

Memahami Formulir HTML dan Input Pengguna

Memahami formulir HTML sangat penting untuk membuat website interaktif yang dapat mengumpulkan informasi dari pengguna. Formulir digunakan untuk segala hal mulai dari halaman login dan formulir kontak hingga survei dan keranjang belanja online. Pelajaran ini akan memperkenalkan Anda pada konsep fundamental formulir HTML, berfokus pada elemen `<form>` dan atribut pentingnya, serta tipe input umum seperti text, password, dan email. Anda juga akan belajar tentang `<textarea>` untuk input teks yang lebih besar dan `<button>` untuk pengiriman formulir, serta tag `<select>` dan `<option>` untuk membuat menu dropdown. Di akhir pelajaran ini, Anda akan dapat membuat formulir HTML dasar dan memahami bagaimana mereka digunakan untuk mengumpulkan input pengguna.

Fondasi: Elemen `<form>`

Elemen `<form>` adalah landasan dari setiap formulir HTML. Ini bertindak sebagai wadah untuk semua elemen input berbeda yang membentuk formulir. Anggap saja sebagai fondasi di mana Anda membangun sistem pengumpulan data interaktif Anda. Tanpanya, field input Anda hanya akan menjadi elemen yang berdiri sendiri, tidak terhubung atau mampu mengirim data ke server.

Atribut Penting dari Elemen `<form>`

Elemen `<form>` memiliki beberapa atribut penting yang mendefinisikan bagaimana formulir berperilaku dan ke mana data yang dikumpulkan dikirim. Mari kita jelajahi dua yang paling penting: `action` dan `method`.

action: Atribut ini menentukan URL ke mana data formulir akan dikirim untuk diproses. URL ini biasanya menunjuk ke script server-side (misalnya, ditulis dalam PHP, Python, Node.js, dll.) yang akan menangani data.

Contoh: `<form action="/submit-form.php" method="post">` Dalam kasus ini, data formulir akan dikirim ke file `/submit-form.php` di server.

method: Atribut ini mendefinisikan metode HTTP yang digunakan untuk mengirim data formulir. Dua metode paling umum adalah get dan post.

get: Metode get menambahkan data formulir ke URL sebagai query string. Metode ini cocok untuk jumlah data kecil yang tidak sensitif, seperti kueri pencarian. Data yang dikirim menggunakan get terlihat di URL.

Contoh: Jika formulir dengan `method="get"` memiliki field bernama "search" dengan nilai "HTML Forms", URL mungkin terlihat seperti ini: `?search=HTML+Forms`. Tanda + mewakili spasi dalam encoding URL.

post: Metode post mengirim data formulir di body dari permintaan HTTP. Metode ini lebih aman dan cocok untuk jumlah data yang lebih besar atau informasi sensitif, seperti password. Data yang dikirim menggunakan post tidak terlihat di URL.

Contoh: `<form action="/process_data" method="post"> </form>`. Ketika formulir dikirim, data dikirim di balik layar ke `/process_data`.

Sangat penting untuk memilih metode yang tepat tergantung pada sensitivitas dan ukuran data yang akan dikumpulkan. Menggunakan GET untuk password akan menjadi kerentanan keamanan yang besar.

Contoh Praktis

Mari kita buat formulir sederhana yang mengumpulkan nama dan alamat email pengguna.

html

```
<form action="/submit-form.php" method="post">
  <label for="name">Nama:</label><br>
  <input type="text" id="name" name="name"><br><br>
  <label for="email">Email:</label><br>
  <input type="email" id="email" name="email"><br><br>
  <button type="submit">Kirim</button>
</form>
```

Dalam contoh ini:

- Elemen `<form>` menentukan bahwa data akan dikirim ke `/submit-form.php` menggunakan metode post.
- Kita memiliki dua field input: satu untuk nama pengguna dan satu untuk alamat email mereka. Perhatikan atribut name pada field `<input>`; ini sangat penting, karena digunakan sebagai kunci ketika data dikirim ke server.
- Elemen `<label>` memberikan teks deskriptif untuk setiap field input.
- Elemen `<button>` memungkinkan pengguna untuk mengirim formulir.

Field Input: Tag `<input>` dan Tipenya

Tag `<input>` adalah salah satu elemen paling serbaguna dalam formulir HTML. Atribut type-nya menentukan jenis field input yang ditampilkan kepada pengguna. Mari kita periksa beberapa tipe input yang paling umum dan berguna.

Tipe Input Umum

text: Membuat field input teks satu baris. Ini cocok untuk mengumpulkan data teks umum seperti nama, alamat, atau kueri pencarian.

html

```
<label for="username">Nama Pengguna:</label><br>
<input type="text" id="username" name="username"><br>
```

password: Membuat field input password. Teks yang dimasukkan dalam field ini disembunyikan (biasanya dengan asterisk atau titik) untuk melindunginya agar tidak terlihat oleh orang lain.

html

```
<label for="password">Password:</label><br>
<input type="password" id="password" name="password"><br>
```

email: Membuat field input email. Tipe input ini memberikan validasi dasar untuk memastikan bahwa teks yang dimasukkan dalam format email yang valid (misalnya, berisi simbol @ dan domain).

html

```
<label for="email">Email:</label><br>
```

```
<input type="email" id="email" name="email"><br>
```

Meskipun browser melakukan pemeriksaan dasar, Anda harus selalu melakukan validasi yang lebih kuat di sisi server.

Atribut untuk Tag `<input>`

Selain `type`, tag `<input>` memiliki beberapa atribut penting lainnya yang mengontrol perilaku dan tampilannya.

id: Menentukan identifier unik untuk field input. Ini terutama digunakan untuk mengaitkan elemen `<label>` dengan field input dan untuk styling dengan CSS atau manipulasi dengan JavaScript.

name: Menentukan nama field input. Ini sangat penting karena nama digunakan sebagai kunci ketika data formulir dikirim ke server. Tanpa nama, data yang dimasukkan dalam field input tidak akan dikirim.

value: Menentukan nilai awal field input. Untuk tipe `text`, `email`, dan `password`, ini mengatur teks default di field.

placeholder: Menentukan petunjuk singkat yang menjelaskan nilai yang diharapkan dari field input. Teks placeholder ditampilkan di dalam field input ketika kosong dan menghilang ketika pengguna mulai mengetik.

required: Menentukan bahwa field input harus diisi sebelum formulir dapat dikirim. Browser modern akan mencegah pengiriman formulir jika field yang diperlukan kosong.

Contoh Menggunakan Atribut Input

html

```
<label for="city">Kota:</label><br>
<input type="text" id="city" name="city" placeholder="Masukkan kota Anda" required><br>
```

Dalam contoh ini:

- Atribut `placeholder` memberikan petunjuk kepada pengguna tentang apa yang harus dimasukkan dalam field.

- Atribut required memastikan bahwa pengguna harus memasukkan kota sebelum mengirim formulir.

Area Teks: Tag <textarea>

Tag <textarea> membuat field input teks multi-baris, yang berguna untuk mengumpulkan jumlah teks yang lebih besar, seperti komentar, deskripsi, atau pesan.

Atribut Kunci dari Tag <textarea>

id: Menentukan identifier unik untuk area teks. Mirip dengan <input>, ini penting untuk asosiasi <label> dan styling.

name: Menentukan nama area teks, digunakan untuk mengidentifikasi data ketika dikirim ke server.

rows: Menentukan tinggi yang terlihat dari area teks dalam baris teks.

cols: Menentukan lebar yang terlihat dari area teks dalam karakter.

placeholder: Menentukan petunjuk singkat yang menjelaskan nilai yang diharapkan dari area teks.

required: Menentukan bahwa area teks harus diisi sebelum formulir dapat dikirim.

Contoh Area Teks

html

```
<label for="message">Pesan:</label><br>
<textarea id="message" name="message" rows="4" cols="50" placeholder="Masukkan pesan
Anda di sini" required></textarea><br>
```

Dalam contoh ini:

- Atribut rows mengatur tinggi yang terlihat dari area teks menjadi 4 baris.
- Atribut cols mengatur lebar yang terlihat dari area teks menjadi 50 karakter.
- Atribut placeholder memberikan petunjuk yang membantu kepada pengguna.

Tombol: Tag `<button>`

Tag `<button>` membuat tombol yang dapat diklik yang dapat digunakan untuk mengirim formulir atau memicu tindakan lain di halaman web. Meskipun Anda dapat menggunakan `<input type="button">`, tag `<button>` menawarkan lebih banyak fleksibilitas, termasuk kemampuan untuk menyertakan konten HTML di dalam tombol (misalnya, ikon atau gambar).

Tipe Tombol

Atribut `type` dari tag `<button>` sangat penting untuk mendefinisikan perilakunya dalam formulir.

submit: Membuat tombol yang mengirim data formulir ke server. Ini adalah tipe default jika tidak ada tipe yang ditentukan.

html

```
<button type="submit">Kirim</button>
```

reset: Membuat tombol yang mereset semua field formulir ke nilai defaultnya.

html

```
<button type="reset">Reset</button>
```

button: Membuat tombol generik yang dapat digunakan untuk memicu fungsi JavaScript kustom. Tipe tombol ini tidak memiliki perilaku default yang terkait dengan pengiriman formulir.

html

```
<button type="button" onclick="myFunction()">Klik Saya</button>
```

(Catatan: `onclick` adalah event handler JavaScript; kita akan mengeksplorasi integrasi JavaScript di modul selanjutnya.)

