

## **TUGAS PERTEMUAN 2**

Disusun Guna Memenuhi Tugas Mata Kuliah Pemrograman Berbasis Framework

Dosen Pengampu : Dimas Wahyu Wibowo, S.T., M.T.



Disusun Oleh :

Akhmad Ramadani (2041720002)

TI-3B

**PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNOLOGI INFORMASI  
POLITEKNIK NEGERI MALANG**

**2023**

## 1. Latihan

### a. Memproses Object pada JS

#### i. Kode Program

object.html

```
<!-- *** Author : Akhmad Ramadani *** -->
<!-- *** NIM : 2041720002 *** -->
<!-- *** Kelas : TI-3B *** -->

<!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>React Uncover</title>
</head>
<body>
  <h1>Belajar Javascript</h1>
  <script>
    let mahasiswa = {
      nama: "Eka",
      umur: 19,
      jurusan: "Teknik Informatika",
      getInfo(){
        return `${this.nama} (${this.umur}
tahun) dari jurusan ${this.jurusan}`;
      }
    }

    console.log(mahasiswa);

    mahasiswa.umur = 20;
    mahasiswa.tempatLahir = "Jakarta";

    console.log(mahasiswa);
  </script>
</body>
</html>
```

ii. Hasil

## Belajar Javascript



iii. Penjelasan

1. Kode di atas mendeklarasikan objek menggunakan let. Objek tersebut memiliki tiga properti yaitu nama, umur, dan jurusan, serta satu metode getInfo() yang mengembalikan string dengan mengakses nilai properti objek.
2. Kemudian, setelah dicetak, properti pada objek diubah seperti kode di atas.
3. Setelah diubah, dicetak, dan akan tampak bahwa objek juga sudah berubah.

## b. Memproses Object pada JS 2

### i. Kode Program

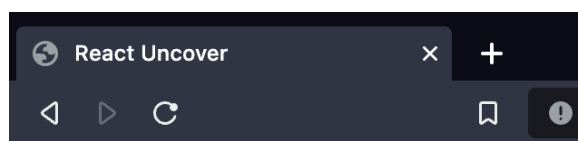
object2.html

```
<!-- *** Author : Akhmad Ramadani *** -->
<!-- *** NIM      : 2041720002 *** -->
<!-- *** Kelas   : TI-3B   *** -->

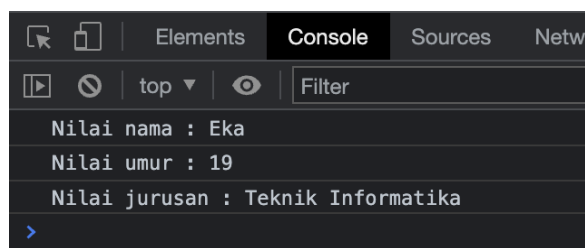
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible"
content="IE=edge">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>React Uncover</title>
</head>
<body>
  <h1>Belajar Javascript</h1>
  <script>
    let mahasiswa = {
      nama: "Eka",
      umur: 19,
      jurusan: "Teknik Informatika",
    }

    for (prop in mahasiswa) {
      console.log(`Nilai ${prop} :
${mahasiswa[prop]}`);
    }
  </script>
</body>
</html>
```

### ii. Hasil



## Belajar Javascript



iii. Penjelasan

1. Kode di atas digunakan untuk menampilkan properti-properti yang ada pada objek mahasiswa yang telah dibuat.
2. Untuk menampilkannya bisa menggunakan perulangan `foreach` seperti pada kode di atas.

### c. Contoh Penggunaan Memproses Object pada JS

#### i. Kode Program

object3.html

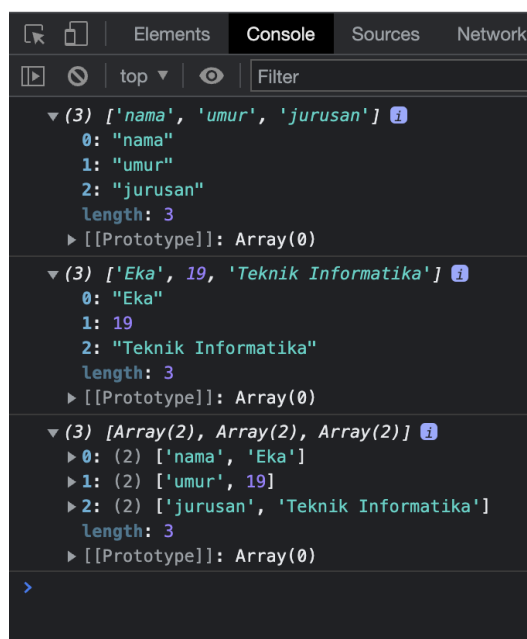
```
<!-- *** Author : Akhmad Ramadani *** -->
<!-- *** NIM      : 2041720002 *** -->
<!-- *** Kelas   : TI-3B   *** -->

<!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>React Uncover</title>
</head>
<body>
  <h1>Belajar Javascript</h1>
  <script>
    let mahasiswa = {
      nama: "Eka",
      umur: 19,
      jurusan: "Teknik Informatika",
    }

    console.log(Object.keys(mahasiswa));
    console.log(Object.values(mahasiswa));

    console.log(Object.entries(mahasiswa));
  </script>
</body>
</html>
```

#### ii. Hasil



iii. Penjelasan

1. `Object.keys()` digunakan untuk mendapatkan array yang berisi nama-nama properti dari sebuah objek.
2. `Object.values()` digunakan untuk mendapatkan array yang berisi nilai-nilai dari setiap properti pada sebuah objek.
3. `Object.entries()` digunakan untuk mendapatkan array yang berisi pasangan key-value dari setiap properti pada sebuah objek.

#### d. Array dan Object Destructuring

##### i. Kode Program

arraydestructuring.html

```
<!-- *** Author : Akhmad Ramadani *** -->
<!-- *** NIM      : 2041720002 *** -->
<!-- *** Kelas   : TI-3B *** -->

<!DOCTYPE html>
<html lang="en">

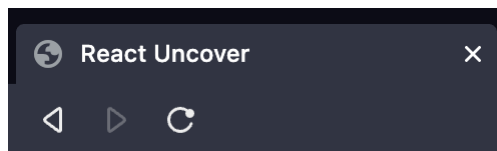
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible"
content="IE=edge">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>React Uncover</title>
</head>

<body>
  <h1>Belajar Javascript</h1>
  <script>
    let mahasiswa = ["Andi", "Lisa", "Eko"];
    let [a, b, c] = mahasiswa;

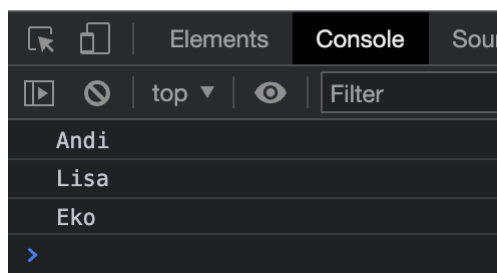
    console.log(a);
    console.log(b);
    console.log(c);
  </script>
</body>

</html>
```

