

TUGAS PERTEMUAN 1

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. Membuat variable dengan let

i. Kode Program

var_let.html

```
<!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 (ES6)</h1>
  <script>
    var foo = "Belajar ECMAScript 6";
    console.log(foo);

    foo = "React Uncover";
    console.log(foo);

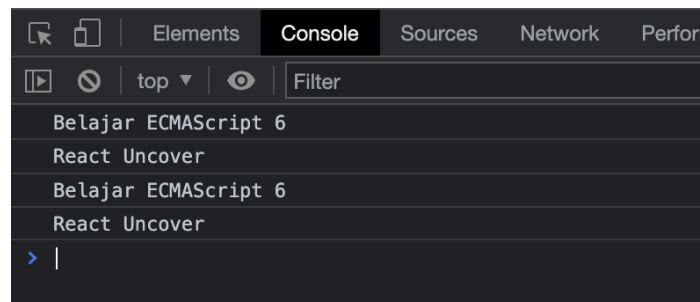
    let bar = "Belajar ECMAScript 6";
    console.log(bar);

    bar = "React Uncover";
    console.log(bar);
  </script>
</body>
</html>
```

ii. Hasil



Belajar Javascript (ES6)



iii. Penjelasan

- Blok kode pertama dimulai dari setelah tag <script>
- Pada baris pertama, kita mendeklarasikan sebuah variabel bernama foo dan memberi nilainya dengan string "Belajar ECMAScript 6". Kemudian, kita mencetak nilai variabel tersebut ke konsol menggunakan console.log.
- Pada baris kedua, kita mengubah nilai variabel foo menjadi string "React Uncover" dan mencetaknya kembali ke konsol.
- Selanjutnya, pada baris ketiga, kita mendeklarasikan variabel lain bernama bar menggunakan kata kunci let. Variabel bar diberi nilai string yang sama dengan foo, yaitu "Belajar ECMAScript 6". Kemudian, kita mencetak nilai variabel bar ke konsol.
- Pada baris keempat, kita mengubah nilai variabel bar menjadi string "React Uncover" dan mencetaknya kembali ke konsol.

b. Perbedaan antara let dan var

i. Kode Program

var_let2.html

```
<!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>

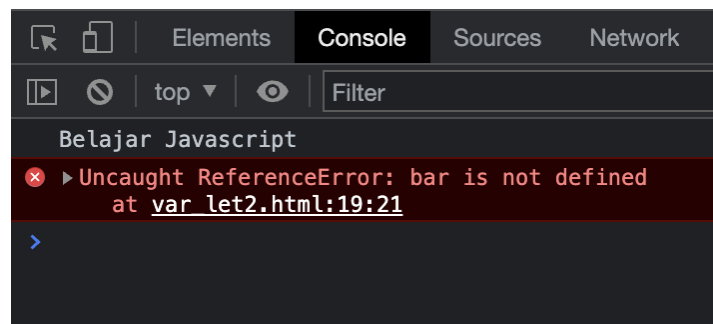
<body>
  <h1>Belajar Javascript (ES6)</h1>
  <script>
    {
      var foo = "Belajar Javascript";
      let bar = "Belajar React";
    }
    console.log(foo);
    console.log(bar);
  </script>
</body>

</html>
```

ii. Hasil



Belajar Javascript (ES6)



iii. Penjelasan

- Blok kode pertama dimulai dari setelah tag `<script>`
- Pada blok kode pertama, kita mendeklarasikan dua variabel yaitu `foo` dan `bar`. Variabel `foo` dideklarasikan dengan kata kunci `var` dan diberi nilai string "Belajar Javascript". Variabel `bar` dideklarasikan dengan kata kunci `let` dan diberi nilai string "Belajar React". Blok kode tersebut dikelilingi oleh kurung `{}` yang menunjukkan sebuah blok kode.
- Pada blok kode berikutnya, kita mencoba mencetak nilai dari variabel `foo` dan `bar` ke konsol menggunakan `console.log`. Namun, kita akan mendapatkan error pada baris `console.log(bar)` karena variabel `bar` hanya dideklarasikan di dalam blok kode sebelumnya dan tidak dapat diakses di luar blok kode tersebut. Variabel `foo`, yang dideklarasikan dengan kata kunci `var`, dapat diakses di seluruh kode setelah deklarasi variabel tersebut.
- Dalam JavaScript, variabel yang dideklarasikan dengan kata kunci `let` atau `const` hanya dapat diakses di dalam blok kode tempat variabel tersebut dideklarasikan. Sementara itu, variabel yang dideklarasikan dengan kata kunci `var` memiliki cakupan (scope) yang lebih luas dan dapat diakses di seluruh fungsi atau blok kode setelah deklarasi variabel tersebut.