Contoh Menggunakan Tag `<button>`

html

```
<button type="submit">Kirim Pesan</button>
```

```
<button type="reset">Hapus Formulir</button>
```

Dalam contoh ini:

- Tombol "Kirim Pesan" mengirim formulir.
- Tombol "Hapus Formulir" mereset field formulir.

Kotak Select: Tag `<select>` dan `<option>`

Tag `<select>` membuat daftar dropdown, memungkinkan pengguna untuk memilih dari serangkaian opsi yang telah ditentukan. Setiap opsi dalam daftar dropdown didefinisikan menggunakan tag `<option>`.

Atribut Kunci

Tag `<select>`:

- **id**: Menentukan identifier unik untuk kotak select.
- **name**: Menentukan nama kotak select, digunakan untuk mengidentifikasi nilai yang dipilih ketika formulir dikirim ke server.
- **required**: Menentukan bahwa nilai harus dipilih sebelum formulir dapat dikirim.

Tag `<option>`:

- **value**: Menentukan nilai yang akan dikirim ke server ketika opsi dipilih.
- **selected**: Menentukan bahwa opsi harus dipilih sebelumnya ketika halaman dimuat.

Contoh Kotak Select

html

```
<label for="country">Negara:</label><br>
<select id="country" name="country" required>
  <option value="">Pilih negara</option>
  <option value="USA">Amerika Serikat</option>
  <option value="Canada">Kanada</option>
  <option value="UK">Britania Raya</option>
  <option value="Germany">Jerman</option>
</select><br>
```

Dalam contoh ini:

- Elemen `<select>` membuat daftar dropdown dengan id "country" dan nama "country".
- Elemen `<option>` pertama dengan nilai kosong bertindak sebagai placeholder, mendorong pengguna untuk memilih negara.
- Atribut required memastikan bahwa pengguna harus memilih negara sebelum mengirim formulir.
- Elemen `<option>` berikutnya memberikan opsi negara yang tersedia.

Menyatukan Semuanya: Contoh Formulir Lengkap

Berikut adalah contoh yang lebih komprehensif yang menggabungkan semua elemen yang telah kita bahas:

html

```
<form action="/process-form.php" method="post">
    <label for="name">Nama:</label><br>
        <input type="text" id="name" name="name" placeholder="Masukkan nama Anda"
required><br><br>
    <label for="email">Email:</label><br>
        <input type="email" id="email" name="email" placeholder="Masukkan email Anda"
required><br><br>
    <label for="country">Negara:</label><br>
    <select id="country" name="country" required>
        <option value="">Pilih negara</option>
        <option value="USA">Amerika Serikat</option>
        <option value="Canada">Kanada</option>
        <option value="UK">Britania Raya</option>
        <option value="Germany">Jerman</option>
    </select><br><br>
    <label for="message">Pesan:</label><br>
        <textarea id="message" name="message" rows="4" cols="50" placeholder="Masukkan
pesan Anda" required></textarea><br><br>
    <button type="submit">Kirim</button>
</form>
```

Formulir ini mengumpulkan nama pengguna, email, negara, dan pesan. Semua field diperlukan, dan placeholder yang sesuai disediakan untuk membimbing pengguna. Ketika formulir dikirim, data akan dikirim ke `/process-form.php` menggunakan metode post.

Latihan

1. **Tingkatkan Halaman Resep:** Rujuk kembali halaman resep yang Anda buat di Modul 2. Tambahkan formulir di bagian bawah halaman yang memungkinkan pengguna untuk mengirimkan komentar atau ulasan untuk resep. Sertakan field untuk nama mereka, email (opsional), dan komentar.
2. **Buat Formulir Survei Sederhana:** Rancang formulir dengan beberapa pertanyaan menggunakan tipe input berbeda. Sertakan field teks untuk jawaban singkat, area teks untuk respons yang lebih panjang, dan kotak select untuk pertanyaan pilihan ganda. Pastikan untuk menggunakan label dan placeholder yang sesuai untuk membimbing pengguna. Tambahkan tombol kirim.
3. **Modifikasi Formulir:** Ambil contoh formulir lengkap yang disediakan dalam pelajaran dan modifikasi untuk menyertakan tipe input tambahan, seperti input number untuk umur, dan tombol radio untuk pemilihan jenis kelamin. Ingatlah untuk menggunakan label deskriptif untuk setiap field input dan pastikan bahwa semua field memiliki atribut name yang sesuai.
4. **Eksperimen dengan Atribut:** Ambil salah satu contoh formulir yang telah Anda buat dan eksperimen dengan atribut berbeda untuk field input, seperti maxlength, minlength, pattern (untuk validasi regular expression), dan disabled. Amati bagaimana atribut ini mempengaruhi perilaku dan tampilan formulir.

Ringkasan

Formulir HTML sekarang dibuat dan siap untuk diberi gaya dan dikirim ke server!

Dalam pelajaran ini, Anda belajar tentang blok bangunan fundamental formulir HTML, termasuk elemen `<form>`, berbagai tipe input, area teks, tombol, dan kotak select. Anda mengeksplorasi atribut kunci dari elemen-elemen ini dan bagaimana mereka mengontrol perilaku dan tampilan formulir. Anda sekarang dilengkapi untuk membuat formulir HTML dasar yang dapat mengumpulkan input pengguna dan mengirimkannya ke server untuk diproses.

Selanjutnya, kita akan mendalami elemen `<form>` dan Atributnya.

Elemen <form> dan Atributnya

Elemen `<form>` adalah landasan interaksi pengguna dalam HTML. Ini bertindak sebagai wadah untuk berbagai elemen input, seperti field teks, tombol, checkbox, dan lainnya, memungkinkan pengguna untuk mengirimkan data ke server untuk diproses. Memahami elemen `<form>` dan atributnya sangat penting untuk membangun aplikasi web yang interaktif dan dinamis. Tanpa formulir, kita tidak akan dapat login ke website, melakukan pembelian online, atau meninggalkan komentar di posting blog. Pelajaran ini akan memberikan tinjauan komprehensif tentang elemen `<form>` dan atribut pentingnya, meletakkan dasar untuk membuat formulir yang kuat dan ramah pengguna.

Memahami Elemen <form>

Elemen `<form>` digunakan untuk membuat formulir HTML untuk input pengguna. Ini adalah elemen block-level, yang berarti menempati lebar penuh yang tersedia dan dimulai pada baris baru. Elemen `<form>` itu sendiri tidak me-render apa pun yang terlihat di halaman; ini adalah wadah untuk elemen lain yang melakukannya.

Berikut adalah contoh dasar dari elemen `<form>`:

```
html
<form>
  <!-- Elemen input akan ada di sini -->
</form>
```

Contoh ini membuat formulir kosong. Untuk membuatnya berguna, kita perlu menambahkan elemen input di dalam tag `<form>`, yang akan dibahas dalam pelajaran berikutnya.

Atribut Kunci dari Elemen <form>

Elemen `<form>` memiliki beberapa atribut penting yang mengontrol perilakunya dan bagaimana berinteraksi dengan server. Mari kita jelajahi yang paling penting:

- **action:** Menentukan URL ke mana data formulir akan dikirim ketika formulir dikirim. Atribut ini wajib untuk formulir fungsional apa pun.

- **method:** Menentukan metode HTTP yang digunakan untuk mengirim data formulir. Dua metode paling umum adalah GET dan POST.
- **accept:** Menentukan tipe MIME yang dapat ditangani server. Ini digunakan untuk membatasi jenis file yang dapat diunggah pengguna.
- **accept-charset:** Menentukan encoding karakter yang dapat ditangani server untuk data formulir.
- **autocomplete:** Menentukan apakah browser harus secara otomatis melengkapi formulir berdasarkan nilai yang telah dimasukkan pengguna sebelumnya.
- **enctype:** Menentukan bagaimana data formulir harus di-encode ketika mengirimkannya ke server.
- **name:** Menentukan nama untuk formulir. Atribut ini terutama digunakan untuk mereferensikan formulir dalam JavaScript.
- **novalidate:** Menentukan bahwa formulir tidak boleh divalidasi ketika dikirim.
- **target:** Menentukan di mana menampilkan respons setelah mengirim formulir.

Mari kita dalami setiap atribut ini dengan penjelasan dan contoh detail:

Atribut action

Atribut action bisa dibilang adalah atribut paling penting dari elemen `<form>`. Ini mendefinisikan URL ke mana data formulir akan dikirim untuk diproses. URL ini biasanya menunjuk ke script atau aplikasi server-side yang menangani data yang dikirimkan.

Contoh:

```
html
<form action="/submit-form" method="POST">
  
  <button type="submit">Kirim</button>
</form>
```

Dalam contoh ini, ketika formulir dikirim, data akan dikirim ke URL `/submit-form` di server. Server di lokasi tersebut diharapkan untuk menangani data formulir yang masuk.

Skenario Hipotetis: Bayangkan formulir kontak sederhana di website pribadi. Atribut action akan menunjuk ke script (misalnya, script PHP) di server yang mengumpulkan nama, email, dan pesan pengguna, dan kemudian mengirim informasi tersebut ke pemilik website melalui email.

Atribut method

Atribut method menentukan metode HTTP yang digunakan untuk mengirim data formulir ke server. Dua metode yang paling umum digunakan adalah GET dan POST.

