

PEMROGRAMAN WEB
MODUL 6
PENGENALAN JAVASCRIPT TINGKAT LANJUT



Informatika
Telkom University
Surabaya

Disusun Oleh:

Jordan Wijayanto

GitHub

[PemWeb/ at main · Lufasu-Adm/PemWeb \(github.com\)](https://github.com/Lufasu-Adm/PemWeb)

PROGRAM STUDI S1 INFORMATIKA
FAKULTAS INFORMATIKA
UNIVERSITAS TELKOM SURABAYA
2024

Praktikum

1. Tipe Data Object

Buat file object_js.html pada sebuah folder, kemudian terapkan kode HTML dan Javascript di bawah ini.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Tipe Data Object</title>
  </head>
  <body>
    <h1>Belajar Javascript: Tipe Data Object</h1>
  </body>
  <script>
    // Object Kosong
    let foo = {};
    console.log("Type of foo: ", typeof foo); // object

    // Object
    let mahasiswa = {
      nama: "Ali",
      jurusan: "Informatika",
      ipk: 3.67,
      semester: 4,
    };
    console.log("Nama Mahasiswa: ", mahasiswa.nama);
    console.log("Jurusan Mahasiswa: ", mahasiswa.jurusan);
    console.log("IPK Mahasiswa: ", mahasiswa.ipk);
    console.log("Semester Mahasiswa: ", mahasiswa.semester);

    // Menambah Properti
    let mahasiswa2 = {
      nama: "Budi",
      jurusan: "Sistem Informasi",
    };
    mahasiswa2.ipk = 3.67;
    mahasiswa2.semester = 4;
    console.log("Nama Mahasiswa2: ", mahasiswa2.nama);
    console.log("Jurusan Mahasiswa2: ", mahasiswa2.jurusan);
    console.log("IPK Mahasiswa2: ", mahasiswa2.ipk);
    console.log("Semester Mahasiswa2: ", mahasiswa2.semester);

    // Mengubah nilai properti
    let mahasiswa3 = {
      nama: "Dodi",
      jurusan: "Kedokteran",
      ipk: 3.67,
      semester: 4,
    };
    console.log("Nama Mahasiswa3: ", mahasiswa3.nama);
```

```

    mahasiswa3.nama = "Joko";
    console.log("Nama Mahasiswa3: ", mahasiswa3.nama);
    console.log("IPK Mahasiswa3: ", mahasiswa3["ipk"]);
    mahasiswa3["ipk"] = 2.9;
    console.log("IPK Mahasiswa3: ", mahasiswa3["ipk"]);

    // Object Method
    let mobil = {
        nama: "Toyota Avanza",
        tipe: "MPV",
        harga: 300000000,
        warna: "merah",
        hidupkan: function () {
            return "Mesin dihidupkan";
        },
    };
    console.log("Hidupkan mobil: ", mobil.hidupkan());

    // Spread Operator
    let mahasiswa4 = {
        nama: "Heri",
        jurusan: "Informatika",
    };
    let mahasiswaBaru = { ...mahasiswa4 };
    mahasiswaBaru.jurusan = "Ekonomi Manajemen";
    console.log("Jurusan Mahasiswa4: ", mahasiswa4.jurusan);
    console.log("Jurusan MahasiswaBaru: ", mahasiswaBaru.jurusan);

