**MODUL III : JavaScript**

**KODE MODUL** : TIK.RPL02.002.001.01

**DESKRIPSI MODUL** : Validasi Form dengan JavaScript

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Elemen Kompetensi | Indikator Kinerja | Jml Jam | halaman |
| 1 | Menerapkan validasi form dengan JavaScript | Form dapat divalidasi untuk memastikan tidak ada kolom yang kosong sebelum dikirim. | 2 |  |
| 2 | Menyediakan feedback visual untuk validasi | Elemen input yang tidak valid akan ditandai dengan warna merah dan pesan kesalahan akan ditampilkan. | 1 |  |

**PRELAB**

Persiapan Modul Praktikum Validasi Form dengan JavaScript:

1. JavaScript (<https://www.w3schools.com/js/>) untuk memahami dasar-dasar JavaScript dan DOM.
2. Validasi Form (<https://www.w3schools.com/js/js_validation.asp>) untuk mempelajari bagaimana melakukan validasi pada form.

**DAFTAR PERTANYAAN**

1. Apa tujuan dari validasi form pada sisi klien menggunakan JavaScript?
2. Apa yang dilakukan oleh trim() dalam kode validasi form, dan mengapa itu penting?
3. Bagaimana script tersebut mencegah form dari disubmit jika ada kolom yang kosong?
4. Mengapa ada pengecekan kondisional individu untuk setiap input field dalam kode tersebut?

**Jawab**

1. Tujuan validasi form yaitu untuk memastikan bahwa data yang dimasukkan oleh pengguna ke dalam form sesuai dengan kriteria yang diinginkan sebelum data tersebut dikirim ke server. Ini membantu untuk meningkatkan pengalaman pengguna dengan memberikan umpan balik instan saat pengguna mengisi form, sehingga menghindari pengiriman data yang tidak valid atau tidak lengkap ke server.
2. Fungsi trim() digunakan untuk menghapus spasi putih (seperti spasi, tab, atau baris baru) dari awal dan akhir sebuah string. Dalam konteks validasi form, trim() digunakan untuk membersihkan input dari spasi putih yang tidak diinginkan sebelum validasi dilakukan.
3. Mencegah form kosong saat disubmit:
   1. Saat formulir diajukan (event submit), event defaultnya dicegah menggunakan event.preventDefault();.
   2. Kemudian, nilai dari setiap input field diambil dan disimpan dalam array inputs.
   3. Script kemudian melakukan iterasi melalui setiap input dalam array tersebut.
   4. Jika nilai dari suatu input adalah string kosong setelah dipanggil trim(), maka warna border input tersebut diubah menjadi merah, dan pengguna diberikan pesan kesalahan yang spesifik tergantung pada input yang kosong (nama, email, atau input lainnya).
   5. Jika ada input yang kosong, variabel isValid diatur menjadi false, dan iterasi berhenti.
   6. Jika panjang nilai dari input kata sandi kurang dari 8 karakter, atau jika nilai dari konfirmasi kata sandi tidak sama dengan kata sandi, warna border input tersebut juga diubah menjadi merah, dan pesan kesalahan yang sesuai ditampilkan.
   7. Jika semua kondisi validasi dipenuhi (tidak ada input yang kosong, dan kata sandi serta konfirmasi kata sandi sesuai), maka variabel isValid tetap true.
4. Pengecekan kondisional individu untuk setiap input field dalam kode tersebut diperlukan karena setiap input field memiliki kriteria validasi yang unik. Dengan melakukan pengecekan kondisional untuk setiap input field, kita dapat memastikan validasi sesuai dengan kebutuhan spesifik dari masing-masing input, memberikan pesan kesalahan yang tepat, dan meningkatkan kualitas keseluruhan dari validasi form.

**TEORI SINGKAT**

Validasi form adalah proses verifikasi bahwa informasi yang dimasukkan pengguna ke dalam form adalah sesuai dengan yang diharapkan sebelum dikirim ke server. Validasi ini bisa meliputi pemeriksaan format, seperti email yang valid atau password dengan kriteria tertentu, dan juga memastikan bahwa informasi esensial tidak ditinggalkan kosong.

JavaScript menyediakan cara untuk melakukan validasi ini di sisi klien, yaitu pada browser pengguna, sebelum data dikirim ke server. Melalui penggunaan JavaScript, developer dapat memprogram script yang secara otomatis memeriksa isi dari setiap elemen form seperti input teks, radio buttons, checkboxes, dan lainnya, untuk memastikan data yang diinput sudah benar dan lengkap.

Fungsi trim() dalam JavaScript digunakan untuk menghilangkan whitespace di awal dan akhir string, yang penting untuk memastikan bahwa pengguna tidak hanya memasukkan spasi sebagai input.

Dalam konteks visual, JavaScript dapat digunakan untuk memberikan umpan balik langsung kepada pengguna, seperti mengubah warna border input menjadi merah atau menampilkan pesan kesalahan. Ini membantu pengguna segera mengidentifikasi kesalahan dan memperbaikinya tanpa harus menunggu respon dari server, yang secara signifikan dapat meningkatkan pengalaman pengguna (user experience).