c. Perbedaan antara let dan var 2

i. Kode Program Menggunakan Var

var_let3.html

```
<!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>

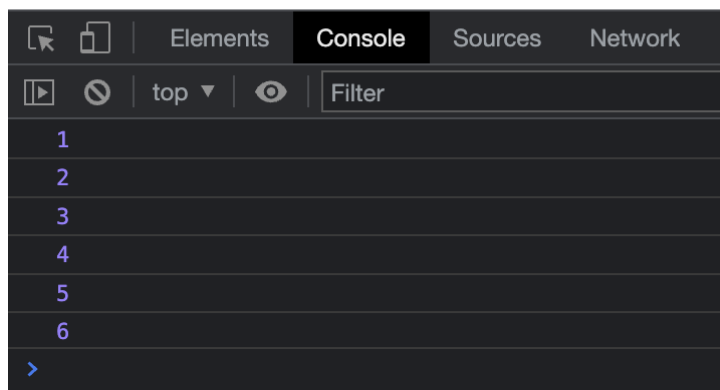
<body>
  <h1>Belajar Javascript (ES6)</h1>
  <script>
    var i = "Belajar Javascript";
    for (var i = 1; i <= 5; i++) {
      console.log(i);
    }
    console.log(i);
  </script>
</body>

</html>
```

ii. Hasil



Belajar Javascript (ES6)



iii. Kode Program Menggunakan Let

```
var_let4.html

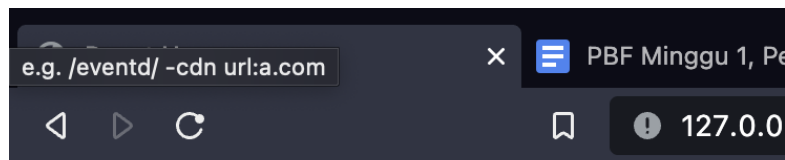
<!DOCTYPE html>
<html lang="en">

<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>

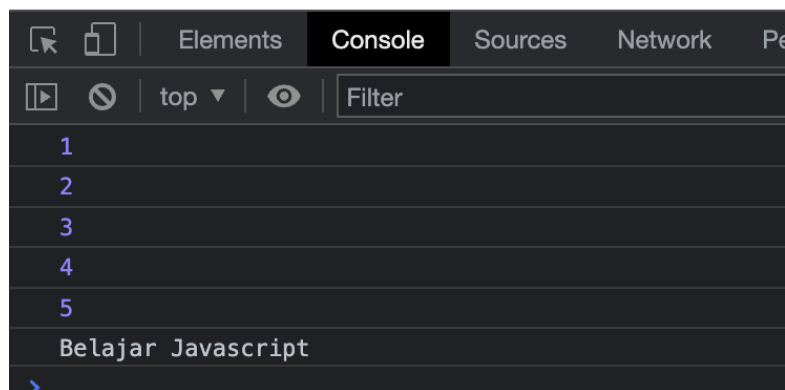
<body>
  <h1>Belajar Javascript (ES6)</h1>
  <script>
    let i = "Belajar Javascript";
    for (let i = 1; i <= 5; i++) {
      console.log(i);
    }
    console.log(i);
  </script>
</body>

</html>
```

iv. Hasil



Belajar Javascript (ES6)



v. **Penjelasan**

- Dalam kode di atas, variabel yang dideklarasikan dengan kata kunci `let` atau `const` hanya dapat diakses di dalam blok kode tempat variabel tersebut dideklarasikan. Sementara itu, variabel yang dideklarasikan dengan kata kunci `var` memiliki cakupan (scope) yang lebih luas dan dapat diakses di seluruh fungsi atau blok kode setelah deklarasi variabel tersebut.

d. **Template String**

i. **Kode Program**

interpolation.html

```
<!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 (ES6)</h1>
  <script>
    const foo = "Javascript";
    const bar = "React";

    console.log("Belajar " + foo + " dan " + bar);

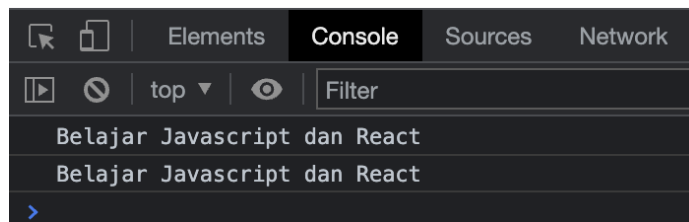
    console.log(`Belajar ${foo} dan ${bar}`);
  </script>
</body>

</html>
```

ii. **Hasil**



Belajar Javascript (ES6)



iii. Penjelasan

- Pada baris pertama, kita mendeklarasikan sebuah konstanta bernama foo dan memberikan nilai string "Javascript". Pada baris kedua, kita mendeklarasikan konstanta lain bernama bar dan memberikan nilai string "React".
- Pada baris ketiga, kita mencetak pesan ke konsol dengan menggabungkan tiga string menggunakan operator +. String pertama adalah "Belajar ", kemudian diikuti dengan nilai dari variabel foo dan bar, masing-masing dipisahkan dengan string " dan ". Output dari baris ini adalah "Belajar Javascript dan React".
- Pada baris keempat, kita mencetak pesan yang sama ke konsol menggunakan string template (template literals). String template dibuat dengan menggunakan tanda backtick (`) sebagai pembatas dan memungkinkan kita untuk menyisipkan variabel langsung ke dalam string dengan menggunakan sintaks `${namaVariabel}`. Output dari baris ini juga adalah "Belajar Javascript dan React".
- Menggunakan string template memudahkan kita untuk membuat string yang kompleks dan memungkinkan kita untuk menggabungkan variabel secara langsung ke dalam string tanpa perlu menggabungkan beberapa string menggunakan operator +.

