran user, bahkan untuk mengirimkan data antar halaman web. Penggunaan form hanya menggunakan HTML saja tidak akan terlalu berguna. Form biasanya hanya berupa *interface* yang disediakan untuk mengumpulkan data dari user, dan akan diproses dengan bahasa pemograman web seperti JavaScript atau PHP, dan disimpan di dalam tabel MySQL. Tag dasar yang akan digunakan untuk membuat form di HTML adalah tag form, input, textarea, select dan option.

1.2.1. Tag <Form>

Sebuah form dalam HTML harus berada di dalam tag form, yang diawali dengan <form> dan diakhiri dengan </form>. Tag form akan membutuhkan beberapa atribut untuk dapat berfungsi dengan seharusnya.

Atribut pertama adalah action, yang berfungsi untuk menjelaskan kemana data form akan dikirimkan. Biasanya nilai dari atribut action ini adalah alamat dari sebuah halaman PHP yang digunakan untuk memproses isi data form.

Atribut kedua adalah method, yang berfungsi untuk menjelaskan bagaimana data isian form akan dikirim oleh web browser. Nilai dari atribut method ini bisa berupa get atau post.

Perbedaan *method get* dan *method post* adalah, jika kita mengisi atribut *method* dengan *get* (dimana ini adalah nilai *default* seandainya atribut *method* tidak ditulis) maka isian form akan terlihat pada *url browser*. Method *get* ini biasanya digunakan untuk *query* pencarian. Method *post* biasanya digunakan untuk data yang lebih sensitif seperti yang berisi *password*, atau *registrasi user*. Data hasil form tidak akan terlihat pada browser.

Struktur dasar form adalah sebagai berikut:

```
<form action="prosesdata.php" method="post">
    ...isi form...
</form>
```

1.2.2. Tag <Input>

Tag *input* merupakan tag paling banyak digunakan di dalam form dan memiliki banyak bentuk, mulai dari isian text biasa, text *password*, *checkbox*, *radio*, sampai dengan tombol *submit*, semuanya dalam bentuk tag *<input>*.

Bentuk-bentuk dari keluarga tag *input* ini dibedakan berdasarkan atribut *type*:

- <input type="text" />* atau bisa juga *<input />* adalah *textbox* inputan biasa yang menerima input berupa text, contohnya digunakan untuk inputan *nama*, *username*, dan inputan yang berupa text pendek. Input type text ini juga bisa memiliki atribut *value* yang bisa diisi nilai tampilan awal dari text.
- <input type="password" />* dalam tampilannya sama dengan type text, namun teks yang diinput tidak akan terlihat, akan berupa bintang atau bulatan. Biasanya hanya digunakan untuk inputan yang sensitif seperti *password*.
- <input type="checkbox" />* adalah inputan berupa *checkbox* yang dapat diceklist atau di centang oleh user. User dapat memilih atau

tidak memilih checkbox ini. Type checkbox memiliki atribut `checked` yang jika ditulis atau diisi dengan nilai `checked`, akan membuat checkbox langsung terpilih pada saat pertama kali halaman ditampilkan. Contoh inputan checkbox berupa *hobi*, yang oleh user dapat dipilih beberapa hobi.

- d. `<input type="radio" />` mirip dengan checkbox, namun user hanya bisa memilih satu diantara pilihan group radio. Type radio ini berada dalam suatu grup dan user hanya bisa memilih salah satunya. Contoh inputan type radio adalah jenis kelamin.
- e. `<input type="submit" />` akan menampilkan tombol untuk memproses form. Biasanya diletakkan pada baris terakhir dari form. Atribut `value` jika diisi akan membuat text tombol submit berubah sesuai inputan nilai `value`.

1.2.3. Tag `<Textarea>`

Tag `textarea` pada dasarnya sama dengan `input type="text"`, namun lebih besar dan dapat berisi banyak baris. Panjang dan banyak baris untuk text area di atur melalui atribut `rows` dan `cols`, atau melalui CSS.