Melakukan validasi di sisi klien tidak hanya meningkatkan responsivitas aplikasi web tetapi juga mengurangi beban pada server karena menghindari prosesing data yang tidak perlu atau format yang salah. Namun, validasi sisi klien tidak boleh digunakan sebagai pengganti validasi sisi server, karena JavaScript dapat dinonaktifkan di browser pengguna atau bisa di-bypass oleh pengguna dengan niat jahat. Sehingga, validasi sisi server juga penting untuk keamanan data.

|  |  |
| --- | --- |
| **Tag** | **Keterangan** |
| <form id="registrationForm"> .... </form> | Tag yang digunakan untuk membuat form registrasi. ID unik memudahkan akses melalui JavaScript untuk validasi. |
| <input type="text" id="name" ...> | Input field untuk nama pengguna. Divalidasi oleh JavaScript untuk memastikan bahwa input tidak dibiarkan kosong. |
| <input type="email" id="email" ...> | Input field untuk alamat email pengguna. JavaScript memvalidasi format email dan memastikan field ini tidak kosong. |
| <input type="password" id="password" ...> | Input field untuk kata sandi. JavaScript memeriksa agar tidak kosong dan memenuhi kriteria keamanan yang ditentukan. |
| <button type="submit"> .... </button> | Tombol submit untuk form. Ketika diklik, JavaScript melakukan validasi form dan mencegah pengiriman jika ada kesalahan. |
| <script> .... </script> | Element script yang berisi JavaScript. Digunakan untuk validasi form pada sisi klien sebelum data dikirim ke server. |

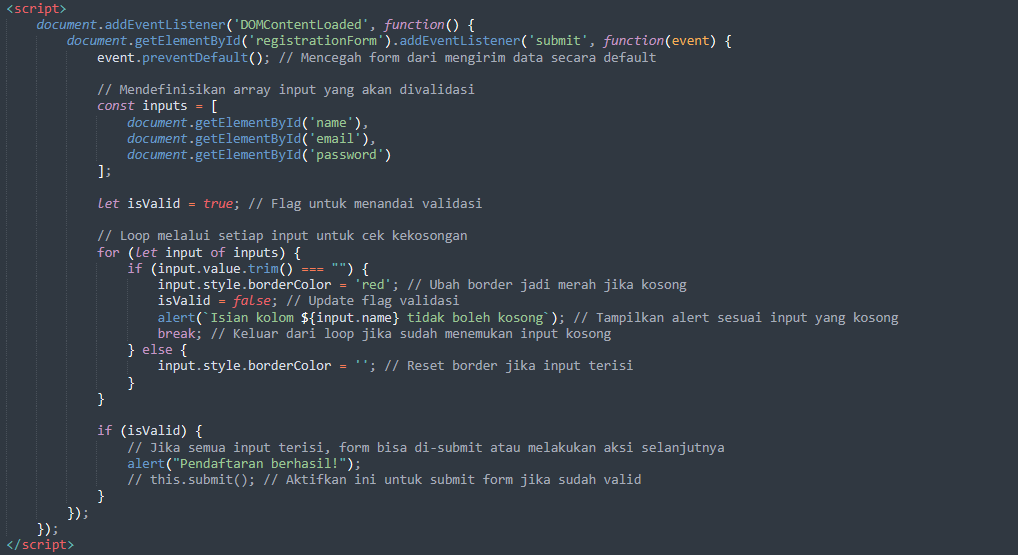
**ELEMEN KOMPETENSI**

**Deskripsi :** Validasi Form dengan JavaScript

**Kompetensi Dasar** : Memahami dan menerapkan teknik validasi form sisi klien menggunakan JavaScript bersamaan dengan HTML dan CSS untuk meningkatkan interaktivitas dan pengalaman pengguna dalam pengisian form registrasi.

**Latihan 1**

* 1. Buka editor web, sublime atau notepad.
  2. Tuliskan JavaScript ini pada code Form yang dibuat kemarin:



* 1. Kemudian tulis form html ini:



* 1. Simpanlah ke dalams \*.html (contoh : web\_NAMA\_ANDA.html) pada sebuah folder.