e. Template String Expression

i. Kode Program

interpolation2.html

```
<!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>

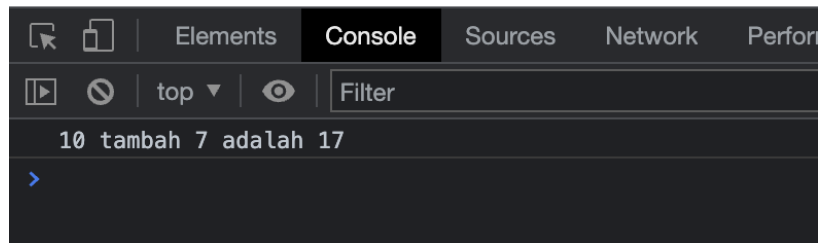
<body>
  <h1>Belajar Javascript (ES6)</h1>
  <script>
    console.log(`10 tambah 7 adalah ${10 + 7}`);
  </script>
</body>

</html>
```

ii. Hasil



Belajar Javascript (ES6)



iii. Penjelasan

- Template string juga bisa digunakan untuk menjumlahkan menggunakan sintaks `${10 + 7}` untuk menyisipkan hasil penjumlahan langsung ke dalam string.

f. Template string function

i. Kode Program

```
interpolation3.html

<!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 (ES6)</h1>
  <script>
    function kuadrat(a) {
      return a * a;
    }

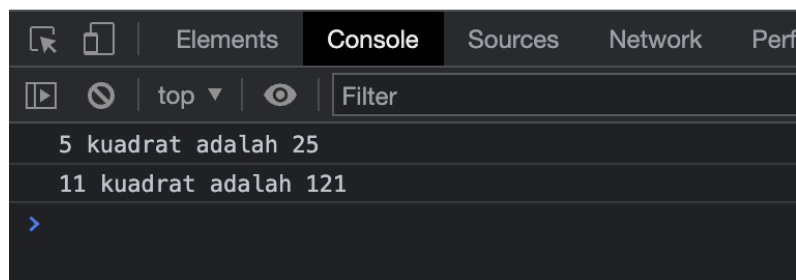
    console.log(`5 kuadrat adalah ${kuadrat(5)}`);
    console.log(`11 kuadrat adalah ${kuadrat(11)}`);
  </script>
</body>

</html>
```

ii. Hasil



Belajar Javascript (ES6)



iii. Penjelasan

- Pada baris kedua dan ketiga, menggunakan fungsi kuadrat yang telah didefinisikan sebelumnya untuk mencetak pesan ke konsol yang berisi hasil kuadrat dari angka 5 dan 11 masing-masing. Kita menggunakan sintaks `${kuadrat(5)}` dan `${kuadrat(11)}` untuk menyisipkan hasil pemangkatan langsung ke dalam string. Output dari baris ini adalah "5 kuadrat adalah 25" dan "11 kuadrat adalah 121".

g. Arrow Notation

i. Kode Program

```
arrow_notation.html

<!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 (ES6)</h1>
  <script>
    const kuadrat = (x) => {
      return x * x;
    };

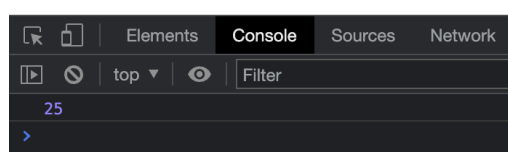
    console.log(kuadrat(5));
  </script>
</body>

</html>
```

ii. Hasil



Belajar Javascript (ES6)



iii. Penjelasan

- Fungsi arrow dinamakan kuadrat dan menerima sebuah parameter x. Di dalam fungsi, kita mengembalikan hasil dari x dipangkatkan dua.

h. Arrow Notation 2

i. Kode Program

arrow_notation2.html

```
<!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 (ES6)</h1>
  <script>
    const generateRandom = () =>
Math.floor(Math.random() * 10) + 1;

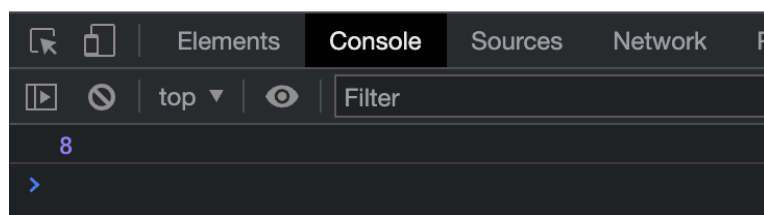
    console.log(generateRandom());
  </script>
</body>

</html>
```

ii. Hasil



Belajar Javascript (ES6)