Contoh penggunaan *textarea* adalah sebagai berikut:

```
<textarea rows="5" cols="20">  
    Text yang diisi dapat mencapai banyak baris  
</textarea>
```

Elemen yang berada diantara tag `textarea` akan ditampilkan sebagai text awal dari form.

1.2.4. Tag `<Select>`

Tag `select` digunakan untuk inputan yang telah tersedia nilainya, dan user hanya dapat memilih dari nilai yang ada. Tag `select` digunakan bersama-sama dengan tag `option` untuk membuat box pilihan.

Contoh penggunaan *tag select* adalah sebagai berikut:

```
<select>  
    <option>Pilihan 1</option>  
    <option>Pilihan 2</option>
```

```
<option value="pilihan ketiga">Pilihan 3</option>
</select>
```

Ketika form dikirim untuk diproses, nilai dari tag `<option>` akan dikirimkan. Nilai ini adalah berupa text diantara tag option, kecuali jika kita memberikan atribut *value*. Jika atribut *value* berisi nilai, maka nilai *value*-lah yang akan dikirim. Ada atau tidaknya atribut *value* ini tidak akan tampak dalam tampilan form.

Tag select memiliki atribut *selected* yang dapat ditambahkan agar tag select berisi nilai awal. Contoh penggunaannya adalah sebagai berikut:

```
<select>
  <option>Pilihan 1</option>
  <option>Pilihan 2</option>
  <option value="pilihan ketiga" selected>Pilihan
3</option>
</select>
```

1.2.5. Atribut Name

Setiap tag inputan di dalam form harus ditambahkan *atribut* name agar dapat diproses oleh web server nantinya. Di dalam halaman proses (yang biasanya berupa bahasa PHP atau ASP), nilai dari atribut *name* inilah yang akan menjadi *variabel form*. Contoh pemakaiannya adalah sebagai berikut:

```
<input type="text" name="username">
<input type="text" name="email">
```

Kedua input diatas akan tampak sama persis, namun pada saat pemrosesan data, masing-masing akan dibedakan menurut *atribut* name.

1.2.6. Struktur Dasar Form HTML

```
<form action="proses.php" method="get">
  Username : <input type="text" name="username" />
  <br />
  Password : <input type="password" name="password"
/>
  <br />
  <input type="submit" value="Login" >
</form>
```

Keterangan :

- a. Perhatikan pada line 1 terdapat attribute action dengan value proses.php yang artinya ketika nanti tombol submit di form diklik, maka akan memanggil file dengan nama proses.php untuk memproses inputan form, jadi kita perlu membuat file dengan nama proses.php untuk memproses inputan dari form.
- b. Berikutnya perhatikan value dari attribute action, valuenya adalah proses.php, sehingga file proses.php harus berada satu folder dengan file formnya, anda juga dapat menambahkan nama folder posisi dari file proses.php jika memang anda ingin file proses.php diletakkan di folder yang berbeda dengan file formnya, semisal action="sistem/proses.php" berarti file proses.php berada didalam folder sistem.
- c. Pada line 1 juga terdapat attribute method dengan value get, attribute method ini bisa anda isi dengan menggunakan 2 nilai yaitu post atau get, method ini nanti yang mengatur bagaimana data dikirim dan dapat diterima oleh server.
- d. Jika anda menggunakan method get maka inputan form akan dikirim melalui alamat url website, tetapi jika anda menggunakan method post maka inputan form yang dikirimkan tidak terlihat di bagian alamat url website.
- e. Berikutnya pada line 2 kita menuliskan tag <input type="text"> dan line 4 kita menuliskan tag <input type="password"> yang artinya kita membuat inputan untuk text di line 2 dan inputan untuk password di line 4, yang terpenting dibagian inputan ini mereka memiliki attribute name, value dari attribute name ini yang digunakan sebagai penanda masing – masing objek form agar dapat diproses oleh php.
- f. Pada line 6 kita menuliskan perintah tag input dengan type="submit" ini yang akan menjadi tombol, dan ketika tombol ini diklik

maka akan mengakses file yang menjadi value dari attribute action yaitu file proses.php untuk memproses data inputannya.