GET: Menambahkan data formulir ke URL dalam query string. Metode ini cocok untuk jumlah data kecil dan ketika data tidak sensitif. Permintaan GET juga di-cache oleh browser, dan terlihat dalam riwayat browser.

POST: Mengirim data formulir dalam body permintaan HTTP. Metode ini cocok untuk jumlah data yang lebih besar dan ketika data sensitif (misalnya, password). Permintaan POST tidak di-cache, dan tidak terlihat dalam riwayat browser.

Contoh:

Menggunakan metode GET:

```
html
<form action="/search" method="GET">
  <input type="text" name="query" placeholder="Cari...">
  <button type="submit">Cari</button>
</form>
```

Dalam kasus ini, jika pengguna memasukkan "example" di kotak pencarian, browser akan menavigasi ke URL seperti /search?query=example.

Menggunakan metode POST:

```
html
<form action="/login" method="POST">
  <input type="text" name="username" placeholder="Nama Pengguna">
  <input type="password" name="password" placeholder="Password">
  <button type="submit">Login</button>
</form>
```

Di sini, nama pengguna dan password akan dikirim ke URL /login dalam body permintaan, tidak terlihat di URL.

Contoh Dunia Nyata:

- **GET:** Formulir pencarian di website e-commerce.
- **POST:** Formulir login di platform media sosial.

Atribut accept

Atribut accept menentukan tipe MIME yang dapat ditangani server. Atribut ini terutama digunakan dengan elemen input upload file untuk membatasi jenis file yang dapat diunggah pengguna.

Contoh:

html

```
<form action="/upload" method="POST" enctype="multipart/form-data">
  <input type="file" name="avatar" accept="image/png, image/jpeg">
  <button type="submit">Unggah</button>
</form>
```

Dalam contoh ini, atribut accept membatasi pengguna untuk hanya mengunggah file gambar PNG dan JPEG. `enctype="multipart/form-data"` diperlukan untuk formulir yang menyertakan upload file.

Contoh Dunia Nyata: Website yang memungkinkan pengguna mengunggah foto profil mungkin menggunakan atribut accept untuk memastikan bahwa hanya file gambar yang diunggah.

Atribut accept-charset

Atribut accept-charset menentukan encoding karakter yang dapat ditangani server untuk data formulir. Encoding karakter mendefinisikan bagaimana karakter direpresentasikan dalam format biner.

Contoh:

html

```
<form action="/submit" method="POST" accept-charset="UTF-8">
  <input type="text" name="name" placeholder="Nama">
  <button type="submit">Kirim</button>
```

```
</form>
```

Dalam contoh ini, atribut accept-charset menentukan bahwa server dapat menangani data yang di-encode UTF-8. UTF-8 adalah encoding karakter yang banyak digunakan yang mendukung berbagai karakter dari bahasa berbeda. Umumnya praktik yang baik untuk menggunakan UTF-8 untuk aplikasi web.

Atribut autocomplete

Atribut autocomplete menentukan apakah browser harus secara otomatis melengkapi formulir berdasarkan nilai yang telah dimasukkan pengguna sebelumnya. Ini dapat digunakan pada elemen `<form>` atau pada elemen input individual.

Contoh:

Pada elemen `<form>`:

```
html
<form action="/submit" method="POST" autocomplete="on">
  <input type="text" name="name" placeholder="Nama">
  <input type="email" name="email" placeholder="Email">
  <button type="submit">Kirim</button>
</form>
```

Ini mengaktifkan autocomplete untuk semua field input dalam formulir.

Pada elemen input individual:

```
html
<form action="/submit" method="POST">
  <input type="text" name="name" placeholder="Nama" autocomplete="name">
  <input type="email" name="email" placeholder="Email" autocomplete="email">
  <button type="submit">Kirim</button>
</form>
```

Ini mengaktifkan autocomplete hanya untuk field input yang ditentukan, dan memungkinkan memberikan petunjuk yang lebih spesifik ke browser (misalnya `name`, `email`, `street-address`, dll.).

Atribut autocomplete dapat memiliki nilai berikut:

- **on**: Mengaktifkan autocomplete.
- **off**: Menonaktifkan autocomplete.

Contoh Dunia Nyata: Keranjang belanja online sering menggunakan autocomplete untuk mengingat alamat pengiriman dan informasi kartu kredit Anda, membuatnya lebih cepat untuk menyelesaikan pembelian di masa depan.

Atribut enctype

Atribut enctype menentukan bagaimana data formulir harus di-encode ketika mengirimkannya ke server. Atribut ini hanya relevan ketika atribut method diatur ke POST.

Atribut enctype dapat memiliki nilai berikut:

- **application/x-www-form-urlencoded**: Ini adalah nilai default. Ini meng-encode data formulir sebagai pasangan key-value yang dipisahkan oleh ampersand (&). Spasi diganti dengan tanda plus (+), dan karakter khusus di-encode.
- **multipart/form-data**: Nilai ini digunakan ketika formulir menyertakan upload file. Ini meng-encode data formulir sebagai beberapa bagian, dengan setiap bagian mewakili field atau file yang berbeda.
- **text/plain**: Nilai ini meng-encode data formulir sebagai teks biasa, tanpa encoding khusus. Jarang digunakan.

Contoh:

Menggunakan application/x-www-form-urlencoded (default):

```
html
<form action="/submit" method="POST" enctype="application/x-www-form-urlencoded">
  <input type="text" name="name" placeholder="Nama">
  <button type="submit">Kirim</button>
</form>
```

Menggunakan multipart/form-data untuk upload file:

```
html
<form action="/upload" method="POST" enctype="multipart/form-data">
```

```
<input type="file" name="avatar">
<button type="submit">Unggah</button>
</form>
```

Penting: Ketika menggunakan upload file, Anda harus mengatur atribut enctype ke multipart/form-data.

Atribut name

Atribut name menentukan nama untuk formulir. Atribut ini terutama digunakan untuk mereferensikan formulir dalam JavaScript.

Contoh:

```
html
<form action="/submit" method="POST" name="myForm">
  <input type="text" name="name" placeholder="Nama">
  <button type="submit">Kirim</button>
</form>
<script>
  const form = document.forms["myForm"];
  // Akses elemen formulir menggunakan JavaScript
</script>
```

Dalam contoh ini, formulir diberi nama "myForm". Ini memungkinkan Anda mengakses formulir menggunakan `document.forms["myForm"]` dalam JavaScript.

Atribut novalidate

Atribut novalidate menentukan bahwa formulir tidak boleh divalidasi ketika dikirim. HTML5 menyediakan fitur validasi formulir built-in, seperti memerlukan field tertentu untuk diisi atau memvalidasi format alamat email. Atribut novalidate menonaktifkan fitur-fitur ini.

Contoh:

```
html
<form action="/submit" method="POST" novalidate>
  <input type="email" name="email" placeholder="Email" required>
```

```
<button type="submit">Kirim</button>  
</form>
```

Dalam contoh ini, meskipun field input email memiliki atribut required (yang biasanya memicu validasi), atribut novalidate pada elemen `<form>` menonaktifkan validasi, dan formulir akan dikirim bahkan jika field email kosong.

Skenario Dunia Nyata: Selama pengujian atau pengembangan, Anda mungkin sementara menonaktifkan validasi formulir menggunakan atribut novalidate untuk melewati error validasi dan dengan cepat mengirim formulir.

Atribut target

Atribut target menentukan di mana menampilkan respons setelah mengirim formulir. Ini bekerja sama seperti atribut target pada tag `<a>` (anchor).

Contoh:

```
html  
<form action="/submit" method="POST" target="_blank">  
  <input type="text" name="name" placeholder="Nama">  
  <button type="submit">Kirim</button>  
</form>
```

Dalam contoh ini, ketika formulir dikirim, respons dari server akan ditampilkan di tab atau jendela browser baru (karena `target="_blank"`).

Atribut target dapat memiliki nilai berikut:

- `_blank`: Membuka respons di tab atau jendela baru.
- `_self`: Membuka respons di frame yang sama (ini adalah default).
- `_parent`: Membuka respons di frame induk.
- `_top`: Membuka respons di body penuh jendela.
- `framename`: Membuka respons di frame bernama.

Contoh Praktis dan Demonstrasi

Mari kita buat beberapa contoh praktis untuk mendemonstrasikan penggunaan elemen `<form>` dan atributnya.

Contoh 1: Formulir Kontak Sederhana

html

```
<form action="/contact" method="POST">
  <label for="name">Nama:</label><br>
  <input type="text" id="name" name="name" placeholder="Nama Anda" required><br><br>
  <label for="email">Email:</label><br>
  <input type="email" id="email" name="email" placeholder="Email Anda" required><br><br>
  <label for="message">Pesan:</label><br>
    <textarea id="message" name="message" rows="4" cols="50" placeholder="Pesan Anda"></textarea><br><br>
  <button type="submit">Kirim Pesan</button>
</form>
```

Contoh ini membuat formulir kontak sederhana dengan field untuk nama, email, dan pesan. Atribut `action` menentukan bahwa data formulir akan dikirim ke URL `/contact` menggunakan metode POST. Atribut `required` pada field input nama dan email memastikan bahwa field ini harus diisi sebelum formulir dapat dikirim.