    // For In Method
    let mobil2 = {
        nama: "Toyota Avanza",
        tipe: "MPV",
        harga: 300000000,
        warna: "merah",
        hidupkan: function () {
            return "Mesin dihidupkan";
        },
    };
    for (let prop in mobil2) {
        console.log("Prop: ", prop);
        console.log("Isi " + prop + " = " + mobil2[prop]);
    }
</script>
</html>

```

Kode di atas adalah contoh penggunaan tipe data Object dalam JavaScript

- **Object Kosong:** let foo = {}; mendeklarasikan sebuah object kosong tanpa properti.
- **Object dengan Properti:** let mahasiswa = {...}; mendeklarasikan object dengan properti seperti nama, jurusan, ipk, dan semester.
- **Menambah Properti:** mahasiswa2.ipk = 3.67; menambahkan properti ipk ke object mahasiswa2.

- **Mengubah Nilai Properti:** mahasiswa3.nama = "Joko"; mengubah nilai properti nama pada object mahasiswa3.
- **Object Method:** let mobil = {...}; mendeklarasikan object mobil dengan method hidupkan yang mengembalikan string "Mesin dihidupkan".
- **Spread Operator:** let mahasiswaBaru = { ...mahasiswa4 }; membuat salinan object mahasiswa4 dan mengubah nilai jurusan pada object mahasiswaBaru.
- **For In Loop:** Loop for (let prop in mobil2) {...} digunakan untuk mengiterasi semua properti dalam object mobil2 dan mencetak nama properti serta nilainya.

Setiap bagian kode menunjukkan cara yang berbeda untuk berinteraksi dengan object dalam JavaScript, termasuk mendeklarasikan, mengakses, memodifikasi properti, dan menggunakan method serta operator spread.

2. . Object Oriented Programming (OOP)

Buat file oop_js.html pada sebuah folder, kemudian terapkan kode HTML dan Javascript di bawah ini.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Object Oriented Programming (OOP)</title>
  </head>
  <body>
    <h1>Belajar Javascript: Object Oriented Programming (OOP)</h1>
  </body>
  <script>
    // Javascript Class
    class Mobil {
      constructor(nama, tipe, harga) {
        this.nama = nama;
        this.tipe = tipe;
        this.harga = harga;
      }

      hidupkan() {
        return `Mesin ${this.nama} dihidupkan`;
      }

      pergi(tempat) {
        return `Pergi ke ${tempat} dengan ${this.nama}`;
      }
    }

    let mobilAndi = new Mobil("Daihatsu Xenia", "MPV", 220000000);
    console.log("Nama Mobil Andi: ", mobilAndi.nama);
    console.log("Mobil Andi Pergi: ", mobilAndi.pergi("Jakarta"));
    let mobilJoko = new Mobil("Toyota Camry", "Sedan", 750000000);
```

```

    console.log("Nama Mobil Joko: ", mobilJoko.nama);
    console.log("Mobil Joko Pergi: ", mobilJoko.pergi("Bali"));
</script>
</html>

```

Kode di atas adalah contoh penggunaan konsep Pemrograman Berorientasi Objek (OOP) dalam JavaScript.

- **Class Mobil:** Ini adalah definisi class atau blueprint untuk membuat objek Mobil. Class ini memiliki tiga properti: nama, tipe, dan harga, yang diinisialisasi melalui konstruktor. Class ini juga memiliki dua metode: hidupkan() dan pergi(tempat).
- **Konstruktor:** Fungsi constructor(nama, tipe, harga) ini dipanggil saat pembuatan objek baru dari class Mobil. Fungsi ini menginisialisasi properti nama, tipe, dan harga berdasarkan argumen yang diberikan.
- **Metode:** hidupkan() dan pergi(tempat) adalah metode dari class Mobil. Metode hidupkan() mengembalikan string yang menggambarkan bahwa mesin mobil dihidupkan. Metode pergi(tempat) mengembalikan string yang menggambarkan mobil pergi ke suatu tempat.
- **Pembuatan Objek:** let mobilAndi = new Mobil("Daihatsu Xenia", "MPV", 220000000); dan let mobilJoko = new Mobil("Toyota Camry", "Sedan", 750000000); adalah pembuatan objek baru dari class Mobil. Objek mobilAndi dan mobilJoko memiliki properti dan metode yang ditentukan dalam class Mobil.
- **Penggunaan Objek:** console.log("Nama Mobil Andi: ", mobilAndi.nama); dan console.log("Mobil Andi Pergi: ", mobilAndi.pergi("Jakarta")); adalah contoh penggunaan objek. Properti nama diakses langsung dari objek, dan metode pergi(tempat) dipanggil pada objek.

3. Built-in Object

Buat file built_in_object.html pada sebuah folder, kemudian terapkan kode HTML dan Javascript di bawah ini.