iii. Penjelasan

- Fungsi arrow pada contoh ini dinamakan generateRandom. Fungsi ini langsung mengembalikan nilai dari `Math.floor(Math.random() * 10) + 1;`

i. Arrow Notation 3

i. Kode Program

```
arrow_notation3.html

<!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 (ES6)</h1>
  <script>
    const generateRandom = (start = 1, end = 10) =>
Math.floor(Math.random() * (end - start + 1)) + start;

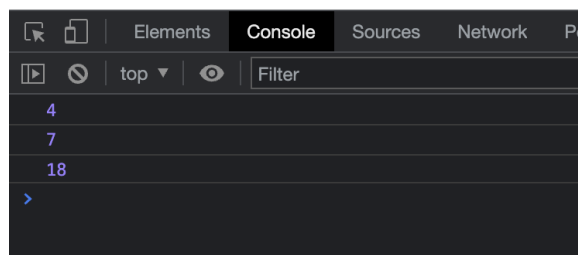
    console.log(generateRandom());
    console.log(generateRandom(5));
    console.log(generateRandom(10, 20));
  </script>
</body>

</html>
```

ii. Hasil



Belajar Javascript (ES6)



iii. Penjelasan

- Pada baris pertama, kita mendefinisikan sebuah fungsi arrow dengan nama generateRandom yang menerima dua parameter opsional yaitu start dan end. Fungsi ini menghasilkan sebuah bilangan bulat acak (random integer) antara start dan end menggunakan objek Math pada JavaScript.

j. Class dan object

i. Kode Program

class_object.html

```
<!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 (ES6)</h1>
  <script>
    class Laptop {
      constructor(merk, milik) {
        this.merk = merk;
        this.milik = milik;
      }

      hidupkanLaptop() {
        console.log(`Hidupkan laptop
${this.merk} milik ${this.milik}`);
      }
    }

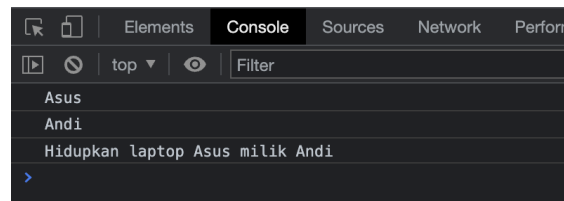
    const laptopAndi = new Laptop("Asus", "Andi");
    console.log(laptopAndi.merk);
    console.log(laptopAndi.milik);
    laptopAndi.hidupkanLaptop();
  </script>
</body>

</html>
```

ii. Hasil



Belajar Javascript (ES6)



iii. Penjelasan

- Pada baris pertama, kita mendefinisikan sebuah kelas (class) dengan nama Laptop. Kelas ini memiliki dua properti yaitu merk dan milik, serta satu method yaitu hidupkanLaptop. Properti merk dan milik didefinisikan dalam constructor yang diinisialisasi dengan nilai yang diberikan pada saat pembuatan objek.
- Pada baris ketiga, method hidupkanLaptop diimplementasikan untuk mencetak pesan pada console yang menyatakan bahwa laptop dengan merk dan milik tertentu telah dinyalakan.
- Pada baris keenam, kita membuat sebuah objek baru dengan menggunakan kelas Laptop dan menyimpannya pada variabel laptopAndi. Objek ini memiliki nilai merk "Asus" dan nilai milik "Andi".
- Pada baris ketujuh dan kedelapan, kita mencetak nilai dari properti merk dan milik dari objek laptopAndi.
- Pada baris kesembilan, kita memanggil method hidupkanLaptop dari objek laptopAndi, sehingga akan tercetak pesan "Hidupkan laptop Asus milik Andi" pada console.

k. Class dan object 2

i. Kode Program

```
class_object2.html

<!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>

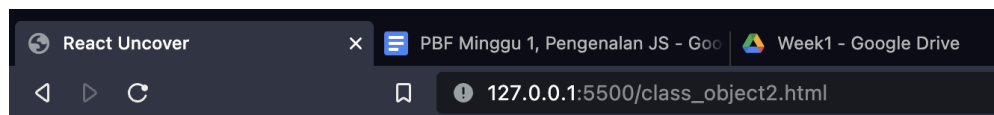
<body>
  <h1>Belajar Javascript (ES6)</h1>
  <script>
    let laptopRama = {
      milik: "Rama",
      merek: "Asus",
      foo () {
        return this
      },
      bar: () => this,
    }

    console.log(laptopRama.foo());
    console.log(laptopRama.bar());
  </script>
</body>

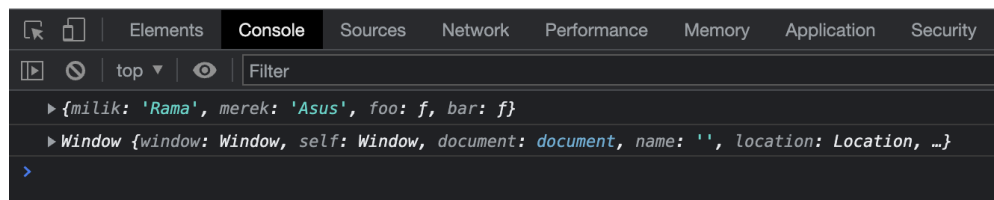
</html>

