

LAPORAN PRAKTIKUM LAB D1
PENGEMBANGAN BERBASIS PLATFORM
“Client Side Server”



DISUSUN OLEH:
RESMA ADI NUGROHO
24060121120021

DEPARTEMEN INFORMATIKA
FAKULTAS SAINS DAN MATEMATIKA
UNIVERSITAS DIPONEGORO
2023

BAB I

PENDAHULUAN

1.1 Tujuan

1. Mahasiswa mampu menerapkan konsep *client-side server* dengan menggunakan javascript
2. Mahasiswa mampu mengimplementasikan basic javascript, *document object model*, dan *browser object model* pada *website*

1.2 Rumusan Masalah

1. Bagaimana cara membuat sebuah *form user input* dengan menerapkan validasi *form* seperti pada gambar di bawah.

User Input

Nama

Gender ☐ Male
☐ Pria

Height

Address

Province

City

Hobby ☐ Travelling
☐ Reading
☐ Swimming
☐ Painting

Captcha

Field	Aturan Validasi
Nama	Harus diisi, minimal 5 karakter, maksimal 30 karakter
Gender	Harus diisi Pilihan: Male Female
Height	Harus diisi, berupa nilai numerik
Address	Harus diisi, minimal 5 karakter, maksimal 100 karakter
Province	Harus diisi Pilihan: West Java Central Java East Java
City	Harus diisi, pilihan sesuai dengan province yang dipilih. Jika city = West Java, pilihan city adalah Bandung Kuningan Indramayu Jika city = Central Java, pilihan city adalah Semarang Ungaran Solo Jika city = East Java, pilihan city adalah Surabaya Sidoarjo Malang Jika province tidak dipilih, maka dropdown city adalah kosong
Hobby	Harus diisi, minimal hobby yang dipilih adalah 1
Captcha	Harus diisi, berisi angka dari 0-9 sepanjang 5 karakter, di-generate secara random saat halaman di-load.

BAB II

DASAR TEORI

2.1 JavaScript

JavaScript, dalam dunia pemrograman web adalah bahasa pemrograman tingkat tinggi yang digunakan untuk mengontrol perilaku interaktif pada situs web. Javascript merupakan salah satu komponen penting dalam perkembangan dunia web modern. JavaScript memungkinkan adanya interaksi yang interaktif antara pengguna dengan halaman web. JavaScript memungkinkan *developer* untuk mengontrol elemen HTML, mengubah konten secara *real-time*, dan berkomunikasi dengan *server* web tanpa perlu memuat ulang halaman. Hal ini juga memfasilitasi pembuatan aplikasi web berbasis AJAX yang menghadirkan pengalaman pengguna yang lebih dinamis dan responsif. Beberapa keunggulan dari JavaScript antara lain:

1. Bahasa Pemrograman *Universal*

JavaScript bukan hanya bahasa pemrograman untuk web namun JS adalah bahasa pemrograman universal yang dapat digunakan di berbagai platform, termasuk *server-side* (Node.js), *desktop* (Electron), dan bahkan untuk mengembangkan aplikasi mobile (dengan bantuan kerangka kerja seperti *React Native*).

2. Interaksi dengan HTML dan CSS

JavaScript dengan HTML dan CSS untuk menciptakan tampilan dan interaksi dalam halaman web. Hal ini dapat mengubah konten HTML, menganimasikan elemen, mengganti properti CSS, dan lebih banyak lagi. Dengan JavaScript, kita dapat membuat tampilan web yang sangat dinamis dan responsif.

3. *Event Handling*

JavaScript memungkinkan *developer* untuk merespons berbagai peristiwa (*event*) di dalam halaman web, seperti klik tombol, pengisian formulir, atau gerakan mouse. Dengan menanggapi peristiwa ini, kita dapat membuat pengalaman pengguna yang lebih interaktif dan intuitif.

4. Penggunaan *Variabel* dan *Data Types*

Seperti bahasa pemrograman lainnya, JavaScript memungkinkan kita untuk menggunakan variabel untuk menyimpan dan memanipulasi data. JavaScript memiliki beberapa jenis data seperti string, angka, *boolean*, dan objek. JavaScript juga mendukung *array* dan objek, yang memungkinkan kita mengorganisasi data dengan lebih efisien.