##### ii. Hasil



# Belajar Javascript





iii. Penjelasan

1. Pada script di atas, kita membuat sebuah array mahasiswa yang berisi tiga string: "Andi", "Lisa", dan "Eko". Kemudian kita mendestrukturisasi array tersebut menggunakan tanda kurung siku dan memberikan nama variabel a, b, dan c untuk masing-masing elemen di dalam array.
2. Kemudian kita mencetak nilai dari masing-masing variabel a, b, dan c menggunakan fungsi `console.log()`. Hasilnya adalah nilai dari masing-masing elemen di dalam array mahasiswa, yaitu "Andi", "Lisa", dan "Eko".

## e. Array dan Object Destructuring

### i. Kode Program

objecdestructuring.html

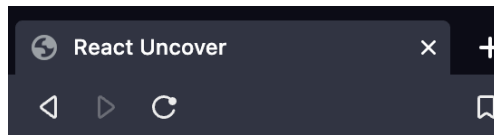
```
<!-- *** Author : Akhmad Ramadani *** -->
<!-- *** NIM      : 2041720002 *** -->
<!-- *** Kelas   : TI-3B   *** -->

<!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>React Uncover</title>
</head>
<body>
  <h1>Belajar Javascript</h1>
  <script>
    let mahasiswa = {
      nama: "Eka",
      umur: 19,
      jurusan: "Teknik Informatika",
    }

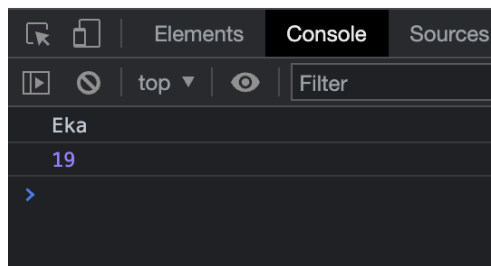
    let {nama, umur} = mahasiswa;

    console.log(nama);
    console.log(umur);
  </script>
</body>
</html>
```

### ii. Hasil



# Belajar Javascript



iii. Penjelasan

1. Di atas adalah kode untuk destrukturisasi objek. Pada kode di atas, terdapat sebuah objek bernama "mahasiswa" dengan tiga properti yaitu "nama", "umur", dan "jurusan". Kemudian, dilakukan destrukturisasi objek untuk mengambil nilai dari properti "nama" dan "umur" dan menyimpannya ke dalam variabel "nama" dan "umur" secara langsung. Selanjutnya, nilai dari variabel "nama" dan "umur" dicetak ke console.

## f. Array dan Object Destructuring

### i. Kode Program

objecdestructuring2.html

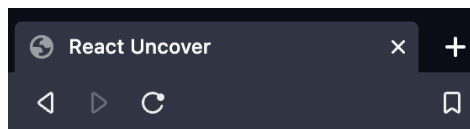
```
<!-- *** Author : Akhmad Ramadani *** -->
<!-- *** NIM      : 2041720002 *** -->
<!-- *** Kelas   : TI-3B *** -->

<!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>React Uncover</title>
</head>
<body>
  <h1>Belajar Javascript</h1>
  <script>
    const dataMahasiswa = () => {
      return {
        nama: "Eka",
        umur: 19,
        jurusan: "Teknik Informatika",
      }
    }

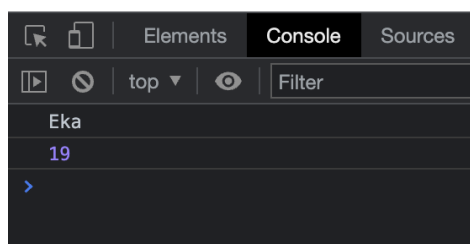
    let {nama, umur} = dataMahasiswa();

    console.log(nama);
    console.log(umur);
  </script>
</body>
</html>
```

### ii. Hasil



## Belajar Javascript



iii. Penjelasan

1. Code di atas adalah contoh penggunaan arrow function dan object destructuring di dalam JavaScript.
2. Fungsi dataMahasiswa() adalah sebuah arrow function yang mengembalikan sebuah objek yang berisi data mahasiswa.
3. Kemudian, menggunakan object destructuring untuk mengambil nilai dari objek yang dikembalikan oleh fungsi dataMahasiswa(). Dengan destructuring, kita dapat mengambil nilai nama dan umur dari objek tersebut.

## g. Array dan Object Destructuring

### i. Kode Program

objecdestructuring3.html

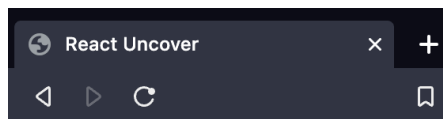
```
<!-- *** Author : Akhmad Ramadani *** -->
<!-- *** NIM      : 2041720002 *** -->
<!-- *** Kelas   : TI-3B   *** -->

<!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>React Uncover</title>
</head>
<body>
  <h1>Belajar Javascript</h1>
  <script>
    const dataMahasiswa = () => {
      return {
        nama: "Eka",
        umur: 19,
        jurusan: "Teknik Informatika",
      }
    }

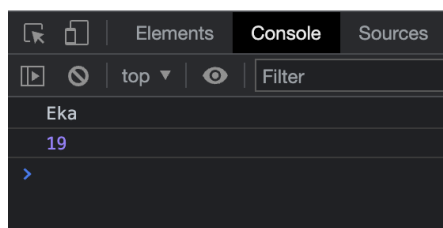
    let {nama: namaMahasiswa, umur: umurMahasiswa} =
dataMahasiswa();

    console.log(namaMahasiswa);
    console.log(umurMahasiswa);
  </script>
</body>
</html>
```

### ii. Hasil



# Belajar Javascript



iii. Penjelasan

1. Objek tersebut kemudian di-destrukturisasi pada baris selanjutnya menggunakan syntax {nama: namaMahasiswa, umur: umurMahasiswa}. Ini berarti bahwa nilai dari properti nama akan disimpan dalam variabel namaMahasiswa, dan nilai dari properti umur akan disimpan dalam variabel umurMahasiswa.