```

ii. Hasil



Belajar Javascript (ES6)



iii. Penjelasan

- Pada baris pertama, kita mendefinisikan sebuah objek dengan nama laptopRama yang memiliki tiga properti yaitu milik, merek, foo dan bar.
- Pada baris kedua, properti milik diisi dengan nilai "Rama", dan properti merek diisi dengan nilai "Asus".
- Pada baris ketiga, properti foo diisi dengan sebuah fungsi yang mengembalikan objek saat ini (dalam hal ini objek laptopRama) menggunakan this.
- Pada baris keempat, properti bar diisi dengan sebuah fungsi arrow yang mengembalikan this. Perlu dicatat bahwa fungsi arrow tidak memiliki konteks this sendiri, sehingga nilai this di dalam fungsi arrow akan menjadi sama dengan nilai this pada lingkup terluas, yaitu lingkup global.

I. Soal

i. Kode Program

Soal.html

```
<!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 (ES6)</h1>
  <script>
    let laptopRama = {
      milik: "Rama",
      merek: "Asus",
      hidupkanLaptop: () => `Hidupkan laptop
${this.merek} milik ${this.milik}`,
    }

    console.log(laptopRama.merek);
    console.log(laptopRama.milik);
    console.log(laptopRama.hidupkanLaptop());

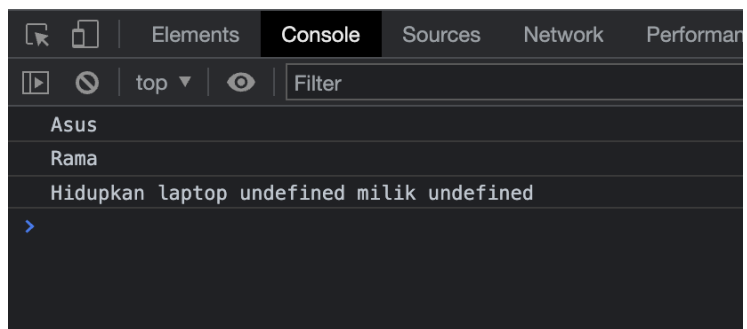
  </script>
</body>

</html>
```

ii. Hasil



Belajar Javascript (ES6)



- iii. **Jawab**
Tidak terdefinisi, karena merek dan milik tidak terdefinisi dalam lingkup global {this}. Tapi, didefinisikan dalam objek laptopRama. Jadi, kita bisa mengaksesnya dengan menggunakan laptopRama.merek dan laptopRama.milik
- iv. **Kode Program**

Soal_Revisi.html

```
<!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 (ES6)</h1>
  <script>
    let laptopRama = {
      milik: "Rama",
      merek: "Asus",
      hidupkanLaptop: () => `Hidupkan laptop
${laptopRama.merek} milik ${laptopRama.milik}`,
    }

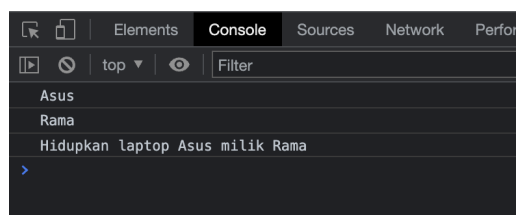
    console.log(laptopRama.merek);
    console.log(laptopRama.milik);
    console.log(laptopRama.hidupkanLaptop());

  </script>
</body>
</html>
```

- v. **Hasil**



Belajar Javascript (ES6)



2. Praktikum

a. Membuat variabel menggunakan const

o Kode Program

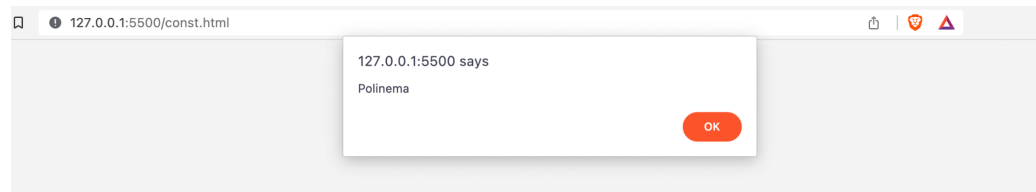
const.html

```
<!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>learning modern JavaScript</title>
</head>
<body>
  <script src="const.js"></script>
</body>
</html>
```

const.js

```
const name = 'Polinema';
alert(name);
```

o Hasil



o Penjelasan

- Pada const.js baris pertama kita mendeklarasikan sebuah variabel dengan nama name dan value "Polinema";
- Kemudian, pada baris kedua kita memanggil sebuah alert dengan isian variabel name.
- Kode tersebut akan mengeluarkan output "Polinema".