Contoh 2: Formulir Pendaftaran Pengguna

html

```
<form action="/register" method="POST" autocomplete="on">
  <label for="username">Nama Pengguna:</label><br>
    <input type="text" id="username" name="username" placeholder="Nama Pengguna" required><br><br>
  <label for="password">Password:</label><br>
    <input type="password" id="password" name="password" placeholder="Password" required><br><br>
  <label for="confirm_password">Konfirmasi Password:</label><br>
    <input type="password" id="confirm_password" name="confirm_password" placeholder="Konfirmasi Password" required><br><br>
  <label for="email">Email:</label><br>
    <input type="email" id="email" name="email" placeholder="Email" autocomplete="email"><br><br>
```

```
<button type="submit">Daftar</button>  
</form>
```

Contoh ini membuat formulir pendaftaran pengguna dengan field untuk nama pengguna, password, konfirmasi password, dan email. Atribut autocomplete digunakan untuk mengaktifkan autocomplete untuk formulir. `autocomplete="email"` pada field email memberikan petunjuk spesifik ke browser.

Contoh 3: Formulir Upload File

html

```
<form action="/upload" method="POST" enctype="multipart/form-data">  
  <label for="file">Pilih File:</label><br>  
  <input type="file" id="file" name="file" accept=".pdf, .doc, .docx"><br><br>  
  <button type="submit">Unggah</button>  
</form>
```

Contoh ini membuat formulir upload file. Atribut enctype diatur ke multipart/form-data untuk mengaktifkan upload file. Atribut accept membatasi pengguna untuk hanya mengunggah file PDF, DOC, dan DOCX.

Latihan dan Aktivitas Praktik

- Modifikasi Formulir Kontak:** Tambahkan field "Subjek" ke contoh formulir kontak dan modifikasi kode HTML yang sesuai.
- Buat Formulir Pencarian:** Rancang formulir pencarian yang menggunakan metode GET untuk mengirimkan kueri pencarian ke mesin pencari (misalnya, Google). Sertakan URL action yang sesuai.
- Implementasikan Validasi Formulir:** Menggunakan atribut validasi built-in HTML5 (misalnya, required, pattern), tambahkan validasi ke formulir pendaftaran pengguna untuk memastikan bahwa field password dan konfirmasi password cocok dan bahwa nama pengguna setidaknya 8 karakter. (Catatan: atribut pattern memungkinkan Anda menentukan regular expression untuk memvalidasi input).
- Eksperimen dengan Atribut target:** Buat formulir sederhana dan eksperimen dengan nilai berbeda dari atribut target (_blank, _self, _parent, _top) untuk melihat bagaimana mereka mempengaruhi di mana respons formulir ditampilkan.

Anda dapat menggunakan layanan seperti httpbin.org untuk menguji pengiriman formulir.

Ringkasan dan Langkah Selanjutnya

Dalam pelajaran ini, kita mengeksplorasi elemen `<form>` dan atribut kuncinya. Kita belajar cara menggunakan atribut action untuk menentukan URL ke mana data formulir dikirim, atribut method untuk menentukan metode HTTP yang digunakan untuk mengirim data, dan atribut lain seperti accept, autocomplete, enctype, name, novalidate, dan target untuk mengontrol perilaku formulir. Kita juga bekerja melalui contoh praktis membuat formulir kontak, formulir pendaftaran, dan formulir upload file.

Di pelajaran mendatang, kita akan mendalami berbagai elemen input yang dapat digunakan dalam `<form>`, seperti field teks, field password, checkbox, tombol radio, dan kotak select. Kita juga akan mengeksplorasi cara menggunakan elemen-elemen ini untuk membuat formulir yang lebih kompleks dan interaktif. Kita akan belajar tentang tag `<input>` dan berbagai tipenya, tag `<textarea>` untuk input teks multi-baris, dan tag `<button>` untuk membuat tombol formulir. Kemudian kita akan beralih ke tag `<select>` dan `<option>` untuk menu dropdown.

Field Input: Tag `<input>` dan Tipe (text, password, email)

Formulir HTML sangat penting untuk mengumpulkan data dari pengguna di website Anda. Elemen `<input>` adalah landasan pembuatan formulir, memungkinkan Anda mengumpulkan berbagai jenis informasi. Pelajaran ini akan mendalami tag `<input>` dan mengeksplorasi tipe fundamentalnya: text, password, dan email. Kita akan membahas atribut yang mengontrol perilaku mereka dan bagaimana mereka berkontribusi untuk membangun formulir yang fungsional dan ramah pengguna.

Tag `<input>`: Alat yang Serbaguna

Tag `<input>` adalah elemen kosong (artinya tidak memiliki tag penutup) yang digunakan untuk membuat kontrol interaktif untuk formulir berbasis web untuk menerima data dari

pengguna; data seperti teks, password, atau bahkan file. Atribut terpenting dari tag `<input>` adalah atribut `type`. Atribut ini mendefinisikan jenis kontrol input yang akan ditampilkan. Mari kita jelajahi beberapa tipe input fundamental.

Input Teks: Mengumpulkan Informasi Dasar

Atribut `type="text"` membuat field input teks satu baris. Ini adalah tipe input paling dasar dan cocok untuk mengumpulkan nama, alamat, atau informasi berbasis teks pendek lainnya.

html

```
<label for="firstName">Nama Depan:</label><br>
<input type="text" id="firstName" name="firstName"><br>
```

- **label:** Tag `<label>` memberikan caption yang ramah pengguna untuk field input. Atribut `for` dari `<label>` harus cocok dengan `id` dari elemen `<input>`, mengaitkan label dengan input. Ini meningkatkan aksesibilitas, memungkinkan pengguna untuk mengklik label untuk fokus pada field input.
- **input `type="text"`:** Ini mendefinisikan field input sebagai input teks satu baris.
- **id:** Atribut `id` memberikan identifier unik untuk elemen input dalam dokumen HTML.
- **name:** Atribut `name` sangat penting untuk pemrosesan formulir. Ini menentukan nama field input, yang digunakan untuk mengidentifikasi data ketika formulir dikirim ke server.

Atribut untuk Input Teks

Beberapa atribut dapat digunakan untuk menyesuaikan input teks:

maxlength: Menentukan jumlah maksimum karakter yang diizinkan dalam field input.

html

```
<input type="text" id="city" name="city" maxlength="50">
```

Dalam contoh ini, pengguna dapat memasukkan maksimum 50 karakter dalam field input "city".

size: Menentukan lebar yang terlihat dari field input (dalam karakter). Perhatikan bahwa size tidak membatasi jumlah karakter yang dapat dimasukkan pengguna (gunakan maxlength untuk itu); ini hanya mempengaruhi ukuran visual kotak input.

html

```
<input type="text" id="state" name="state" size="2">
```

Ini membuat field input teks yang secara visual lebar dua karakter.

placeholder: Memberikan petunjuk di dalam field input, yang menghilang ketika pengguna mulai mengetik.

html

```
<input type="text" id="search" name="search" placeholder="Masukkan kueri pencarian Anda">
```

Teks "Masukkan kueri pencarian Anda" akan ditampilkan dalam field input sampai pengguna mulai mengetik.

required: Menentukan bahwa field input harus diisi sebelum formulir dapat dikirim.

html

```
<input type="text" id="lastName" name="lastName" required>
```

Formulir tidak dapat dikirim jika field input "lastName" kosong.

value: Mengatur nilai default untuk field input.

html

```
<input type="text" id="country" name="country" value="USA">
```

Field input akan diisi sebelumnya dengan "USA". Pengguna dapat mengubah nilai ini.

Contoh

Mari kita bangun contoh halaman resep sederhana yang secara hipotetis kita buat di Modul 2. Bayangkan kita ingin memungkinkan pengguna untuk mengirimkan resep mereka sendiri. Field input teks akan sempurna untuk mengumpulkan nama resep:

```
html
<label for="recipeName">Nama Resep:</label><br>
<input type="text" id="recipeName" name="recipeName" placeholder="Masukkan nama
resep" required>
```

Input Password: Mengumpulkan Data Sensitif Secara Aman

Atribut `type="password"` membuat field input password. Teks yang dimasukkan ke dalam field ini disembunyikan (biasanya dengan asterisk atau titik) untuk melindunginya agar tidak dilihat oleh orang lain.

```
html
<label for="password">Password:</label><br>
<input type="password" id="password" name="password">
```

Pertimbangan Keamanan

Sangat penting untuk memahami bahwa `type="password"` hanya memberikan keamanan visual. Data masih dikirimkan sebagai teks biasa kecuali formulir dikirimkan melalui HTTPS (SSL/TLS). Selalu gunakan HTTPS untuk halaman apa pun yang berisi formulir password untuk mengenkripsi data selama transmisi. Selain itu, Anda tidak boleh pernah menyimpan password dalam teks biasa di server Anda. Selalu gunakan teknik hashing dan salting yang tepat.

Atribut untuk Input Password

Input password mendukung atribut yang sama dengan input teks, seperti `maxlength`, `size`, `placeholder`, dan `required`.

minlength: Menentukan jumlah minimum karakter yang diperlukan dalam field password.

html

```
<input type="password" id="password" name="password" minlength="8" required>
```

Password harus setidaknya 8 karakter sebelum formulir dapat dikirim.