5. Fungsi dan Modularitas

Kita dapat mengorganisasi kode JavaScript dalam fungsi-fungsi yang dapat digunakan kembali. Hal ini memungkinkan pengembangan modular, yang memudahkan pemeliharaan dan pengembangan lebih lanjut.

6. Kerangka Kerja (*Framework*) dan Perpustakaan (*Library*)

Ada banyak kerangka kerja (seperti React, Angular, dan Vue.js) dan *library* (seperti jQuery) yang mempermudah pengembangan dengan JavaScript. Mereka menyediakan alat dan struktur yang dapat membantu dalam pembuatan aplikasi web yang kompleks.

7. Pengolahan Data di Sisi Klien

JavaScript berjalan di sisi klien, yang berarti kode tersebut dijalankan di perangkat pengguna. Hal ini memungkinkan pemrosesan data di sisi klien, mengurangi beban *server* dan meningkatkan responsivitas aplikasi.

8. Penggunaan Modern JavaScript

JavaScript telah berkembang dengan pesat dalam beberapa tahun terakhir, dengan penambahan fitur-fitur modern seperti *async/await*, modul ECMAScript 6, dan banyak lagi. Hal ini memungkinkan pengembangan web yang lebih efisien dan aman.

2.2 Document Object Model

Document Object Model (DOM) adalah representasi struktur hierarki dari sebuah halaman web dalam bentuk objek yang dapat dimanipulasi menggunakan bahasa pemrograman seperti JavaScript. Hal ini memungkinkan pengembang web untuk

mengakses, memodifikasi, dan menginteraksikan dengan elemen-elemen HTML, CSS, dan konten lainnya secara dinamis. DOM menggambarkan halaman web sebagai pohon struktur, dengan elemen HTML sebagai simpul-simpul dalam pohon tersebut. Ini berarti bahwa setiap elemen seperti judul, paragraf, gambar, dan tautan adalah objek yang dapat diakses dan dimanipulasi menggunakan JavaScript. Dengan DOM, pengembang dapat membuat halaman web yang responsif, berinteraksi dengan pengguna, dan merubah isi halaman secara dinamis, tanpa perlu mereload seluruh halaman.

Selain sebagai alat untuk memanipulasi elemen HTML, DOM juga memungkinkan pemasangan *event handler*. Hal ini memungkinkan developer untuk merespons peristiwa seperti klik tombol, mengisi formulir, atau menggulir halaman, menjadikan halaman web lebih interaktif. DOM juga memainkan peran penting dalam pengembangan aplikasi web modern, karena memungkinkan komunikasi antara aplikasi web dan server melalui teknologi seperti AJAX (*Asynchronous JavaScript and XML*). Secara keseluruhan, DOM adalah bagian integral dari pengembangan web, memungkinkan pembuatan pengalaman web yang lebih kaya dan dinamis.

2.3 Browser Object Model

Browser Object Model (BOM) adalah bagian dari lingkungan pengembangan web yang menggambarkan objek-objek yang terkait dengan browser yang digunakan untuk mengakses dan menampilkan halaman web. Ini berbeda dari Document Object Model (DOM) yang fokus pada representasi struktur halaman web, BOM berfokus pada interaksi antara browser dan halaman web. Dalam BOM, terdapat berbagai objek yang memberikan akses dan kontrol terhadap berbagai fitur browser, seperti jendela (*window*), dokumen (*document*), riwayat peramban (*browser history*), serta obyek yang mendukung komunikasi jaringan seperti XMLHttpRequest (XHR) untuk pengambilan data tanpa perlu me-*refresh* halaman.