## h. Array Method: `forEach()` dan `map()`

### i. Kode Program

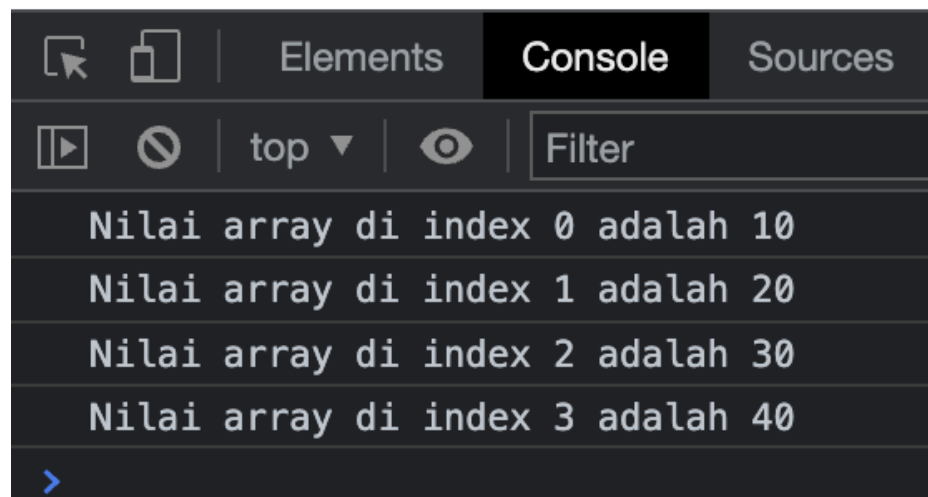
foreach.html

```
<!-- *** Author : Akhmad Ramadani *** -->
<!-- *** NIM : 2041720002 *** -->
<!-- *** Kelas : TI-3B *** -->

<!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>React Uncover</title>
</head>
<body>
  <h1>Belajar Javascript</h1>
  <script>
    let arr = [10, 20, 30, 40];

    arr.forEach(function (val, key){
      console.log(`Nilai array di index ${key} adalah
${val}`);
    })
  </script>
</body>
</html>
```

### ii. Hasil



### iii. Penjelasan

1. Method `forEach()` digunakan untuk melakukan iterasi pada setiap elemen pada array, kemudian menjalankan fungsi callback yang diberikan pada setiap elemen tersebut.
2. Dalam kode di atas, fungsi callback yang diberikan menggunakan `function(val, key)` yang akan menerima dua



parameter. Parameter `val` akan merepresentasikan nilai elemen pada array, sedangkan parameter `key` akan merepresentasikan index elemen pada array.

## i. Array Method: `forEach()` dan `map()`

### i. Kode Program

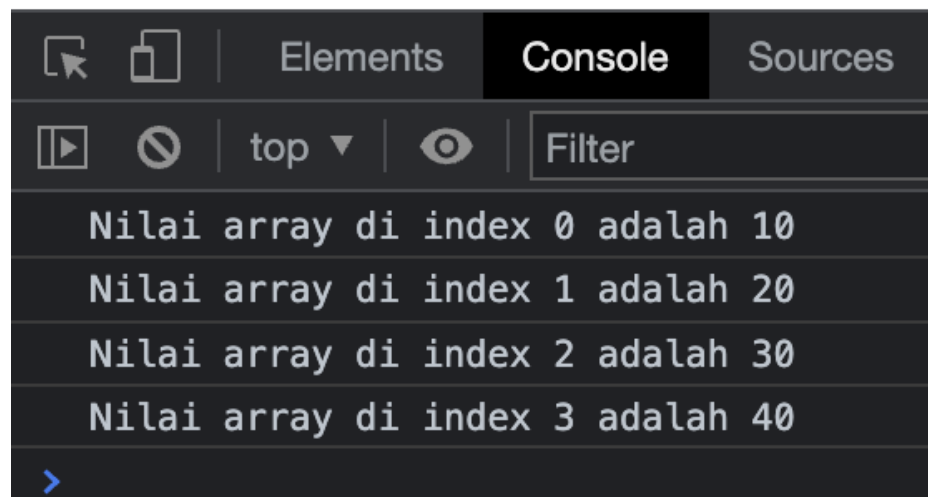
foreach2.html

```
<!-- *** Author : Akhmad Ramadani *** -->
<!-- *** NIM : 2041720002 *** -->
<!-- *** Kelas : TI-3B *** -->

<!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>React Uncover</title>
</head>
<body>
  <h1>Belajar Javascript</h1>
  <script>
    let arr = [10, 20, 30, 40];

    arr.forEach((val, key)=>
      console.log(`Nilai array di index ${key} adalah
${val}`)
    )
  </script>
</body>
</html>
```

### ii. Hasil



### iii. Penjelasan

1. Penjelasan sama dengan kode di atas. Namun, kali ini `forEach` menggunakan `arrow function`, yang memiliki fungsi sama dengan kode sebelumnya.

j. **Array Method: forEach() dan map()**

i. Kode Program

foreach3.html

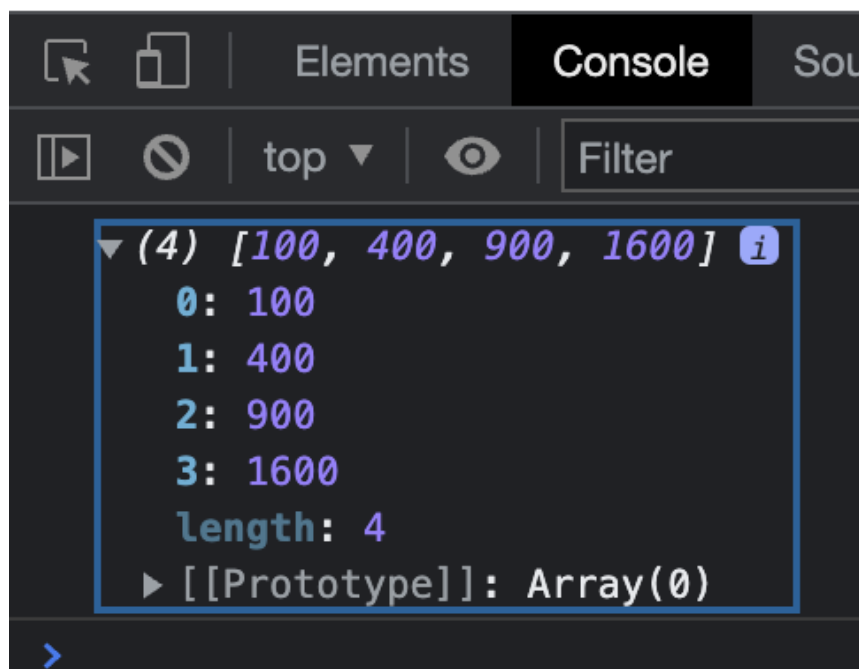
```
<!-- *** Author : Akhmad Ramadani *** -->
<!-- *** NIM      : 2041720002 *** -->
<!-- *** Kelas   : TI-3B *** -->

<!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>React Uncover</title>
</head>
<body>
  <h1>Belajar Javascript</h1>
  <script>
    let arr = [10, 20, 30, 40];
    arrKuadrat = [];

    arr.forEach((val) => {
      arrKuadrat.push(val * val);
    })

    console.log(arrKuadrat);
  </script>
</body>
</html>
```

ii. Hasil



iii. Penjelasan

1. Script di atas mendefinisikan sebuah array `arr` yang terdiri dari empat elemen. Kemudian, sebuah array kosong `arrKuadrat` didefinisikan. Selanjutnya, menggunakan metode `forEach()`, setiap elemen dari array `arr` dikalikan dengan dirinya sendiri dan hasilnya dimasukkan ke dalam array `arrKuadrat`.