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Number, Math, String, Array</title>
  </head>
  <body>
    <h1>Belajar Javascript: Number, Math, String, Array</h1>
  </body>
  <script>
    // Number Static Property & Static Method
    console.log("Max Value: ", Number.MAX_VALUE);
    console.log('Is "A" integer? ', Number.isInteger("A"));
  </script>

```

```

console.log("Is 12 integer? ", Number.isInteger(12));

// Number Object Instance Method
let foo = new Number(5.123456);
console.log("toPrecision : ", foo.toPrecision(3));

// Math Static Property
console.log("Math.E = " + Math.E); // 2.718281828459045
console.log("Math.LN10 = " + Math.LN10); // 2.302585092994046
console.log("Math.LN2 = " + Math.LN2); // 0.6931471805599453
console.log("Math.LOG10E = " + Math.LOG10E); // 0.4342944819032518
console.log("Math.LOG2E = " + Math.LOG2E); // 1.4426950408889634
console.log("Math.PI = " + Math.PI); // 3.141592653589793
console.log("Math.SQRT1_2 = " + Math.SQRT1_2); // 0.7071067811865476
console.log("Math.SQRT2 = " + Math.SQRT2); // 1.4142135623730951

// Math Static Method
let num = 12.5;
console.log("Floor: ", Math.floor(num)); // 12
console.log("Ceil: ", Math.ceil(num)); // 13
console.log("Round: ", Math.round(num)); // 13
console.log("Random: ", Math.random());
console.log("Pow: ", Math.pow(5, 2)); // 25
console.log("Pow: ", Math.pow(2, 8)); // 256

// String Object Instance Property & Method
let text = new String("Selamat Belajar JavaScript");
console.log("String length: ", text.length);
console.log("Lowercase: : ", text.toLowerCase());
console.log("Uppercase: : ", text.toUpperCase());

// Array Static Method
let a = new Array("a", "b", "c", "d", "e");
let b = ["a", "b", "c", "d", "e"];
let c = "aiueo";
let d = 123456;
let e = true;
console.log("A: ", Array.isArray(a)); // true
console.log("B: ", Array.isArray(b)); // true
console.log("C: ", Array.isArray(c)); // false
console.log("D: ", Array.isArray(d)); // false
console.log("E: ", Array.isArray(e)); // false
console.log("isArray: ", Array.isArray([1, 2, 3])); // true

// Array Instance Property
let arr = ["a", "b", "c", "d", "e"];
console.log("Arr length: ", arr.length); // 5
let arr2 = ["apel", "pisang", "anggur", "jambu"];
console.log("Arr2 length: ", arr2.length); // 4
console.log("Array length: ", ["x", "y", "z"].length); // 3