BAB III

PEMBAHASAN

3.1 Form User Input

- File form_validation.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Tugas PBP Pertemuan 1</title>
    <script src="js/form.js"></script>
  </head>
  <body onload="generateCaptcha()">
    <h2>User Input</h2>
    <form
      name="formRegistration"
      method="GET"
      action=""
      onsubmit="return validateForm()"
    >
      <table>
        <tr>
          <td><label for="name">Nama</label></td>
          <td>
            <!-- validasi input name dengan panjang minimal 5 karakter dan
            maksimal 30 karakter dengan atribut min dan max length-->
            <input
              type="text"
              name="name"
              id="name"
              minlength="5"
              maxlength="30"
              placeholder="Masukan nama anda"
            />
          </td>
        </tr>
        <tr>
          <td>Gender</td>
          <td>
            <input type="radio" name="gender" id="male" value="male" />
            <label for="male">Male</label><br />
            <input type="radio" name="gender" id="female" value="female" />
            <label for="female">Female</label>
          </td>
        </tr>
        <tr>
          <td><label for="height">Height</label></td>
          <td>
            <!-- input height tipe number, default value diset 0 agar
            memudahkan dalam validasi ketika dalam js
            optinal sebenarnya bisa menggunakan type text kemudian terdapat
            pengondisian apabila bernilai numerik
            namun dapat juga menggunakan type number yang secara tidak
            langsung akan memaksa user untuk memasukan numerik -->
            <input
              type="number"
              value="0"
              name="height"
              id="height"
            />
          </td>
        </tr>
      </table>
    </form>
  </body>
</html>
```

```

        size="3"
        placeholder="dalam cm"
    />
</td>
</tr>
<tr>
    <td>Address</td>
    <td>
        <!-- validasi input address dengan panjang minimal 5 karakter
dan maksimal 100 karakter-->
        <textarea
            name="address"
            rows="5"
            cols="30"
            minlength="5"
            maxlength="100"
            placeholder="Masukan alamat anda"
        ></textarea>
    </td>
</tr>
<tr>
    <td><label for="province">Province</label></td>
    <td>
        <select name="province" id="province" onchange="get_city()">
            <option value="">--Select a province--</option>
            <option value="West Java">West Java</option>
            <option value="Central Java">Central Java</option>
            <option value="East Java">East Java</option>
        </select>
    </td>
</tr>
<tr>
    <td><label for="city">City</label></td>
    <td>
        <select name="city" id="city">
            <option value="">--Select a city--</option>
        </select>
    </td>
</tr>
<tr>
    <td><label for="hobby">Hobby</label></td>
    <td>
        <input
            type="checkbox"
            name="hobby"
            id="travelling"
            value="travelling"
        />
        <label for="travelling">Travelling</label><br />
        <input type="checkbox" name="hobby" id="reading" value="reading"
/>

        <label for="reading">Reading</label><br />
        <input
            type="checkbox"
            name="hobby"
            id="swimming"
            value="swimming"
        />
        <label for="swimming">Swimming</label><br />
        <input
            type="checkbox"
            name="hobby"
            id="painting"
            value="painting"
        />
        <label for="painting">Painting</label><br />
    </td>
</tr>

```