1.2.7. Perbedaan Metod Get Dan Post

Berikutnya kita akan membahas mengenai perbedaan dari penggunaan method get dan method post, perbedaan yang paling terlihat dari penggunaan 2 method ini adalah bagaimana cara data dikirimkan agar dapat di proses oleh server.

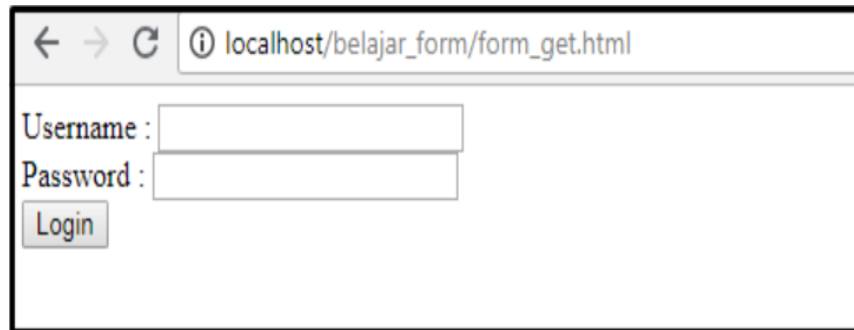
Pada contoh ini karena nantinya kita menggunakan php untuk menangani prosesnya inputan form, sehingga file formnya kita harus meletakkan di dalam web direktori dalam contoh ini adalah di c:xampp/htdocs, dalam contoh ini kita membuat folder dengan nama belajar_form, sehingga nanti kita memanggilnya dengan alamat http://localhost/belajar_form/form.html, sebagai contoh perhatikan skrip dibawah ini :

a. Menggunakan Method Get

Kita membuat file dengan nama form_get.html dan disimpan didalam folder belajar_form didalam folder htdocs, dengan isi skrip sebagai berikut :

```
<form action="proses.php" method="get">
  Username : <input type="text" name="username"
/>
  <br />
  Password : <input type="password"
name="password" />
  <br />
  <input type="submit" value="Login" >
</form>
```

kita dapat mengaksesnya dengan alamat http://localhost/belajar_form/form_get.html tampilannya seperti berikut ini :



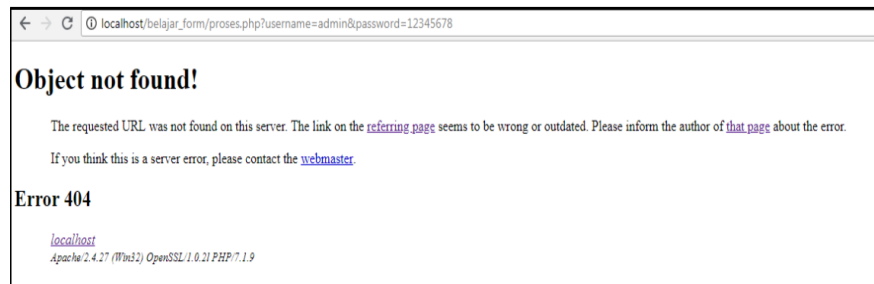
← → ↻ ⓘ localhost/belajar_form/form_get.html

Username :

Password :

Login

Jika tombol Login di klik maka akan memanggil file dengan nama proses.php, serta nilai inputan formnya akan akan ditampilkan di bagian url websitenya.