## k. Array Method: `forEach()` dan `map()`

### i. Kode Program

map.html

```
<!-- *** Author : Akhmad Ramadani *** -->
<!-- *** NIM : 2041720002 *** -->
<!-- *** Kelas : TI-3B *** -->

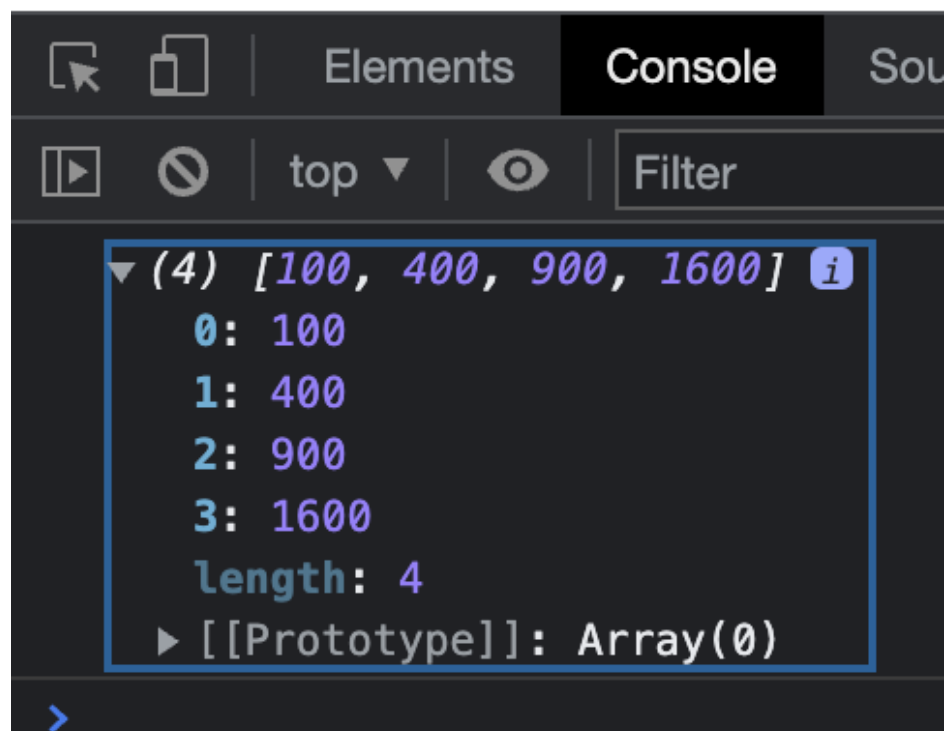
<!DOCTYPE html>
<html lang="en">

<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>React Uncover</title>
</head>

<body>
  <h1>Belajar Javascript</h1>
  <script>
    let arr = [10, 20, 30, 40];
    arrKuadrat = arr.map((val) => val * val);
    console.log(arrKuadrat);
  </script>
</body>

</html>
```

### ii. Hasil



iii. Penjelasan

1. Fungsi `map()` digunakan untuk mengubah setiap elemen dalam array tanpa mengubah ukuran array itu sendiri. Di sini, `map()` digunakan untuk mengembalikan setiap elemen array yang sudah dihitung kuadratnya, lalu disimpan ke dalam array `arrKuadrat`.

## I. Array Method: `forEach()` dan `map()`

### i. Kode Program

map2.html

```
<!-- *** Author : Akhmad Ramadani *** -->
<!-- *** NIM      : 2041720002 *** -->
<!-- *** Kelas   : TI-3B   *** -->

<!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>React Uncover</title>
</head>

<body>
  <h1>Belajar Javascript</h1>
  <script>
    const mahasiswa = [
      {
        nama: "Eka",
        umur: 19,
        jurusan: "Teknik Informatika",
      },
      {
        nama: "Lisa",
        umur: 18,
        jurusan: "Sistem Informasi",
      },
      {
        nama: "Rudi",
        umur: 19,
        jurusan: "Teknik Elektro",
      }
    ];

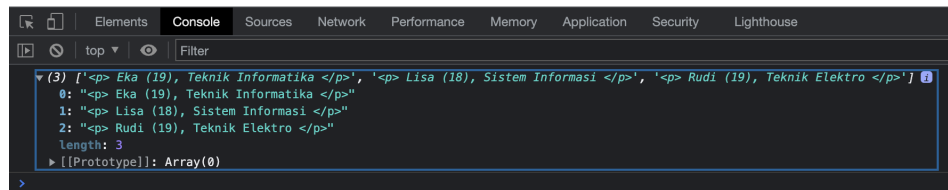
    const prosesMahasiswa = (mahasiswa) =>
      `<p> ${mahasiswa.nama} (${mahasiswa.umur}),
${mahasiswa.jurusan} </p>`;

    const formatMahasiswa =
(mahasiswa).map(prosesMahasiswa);

    console.log(formatMahasiswa);
  </script>
</body>

</html>
```

## ii. Hasil



```
(3) ['<p> Eka (19), Teknik Informatika </p>', '<p> Lisa (18), Sistem Informasi </p>', '<p> Rudi (19), Teknik Elektro </p>']  
0: "<p> Eka (19), Teknik Informatika </p>"  
1: "<p> Lisa (18), Sistem Informasi </p>"  
2: "<p> Rudi (19), Teknik Elektro </p>"  
length: 3  
__proto__: Array(0)
```

## iii. Penjelasan

1. Pada contoh di atas, setiap elemen array mahasiswa berupa objek dengan atribut nama, umur, dan jurusan. Kemudian, sebuah fungsi bernama prosesMahasiswa didefinisikan untuk memformat tiap objek mahasiswa menjadi sebuah string dengan pola tertentu. Fungsi tersebut kemudian dipanggil menggunakan method map() pada array mahasiswa, sehingga setiap objek pada array diolah menggunakan fungsi prosesMahasiswa, dan hasilnya disimpan pada array formatMahasiswa.