```

<tr>
  <td><label for="captcha">Captcha</label></td>
  <td>
    <input type="text" name="captcha_text" size="10" disabled />
    <input
      type="text"
      name="captcha_input"
      id="captcha_input"
      size="10"
    />
  </td>
</tr>
<script></script>
<tr>
  <td colspan="2">
    <br /><input type="submit" name="submit" value="Submit" />&nbsp;  
    <input type="reset" name="reset" value="Reset" />
  </td>
</tr>
</table>
</form>
</body>
</html>

```

Penjelasan kode, kerangka dan elemen html yang digunakan:

- Ketika halaman diakses, pada kerangka *body* terdapat fungsi **generateCaptcha** pada atribut **onload** yang berarti bahwa akan memanggil fungsi **generateCaptcha** yang akan menghasilkan *random captcha* pada form yang sudah disediakan.
- Selanjutnya terdapat pula elemen *form* yang digunakan untuk menampilkan *user input form*. Di dalam form nanti akan tersedia beberapa input yang memiliki beberapa jenis juga untuk user perlu masukan. Pada elemen form diberi nama **formRegistration** dengan method **POST** yang memiliki arti bahwa form tersebut akan mensubmit atau mengumpulkan data user. Kemudian selain itu terdapat pula fungsi **validateForm()** pada atribut onsubmit yang artinya ketika form tersebut disubmit akan memanggil fungsi **validateForm()**
 - **Input Nama:** komponen pertama pada form adalah sebuah input yang memiliki tipe **text** untuk memasukan nama user. Terdapat beberapa validasi yang diinputkan melalui atribut **minLength** dan **maxLength** yang telah disediakan oleh html. Kemudian terdapat juga **placeholder** agar membantu user ketika akan memasukan data ke dalam form
 - **Input Gender:** komponen selanjutnya adalah gender dengan menggunakan input bertipe radio.

- **Input Height:** komponen selanjutnya adalah height dengan tipenya berupa number sehingga ketika user akan memasukan data maka dapat dipastikan data akan selalu bertipe number. Untuk atribut lain pada input bertipe number terdapat value yang diset nilai 0 sebagai initial valuenya kemudian terdapat placeholder kembali agar user dapat lebih mengerti form yang harus diisi harus sesuai format atau ketentuan yang sudah ditetapkan.
- **Input Address:** komponen selanjutnya adalah input address dengan menggunakan komponen textarea karena untuk panjang data address akan melebihi dari input-input sebelumnya. Terdapat juga placeholder untuk membantu user untuk memahami form.
- **Input Province dan City:** selanjutnya terdapat komponen province dan city yang digunakan untuk memilih provinsi dan kota. Untuk provinsi yang disediakan hanya 3, kemudian apabila untuk city akan berubah secara dinamis bergantung dengan provinsinya dengan fungsi `get_city` yang ditempatkan pada atribut `onchange`. Atribut `onchange` berguna apabila data pada input berubah maka fungsi akan dijalankan.
- **Input Hobby:** komponen selanjutnya yakni input hobby dengan inputnya bertipe checkbox artinya user dapat memilih lebih dari satu pilihan.
- **Input Captcha:** komponen yang tidak kalah pentingnya yakni input captcha beserta dengan view captchanya. Komponen ini akan menampilkan data captcha yang akan dihasilkan secara random pada saat awal halaman web dimuat atau secara spesifik ketika elemen body dimuat.
- **Button Submit dan Reset:** button submit berguna untuk mengirimkan atau mensubmit data, sedangkan reset maka button ini akan mereset semua nilai form yang telah diinputkan.

- File `form.js`

```
function validateForm() {
    //validasi name tidak boleh kosong
    if (document.forms["formRegistration"]["name"].value == "") {
        alert("Please fill your name.");
        document.forms["formRegistration"]["name"].focus();
        return false;
    }
    //validasi gender tidak boleh kosong
    if (document.forms["formRegistration"]["gender"].value == "") {
        alert("Please select your gender.");
        return false;
    }
}
```

```

//validasi height tidak boleh bernilai 0
if (document.forms["formRegistration"]["height"].value == 0) {
    alert("Please fill your height.");
    document.forms["formRegistration"]["height"].focus();
    return false;
}
//validasi height harus numerik
if (Number.isNaN(document.forms["formRegistration"]["height"].value)) {
    alert("Height must be numeric");
    document.forms["formRegistration"]["height"].value = "";
    document.forms["formRegistration"]["height"].focus();
    return false;
}
//validasi alamat tidak boleh kosong
if (document.forms["formRegistration"]["address"].value == "") {
    alert("Please fill your address.");
    document.forms["formRegistration"]["address"].focus();
    return false;
}
//validasi kota tidak boleh kosong
if (document.forms["formRegistration"]["city"].value == "") {
    alert("Please select your city.");
    document.forms["formRegistration"]["city"].focus();
    return false;
}
//validasi hobby, minimal pilih salah satu
var hobby = document.forms["formRegistration"]["hobby"];
if (hobby[0].checked == false && hobby[1].checked == false &&
hobby[2].checked == false && hobby[3].checked == false) {
    alert("Please select at least one hobby.");
    return false;
}
//validasi text captcha harus sesuai dengan captcha yang di-generate
if (document.forms["formRegistration"]["captcha_input"].value !=
document.forms["formRegistration"]["captcha_text"].value) {
    alert("Please type Captcha correctly.");
}
alert('Success! Thank you for filling the form.')
return true;
}
//fungsi untuk mengisi elemen dropdown city sesuai nilai province yang
dipilih
function get_city() {
    let province = document.forms["formRegistration"]["province"].value;
    if (province == "West Java") {
        document.getElementById("city").innerHTML =
            '<option value="Bandung">Bandung</option>' +
            '<option value="Kuningan">Kuningan</option>' +
            '<option value="Indramayu">Indramayu</option>';
    }
    if (province == "Central Java") {
        document.getElementById("city").innerHTML =
            '<option value="Semarang">Semarang</option>' +
            '<option value="Ungaran">Ungaran</option>' +
            '<option value="Solo">Solo</option>';
    }
    if (province == "East Java") {
        document.getElementById("city").innerHTML =
            '<option value="Surabaya">Surabaya</option>' +
            '<option value="Sidoarjo">Sidoarjo</option>' +
            '<option value="Malang">Malang</option>';
    }
}
}

```