// Array For Each
let arr3 = ["a", "b", "c", "d", "e"];
arr3.forEach(function (element) {
    console.log("Arr3 element: ", element);
});
arr3.forEach((element) =>
    console.log("Arr3 element from Arrow: ", element)
);
</script>
</html>

```

Kode di atas adalah contoh penggunaan beberapa fitur dalam JavaScript, termasuk Number, Math, String, dan Array. Berikut penjelasannya dalam Bahasa Indonesia:

- **Number Static Property & Static Method:** Bagian ini menunjukkan penggunaan properti statis dan metode statis dari objek Number. Misalnya, Number.MAX_VALUE mengembalikan nilai maksimum yang dapat direpresentasikan oleh Number, dan Number.isInteger() digunakan untuk mengecek apakah suatu nilai adalah integer.
- **Number Object Instance Method:** Bagian ini menunjukkan penggunaan metode instance dari objek Number. Misalnya, foo.toPrecision(3) digunakan untuk mengubah angka menjadi string dengan presisi tertentu.
- **Math Static Property:** Bagian ini menunjukkan penggunaan properti statis dari objek Math. Misalnya, Math.PI mengembalikan nilai Pi.
- **Math Static Method:** Bagian ini menunjukkan penggunaan metode statis dari objek Math. Misalnya, Math.floor(num) digunakan untuk membulatkan angka ke bawah, dan Math.pow(5, 2) digunakan untuk menghitung pangkat.
- **String Object Instance Property & Method:** Bagian ini menunjukkan penggunaan properti dan metode instance dari objek String. Misalnya, text.length digunakan untuk mendapatkan panjang string, dan text.toLowerCase() dan text.toUpperCase() digunakan untuk mengubah string menjadi huruf kecil dan huruf besar.
- **Array Static Method:** Bagian ini menunjukkan penggunaan metode statis dari objek Array. Misalnya, Array.isArray(a) digunakan untuk mengecek apakah suatu nilai adalah array.
- **Array Instance Property:** Bagian ini menunjukkan penggunaan properti instance dari objek Array. Misalnya, arr.length digunakan untuk mendapatkan jumlah elemen dalam array.
- **Array For Each:** Bagian ini menunjukkan penggunaan metode forEach() dari objek Array. Metode ini digunakan untuk melakukan iterasi pada setiap elemen dalam array. Dalam contoh ini, setiap elemen dalam array dicetak ke konsol.

4. Global Property dan Function

Buat file global_js.html pada sebuah folder, kemudian terapkan kode HTML dan Javascript di bawah ini.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Global Property dan Function</title>
  </head>
  <body>
    <h1>Belajar Javascript: Global Property dan Function</h1>
```

```

</body>
<script>
  // Global Property
  let nan = NaN;
  console.log("nan: ", nan); // NaN
  let infinity = Infinity;
  console.log("infinity: ", infinity); // Infinity
  let undfn = undefined;
  console.log("undfn", undfn); // undefined
  let n = null;
  console.log("n: ", n); // null

  // Global Function
  let foo = "1234.567";
  console.log("parseInt: ", parseInt(foo)); // 1234
  console.log("parseInt: ", parseInt(99.99)); // 99
  let foo2 = "1234";
  console.log("Type of foo2: ", typeof foo2); // string
  foo2 = parseFloat(foo2);
  console.log("foo2: ", foo2); // 1234
  console.log("Type of foo2: ", typeof foo2); // number
</script>
</html>

```

Kode di atas adalah contoh penggunaan beberapa properti dan fungsi global dalam JavaScript.

1. **Global Property:** Bagian ini menunjukkan penggunaan beberapa properti global dalam JavaScript, seperti NaN (Not a Number), Infinity, undefined, dan null.
2. **Global Function:** Bagian ini menunjukkan penggunaan beberapa fungsi global dalam JavaScript. Misalnya, `parseInt(foo)` digunakan untuk mengubah string menjadi integer. Fungsi `parseFloat(foo2)` digunakan untuk mengubah string menjadi floating point number. Fungsi `typeof foo2` digunakan untuk mendapatkan tipe data dari variabel `foo2`.

Secara keseluruhan, kode ini memberikan contoh bagaimana menggunakan beberapa properti dan fungsi global yang disediakan oleh JavaScript. Properti dan fungsi ini dapat digunakan di mana saja dalam kode ini merupakan bagian integral dari bahasa pemrograman JavaScript.

5. Form Validation

Buat file `form.html` pada sebuah folder, kemudian terapkan kode HTML dan Javascript di bawah ini.

```

<!DOCTYPE html>
<html lang="id">
  <head>
    <meta charset="utf-8" />
    <title>Form Validation</title>
    <style>
      .error {
        color: red;

```



```

        width: 300px;
        text-align: center;
        padding: 2px 10px;
        margin-left: 10px;
    }
</style>
</head>
<body>
    <h1>Belajar Javascript: Form Validation</h1>
    <form id="formKu" name="formKu" method="get" action="proses.php">
        <p>
            Username: <input type="text" name="username" id="username" />
            <span id="usernameSpan"></span>
        </p>
        <p>
            Password: <input type="password" name="pass" id="pass" />
            <span id="passSpan"></span>
        </p>
        <p>
            Konfirmasi Password:
            <input type="password" name="konfPass" id="konfPass" /><span
            id="konfPassSpan"
            ></span>
        </p>
        <p>
            <input type="checkbox" name="syarat" id="syarat" /> Saya menyetujui
            syarat dan ketentuan<span id="syaratSpan"></span>
        </p>
        <p>
            <input type="submit" name="submit" id="submit" value="Kirim Data" />
        </p>
    </form>
    <script>
        let formKuNode = document.getElementById("formKu");

        let usernameNode = document.getElementById("username");
        let usernameSpanNode = document.getElementById("usernameSpan");

        let passNode = document.getElementById("pass");
        let passSpanNode = document.getElementById("passSpan");

        let konfPassNode = document.getElementById("konfPass");
        let konfPassSpanNode = document.getElementById("konfPassSpan");

        let syaratNode = document.getElementById("syarat");
        let syaratSpanNode = document.getElementById("syaratSpan");

        const diProses = (e) => {
            //===== Untuk Validasi username ===== //
            let usernameError = "";

            if (usernameNode.value.trim() === "") {
                usernameError = "Username harus diisi";
            } else if (/^\w/.test(usernameNode.value.trim())) {
                usernameError = "Hanya bisa diisi karakter alfanumerik";
            } else if (usernameNode.value.trim().length < 6) {
                usernameError = "Username minimal 6 karakter";
            }
        }
    </script>