Contoh

Memperluas formulir pengiriman resep kita, kita dapat menambahkan field password untuk akun pengguna:

html

```
<label for="password">Password:</label><br>
<input type="password" id="password" name="password" minlength="8" required
placeholder="Masukkan password Anda"><br>
```

Input Email: Memvalidasi Alamat Email

Atribut `type="email"` membuat field input khusus untuk alamat email. Browser modern menyediakan validasi built-in untuk memastikan bahwa teks yang dimasukkan dalam format email yang valid (misalnya, berisi simbol @ dan domain). Namun, ini bukan validasi yang sempurna dan tidak boleh diandalkan sebagai satu-satunya metode verifikasi email. Validasi server-side masih diperlukan.

html

```
<label for="email">Email:</label><br>
<input type="email" id="email" name="email" required>
```

Cara Kerja Validasi Email

Ketika formulir yang berisi input email dikirim, browser memeriksa apakah nilai yang dimasukkan sesuai dengan struktur alamat email dasar. Jika input tidak dalam format yang valid, browser akan menampilkan pesan error, mencegah formulir dikirim. Misalnya, "test" atau "test@" akan ditandai sebagai tidak valid, sementara "test@example.com" akan dianggap valid (setidaknya oleh validasi dasar browser).

Atribut untuk Input Email

Input email juga mendukung atribut seperti maxlength, size, placeholder, dan required. Atribut multiple spesifik untuk tipe input email.

multiple: Memungkinkan pengguna memasukkan beberapa alamat email, dipisahkan dengan koma.

html

```
<label for="emails">Email:</label><br>
<input type="email" id="emails" name="emails" multiple>
```

Pengguna dapat memasukkan "user1@example.com, user2@example.net" ke dalam field input.

Contoh

Mari kita gabungkan input email ke dalam formulir pengiriman resep kita:

html

```
<label for="email">Email:</label><br>
<input type="email" id="email" name="email" required placeholder="Masukkan email
Anda"><br>
```

Aktivitas Praktik

1. **Formulir Informasi Pribadi:** Buat formulir HTML sederhana untuk mengumpulkan informasi pribadi. Sertakan field input untuk:
 - Nama Depan (text, required)
 - Nama Belakang (text, required)
 - Email (email, required)
 - Password (password, required, minlength="8")
 - Kota (text, opsional, maxlength="50")
 - Provinsi (text, opsional, size="2")
2. **Formulir Pendaftaran Newsletter:** Rancang formulir pendaftaran newsletter dengan field berikut:
 - Alamat Email (email, required, placeholder="Masukkan alamat email Anda")

3. Peningkatan Formulir Pengiriman Resep: Tambahkan field input berikut ke contoh formulir pengiriman resep kita yang sedang berlangsung:

- Nama Anda (text, required)
- Konfirmasi Password (password, required, harus cocok dengan field "Password" - Anda akan memerlukan Javascript untuk mengimplementasikan pencocokan ini di modul selanjutnya)
- Website (text, opsional, placeholder="URL website Anda")

Mempersiapkan Pelajaran Masa Depan

Dalam pelajaran mendatang, kita akan mengeksplorasi tipe input lain, seperti area teks (untuk input teks yang lebih panjang), tombol, dan kotak select. Kita juga akan mendalami atribut elemen `<form>`, yang mengontrol bagaimana data formulir dikirim. Selain itu, kita akan melihat cara menggunakan CSS untuk memberi gaya pada elemen formulir ini dan Javascript untuk memvalidasi input di sisi klien sebelum pengiriman.

Area Teks: Tag `<textarea>`

Tag `<textarea>` adalah bagian fundamental dari formulir HTML, memungkinkan pengguna memasukkan teks multi-baris. Tidak seperti field input teks sederhana (`<input type="text">`), tag `<textarea>` menyediakan ruang yang lebih besar yang cocok untuk pesan yang lebih panjang, komentar, deskripsi, atau jenis teks apa pun yang melebihi satu baris. Memahami cara mengimplementasikan dan menyesuaikan elemen `<textarea>` sangat penting untuk membuat formulir web yang interaktif dan ramah pengguna.

Memahami Tag `<textarea>`

Tag `<textarea>` mendefinisikan kontrol input teks multi-baris. Pengguna dapat memasukkan jumlah teks yang tidak terbatas dalam `<textarea>`. Teks di dalam elemen `<textarea>` adalah yang akan dikirim ke server ketika formulir dikirimkan.

Sintaks Dasar

Sintaks dasar dari tag `<textarea>` adalah sebagai berikut:

```
html
<textarea name="name" rows="number" cols="number">
    Konten teks awal
</textarea>
```

- **<textarea>**: Tag pembuka yang memulai elemen area teks.
- **</textarea>**: Tag penutup yang mengakhiri elemen area teks.
- **atribut name**: Menentukan nama area teks, yang digunakan untuk mengidentifikasi data ketika formulir dikirim.
- **atribut rows**: Menentukan tinggi yang terlihat dari area teks dalam hal jumlah baris teks.
- **atribut cols**: Menentukan lebar yang terlihat dari area teks dalam hal lebar karakter rata-rata.
- **Konten teks awal**: Teks yang muncul di dalam area teks ketika halaman dimuat. Konten ini dapat dimodifikasi atau dihapus oleh pengguna.

Atribut Kunci

Beberapa atribut dapat digunakan untuk menyesuaikan perilaku dan tampilan elemen **<textarea>**.

name: Seperti yang disebutkan sebelumnya, ini sangat penting untuk mengidentifikasi data area teks ketika formulir dikirim. Tanpa nama, data **<textarea>** tidak akan disertakan dalam data formulir yang dikirim.

rows: Menentukan jumlah baris teks yang terlihat untuk area teks. Jika konten melebihi area yang terlihat, scrollbar akan muncul.

cols: Menentukan lebar yang terlihat dari area teks dalam lebar karakter rata-rata. Mirip dengan rows, konten yang melebihi lebar ini akan memicu scrollbar.

placeholder: Memberikan petunjuk kepada pengguna tentang jenis informasi apa yang harus dimasukkan dalam area teks. Teks placeholder menghilang ketika pengguna mulai mengetik.

required: Menentukan bahwa area teks harus diisi sebelum formulir dapat dikirim.

readonly: Menentukan bahwa area teks hanya baca, artinya pengguna tidak dapat memodifikasi kontennya.

disabled: Menentukan bahwa area teks dinonaktifkan, artinya tidak dapat digunakan atau dimodifikasi, dan nilainya tidak akan dikirim dengan formulir.

maxlength: Menentukan jumlah maksimum karakter yang diizinkan dalam area teks.

wrap: Mengontrol bagaimana teks dalam area teks dibungkus ketika dikirim. Ini dapat memiliki nilai seperti soft (teks dibungkus tetapi pemisah baris tidak termasuk dalam data yang dikirim) atau hard (teks dibungkus, dan pemisah baris termasuk dalam data yang dikirim). Nilai hard memerlukan atribut cols untuk ditentukan. Disarankan untuk menghindari atribut wrap dan menangani pembungkusan teks melalui CSS.

Contoh dengan Atribut

html

```
<textarea name="message" rows="10" cols="30" placeholder="Masukkan pesan Anda di sini"
required maxlength="500">
Ini adalah beberapa teks awal.
</textarea>
```

Dalam contoh ini:

- Area teks diberi nama "message".
- Memiliki 10 baris yang terlihat dan 30 kolom yang terlihat.
- Teks placeholder "Masukkan pesan Anda di sini" ditampilkan.
- Field diperlukan, artinya formulir tidak dapat dikirim tanpa mengisinya.
- Panjang maksimum yang diizinkan adalah 500 karakter.
- Area teks awalnya berisi teks "Ini adalah beberapa teks awal."

Contoh Praktis dan Demonstrasi

Mari kita jelajahi beberapa contoh praktis untuk mendemonstrasikan penggunaan tag `<textarea>` dalam berbagai skenario.

Area Teks Dasar

Contoh ini membuat area teks sederhana untuk mengumpulkan komentar pengguna.

html

```
<form>
  <label for="comment">Masukkan komentar Anda:</label><br>
  <textarea id="comment" name="comment" rows="4" cols="50"></textarea><br><br>
  <input type="submit" value="Kirim">
</form>
```

Kode ini membuat formulir dengan label dan area teks. Area teks memiliki id "comment" yang sesuai dengan atribut for label, yang membuat tautan antara label dan elemen input. Atribut rows diatur ke 4, dan atribut cols diatur ke 50, mendefinisikan ukuran awal area teks.

Area Teks dengan Placeholder dan Atribut Required

Contoh ini menunjukkan cara menggunakan atribut placeholder dan required untuk memberikan panduan kepada pengguna dan memastikan bahwa field diisi.

```
html
<form>
  <label for="description">Deskripsi:</label><br>
  <textarea id="description" name="description" rows="5" cols="40" placeholder="Masukkan
deskripsi singkat" required></textarea><br><br>
  <input type="submit" value="Kirim">
</form>
```

Atribut placeholder menampilkan teks "Masukkan deskripsi singkat" di dalam area teks sampai pengguna mulai mengetik. Atribut required memastikan bahwa pengguna harus memasukkan beberapa teks sebelum mengirim formulir. Jika pengguna mencoba mengirim formulir tanpa mengisi area teks, browser akan menampilkan pesan error.

Area Teks dengan Batas Karakter

Contoh ini menunjukkan cara membatasi jumlah karakter yang dapat dimasukkan ke dalam area teks menggunakan atribut maxlength.

```
html
<form>
  <label for="feedback">Umpulan Balik:</label><br>
```

```
<textarea id="feedback" name="feedback" rows="3" cols="60"  
maxlength="200"></textarea><br><br>  
<input type="submit" value="Kirim">  
</form>
```

Dalam kasus ini, atribut maxlength diatur ke 200, yang berarti pengguna dapat memasukkan maksimum 200 karakter dalam area teks. Browser biasanya mencegah pengguna memasukkan lebih banyak karakter dari yang ditentukan, dan beberapa memberikan indikasi visual dari karakter yang tersisa.