```
//fungsi untuk genereta captcha
function generateCaptcha() {
    let a = Math.ceil(Math.random() * 9) + '';
    let b = Math.ceil(Math.random() * 9) + '';
    let c = Math.ceil(Math.random() * 9) + '';
    let d = Math.ceil(Math.random() * 9) + '';
    let e = Math.ceil(Math.random() * 9) + '';
    let code = a + b + c + d + e;
    document.forms["formRegistration"]["captcha_text"].value = code;
}
```

Penjelasan kode javascript yang digunakan pada halaman web html:

- Fungsi validateForm** : fungsi ini berguna untuk melakukan validasi pada masing-masing input yang telah dibuat pada file html. Dalam fungsi ini terdapat pengecekan apakah input bernilai kosong atau tidak dengan melakukan pengondisian pada masing-masing input. Contohnya untuk input nama maka kode yang perlu digunakan adalah
`document.forms["formRegistration"]["name"].value == ""`
Kemudian setelah input tersebut dicek apabila bernilai true maka popup alert akan muncul. Kode diatas digunakan untuk beberapa input lain juga namun terdapat beberapa hal yang berbeda pada beberapa input seperti untuk input height maka kondisi yang perlu dicek adalah apakah valuenya bernilai 0. Kemudian untuk hobby akan dicek apakah user setidaknya sudah memilih satu hobby. Sedangkan untuk captcha akan dicek apakah nilai captcha sama dengan nilai captcha yang digenerate. Setelah kondisi sudah dipenuhi semua maka akan muncul popup bahwa form telah berhasil diisi.
- Fungsi get_city** : fungsi ini berguna untuk menampilkan data city ketika user telah memilih provinsi. Karena fungsi ini sudah ditempatkan pada atribut onchange maka ketika terdapat perubahan pada nilai provinsi maka data city juga akan berubah. Terdapat 3 kondisional untuk masing-masing provinsi yang akan menampilkan city masing-masing dengan menambahkan elemen option pada innerHTML elemen city.
- Fungsi generateCaptcha** : fungsi ini akan menghasilkan captcha secara random kemudian akan digabungkan menjadi sebuah string yang nantinya akan ditampilkan pada form captcha.

BAB IV

PENUTUP

JavaScript merupakan inti dari pengembangan web modern. Hal ini memungkinkan kita untuk menciptakan halaman web yang dinamis dan interaktif dengan mengakses dan memanipulasi elemen-elemen dalam struktur halaman yang direpresentasikan oleh DOM. Sementara itu, BOM memungkinkan kita untuk berinteraksi dengan browser dan mengendalikan perilaku jendela peramban. Gabungan dari ketiganya memberikan alat yang sangat kuat untuk pengembang web, memungkinkan untuk menciptakan situs web yang meresponsif, dinamis, dan sesuai dengan kebutuhan pengguna modern. Penting untuk terus mengikuti perkembangan dalam dunia pengembangan web, karena teknologi ini terus berkembang dan berubah. Dengan pemahaman yang mendalam tentang JavaScript, DOM, dan BOM, kita dapat menciptakan pengalaman web yang inovatif dan memenuhi tuntutan masa kini.

DAFTAR PUSTAKA

Flanagan, D. (2019). JavaScript: The Definitive Guide. O'Reilly Media.

Duckett, J. (2014). JavaScript and jQuery: Interactive Front-End Web Development. Wiley.

McFarland, D. (2015). JavaScript & jQuery: The Missing Manual. O'Reilly Media.

Pilgrim, M. (2011). HTML5: Up and Running. O'Reilly Media.

Zakas, N. C. (2015). Understanding ECMAScript 6: The Definitive Guide for JavaScript Developers. No Starch Press.