### m. Array Method: `forEach()` dan `map()`

#### i. Kode Program

map3.html

```
<!-- *** Author : Akhmad Ramadani *** -->
<!-- *** NIM      : 2041720002 *** -->
<!-- *** Kelas   : TI-3B *** -->

<!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>React Uncover</title>
</head>

<body>
  <h1>Belajar Javascript</h1>
  <div id="root"></div>
  <button onclick="renderData()">Tampilkan Data
Mahasiswa</button>
  <script>
    const mahasiswa = [
      {
        nama: "Eka",
        umur: 19,
        jurusan: "Teknik Informatika",
      },
      {
        nama: "Lisa",
        umur: 18,
        jurusan: "Sistem Informasi",
      },
      {
        nama: "Rudi",
        umur: 19,
        jurusan: "Teknik Elektro",
      }
    ];

    const prosesMahasiswa = (mahasiswa) =>
      `<p> ${mahasiswa.nama} (${mahasiswa.umur} tahun)
- <i>${mahasiswa.jurusan}</i> </p>`;

    const formatMahasiswa =
(mahasiswa).map(prosesMahasiswa);

    const renderData = () => {
      document.getElementById('root').innerHTML =
formatMahasiswa.join('');
    }
  </script>
</body>

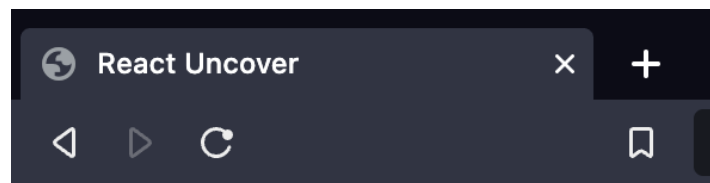
</html>
```

ii. Hasil



# Belajar Javascript

Tampilkan Data Mahasiswa



# Belajar Javascript

Eka (19 tahun) - *Teknik Informatika*

Lisa (18 tahun) - *Sistem Informasi*

Rudi (19 tahun) - *Teknik Elektro*

Tampilkan Data Mahasiswa

iii. Penjelasan

1. Pada saat halaman web dibuka, terdapat sebuah button yang memiliki event listener onclick yang memanggil function `renderData()`. Function `renderData()` kemudian akan mengeksekusi kode yang mengubah isi dari elemen dengan id root pada halaman web dengan menggunakan method `innerHTML`, sehingga data mahasiswa akan ditampilkan pada halaman web tersebut.

## n. Asynchronous JavaScript (Promise / Async - Await)

### i. Kode Program

async.html

```
<!-- *** Author : Akhmad Ramadani *** -->
<!-- *** NIM    : 2041720002 *** -->
<!-- *** Kelas  : TI-3B   *** -->

<!DOCTYPE html>
<html lang="en">

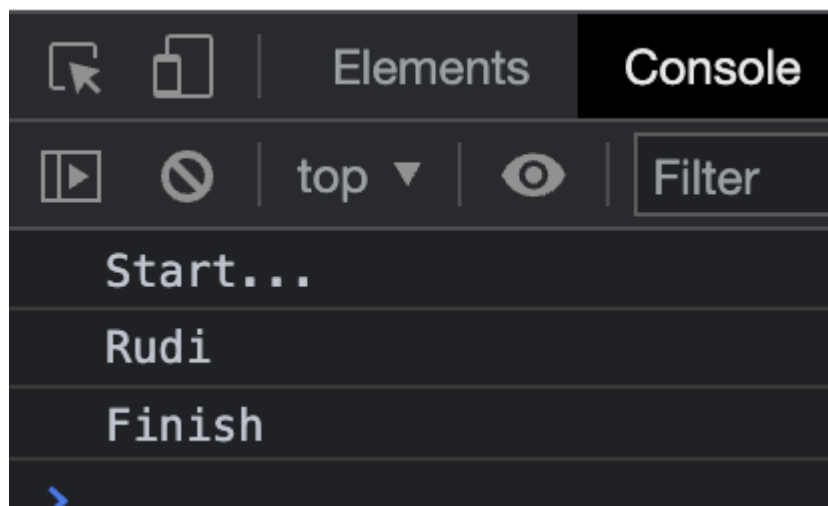
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible"
content="IE=edge">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>React Uncover</title>
</head>

<body>
  <h1>Belajar Javascript</h1>
  <script>
    const getUser = () => "Rudi";

    console.log("Start...");
    console.log(getUser());
    console.log("Finish");
  </script>
</body>

</html>
```

### ii. Hasil



### iii. Penjelasan

1. Script di atas menunjukkan sebuah fungsi JavaScript `getUser()` yang mengembalikan sebuah string "Rudi" yang akan dicetak setelah "Start..."

## o. Asynchronous JavaScript (Promise / Async - Await)

### i. Kode Program

async2.html

```
<!-- *** Author : Akhmad Ramadani *** -->
<!-- *** NIM : 2041720002 *** -->
<!-- *** Kelas : TI-3B *** -->

<!DOCTYPE html>
<html lang="en">

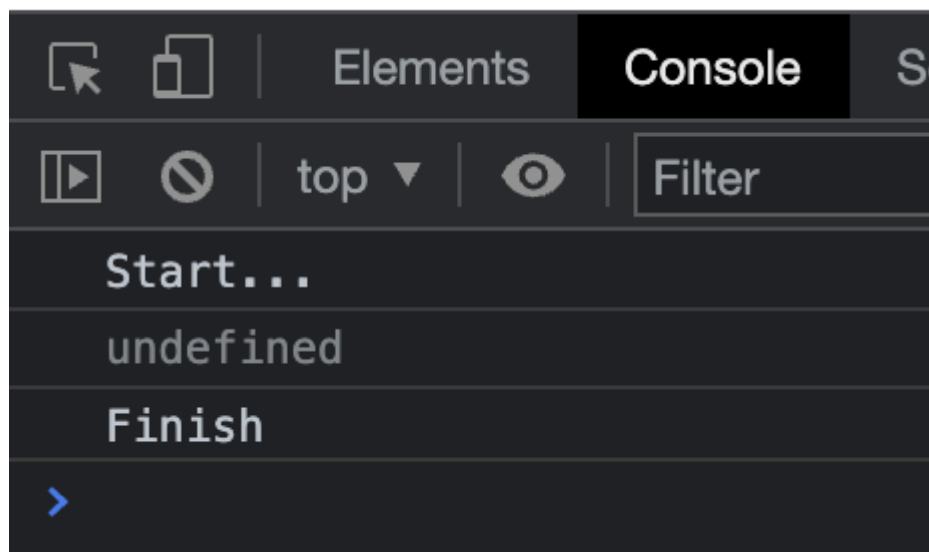
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible"
content="IE=edge">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>React Uncover</title>
</head>

<body>
  <h1>Belajar Javascript</h1>
  <script>
    const getUser = () => {
      setTimeout(() => {
        return "Rudi";
      }, 2000);
    }

    console.log("Start...");
    console.log(getUser());
    console.log("Finish");
  </script>
</body>

</html>
```

### ii. Hasil



iii. Penjelasan

1. Pada kode tersebut, fungsi `getUser` menggunakan `setTimeout` yang berfungsi untuk menunda eksekusi fungsi selama 2 detik. Namun, karena `setTimeout` adalah operasi asinkronus, fungsi `getUser` tidak mengembalikan nilai apa pun dalam waktu 2 detik. Oleh karena itu, ketika `getUser()` dipanggil di baris ke-8, itu akan mengembalikan nilai `undefined`.