Keterangan :

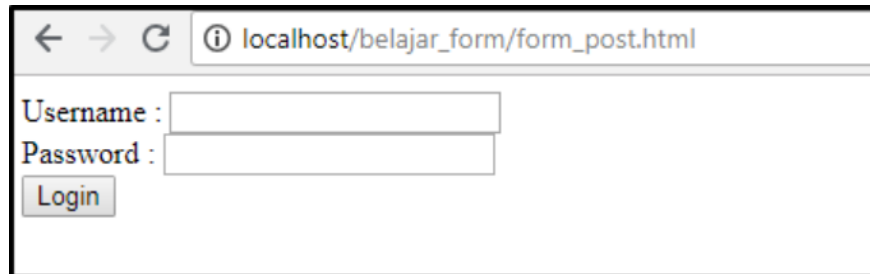
- Perhatikan untuk alamat urlnya tertulis ***proses.php?username=admin&password=12345678***, arti dari tulisan username dan password adalah value dari attribute name dimasing – masing inputan. Lalu tulisan admin dan 12345678 adalah value yang diinputkan dibagian form.
- Jadi kalau kita artikan karakter diatas mengirimkan 2 data, yang pertama data username dengan value admin, dan data password dengan value 12345678
- Pada gambar diatas terdapat tulisan not found dikarenakan untuk file proses.php memang belum dibuat jadi keterangannya file proses.php tidak ditemukan.

b. Menggunakan Post

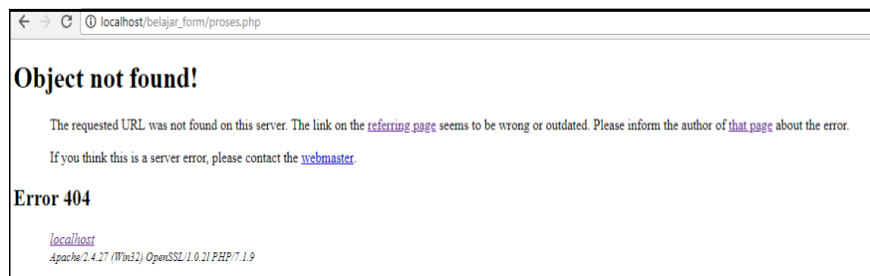
Berikutnya kita membuat file dengan nama form_post.html didalam folder belajar_form didalam folderhtdocs.

```
<form action="proses.php" method="post">
    Username : <input type="text" name="username"
/>
    <br />
    Password : <input type="password"
name="password" />
    <br />
    <input type="submit" value="Login" >
</form>
```

anda dapat mengaksesnya dengan menggunakan alamat http://localhost/belajar_form/form_post.html jika dibuka tampilannya seperti gambar dibawah ini :



Tetapi ketika di klik tombol login, maka tampilannya adalah seperti berikut ini :



Keterangan :

Form akan mengakses file dengan nama proses.php, tetapi di url website tidak terlihat nilai yang dikirimkan, hal ini karena kita menggunakan post sehingga data yang dikirimkan tidak terlihat di bagian url dari website

BAB II

PEMBAHASAN

Code Program HTML

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Halaman Login</title>
    <link rel="stylesheet" href="style.css">
  </head>

  <body>
    <div class="container">
      <h1>Login</h1>
      <form>
        <label>Username</label><br>
        <input type="text"><br>
        <label>Password</label><br>
        <input type="password"><br>
        <button>Log in</button>
      </form>
    </div>
  </body>
</html>
```

Code Program CSS

```
*{
  margin: 0;
  padding: 0;
  outline: 0;
  font-family: 'Open Sans', sans-serif;
}
body{
  height: 100vh;
  background-color: turquoise;
  background-size: cover;
```

```
background-position: center;
background-repeat: no-repeat;
}

.container{
  position: absolute;
  left: 50%;
  top: 50%;
  transform: translate(-50%,-50%);
  padding: 20px 25px;
  width: 300px;

  background-color: rgba(0,0,0,.7);
  box-shadow: 0 0 10px rgba(255,255,255,.3);
}

.container h1{
  text-align: left;
  color: #fafafa;
  margin-bottom: 30px;
  text-transform: uppercase;
  border-bottom: 4px solid #2979ff;
}