b. Membuat variabel menggunakan let

o Kode Program

let.html

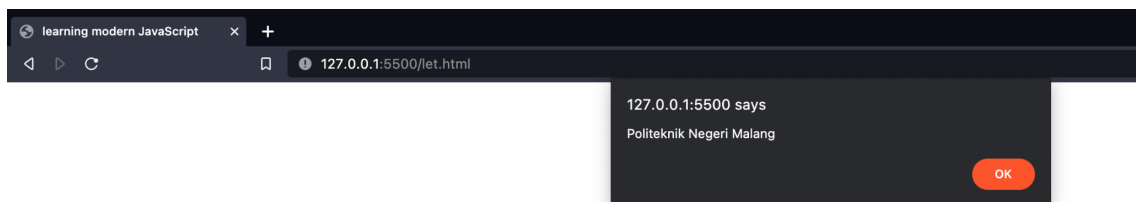
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>learning modern JavaScript</title>
</head>
<body>
  <script src="let.js"></script>
```

```
</body>
</html>
```

let.js

```
if (true) {
  let name = 'Polinema';
  name = 'Politeknik Negeri Malang';
  alert(name);
}
```

o Hasil



o Penjelasan

- Pada kondisi if pada blok statement bernilai true, maka kode di bawahnya akan dijalankan
- Pada baris pertama dalam blok if statement, kita mendeklarasikan variabel name dengan kata kunci let dan memberikan nilai awal 'Polinema' pada variabel tersebut.
- Pada baris kedua, nilai dari variabel name diubah menjadi 'Politeknik Negeri Malang'.
- Pada baris ketiga, kita menampilkan pesan 'Politeknik Negeri Malang' pada sebuah kotak dialog menggunakan fungsi alert.

c. Membuat Template Strings

o Kode Program

template.html

```
<!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>learning modern JavaScript</title>
</head>
<body>
  <script src="template.js"></script>
</body>
</html>
```

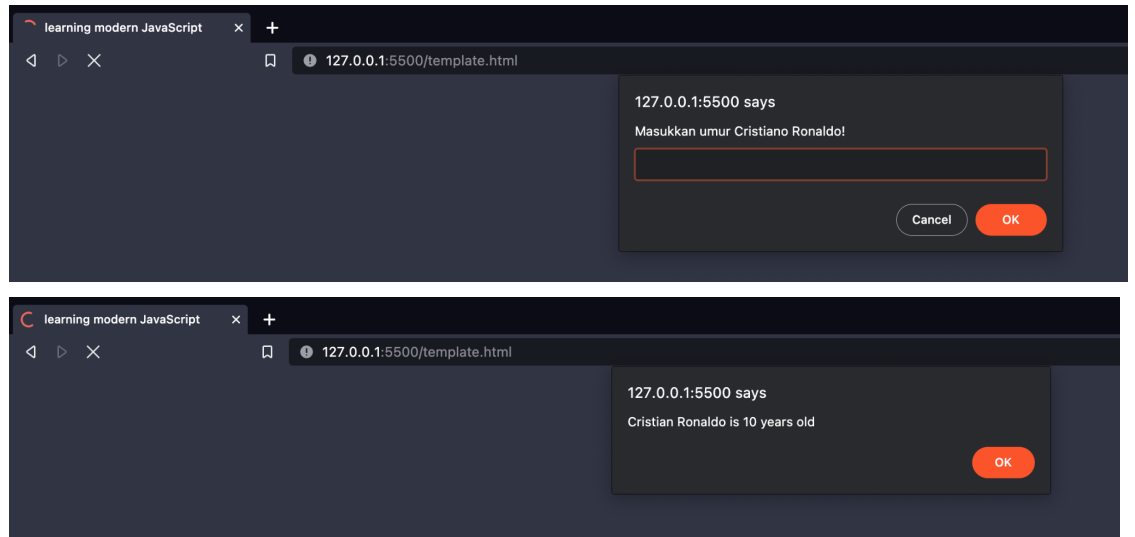
template.js

```
let fname = 'Cristian';
let lname = 'Ronaldo';
```

```
let age = prompt('Masukkan umur Cristiano Ronaldo!');

let result = `${fname} ${lname} is ${age} years old`;
alert(result);
```

o Hasil



o Penjelasan

- Kode di atas mendeklarasikan tiga variabel, yaitu fname, lname, dan age.
- Variabel fname diberikan nilai 'Cristiano' dan variabel lname diberikan nilai 'Ronaldo'.
- Pada baris ketiga, user diminta untuk memasukkan umur dari Cristiano Ronaldo melalui sebuah kotak dialog dengan menggunakan fungsi prompt. Hasil dari input tersebut disimpan pada variabel age.
- Kemudian, pada baris keempat, kita menggabungkan nilai dari variabel-variabel tersebut ke dalam sebuah string menggunakan template literals pada JavaScript. String ini akan disimpan pada variabel result.
- Pada baris terakhir, pesan yang telah digabungkan tersebut ditampilkan pada kotak dialog menggunakan fungsi alert.

d. Membuat default parameters

o Kode Program

defaultParameters.html