**p. Asynchronous JavaScript (Promise / Async - Await)**

**i. Kode Program**

async3.html

```
<!-- *** Author : Akhmad Ramadani *** -->
<!-- *** NIM : 2041720002 *** -->
<!-- *** Kelas : TI-3B *** -->

<!DOCTYPE html>
<html lang="en">

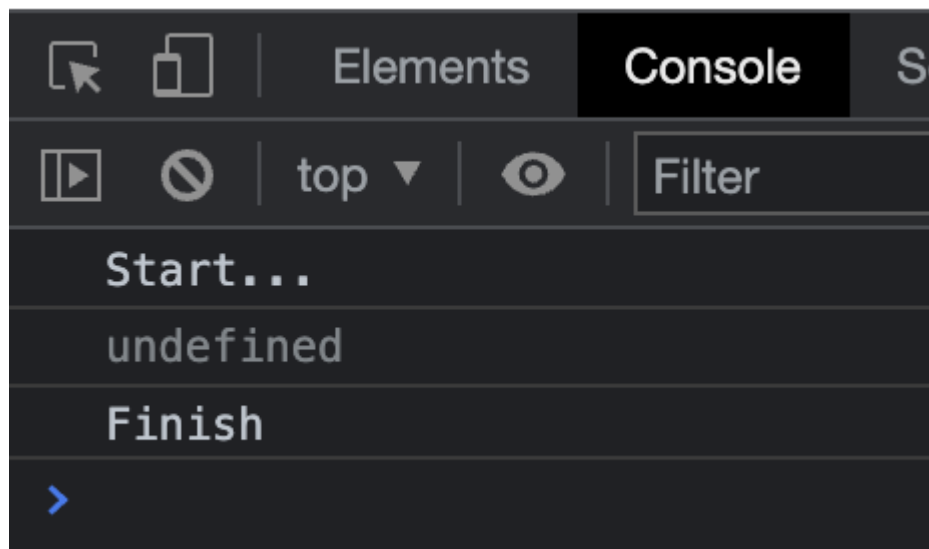
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible"
content="IE=edge">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>React Uncover</title>
</head>

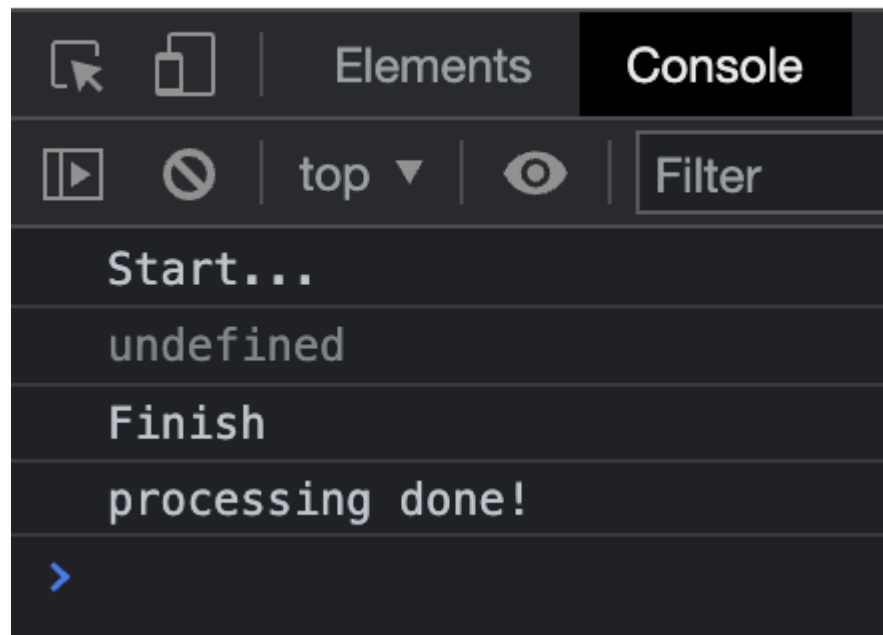
<body>
  <h1>Belajar Javascript</h1>
  <script>
    const getUser = () => {
      setTimeout(() => {
        console.log("processing done!");
        return "Rudi";
      }, 2000);
    }

    console.log("Start...");
    console.log(getUser());
    console.log("Finish");
  </script>
</body>

</html>
```

**ii. Hasil**





iii. Penjelasan

1. Pada kode di atas, terdapat sebuah fungsi bernama `getUser` yang menggunakan `setTimeout` untuk menunda eksekusi sebanyak 2 detik. Setelah 2 detik, fungsi akan mengeksekusi callback function yang berisi perintah untuk mencetak teks "processing done!" dan mengembalikan nilai "Rudi".
2. Ketika kode tersebut dijalankan, pertama-tama akan dicetak teks "Start..." pada console. Kemudian, fungsi `getUser` dipanggil dan mengembalikan `undefined` karena callback function di dalam `setTimeout` dijalankan secara asynchronous dan tidak langsung mengembalikan nilai. Selanjutnya, teks "Finish" dicetak pada console.
3. Setelah 2 detik, callback function di dalam `setTimeout` dijalankan dan mencetak teks "processing done!" pada console. Namun, nilai "Rudi" yang dihasilkan tidak ditampilkan pada console karena pada saat fungsi `getUser` dipanggil, nilai tersebut belum tersedia.