```

```
    }
```

```
    if (usernameError !== "") {  
        usernameSpanNode.innerHTML = usernameError;  
        usernameSpanNode.className = "error";  
        usernameNode.style.border = "2px solid red";  
        e.preventDefault();  
    }
```

```
    //===== Untuk Validasi Password ===== //  
    let passError = "";  
    if (passNode.value.trim() === "") {  
        passError = "Password harus diisi";  
    } else if (passNode.value.trim().length < 6) {  
        passError = "Password minimal 6 karakter";  
    }  
}
```

```
    if (passError !== "") {  
        passSpanNode.innerHTML = passError;  
        passSpanNode.className = "error";  
        passNode.style.border = "2px solid red";  
        e.preventDefault();  
    }
```

```
    //===== Untuk Validasi Konfirmasi Password ===== //  
    let konfPassError = "";  
    if (konfPassNode.value.trim() === "") {  
        konfPassError = "Konfirmasi Password harus diisi";  
    } else if (konfPassNode.value.trim().length < 6) {  
        konfPassError = "Konfirmasi Password minimal 6 karakter";  
    } else if (konfPassNode.value !== passNode.value) {  
        konfPassError = "Konfirmasi Password tidak sama";  
    }  
}
```

```
    if (konfPassError !== "") {  
        konfPassSpanNode.innerHTML = konfPassError;  
        konfPassSpanNode.className = "error";  
        konfPassNode.style.border = "2px solid red";  
        e.preventDefault();  
    }
```

```
    //===== Untuk Validasi Checkbox Syarat ===== //  
    let syaratError = "";  
    if (syaratNode.checked !== true) {  
        syaratError = "Syarat dan ketentuan harus di setujui";  
    }  
}
```

```
    if (syaratError !== "") {  
        syaratSpanNode.innerHTML = syaratError;  
        syaratSpanNode.className = "error";  
        e.preventDefault();  
    }  
};
```

```
const hapusError = (e) => {  
    e.target.style.border = "";  
    e.target.parentElement.lastElementChild.innerHTML = "";
```

```

    });

    formKuDNode.addEventListener("submit", diProses);
    usernameNode.addEventListener("focus", hapusError);
    passNode.addEventListener("focus", hapusError);
    konfPassNode.addEventListener("focus", hapusError);
    syaratNode.addEventListener("focus", hapusError);
  </script>
</body>
</html>

```

Kode di atas adalah contoh penggunaan validasi form dalam JavaScript.

- **Form:** Form ini memiliki beberapa field input seperti username, password, konfirmasi password, dan checkbox untuk syarat dan ketentuan. Setiap field memiliki elemen span yang digunakan untuk menampilkan pesan error.
- **Global Variables:** Variabel-variabel seperti formKuDNode, usernameNode, usernameSpanNode, dan lainnya digunakan untuk menyimpan referensi ke elemen-elemen HTML.
- **Function diProses(e):** Fungsi ini dipanggil saat form disubmit. Fungsi ini melakukan validasi pada setiap field input. Jika terdapat error, maka pesan error akan ditampilkan dan border input akan berubah menjadi merah.
- **Function hapusError(e):** Fungsi ini dipanggil saat fokus pada field input. Fungsi ini menghapus border merah dan pesan error.
- **Event Listeners:** Event listener ditambahkan ke form untuk event submit dan ke setiap field input untuk event focus. Ketika form disubmit, fungsi diProses(e) akan dipanggil. Ketika field input mendapatkan fokus, fungsi hapusError(e) akan dipanggil.