```
<!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>learning modern JavaScript</title>
```

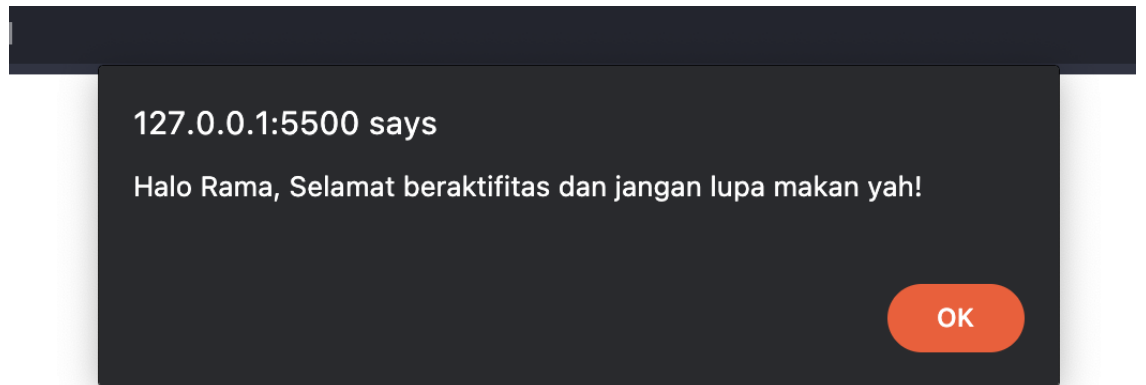
```
</head>
<body>
  <script src="defaultParameters.js"></script>
</body>
</html>
```

defaultParameters.js

```
function welcome(user = 'Rama', message = 'Selamat
beraktifitas dan jangan lupa makan yah!') {
  alert(`Halo ${user}, ${message}`);
}

welcome();
```

o Hasil



o Penjelasan

- Fungsi welcome memiliki dua parameter, yaitu user dan message. Parameter tersebut memiliki nilai default, yaitu 'Rama' dan 'Selamat beraktifitas dan jangan lupa makan yah!'.
- Jika fungsi welcome dipanggil tanpa memberikan argumen, maka nilai default akan digunakan. Dalam hal ini, fungsi akan menampilkan pesan "Halo Rama, Selamat beraktifitas dan jangan lupa makan yah!" pada konsol.

e. Membuat Arrow Function 1

o Kode Program

arrow.html

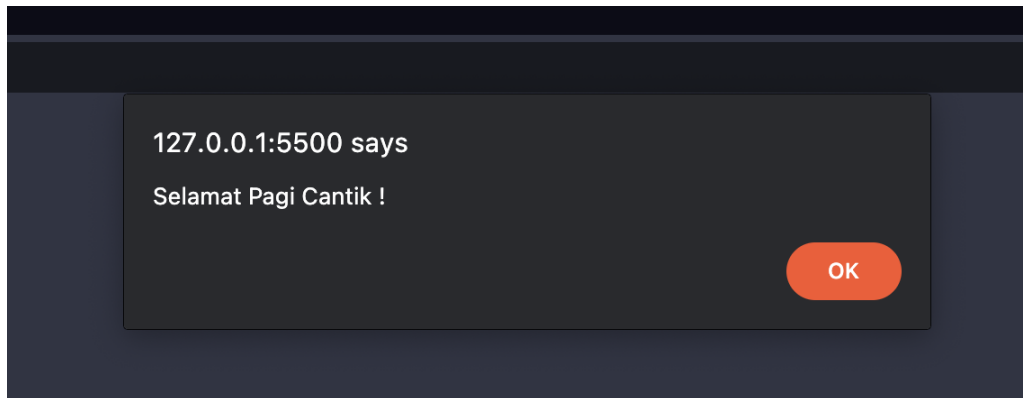
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>learning modern JavaScript</title>
</head>
<body>
  <script src="arrow.js"></script>
```

```
</body>
</html>
```

arrow.js

```
let greeting = message => alert(`${message} Cantik !`);
greeting('Selamat Pagi');
```

o Hasil



o Penjelasan

- Kode di atas mendefinisikan sebuah fungsi arrow yang disimpan dalam variabel greeting. Fungsi ini memiliki satu parameter yang disebut message.
- Fungsi ini akan menampilkan pesan pada jendela pop-up dengan menggunakan fungsi alert(). Pesan yang ditampilkan adalah gabungan antara nilai parameter message yang diberikan dan string "Cantik !".
- Ketika kita memanggil fungsi greeting dengan greeting('Selamat Pagi'), maka fungsi tersebut akan menampilkan pesan "Selamat Pagi Cantik !" pada jendela pop-up.