**Latihan 2**

* 1. Validasi Panjang Karakter Kata Sandi
     1. Tambahkan Validasi untuk memastikan bahwa kata sandi memiliki minimal 8 Karakter
     2. Jika Panjang kata sandi kurang dari 8 karakter, ubah border input menjadi warna merah dan tampilkan pesan dibawah input field yang menginstruksikan pengguna untuk memasukkan lebih banyak karakter
  2. Validasi Konfirmasi Kata Sandi
     1. Tambahkan field input baru untuk konfirmasi kata sandi
     2. Implementasikan logika untuk memvalidasi bahwa kata sandi dan konfirmasi kata sandi sama
     3. Jika kata sandi dan konfirmasi kata sandi tidak cocok, ubah border kedua input field menjadi warna merah dan tampilkan pesan kesalahan

**CODE**

Html

|  |
| --- |
| <html>  <head>      <title>Kelompok 3</title>      <link rel="stylesheet" type="text/css" href="style.css">    </head>  <body>      <form id="registrationForm">          <h2>Daftar Akun Baru</h2>          <label for="name">Nama:</label>          <input type="text" id="name" name="name">          <span class="error"></span>          <label for="email">Email:</label>          <input type="email" id="email" name="email">          <span class="error"></span>          <label for="password">Kata Sandi:</label>          <input type="password" id="password" name="password">          <span class="error"></span>          <label for="confirmPassword">Konfirmasi Kata Sandi:</label>          <input type="password" id="confirmPassword" name="confirmPassword">          <span class="error"></span>          <button type="submit">Daftar</button>      </form>      <script src="script.js"></script>  </body>  </html> |

CSS

|  |
| --- |
| body {      margin: 0;      padding: 0;      display: flex;      justify-content: center;      align-items: center;      min-height: 100vh;      font-family: Arial, sans-serif;      background: linear-gradient(to bottom, #0f0c29, #302b63, #24243e);      background-image: url('background.jpg');  }  form {      background-color: rgba(242, 242, 242, 0.7); /\* Adjusted background color with transparency \*/      padding: 20px;      border-radius: 10px;      box-shadow: 5px 20px 50px rgba(255, 255, 255, 0.5); /\* Adjusted box-shadow \*/      width: 350px;      max-width: 90%;  }  h2 {      color: #fff; /\* Adjusted heading color \*/      text-align: center;  }  label {      display: block;      margin-top: 10px;      color: #fff; /\* Adjusted label color \*/  }  input[type=text], input[type=email], input[type=password] {      width: calc(100% - 20px);      padding: 10px;      margin-top: 5px;      margin-bottom: 20px;      border: 1px solid #ccc;      border-radius: 4px;  }  button {      background-color: #573b8a; /\* Adjusted button color \*/      color: #fff;      padding: 10px 20px;      border: none;      border-radius: 4px;      cursor: pointer;      width: 100%;  }  button:hover {      background-color: #a000c1; /\* Adjusted button hover color \*/  }  /\* Media Queries \*/  @media (max-width: 768px) {      body {          padding: 20px;          display: flex;          justify-content: center;          align-items: center;          flex-direction: column;      }        form {          width: 100%;          margin: 0;          box-shadow: none;      }  } |

JAVASCRIPT

|  |
| --- |
| document.addEventListener('DOMContentLoaded', function() {      document.getElementById('registrationForm').addEventListener('submit', function(event) {          event.preventDefault();          const inputs = [              document.getElementById('name'),              document.getElementById('email'),              document.getElementById('password'),              document.getElementById('confirmPassword')          ];          let isValid = true;          for (let input of inputs) {              if (input.value.trim() === "") {                  input.style.borderColor = 'red';                  if (input.id === 'name') {                      alert('Kolom nama tidak boleh kosong');                  } else if (input.id === 'email') {                      alert('Kolom email tidak boleh kosong');                  } else {                      alert('Isian kolom tidak boleh kosong');                  }              }              if (input.id === 'password' && input.value.trim().length < 8) {                  input.style.borderColor = 'red';                  alert('Kata sandi harus memiliki minimal 8 karakter');                  isValid = false;                  break;              } else {                  input.style.borderColor = '';              }              if (input.id === 'confirmPassword' && input.value !== document.getElementById('password').value) {                  input.style.borderColor = 'red';                  alert('Konfirmasi kata sandi harus sama dengan kata sandi');                  isValid = false;                  break;              } else {                  input.style.borderColor = '';              }          }          if (isValid) {              alert("Pendaftaran berhasil!");          }      });  }); |

**OUTPUT**

|  |
| --- |
|  |

**Github**

|  |
| --- |
|  |

**KESIMPULAN**

**Berikan kesimpulan berdasarkan apa yang sudah di kerjakan pada modul ini minimal 200 kata**

**CEK LIST (√)**

1. Menerapkan validasi pada form input data dengan JavaScript ( **√** )
2. Menangani event submit form untuk memvalidasi data sebelum pengiriman ( **√** )

**FORM UMPAN BALIK**

|  |  |  |  |
| --- | --- | --- | --- |
| **Elemen Kompetensi** | **Tingkat Kesulitan**  **(1: sangat mudah,2: mudah,3: biasa,4: sulit,5: sangat sulit)** | **Tingkat Ketertarikan**  **(1: tidak tertarik, 2: cukup tertarik, 3: tertarik; 4: sangat tertarik)** | **Waktu Penyelesaian**  **Dalam menit** |
| **Menangani event submit form untuk memvalidasi data sebelum pengiriman** | **Biasa** | **Sangat Tertarik** | **10 menit** |
| **Menerapkan validasi pada form input data dengan JavaScript** | **Biasa** | **Sangat Tertarik** | **30 menit** |