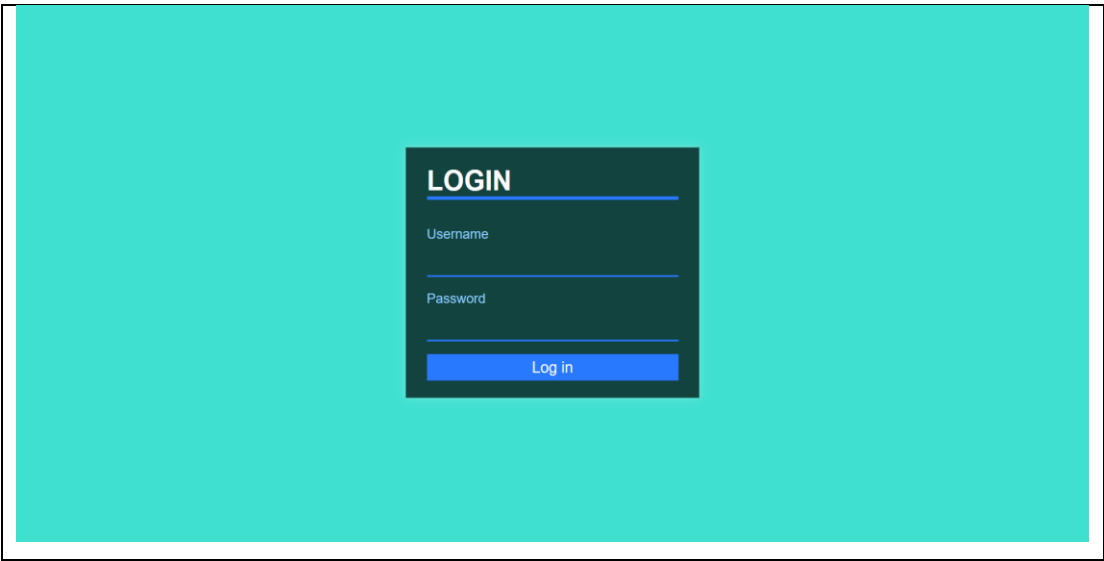
.container label{
  text-align: left;
  color: #90caf9;
}

.container form input{
  width: calc(100% - 20px);
  padding: 8px 10px;
  margin-bottom: 15px;
  border: none;
  background-color: transparent;
  border-bottom: 2px solid #2979ff;
  color: #fff;
  font-size: 20px;
}

.container form button{
  width: 100%;
```

```
padding: 5px 0;  
border: none;  
background-color:#2979ff;  
font-size: 18px;  
color: #fafafa;  
}
```

Output Program



Pembahasan Program

2.1. Code Program HTML

Mendeklarasikan HTML

```
<!DOCTYPE HTML>  
<html>  
</html>
```

Mendeklarasikan dokumen HTML. Deklarasi ini akan memberitahu web browser bahwa Anda menuliskan instruksi dalam versi HTML. Penulisannya sendiri ditandai dengan adanya **DOCTYPE** atau **Document Type Declaration (DTD)** pada baris pertama text editor. Setelah mendeklarasikan dokumen HTML, selanjutnya memulai proses pembuatan form dengan **tag <html>**.

Membuat kepala halaman

```
<!DOCTYPE HTML>
<html>
  <head>
  </head>
</html>
```

Memasukan **tag<head>** didalam **tag< html >**. Tag head mengandung berbagai metadata atau informasi teknis terkait dokumen HTML yang dibuat. Contohnya seperti judul halaman, deskripsi halaman, sumber file eksternal, dan lain-lain.

Memberi judul halaman

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Halaman Login</title>
  </head>
</html>
```

Seperti pembahasan pada poin sebelumnya, kita dapat mengisi metadata di dalam tag <head>. Salah satu komponennya yaitu judul. Untuk membuat judul halaman, kita menggunakan **tag <title>**.