Secara keseluruhan, kode ini menunjukkan bagaimana melakukan validasi form dalam JavaScript. Jika pengguna mencoba mengirim form dengan data yang tidak valid, maka pesan error akan ditampilkan dan form tidak akan dikirim sampai semua error diperbaiki.

6. Asynchronous Javascript and XML (AJAX)

Buat file salam.txt kemudian isi dengan “Selamat Pagi!”. Kemudian buat file ajax.html pada sebuah folder, kemudian terapkan kode HTML dan Javascript di bawah ini.

```

<!DOCTYPE html>
<html lang="id">
  <head>
    <meta charset="utf-8" />
    <title>AJAX</title>
  </head>
  <body>
    <h1>Belajar Javascript: Asynchronous Javascript and XML (AJAX)</h1>
    <button id="tombol">Ambil Data</button>
    <p>Hasil: <span id="hasil"></span></p>
    <script>

```

```

    let tombolNode = document.getElementById("tombol");
    let hasilNode = document.getElementById("hasil");

    const getAJAX = () => {
        let request = new XMLHttpRequest();
        request.open("GET", "salam.txt", false);
        request.send();
        hasilNode.innerHTML = request.responseText;
    };

    tombolNode.addEventListener("click", getAJAX);
</script>
</body>
</html>

```

Kode di atas adalah contoh penggunaan AJAX (Asynchronous JavaScript and XML) dalam JavaScript.

- **Tombol dan Hasil:** Ada sebuah tombol dengan id “tombol” dan sebuah elemen span dengan id “hasil” di HTML. Tombol ini digunakan untuk memicu permintaan AJAX, dan elemen span digunakan untuk menampilkan hasil dari permintaan tersebut.
- **Variabel:** Variabel tombolNode dan hasilNode digunakan untuk menyimpan referensi ke elemen tombol dan span.
- **Fungsi getAJAX:** Fungsi ini membuat sebuah objek XMLHttpRequest, yang digunakan untuk berkomunikasi dengan server. Fungsi open digunakan untuk menyiapkan permintaan ke server. Dalam hal ini, metode GET digunakan dan file yang diminta adalah “salam.txt”. Fungsi send digunakan untuk mengirim permintaan. Setelah permintaan selesai, respons dari server (teks dari file “salam.txt”) ditampilkan di elemen span.
- **Event Listener:** Event listener ditambahkan ke tombol untuk event “click”. Ketika tombol diklik, fungsi getAJAX akan dipanggil.

Perlu diperhatikan bahwa dalam contoh ini, permintaan AJAX dibuat secara sinkronis (parameter ketiga dari fungsi open adalah false). Ini berarti JavaScript akan menunggu sampai permintaan selesai sebelum melanjutkan eksekusi kode berikutnya.

7. JSON, API, Fetch

Buat file json_api_fetch.html pada sebuah folder, kemudian terapkan kode HTML dan Javascript di bawah ini.

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>JSON, API, Fetch</title>

```

```

</head>
<body>
  <h1>Belajar Javascript: JSON, API, Fetch</h1>
</body>
<script>
  // JSON Parse
  let mahasiswaJSON = `{"nim":"18010245","nama":"Budi Susanto",
    "jurusan":"Informatika","asalProvinsi":"DKI Jakarta"}`;
  let mahasiswa = JSON.parse(mahasiswaJSON);
  console.log("Mahasiwa: ", mahasiswa);
  console.log("Nama Mahasiswa: ", mahasiswa.nama);

  // JSON Stringify
  let mahasiswa2 = {
    nim: "18010245",
    nama: "Budi Susanto",
    jurusan: "Informatika",
    asalProvinsi: "DKI Jakarta",
  };
  let mahasiswa2JSON = JSON.stringify(mahasiswa2);
  console.log("Mahasiswa2 JSON", mahasiswa2JSON);