Area Teks Read-Only

Contoh ini menunjukkan cara membuat area teks hanya baca menggunakan atribut readonly.

html

```
<form>  
  <label for="terms">Syarat dan Ketentuan:</label><br>  
  <textarea id="terms" name="terms" rows="10" cols="50" readonly>  
    Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do eiusmod tempor incididunt  
    ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation  
    ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in  
    reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint  
    occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est  
    laborum.  
  </textarea><br><br>  
  <input type="submit" value="Kirim">  
</form>
```

Atribut readonly mencegah pengguna memodifikasi konten area teks. Ini dapat berguna untuk menampilkan syarat dan ketentuan, penafian hukum, atau jenis teks statis lainnya yang tidak boleh dapat diedit.

Area Teks yang Dinonaktifkan

Contoh ini menunjukkan cara menonaktifkan area teks menggunakan atribut disabled.

html

```
<form>
  <label for="notes">Catatan:</label><br>
    <textarea id="notes" name="notes" rows="3" cols="50" disabled>Area teks ini
dinonaktifkan.</textarea><br><br>
  <input type="submit" value="Kirim">
</form>
```

Atribut `disabled` mencegah pengguna berinteraksi dengan area teks. Area teks yang dinonaktifkan biasanya muncul abu-abu, dan nilainya tidak dikirim dengan formulir. Ini dapat berguna untuk sementara menonaktifkan field atau untuk menunjukkan bahwa field saat ini tidak berlaku.

Latihan dan Aktivitas Praktik

- Formulir Umpan Balik:** Buat formulir umpan balik dengan area teks bagi pengguna untuk mengirimkan komentar mereka. Sertakan atribut `placeholder` untuk membimbing pengguna dan atribut `maxlength` untuk membatasi jumlah karakter. Buat field diperlukan.
- Input Alamat:** Rancang formulir yang menyertakan area teks bagi pengguna untuk memasukkan alamat lengkap mereka. Atur atribut `rows` dan `cols` yang sesuai untuk tampilan optimal.
- Perjanjian Read-Only:** Buat formulir yang menampilkan perjanjian hanya baca menggunakan elemen `<textarea>`. Sertakan checkbox yang harus dicentang pengguna untuk menyetujui persyaratan sebelum mengirim formulir (kita akan belajar tentang checkbox dalam pelajaran mendatang).
- Deskripsi Profil:** Bangun formulir profil pengguna sederhana yang menyertakan area teks bagi pengguna untuk menulis bio singkat. Implementasikan penghitung karakter menggunakan JavaScript (ini di luar cakupan pelajaran ini, tetapi teliti cara melakukannya!).
- Modifikasi Halaman Resep:** Mengacu kembali pada halaman resep yang Anda buat di Modul 2, tambahkan `<textarea>` untuk memungkinkan pengguna mengirimkan modifikasi atau instruksi alternatif untuk resep. Sertakan label dan atribut yang sesuai untuk kemudahan penggunaan.

Ringkasan dan Langkah Selanjutnya

Dalam pelajaran ini, kita mengeksplorasi tag `<textarea>`, elemen penting untuk mengumpulkan input teks multi-baris dalam formulir HTML. Kita belajar tentang sintaksnya, atribut kunci, dan aplikasi praktis melalui berbagai contoh. Anda sekarang memahami cara membuat dan menyesuaikan area teks untuk mengumpulkan komentar pengguna, deskripsi, umpan balik, dan jenis data tekstual lainnya.

Di pelajaran berikutnya, kita akan mendalami tag `<button>`, komponen penting lain dari formulir HTML, yang memungkinkan pengguna memicu tindakan dan mengirimkan data. Memahami tombol sangat penting untuk membuat formulir web yang sepenuhnya fungsional dan interaktif.

Baik, saya akan merapikan dan menerjemahkan dokumen ketiga puluh dua ini ke Bahasa Indonesia:

Tombol: Tag `<button>`

Dalam ranah formulir HTML, tombol berfungsi sebagai elemen interaktif penting yang memicu tindakan dan memungkinkan pengiriman pengguna. Meskipun tampaknya sederhana, tag `<button>` menawarkan berbagai fungsionalitas dan opsi kustomisasi yang melampaui pengiriman formulir dasar. Pelajaran ini akan mengeksplorasi seluk-beluk tag `<button>`, atributnya, berbagai tipe tombol, dan cara mengintegrasikannya secara efektif ke dalam formulir HTML Anda. Menguasai tag `<button>` sangat penting untuk membuat pengalaman web yang ramah pengguna dan interaktif.

Memahami Tag `<button>`

Tag `<button>` dalam HTML mendefinisikan elemen tombol yang dapat diklik, yang dapat digunakan dalam formulir untuk mengirim data atau di mana saja dalam dokumen HTML untuk tujuan aksesibilitas (dengan JavaScript, yang akan kita bahas di kursus selanjutnya). Tidak seperti tag `<input>` dengan atribut `type` yang diatur ke "button", "submit", atau "reset", tag `<button>` menawarkan lebih banyak fleksibilitas dalam hal konten dan styling.

Sintaks Dasar:

html

```
<button type="button">Klik Saya!</button>
```

Atribut Kunci dari Tag <button>

Tag `<button>` mendukung beberapa atribut yang mengontrol perilaku dan tampilannya:

type: Menentukan jenis tombol. Nilai umum termasuk:

- **button**: Tombol generik tanpa perilaku default. Sering digunakan dengan JavaScript untuk memicu tindakan kustom.
- **submit**: Mengirimkan data formulir ke server. Ini adalah default jika atribut type tidak ditentukan.
- **reset**: Mereset semua kontrol formulir ke nilai awal mereka.

name: Menentukan nama untuk tombol. Ini digunakan untuk mengidentifikasi tombol ketika formulir dikirim.

value: Menentukan nilai yang terkait dengan tombol. Nilai ini dikirim ke server ketika formulir dikirim.

disabled: Menonaktifkan tombol, mencegahnya diklik.

form: Menentukan satu atau lebih formulir yang dimiliki tombol.

formaction: Menentukan ke mana mengirim data formulir ketika formulir dikirim. Menggantikan atribut action formulir.

formenctype: Menentukan bagaimana data formulir harus di-encode sebelum mengirimkannya ke server. Hanya untuk type="submit". Menggantikan atribut enctype formulir.

formmethod: Mendefinisikan metode HTTP yang digunakan ketika mengirimkan data formulir. Hanya untuk type="submit". Menggantikan atribut method formulir.

formnovalidate: Menentukan bahwa data formulir tidak boleh divalidasi saat pengiriman. Hanya untuk type="submit". Menggantikan atribut novalidate formulir.

formtarget: Menentukan nama atau kata kunci yang menunjukkan di mana menampilkan respons yang diterima setelah mengirimkan formulir. Hanya untuk type="submit". Menggantikan atribut target formulir.

Contoh:

```
html
<form action="/submit" method="post">
    <button type="submit" name="submit_button" value="submit_value">Kirim
Formulir</button>
    <button type="reset">Reset Formulir</button>
    <button type="button" onclick="alert('Tombol Diklik!')">Klik Saya</button>
</form>
```

Dalam contoh di atas, tombol pertama mengirimkan formulir, yang kedua mereset formulir, dan yang ketiga mengeksekusi alert JavaScript ketika diklik. Atribut name dan value dari tombol submit akan dikirim ke server bersama dengan data formulir lainnya ketika formulir dikirim. (Kita akan membahas JavaScript di modul mendatang.)

Tipe Tombol: submit, reset, dan button

Memahami tipe tombol yang berbeda sangat penting untuk membuat formulir yang fungsional.

Tombol submit

Tombol submit, ditandai dengan `type="submit"`, adalah tipe yang paling umum. Ketika diklik, ini memicu proses pengiriman formulir. Data formulir kemudian dikirim ke URL yang ditentukan dalam atribut action formulir, menggunakan metode yang ditentukan dalam atribut method (baik GET atau POST, seperti yang diperkenalkan dalam pelajaran sebelumnya).

Contoh:

```
html
<form action="/process_form" method="post">
    <label for="name">Nama:</label><br>
```

```
<input type="text" id="name" name="name"><br><br>
<button type="submit">Kirim</button>

</form>
```

Mengklik tombol "Kirim" dalam contoh ini akan mengirim data formulir (nilai yang dimasukkan dalam field input "name") ke URL `/process_form` menggunakan metode POST. Jika atribut `type` tidak ditentukan, tombol default ke `type` "submit".

Tombol reset

Tombol reset, ditandai dengan `type="reset"`, mereset semua elemen formulir ke nilai default mereka. Ini dapat berguna untuk menghapus data formulir dan memulai dari awal.

Contoh:

```
html
<form action="/process_form" method="post">
  <label for="email">Email:</label><br>
  <input type="email" id="email" name="email" value="example@email.com"><br><br>
  <button type="reset">Reset</button>

</form>
```

Mengklik tombol "Reset" akan menghapus nilai yang dimasukkan dalam field `email` dan mengaturnya kembali ke nilai default, "example@email.com".