## q. JavaScript Promise

### i. Kode Program

promise.html

```
<!-- *** Author : Akhmad Ramadani *** -->
<!-- *** NIM    : 2041720002 *** -->
<!-- *** Kelas  : TI-3B *** -->

<!DOCTYPE html>
<html lang="en">

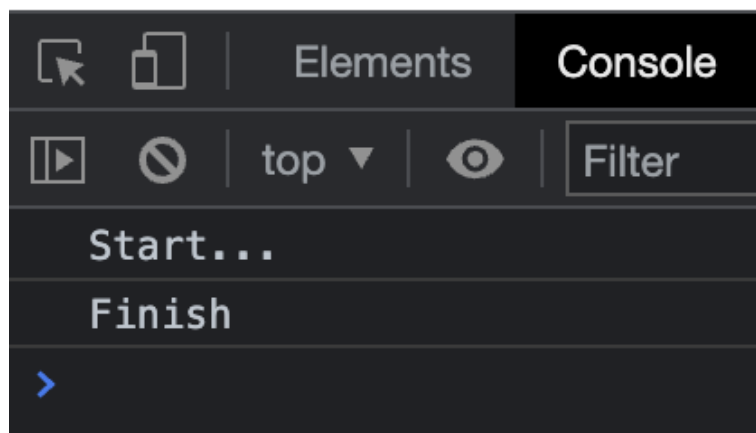
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible"
content="IE=edge">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>React Uncover</title>
</head>

<body>
  <h1>Belajar Javascript</h1>
  <script>
    const getUser = () => {
      return new Promise((resolve, ) => {
        setTimeout(() => {
          resolve("Rudi");
        }, 2000);
      });
    }

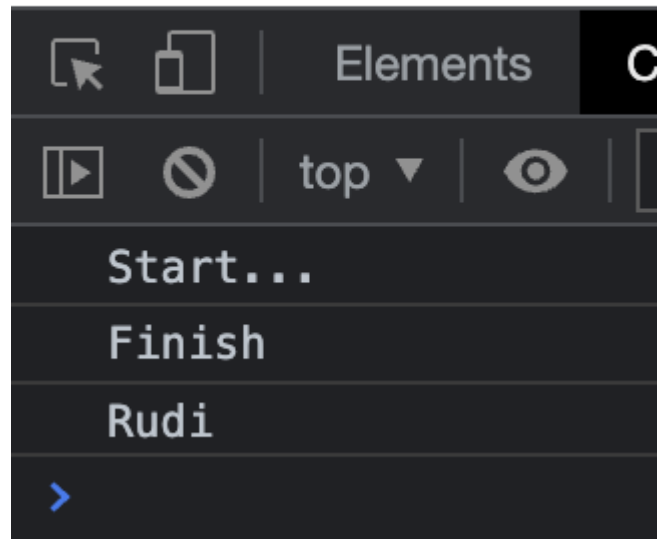
    console.log("Start...");
    getUser().then((username) => {
      console.log(username);
    });
    console.log("Finish");
  </script>
</body>

</html>
```

### ii. Hasil







iii. Penjelasan

1. Pada kode di atas, terdapat sebuah fungsi `getUser` yang mengembalikan sebuah `Promise` yang di dalamnya terdapat sebuah `setTimeout` yang akan menyelesaikan `Promise` setelah 2 detik dengan mengembalikan string `"Rudi"`.
2. Pada `console.log` pertama, `"Start..."` akan ditampilkan terlebih dahulu. Kemudian, `getUser()` akan dipanggil dan diikuti dengan sebuah `.then()` yang akan menunggu `Promise` yang dikembalikan oleh `getUser()` selesai. Ketika `Promise` selesai, `then()` akan dieksekusi dan menampilkan nilai yang dihasilkan oleh `Promise`, yaitu string `"Rudi"`.
3. Setelah itu, `console.log` terakhir `"Finish"` akan ditampilkan. Oleh karena itu, urutan tampilan di console adalah `"Start..."`, `"Finish"`, kemudian `"Rudi"`.

## r. JavaScript Promise

### i. Kode Program

promise2.html

```
<!-- *** Author : Akhmad Ramadani *** -->
<!-- *** NIM    : 2041720002 *** -->
<!-- *** Kelas  : TI-3B   *** -->

<!DOCTYPE html>
<html lang="en">

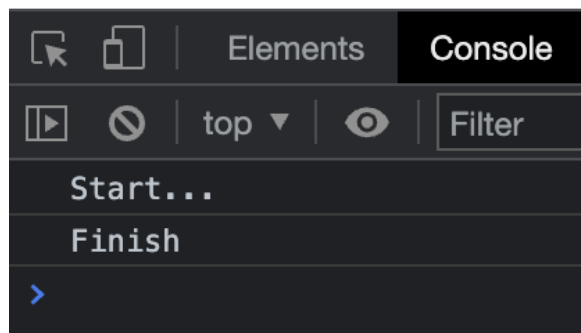
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible"
content="IE=edge">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>React Uncover</title>
</head>

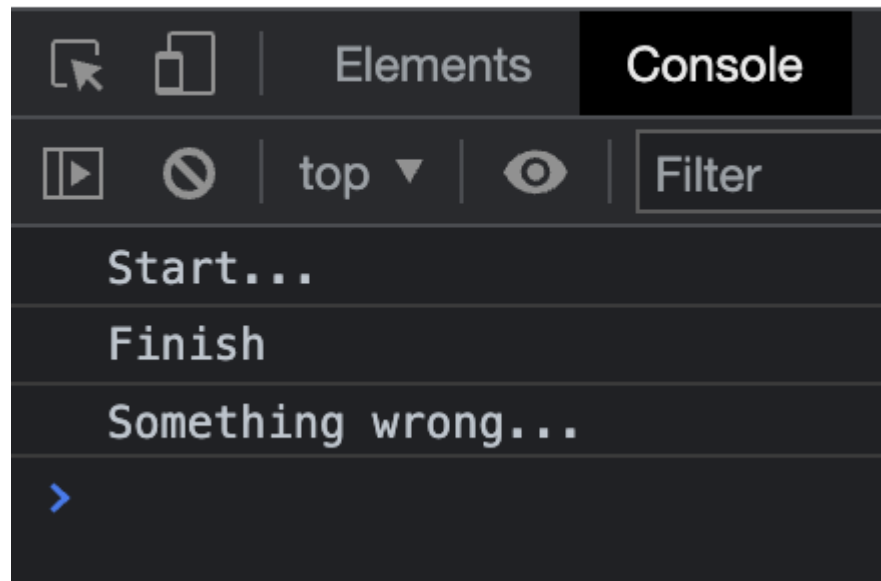
<body>
  <h1>Belajar Javascript</h1>
  <script>
    const getUser = () => {
      return new Promise((resolve, reject) => {
        setTimeout(() => {
          // resolve("Rudi");
          reject("Something wrong...")
        }, 2000);
      });
    }

    console.log("Start...");
    getUser().then((username) => {
      console.log(username);
    }).catch((error) => {
      console.log(error);
    });
    console.log("Finish");
  </script>
</body>

</html>
```

### ii. Hasil





iii. Penjelasan

1. Pada script tersebut, terdapat sebuah fungsi bernama `getUser` yang mengembalikan sebuah promise. Promise ini akan menyelesaikan proses setelah 2 detik menggunakan fungsi `setTimeout`. Apabila proses telah selesai, maka promise akan mengembalikan nilai "Rudi" melalui fungsi `resolve`. Namun, jika terjadi kesalahan, promise akan menolak (`reject`) dan mengirimkan pesan "Something wrong...".