  // Fetch
  fetch("https://reqres.in/api/users/1")
    .then((response) => response.json())
    .then((data) => {
      console.log("Data: ", data);
      let user = data.data;
      console.log("ID: ", user.id);
      console.log("Firstname: ", user.first_name);
      console.log("Lastname: ", user.last_name);
      console.log("Email: ", user.email);
      console.log("Avatar: ", user.avatar);
    });
</script>
</html>

```

Kode di atas adalah contoh penggunaan JSON, API, dan Fetch dalam JavaScript.

- **JSON Parse:** Bagian ini menunjukkan penggunaan `JSON.parse()`. Fungsi ini digunakan untuk mengubah string JSON menjadi objek JavaScript. Dalam contoh ini, string JSON yang berisi data mahasiswa diubah menjadi objek JavaScript dan kemudian dicetak ke konsol.
- **JSON Stringify:** Bagian ini menunjukkan penggunaan `JSON.stringify()`. Fungsi ini digunakan untuk mengubah objek JavaScript menjadi string JSON. Dalam contoh ini, objek JavaScript yang berisi data mahasiswa diubah menjadi string JSON dan kemudian dicetak ke konsol.
- **Fetch:** Bagian ini menunjukkan penggunaan `fetch()`. Fungsi ini digunakan untuk membuat permintaan ke server. Dalam contoh ini, permintaan dibuat ke endpoint API “`https://reqres.in/api/users/1`”. Setelah permintaan selesai, respons dari server diubah menjadi objek JavaScript menggunakan `response.json()`. Kemudian, data dari respons dicetak ke konsol.

Secara keseluruhan, kode ini menunjukkan bagaimana menggunakan JSON untuk serialisasi dan deserialisasi data, serta bagaimana menggunakan Fetch untuk berinteraksi dengan API.

Tugas

```
<!DOCTYPE html>
<html lang="id">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Form Registrasi</title>
<style>
    .error {
        color: red;
    }
</style>
</head>
<body>

<h2>Form Registrasi</h2>
<form id="registrationForm" action="#" method="post">
    <div>
        <label for="nama">Nama:</label><br>
        <input type="text" id="nama" name="nama"><span id="namaError"
class="error"></span><br>
    </div>
    <div>
        <label for="username">Username:</label><br>
        <input type="text" id="username" name="username"><span
id="usernameError" class="error"></span><br>
    </div>
    <div>
        <label for="email">Email:</label><br>
        <input type="email" id="email" name="email"><span id="emailError"
class="error"></span><br>
    </div>
    <div>
        <label for="password">Password:</label><br>
        <input type="password" id="password" name="password"><br>
    </div>
    <div>
        <label for="passwordConfirmation">Konfirmasi Password:</label><br>
        <input type="password" id="passwordConfirmation"
name="passwordConfirmation"><span id="passwordError" class="error"></span><br>
    </div>
    <div>
        <label for="telepon">No Telepon:</label><br>
        <input type="tel" id="telepon" name="telepon"><span id="teleponError"
class="error"></span><br>
    </div>
```

```

<div>
  <label for="jenisKelamin">Jenis Kelamin:</label><br>
  <input type="radio" id="pria" name="jenisKelamin" value="pria">
  <label for="pria">Pria</label>
  <input type="radio" id="wanita" name="jenisKelamin" value="wanita">
  <label for="wanita">Wanita</label>
  <span id="jenisKelaminError" class="error"></span><br>
</div>
<div>
  <label for="website">Alamat Website:</label><br>
  <input type="url" id="website" name="website"><span id="websiteError"
class="error"></span><br>
</div>
<div>
  <input type="submit" value="Submit">
</div>
</form>

<script>
  document.getElementById('registrationForm').addEventListener('submit',
function(event) {
    var isValid = true;

    // Validasi Nama
    var namaInput = document.getElementById('nama');
    var namaError = document.getElementById('namaError');
    if (namaInput.value.trim() === '') {
      namaError.textContent = 'Nama harus diisi';
      isValid = false;
    } else {
      namaError.textContent = '';
    }

    // Validasi Username
    var usernameInput = document.getElementById('username');
    var usernameError = document.getElementById('usernameError');
    if (usernameInput.value.trim() === '') {
      usernameError.textContent = 'Username harus diisi';
      isValid = false;
    } else {
      usernameError.textContent = '';
    }