Tombol button

Tombol button, ditandai dengan `type="button"`, tidak memiliki perilaku default. Biasanya digunakan dengan JavaScript untuk melakukan tindakan kustom, seperti menampilkan jendela modal, membuat permintaan AJAX, atau memanipulasi DOM. (Kita akan membahas JavaScript dan AJAX di modul mendatang.)

Contoh:

```
html
<button type="button" onclick="alert('Halo, dunia!')>Ucapkan Halo</button>
```

Mengklik tombol "Ucapkan Halo" akan memicu kotak alert JavaScript yang menampilkan pesan "Halo, dunia!".

Atribut Tombol Lanjutan dan Kasus Penggunaan

Selain atribut dasar, beberapa atribut lain menawarkan lebih banyak kontrol atas perilaku tombol, terutama ketika digunakan dalam formulir.

Atribut form

Atribut form memungkinkan elemen `<button>` untuk dikaitkan dengan elemen `<form>`, bahkan jika `<button>` tidak bersarang dalam `<form>`. Ini dapat berguna untuk layout kompleks di mana tombol perlu ditempatkan di luar struktur formulir.

Contoh:

```
html
<form id="myForm" action="/submit" method="post">
  <label for="username">Nama Pengguna:</label>
  <input type="text" id="username" name="username"><br><br>
</form>

<button type="submit" form="myForm">Kirim Formulir</button>
```

Dalam contoh ini, tombol berada di luar elemen `<form>`, tetapi atribut `form="myForm"` mengaitkannya dengan formulir yang memiliki ID "myForm". Mengklik tombol akan mengirimkan data dari formulir yang terkait.

Mengganti Atribut Formulir: `formaction`, `formenctype`, `formmethod`, `formnovalidate`, dan `formtarget`

Atribut `formaction`, `formenctype`, `formmethod`, `formnovalidate`, dan `formtarget` memberikan cara untuk mengganti atribut yang sesuai dari elemen `<form>` untuk tombol tertentu. Ini memungkinkan Anda memiliki beberapa tombol submit dalam formulir yang sama yang melakukan tindakan berbeda atau menggunakan metode pengiriman berbeda.

formaction: Menentukan URL yang berbeda untuk mengirimkan data formulir.

formenctype: Menentukan jenis encoding yang berbeda untuk data formulir. Nilai umum termasuk application/x-www-form-urlencoded (default), multipart/form-data (digunakan untuk upload file), dan text/plain.

formmethod: Menentukan metode HTTP yang berbeda (GET atau POST) untuk digunakan ketika mengirimkan data formulir.

formnovalidate: Menentukan bahwa data formulir tidak boleh divalidasi ketika tombol ini diklik. Ini berguna jika Anda ingin memungkinkan pengguna menyimpan draft formulir tanpa mengharuskan semua field diisi. (Kita akan membahas validasi formulir di pelajaran berikutnya.)

formtarget: Menentukan konteks browsing yang berbeda (misalnya, tab atau jendela baru) untuk menampilkan hasil pengiriman formulir.

Contoh:

html

```
<form action="/default_submit" method="post">
  <label for="comment">Komentar:</label><br>
  <textarea id="comment" name="comment"></textarea><br><br>
  <button type="submit">Kirim</button>
    <button type="submit" formaction="/save_draft">
      Simpan Draft</button>
</form>
```

Dalam contoh ini, tombol "Kirim" pertama mengirimkan formulir ke `/default_submit` dan memvalidasi data formulir. Tombol "Simpan Draft" kedua mengirimkan formulir ke `/save_draft` dan melewati validasi formulir.

Styling Tombol dengan CSS

Meskipun kursus ini fokus pada HTML, penting untuk dicatat bahwa tombol dapat diberi gaya secara ekstensif menggunakan CSS. Anda dapat mengubah tampilannya, termasuk warna, font, border, dan spacing mereka. Selektor CSS dapat menargetkan elemen `<button>` secara langsung atau menggunakan atribut name atau id untuk styling yang lebih spesifik. (Kita akan membahas CSS di kursus mendatang).

Contoh:

```
html
<style>
button {
    background-color: #4CAF50; /* Hijau */
    border: none;
    color: white;
    padding: 15px 32px;
    text-align: center;
    text-decoration: none;
    display: inline-block;
    font-size: 16px;
    cursor: pointer;
}
button:hover {
    background-color: #3e8e41;
}
</style>

<button type="button">Tombol Bergaya</button>
```

Contoh ini menunjukkan styling CSS dasar untuk tombol, termasuk warna latar belakang, warna teks, padding, dan ukuran font. Pseudo-class :hover mengubah warna latar belakang ketika kursor mouse melayang di atas tombol.

Pertimbangan Aksesibilitas untuk Tombol

Ketika membuat tombol, sangat penting untuk mempertimbangkan aksesibilitas untuk memastikan bahwa pengguna dengan disabilitas dapat berinteraksi dengan mereka secara efektif.

Gunakan HTML semantik: Gunakan elemen `<button>` untuk tombol, daripada menggunakan elemen `<div>` atau `` dan memberi gaya agar terlihat seperti tombol. Ini memberikan fitur aksesibilitas built-in.

Berikan teks yang jelas dan deskriptif: Teks di dalam tombol harus dengan jelas menunjukkan tujuannya. Hindari istilah yang samar seperti "Klik Di Sini."

Gunakan atribut ARIA bila perlu: Atribut ARIA (Accessible Rich Internet Applications) dapat digunakan untuk memberikan informasi tambahan kepada teknologi asistif, seperti pembaca layar. Misalnya, Anda dapat menggunakan atribut aria-label untuk memberikan deskripsi yang lebih detail tentang tujuan tombol. (Kita tidak akan membahas atribut ARIA dalam kursus ini tetapi itu adalah sesuatu untuk diteliti sendiri.)

Pastikan aksesibilitas keyboard: Pastikan bahwa tombol dapat diakses dan diaktifkan menggunakan keyboard. Elemen `<button>` dapat difokuskan secara default, tetapi Anda mungkin perlu menambahkan atribut tabindex untuk memastikan bahwa tombol berada dalam urutan yang logis.

Berikan petunjuk visual: Gunakan CSS untuk memberikan petunjuk visual yang menunjukkan status tombol (misalnya, fokus, hover, aktif).

Contoh:

html

```
<button type="button" aria-label="Kirim formulir">Kirim</button>
```

Dalam contoh ini, atribut aria-label memberikan label yang lebih deskriptif untuk tombol, yang dapat membantu untuk pengguna dengan pembaca layar.

Aktivitas Praktik

- Buat formulir dengan tipe tombol yang berbeda:** Buat formulir HTML dengan field input untuk nama, email, dan pesan. Sertakan tombol submit, tombol reset, dan tombol kustom yang menampilkan pesan alert ketika diklik.
- Gunakan atribut form:** Buat dua formulir terpisah, masing-masing dengan serangkaian field input yang berbeda. Buat tombol di luar kedua formulir yang mengirimkan data dari formulir pertama menggunakan atribut form.
- Ganti atribut formulir:** Buat formulir dengan URL dan metode pengiriman default. Tambahkan dua tombol submit: satu yang mengirimkan formulir ke URL dan metode default, dan yang lain yang mengganti atribut formaction dan formmethod untuk mengirimkan formulir ke URL yang berbeda menggunakan metode yang berbeda.
- Beri gaya tombol dengan CSS:** Buat serangkaian tombol dengan gaya yang berbeda, menggunakan CSS untuk mengontrol tampilannya (misalnya, warna, font, border).

Ringkasan

Tag `<button>` adalah elemen fundamental untuk membuat formulir interaktif dan antarmuka pengguna dalam HTML. Dengan memahami atributnya, tipe, dan pertimbangan aksesibilitas, Anda dapat membuat tombol yang fungsional dan ramah pengguna. Ingatlah untuk menggunakan tipe tombol yang sesuai untuk perilaku yang diinginkan, dan pertimbangkan aksesibilitas untuk memastikan bahwa tombol Anda dapat digunakan oleh semua orang. Saat Anda terus belajar HTML, CSS, dan JavaScript, Anda akan dapat membuat aplikasi web yang semakin kompleks dan interaktif dengan fungsionalitas tombol yang canggih.

Sempurna! Saya akan merapikan dan menerjemahkan dokumen ketiga puluh tiga ini ke Bahasa Indonesia:

Kotak Select: Tag `<select>` dan `<option>`

Dalam ranah formulir HTML, kotak select menyediakan cara yang ramah pengguna untuk menyajikan daftar opsi dari mana pengguna dapat memilih satu atau lebih. Tag `<select>` dan `<option>` adalah fundamental untuk membuat elemen interaktif ini. Pelajaran ini mendalami seluk-beluk tag-tag ini, mengeksplorasi atribut, penggunaan, dan praktik terbaik mereka. Memahami kotak select sangat penting untuk membangun formulir yang menawarkan input pengguna yang terkontrol dan dapat diprediksi, yang penting untuk akurasi data dan pengalaman pengguna yang positif.

Tag `<select>`: Membuat Wadah

Tag `<select>` mendefinisikan daftar drop-down, juga dikenal sebagai kotak select. Ini bertindak sebagai wadah untuk tag `<option>`, yang mewakili pilihan individual dalam daftar.