## s. Async - Await

### i. Kode Program

async\_await.html

```
<!-- *** Author : Akhmad Ramadani *** -->
<!-- *** NIM : 2041720002 *** -->
<!-- *** Kelas : TI-3B *** -->

<!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>React Uncover</title>
</head>

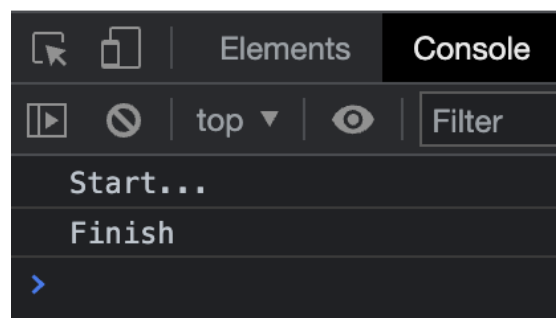
<body>
  <h1>Belajar Javascript</h1>
  <script>
    const getUser = () => {
      return new Promise((resolve, reject) => {
        setTimeout(() => {
          resolve("Rudi");
          reject("Something wrong...")
        }, 2000);
      });
    }

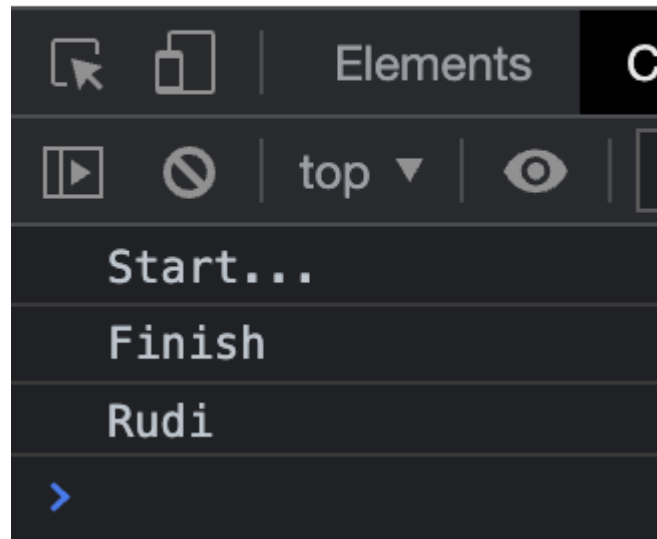
    const tryGetName = async () => {
      let username = await getUser();
      console.log(username);
    }

    console.log("Start...");
    tryGetName();
    console.log("Finish");
  </script>
</body>

</html>
```

### ii. Hasil





iii. Penjelasan

1. Pada script tersebut terdapat sebuah fungsi `getUser` yang mengembalikan Promise yang di dalamnya terdapat `setTimeout` yang menunda resolve Promise selama 2 detik, kemudian pada blok kode utama terdapat sebuah fungsi `tryGetName` yang menggunakan `async/await` untuk menunggu Promise `getUser` resolve, lalu mencetak nilai resolve tersebut pada console.

## t. Async - Await

### i. Kode Program

async\_await2.html

```
<!-- *** Author : Akhmad Ramadani *** -->
<!-- *** NIM      : 2041720002 *** -->
<!-- *** Kelas   : TI-3B   *** -->

<!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>React Uncover</title>
</head>

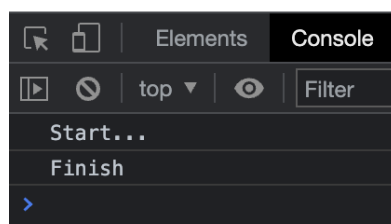
<body>
  <h1>Belajar Javascript</h1>
  <script>
    const getUser = () => {
      return new Promise((resolve, reject) => {
        setTimeout(() => {
          // resolve("Rudi");
          reject("Something wrong...")
        }, 2000);
      });
    }

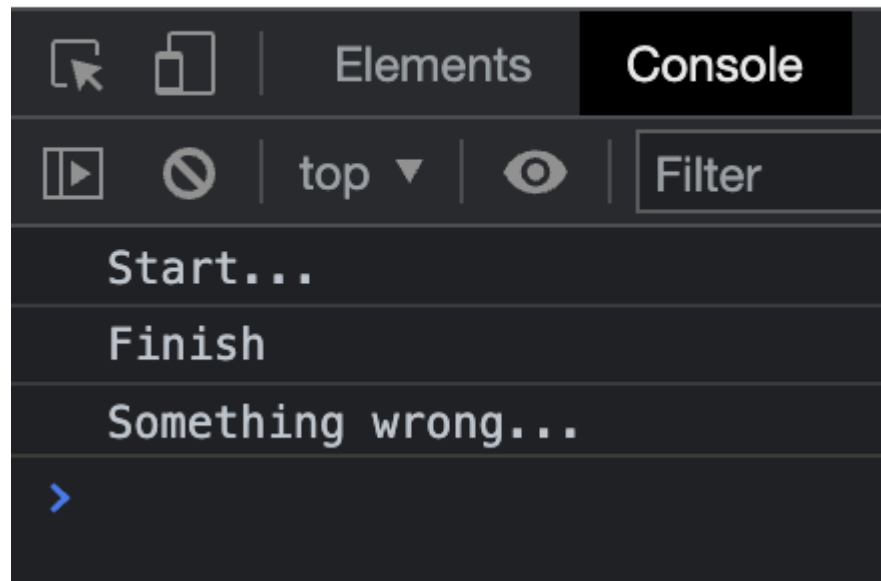
    const tryGetName = async () => {
      try {
        let username = await getUser();
        console.log(username);
      } catch (error) {
        console.log(error);
      }
    }

    console.log("Start...");
    tryGetName();
    console.log("Finish");
  </script>
</body>

</html>
```

### ii. Hasil





iii. Penjelasan

1. Pada script di atas, terdapat sebuah function `getUser` yang mengembalikan sebuah Promise. Promise tersebut akan menunggu selama 2 detik, lalu menolak (reject) dengan pesan "Something wrong...".
2. Kemudian, terdapat sebuah function `tryGetName` yang merupakan sebuah async function. Function ini memanggil `getUser` dengan menggunakan `await`, yang akan membuat JavaScript menunggu hingga Promise tersebut diselesaikan. Jika Promise berhasil dipenuhi (resolve), maka nilai yang dihasilkan akan ditampung pada variabel `username` dan dicetak pada console. Jika Promise menolak (reject), maka pesan error akan dicetak pada console.