    // Validasi Email
    var emailInput = document.getElementById('email');
    var emailError = document.getElementById('emailError');
    var emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
    if (!emailRegex.test(emailInput.value)) {

```



```

        emailError.textContent = 'Email tidak valid';
        isValid = false;
    } else {
        emailError.textContent = '';
    }

    // Validasi Password
    var passwordInput = document.getElementById('password');
    var passwordConfirmationInput =
document.getElementById('passwordConfirmation');
    var passwordError = document.getElementById('passwordError');
    if (passwordInput.value === '') {
        passwordError.textContent = 'Password harus diisi';
        isValid = false;
    } else if (passwordInput.value !== passwordConfirmationInput.value) {
        passwordError.textContent = 'Konfirmasi password tidak cocok';
        isValid = false;
    } else {
        passwordError.textContent = '';
    }

    // Validasi No Telepon
    var teleponInput = document.getElementById('telepon');
    var teleponError = document.getElementById('teleponError');
    var teleponRegex = /^\\d{10,12}$/; // Angka dengan panjang 10-12 digit
    if (!teleponRegex.test(teleponInput.value)) {
        teleponError.textContent = 'No Telepon tidak valid';
        isValid = false;
    } else {
        teleponError.textContent = '';
    }

    // Validasi Jenis Kelamin
    var priaInput = document.getElementById('pria');
    var wanitaInput = document.getElementById('wanita');
    var jenisKelaminError = document.getElementById('jenisKelaminError');
    if (!priaInput.checked && !wanitaInput.checked) {
        jenisKelaminError.textContent = 'Jenis Kelamin harus dipilih';
        isValid = false;
    } else {
        jenisKelaminError.textContent = '';
    }

    // Validasi Alamat Website
    var websiteInput = document.getElementById('website');
    var websiteError = document.getElementById('websiteError');
    var urlRegex = /^(ftp|http|https):\\/\\/[^ "]+$/;
    if (!urlRegex.test(websiteInput.value)) {

```

```

        websiteError.textContent = 'Alamat Website tidak valid';
        isValid = false;
    } else {
        websiteError.textContent = '';
    }

    // Hentikan submit jika ada kesalahan validasi
    if (!isValid) {
        event.preventDefault();
    }
});
</script>

</body>
</html>

```

Kode di atas adalah contoh implementasi validasi form menggunakan JavaScript.

- **Form:** Form ini memiliki beberapa field input seperti nama, username, email, password, konfirmasi password, no telepon, jenis kelamin, dan alamat website. Setiap field memiliki elemen span yang digunakan untuk menampilkan pesan error.
- **Variabel:** Variabel seperti namaInput, namaError, dan lainnya digunakan untuk menyimpan referensi ke elemen-elemen HTML.
- **Fungsi function(event):** Fungsi ini dipanggil saat form disubmit. Fungsi ini melakukan validasi pada setiap field input. Jika terdapat error, maka pesan error akan ditampilkan dan form tidak akan dikirim sampai semua error diperbaiki.
- **Validasi:** Setiap field input divalidasi dengan cara yang berbeda. Misalnya, field nama dan username divalidasi apakah kosong atau tidak, field email dan website divalidasi apakah formatnya benar atau tidak, field password divalidasi apakah kosong dan apakah sama dengan field konfirmasi password, dan seterusnya.
- **Event Listener:** Event listener ditambahkan ke form untuk event submit. Ketika form disubmit, fungsi validasi akan dipanggil.

Secara keseluruhan, kode ini menunjukkan bagaimana melakukan validasi form dalam JavaScript. Jika pengguna mencoba mengirim form dengan data yang tidak valid, maka pesan error akan ditampilkan dan form tidak akan dikirim sampai semua error diperbaiki.



127.0.0.1:5500/Minggu_7/tugas/tgs.html

Form Registrasi

Nama:

Nama harus diisi

Username:

Username harus diisi

Email:

Email tidak valid

Password:

Konfirmasi Password:

Password harus diisi

No Telepon:

No Telepon tidak valid

Jenis Kelamin:

☐ Pria ☐ Wanita Jenis Kelamin harus dipilih

Alamat Website:

Alamat Website tidak valid

Submit