Mensisipkan file CSS

Menyisipkan file CSS. Masih di dalam tag <head>, sekarang kita akan menyisipkan file CSS ke dalam dokumen HTML. File CSS berada terpisah dari dokumen HTML. Karena itu, dibutuhkan jembatan untuk menghubungkan kedua file ini. Jika tidak terhubung, browser akan gagal menyatukan desain dari file CSS. Berikut script untuk menyisipkan file CSS ke dalam dokumen HTML:

```
<!DOCTYPE HTML>
<html>
  <head>
    <link rel="stylesheet" href="style.css">
  </head>
</html>
```

Membuat badan program

```
<!DOCTYPE HTML>
<html>
  <head>
    <link rel="stylesheet" href="style.css">
  </head>
```

```
<body>
</body>
</html>
```

Seusai membuat kepala form login, tahap berikutnya yaitu membuat badan. Untuk itu, Anda membutuhkan **tag <body>**. Bisa dibilang, ini adalah wadah untuk meletakkan seluruh komponen inti form. Caranya, cukup letakkan tag <body> di bawah tag <head>. Jika sudah, berarti Anda siap untuk melangkah ke tahap berikutnya.

Layout Program

Membuat tag <body> saja belum cukup untuk membuat form login yang baik. Alasannya, kemungkinan besar dokumen tidak akan rapi. Karena itu, dibutuhkan layout atau wadah untuk membungkus organ-organ form Anda.

Cara membuat layout ini yaitu menggunakan **elemen <div> dengan class = “container”**. Tag ini akan mengelompokkan atau menampung komponen tertentu ke dalam satu grup.

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Halaman Login</title>
    <link rel="stylesheet" href="style.css">
  </head>

  <body>
    <div class="container">
    </div>
  </body>
</html>
```

fungsi dari elemen ini akan lebih terasa nanti ketika kita sudah bermain dengan CSS.

Memberi judul form login

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Halaman Login</title>
    <link rel="stylesheet" href="style.css">
  </head>
```

```
<body>
  <div class="container">
    <h1>Login</h1>
  </div>
</body>
</html>
```

Judul form ini berbeda dengan judul halaman, Judul halaman ibarat nama ruangan yang kita pakai untuk menyimpan dokumen kerja. Sedangkan judul form merupakan nama dokumen kerja yang berada di ruang penyimpanan tersebut.

Membuat kotak form login

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Halaman Login</title>
    <link rel="stylesheet" href="style.css">
  </head>

  <body>
    <div class="container">
      <h1>Login</h1>
      <form>
      </form>
    </div>
  </body>
</html>
```

Mirip seperti tag <div>, tag <form> juga berfungsi untuk mengelompokkan elemen. Bedanya, tag <form> secara khusus menampung seluruh elemen yang berkaitan dengan sebuah form. Seperti kolom username dan password.

Membuat kolom username dan password

Setidaknya, ada tiga tag yang digunakan untuk membuat kolom username dan password. Anda akan membutuhkan **tag: <label>, <input type>, dan
**.

Tag <label> berfungsi untuk memberi keterangan teks pada halaman form.

Tag
 berfungsi sebagai *break* atau penanda baris baru.

Ada dua tipe yang akan digunakan pada form login. Pertama, **type=text**. Tipe ini akan menampilkan tulisan yang diketikkan di dalam kolom. Kedua, **type=password**. Menggunakan tipe ini, teks isian akan ditampilkan dalam bentuk karakter khusus, seperti dots.

```

<!DOCTYPE HTML>
<html>
  <head>
    <title>Halaman Login</title>
    <link rel="stylesheet" href="style.css">
  </head>

  <body>
    <div class="container">
      <h1>Login</h1>
      <form>
        <label>Username</label><br>
        <input type="text"><br>
        <label>Password</label><br>
        <input type="password"><br>
      </form>
    </div>
  </body>
</html>

```

Membuat tombol form login

Cara membuat tombol pada dokumen HTML caranya adalah Di dalam tag <form>, silakan berikan **tag <button>**.