Sintaks Dasar

Sintaks dasar dari tag `<select>` adalah sederhana:

```
html
<select>
</select>
```

Atribut Kunci dari `<select>`

Tag `<select>` mendukung beberapa atribut yang mengontrol perilaku dan tampilannya. Berikut adalah beberapa yang paling penting:

name: Atribut ini menentukan nama kotak select, yang digunakan ketika mengirimkan data formulir ke server. Ini penting untuk mengidentifikasi nilai yang dipilih.

id: Atribut ini memberikan identifier unik untuk kotak select dalam dokumen HTML. Ini terutama digunakan untuk styling dengan CSS atau manipulasi dengan JavaScript.

multiple: Ketika ada, atribut ini memungkinkan pengguna untuk memilih beberapa opsi dari daftar. Browser biasanya akan menampilkan kotak daftar alih-alih menu drop-down dalam kasus ini.

size: Atribut ini menentukan jumlah opsi yang terlihat dalam kotak select berganda. Jika jumlah opsi melebihi ukuran, scrollbar akan ditampilkan.

required: Ketika ada, atribut ini menunjukkan bahwa pengguna harus memilih opsi sebelum mengirimkan formulir.

disabled: Ketika ada, atribut ini menonaktifkan kotak select, mencegah pengguna berinteraksi dengannya.

Contoh: Kotak Select Sederhana

Berikut adalah contoh dasar yang mendemonstrasikan penggunaan tag `<select>` dengan atribut name dan id:

```
html
<label for="fruit">Pilih buah:</label>
<select name="fruit" id="fruit">
  <option value="apple">Apel</option>
```

```
<option value="banana">Pisang</option>
<option value="orange">Jeruk</option>
</select>
```

Dalam contoh ini:

- Elemen `<select>` dibuat dengan atribut name diatur ke "fruit" dan atribut id diatur ke "fruit".
- Label terkait dengan elemen select dengan `for="fruit"`.
- Tiga elemen `<option>` disertakan dalam elemen `<select>`, mewakili pilihan "Apel", "Pisang", dan "Jeruk".
- Setiap `<option>` memiliki atribut value, yang menentukan nilai yang akan dikirimkan ke server ketika opsi tersebut dipilih.

Tag `<option>`: Mendefinisikan Pilihan

Tag `<option>` mewakili pilihan individual dalam kotak `<select>`. Setiap tag `<option>` harus memiliki atribut value, yang menentukan nilai yang akan dikirim ke server ketika formulir dikirim.

Sintaks Dasar

Sintaks dasar dari tag `<option>` adalah sebagai berikut:

html

```
<option value="value">Teks</option>
```

Di sini, "value" adalah nilai yang dikirim ke server, dan "Teks" adalah teks yang ditampilkan kepada pengguna dalam kotak select.

Atribut Kunci dari `<option>`

Tag `<option>` memiliki beberapa atribut yang mengontrol perilakunya:

value: Seperti yang disebutkan sebelumnya, atribut ini menentukan nilai yang akan dikirimkan ke server ketika opsi dipilih. Jika atribut value tidak ditentukan, konten teks

dari tag `<option>` digunakan sebagai nilai. Namun, secara eksplisit mengatur atribut value adalah praktik terbaik.

selected: Ketika ada, atribut ini menunjukkan bahwa opsi harus dipilih sebelumnya ketika halaman dimuat. Hanya satu `<option>` dalam elemen `<select>` yang harus memiliki atribut `selected` kecuali atribut `multiple` digunakan pada tag `<select>`.

disabled: Ketika ada, atribut ini menonaktifkan opsi, mencegah pengguna memilihnya.

Contoh: Opsi dengan Atribut value dan selected

html

```
<label for="car">Pilih mobil:</label>
<select name="car" id="car">
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="mercedes" selected>Mercedes</option>
  <option value="audi">Audi</option>
</select>
```

Dalam contoh ini:

- Opsi "Mercedes" dipilih sebelumnya karena memiliki atribut `selected`.
- Setiap opsi memiliki atribut `value` yang berbeda yang sesuai dengan merek mobil.

Contoh: Opsi yang Dinonaktifkan

html

```
<label for="country">Pilih negara:</label>
<select name="country" id="country">
  <option value="" disabled selected>Pilih negara</option>
  <option value="usa">USA</option>
  <option value="canada">Kanada</option>
  <option value="uk">Inggris</option>
</select>
```

Dalam contoh ini:

- Opsi pertama "Pilih negara" dinonaktifkan dan dipilih sebelumnya. Ini adalah praktik umum untuk memberikan prompt yang jelas kepada pengguna tanpa memungkinkan mereka untuk memilih prompt itu sendiri.

Bekerja dengan Select Berganda

Atribut multiple, ketika ditambahkan ke tag `<select>`, mengubah kotak select menjadi kotak daftar yang memungkinkan pengguna memilih beberapa opsi secara bersamaan.

Contoh: Kotak Select Berganda

html

```
<label for="cars">Pilih mobil Anda:</label>
<select name="cars" id="cars" multiple size="4">
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="mercedes">Mercedes</option>
  <option value="audi">Audi</option>
</select>
```

Dalam contoh ini:

- Atribut multiple ada, memungkinkan pengguna untuk memilih beberapa merek mobil.
- Atribut size diatur ke "4", yang berarti empat opsi akan terlihat sekaligus. Jika ada lebih dari empat opsi, scrollbar akan muncul.

Ketika formulir dengan kotak select berganda dikirim, server menerima array nilai yang sesuai dengan opsi yang dipilih. Atribut name dari tag `<select>` menjadi nama array.

Mengelompokkan Opsi dengan `<optgroup>`

Tag `<optgroup>` digunakan untuk mengelompokkan opsi terkait dalam kotak `<select>`. Ini meningkatkan organisasi dan keterbacaan daftar opsi yang panjang.

Sintaks Dasar

Sintaks dasar dari tag `<optgroup>` adalah sebagai berikut:

```
html
<optgroup label="Nama Grup">
  <option value="value1">Teks1</option>
  <option value="value2">Teks2</option>
</optgroup>
```

Atribut label menentukan nama grup opsi, yang ditampilkan kepada pengguna.

Contoh: Mengelompokkan Merek Mobil berdasarkan Asal

```
html
<label for="cars">Pilih mobil Anda:</label>
<select name="cars" id="cars">
  <optgroup label="Mobil Swedia">
    <option value="volvo">Volvo</option>
    <option value="saab">Saab</option>
  </optgroup>
  <optgroup label="Mobil Jerman">
    <option value="mercedes">Mercedes</option>
    <option value="audi">Audi</option>
    <option value="bmw">BMW</option>
  </optgroup>
</select>
```

Dalam contoh ini:

- Opsi dikelompokkan menjadi "Mobil Swedia" dan "Mobil Jerman" menggunakan tag `<optgroup>`.
- Atribut label dari setiap tag `<optgroup>` menentukan nama grup.

Pertimbangan Aksesibilitas

Ketika membuat kotak select, penting untuk mempertimbangkan aksesibilitas untuk memastikan bahwa pengguna dengan disabilitas dapat menggunakan formulir secara efektif.

Gunakan Label: Selalu kaitkan elemen `<label>` dengan setiap elemen `<select>` menggunakan atribut `for`. Ini memberikan deskripsi yang jelas dari kotak select untuk pengguna pembaca layar.

Berikan Instruksi yang Jelas: Berikan instruksi yang jelas dan ringkas tentang cara menggunakan kotak select.

Gunakan Atribut ARIA: Atribut ARIA (Accessible Rich Internet Applications) dapat digunakan untuk memberikan informasi tambahan kepada pembaca layar. Misalnya, `aria-describedby` dapat digunakan untuk mengaitkan deskripsi dengan kotak select.

Latihan

- Buat kotak select untuk memilih benua:** Buat kotak select dengan atribut `name` diatur ke "continent" dan atribut `id` diatur ke "continent". Sertakan opsi untuk "Afrika", "Amerika", "Asia", "Eropa", dan "Oseania". Buat "Eropa" opsi yang dipilih sebelumnya.
- Buat kotak select berganda untuk memilih bahasa pemrograman:** Buat kotak select berganda dengan atribut `name` diatur ke "languages" dan atribut `id` diatur ke "languages". Sertakan opsi untuk "JavaScript", "Python", "Java", "C++", dan "C#". Atur atribut `size` ke "3".
- Buat kotak select untuk memilih metode transportasi, dikelompokkan berdasarkan kategori:** Buat kotak select dengan atribut `name` diatur ke "transport" dan atribut `id` diatur ke "transport". Kelompokkan opsi menjadi "Darat", "Udara", dan "Air". Sertakan opsi seperti "Mobil", "Kereta", "Pesawat", "Helikopter", "Perahu", dan "Kapal Selam".
- Buat kotak select "required" untuk memilih metode pembayaran:** Buat kotak select dengan atribut `name` diatur ke "payment" dan atribut `id` diatur ke "payment". Sertakan opsi "Kartu Kredit", "PayPal", dan "Transfer Bank". Buat field diperlukan. Sertakan opsi default yang dinonaktifkan yang bertuliskan "Pilih Metode Pembayaran".

Ringkasan

Tag `<select>` dan `<option>` adalah alat penting untuk membuat formulir interaktif dalam HTML. Mereka menyediakan cara yang ramah pengguna untuk menyajikan daftar opsi dari mana pengguna dapat memilih. Dengan memahami atribut dan praktik terbaik yang terkait

dengan tag-tag ini, Anda dapat membuat formulir yang fungsional dan dapat diakses. Di pelajaran berikutnya, kita akan mengeksplorasi tag `<textarea>` yang akan memungkinkan kita membuat area input teks yang lebih besar.