```

<!DOCTYPE HTML>
<html>
  <head>
    <title>Halaman Login</title>
    <link rel="stylesheet" href="style.css">
  </head>

  <body>
    <div class="container">
      <h1>Login</h1>
      <form>
        <label>Username</label><br>
        <input type="text"><br>
        <label>Password</label><br>
        <input type="password"><br>
        <button>Log in</button>
      </form>
    </div>
  </body>
</html>

```


2.2. Code Program CSS

Mendesain Format Layout

```
*{
    margin: 0;
    padding: 0;
    outline: 0;
    font-family: 'Open Sans', sans-serif;
}
```

- **Margin.** Fungsinya untuk mengatur tepi garis luar. Berdasarkan posisinya, ada empat jenis margin: margin-top, margin-right, margin-bottom, margin-left;
- **Padding.** Artinya jarak dalam di antara konten dengan bingkai;
- **Outline.** Berfungsi sebagai garis bingkai form;
- **Font-family.** Berperan untuk mengatur jenis font sesuai dengan yang Anda inginkan.

Membuat Background Login

```
body{
    height: 100vh;
    background-color: turquoise;
    background-size: cover;
    background-position: center;
    background-repeat: no-repeat;
}
```

- **Height.** Fungsinya untuk mengatur ketinggian objek. Height: 100vh berarti tinggi objek sama dengan 100% tinggi layar tab;
- **Background-color.** Berperan untuk menambahkan warna pada background;
- **Background-size.** Berfungsi untuk mengatur ukuran gambar latar belakang. Ada tiga tipe ukuran yang bisa digunakan: auto, cover, dan contain;
- **Background-position.** Fungsinya menetapkan posisi awal gambar latar belakang. Banyak opsi untuk ini, yaitu: left-top, left-center, left-bottom, right-top, right-center, right-bottom, center-top, center, dan center-bottom;
- **Background repeat.** Ada kalanya dimensi gambar yang dipilih terlalu kecil atau memiliki pattern berulang sehingga membutuhkan perulangan supaya

background terisi penuh. Karena itu, ada dua value yang bisa Anda gunakan: repeat dan no-repeat.

Mendesain kotak form login

```
.container{
  position: absolute;
  left: 50%;
  top: 50%;
  transform: translate(-50%,-50%);
  padding: 20px 25px;
  width: 300px;

  background-color: rgba(0,0,0,.7);
  box-shadow: 0 0 10px rgba(255,255,255,.3);}
```

- **Position**, untuk mengatur penempatan objek dalam halaman website. Ada lima macam posisi: static, relative, fixed, absolute, dan sticky;
- **Left**, yaitu penempatan objek mulai dari jarak tepi kiri layar;
- **Top**, merupakan penempatan objek mulai dari tepi atas layar;
- **Transform**, memungkinkan kita membuat efek rotasi dan animasi pada suatu objek. Terdapat beberapa jenis gaya: translate, scale, skew, rotate, dan matrix;
- **Padding**, mengatur spasi atau jarak di dalam konten dengan satuan pixel;
- **Width**, mengatur lebar objek dalam satuan pixel;
- **Background-color**, mendesain pewarnaan pada bagian latar belakang objek;
- **Box-shadow**, fungsinya membuat bayangan pada objek.

Mendesain judul

```
.container h1{
  text-align: left;
  color: #fafafa;
  margin-bottom: 30px;
  text-transform: uppercase;
  border-bottom: 4px solid #2979ff;
}
```

- **Text-align**, yaitu untuk memposisikan teks. Ada beberapa posisi yang bisa Anda pilih: right, left, center, dan justify;
- **Color**, untuk mengatur warna teks;
- **Margin-bottom**, memberikan jarak dalam bagian bawah dalam satuan pixel;

- **Text-transform**, berperan untuk memodifikasi format teks. Ada tiga gaya yang bisa Anda pilih: uppercase, lowercase, dan capitalize;
- **Border-bottom**, fungsinya memberikan garis bingkai pada bagian bawah teks. Anda juga bisa memodifikasi warnanya.

Mendesain label teks

```
.container label{
  text-align: left;
  color: #90caf9;
```

- **Text-align**, untuk mengatur posisi teks. Silakan pilih salah satu di antara: left, right, center, atau justify;
- **Color**, fungsinya mengubah warna teks.

Mendesain kolom input

```
.container form input{
  width: calc(100% - 20px);
  padding: 8px 10px;
  margin-bottom: 15px;
  border: none;
  background-color: transparent;
  border-bottom: 2px solid #2979ff;
  color: #fff;
  font-size: 20px;
}
```

- **Width**, untuk mengatur lebar objek;
- **Padding**, mengatur jarak dalam objek;
- **Margin-bottom**, mengatur jarak objek dari garis tepi bagian bawah;
- **Border**, fungsinya memberikan bingkai pada objek;
- **Background-color**, untuk memodifikasi warna objek;
- **Border-bottom**; memberikan bingkai bawah serta modifikasi warnanya;
- **Color**, mengatur warna teks pada objek;
- **Font-size**, mengubah jenis font di dalam objek.

Mendesain tombol form login

```
.container form button{
  width: 100%;
  padding: 5px 0;
  border: none;
  background-color: #2979ff;
```

```
font-size: 18px;  
color: #fafafa;  
}
```

- **Width**, fungsinya mengatur lebar objek;
- **Padding**, untuk memodifikasi jarak dalam objek;
- **Border**, memberikan bingkai pada objek;
- **Background-color**, mengatur warna gambar latar belakang pada objek;
- **Font-size**, untuk mengatur ukuran font pada objek;
- **Color**, berfungsi untuk mendesain warna objek.

BAB III

KESIMPULAN

Form biasanya digunakan untuk mengumpulkan data dari pengunjung web kita. Mulai dari form untuk login, form kontak, form untuk pendaftaran user, bahkan untuk mengirimkan data antar halaman web. Penggunaan form hanya menggunakan HTML saja tidak akan terlalu berguna. Form biasanya hanya berupa *interface* yang disediakan untuk mengumpulkan data dari user, dan akan diproses dengan bahasa pemrograman web seperti JavaScript atau PHP, dan disimpan di dalam tabel MySQL. Tag dasar yang akan digunakan untuk membuat form di HTML adalah tag form, input, textarea, select dan option.

Sebuah form dalam HTML harus berada di dalam tag form, yang diawali dengan <form> dan diakhiri dengan </form>. Tag form akan membutuhkan beberapa atribut untuk dapat berfungsi dengan seharusnya.

Tag input merupakan tag paling banyak digunakan di dalam form dan memiliki banyak bentuk, mulai dari isian text biasa, text *password*, *checkbox*, *radio*, sampai dengan tombol *submit*, semuanya dalam bentuk tag <input>.

Tag textarea pada dasarnya sama dengan input type text, namun lebih besar dan dapat berisi banyak baris. Panjang dan banyak baris untuk text area di atur melalui atribut rows dan cols, atau melalui CSS.

Tag select digunakan untuk inputan yang telah tersedia nilainya, dan user hanya dapat memilih dari nilai yang ada. Tag select digunakan bersama-sama dengan tag option untuk membuat box pilihan.

Setiap tag inputan di dalam form harus ditambahkan *atribut* name agar dapat diproses oleh web server nantinya. Di dalam halaman proses (yang biasanya berupa bahasa PHP atau ASP), nilai dari atribut *name* inilah yang akan menjadi *variabel form*.

DAFTAR PUSTAKA

Dosen Teknik Informatika. 2021. *Modul Praktikum Pemrograman Web dan Mobile*

1. Print Pdf.

Kelasprogrammer. 2015, *Perbedaan Method Get dan Post Kapan Digunakan.*

<https://kelasprogrammer.com/perbedaan-method-get-dan-post-kapan-digunakan/>

Warungbelajar. 2016, *Penanganan Form di PHP.*

<https://www.warungbelajar.com/penanganan-form-di-php.html>

Duniailkom. 2015, *Belajar HTML Cara Membuat Form di HTML Tag Form.*

<https://www.duniailkom.com/belajar-html-cara-membuat-form-di-html-tag-form/>