f. Membuat Arrow Function 2

o Kode Program

arrow2.html

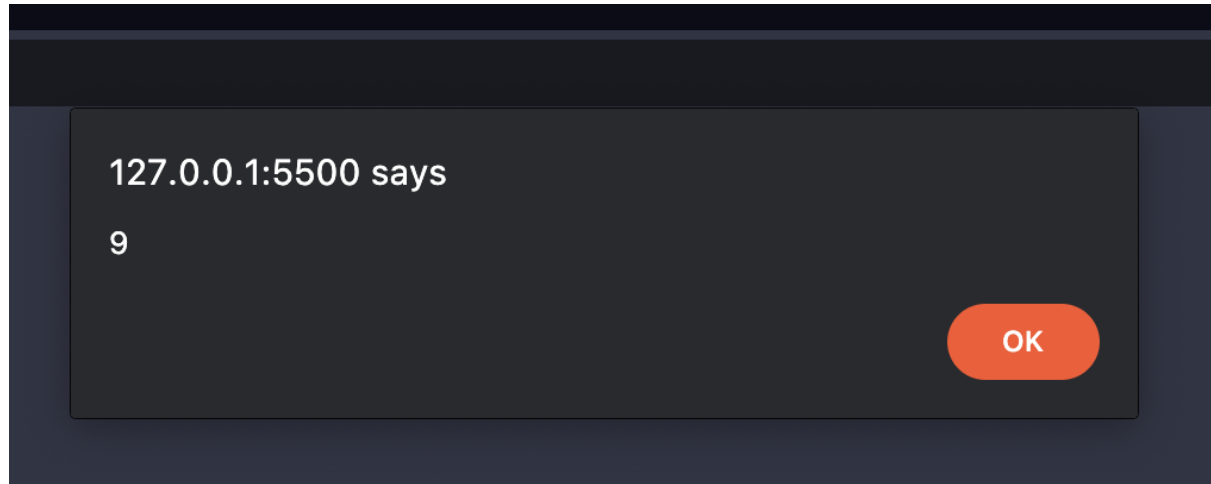
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>learning modern JavaScript</title>
</head>
<body>
  <script src="arrow2.js"></script>
</body>
```

```
</html>
```

arrow2.js

```
const func = (a, b) => {  
    return a + b;  
}  
alert(func(5, 4));
```

○ Hasil



○ Penjelasan

- Kode di atas mendefinisikan sebuah fungsi arrow dengan nama func. Fungsi ini memiliki dua parameter, yaitu a dan b, dan akan mengembalikan hasil penjumlahan antara a dan b.
- Pada baris terakhir, fungsi func dipanggil dengan argumen 5 dan 4 menggunakan fungsi alert. Hasil penjumlahan dari 5 dan 4 akan dikembalikan oleh fungsi func, dan kemudian ditampilkan pada jendela pop-up oleh fungsi alert.
- Sehingga, ketika kode tersebut dijalankan, maka jendela pop-up akan menampilkan angka 9.

g. Membuat Destructuring Object

○ Kode Program

destructuring.html

```
<!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>learning modern JavaScript</title>  
</head>  
<body>  
    <script src="destructuring.js"></script>  
</body>
```

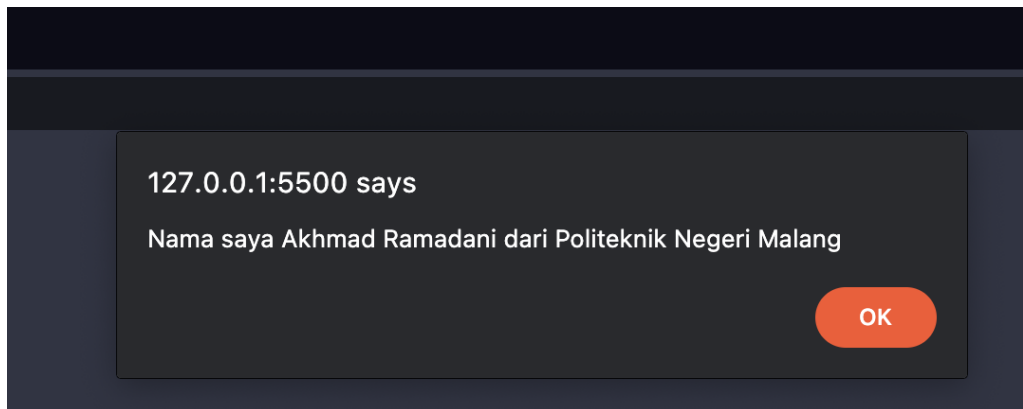


```
</html>
```

destructuring.js

```
let polStudent = ({name, polytechnic}) => {  
    alert(`Nama saya ${name} dari ${polytechnic}`);  
}  
  
polStudent({  
    name: 'Akhmad Ramadani',  
    polytechnic: 'Politeknik Negeri Malang'  
});
```

o Hasil



o Penjelasan

- Kode di atas mendefinisikan sebuah fungsi arrow dengan nama polStudent. Fungsi ini menerima sebuah objek sebagai parameter dan kemudian mengekstrak dua properti objek tersebut yaitu name dan polytechnic. Setelah itu, fungsi akan menampilkan pesan pada jendela pop-up dengan menggunakan fungsi alert(). Pesan yang ditampilkan adalah gabungan antara string "Nama saya", nilai dari properti name, dan string "dari", serta nilai dari properti polytechnic.

h. Membuat Destructuring an Array

o Kode Program

array.html

```
<!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>learning modern JavaScript</title>  
</head>  
<body>  
    <script src="array.js"></script>  
</body>  
</html>
```

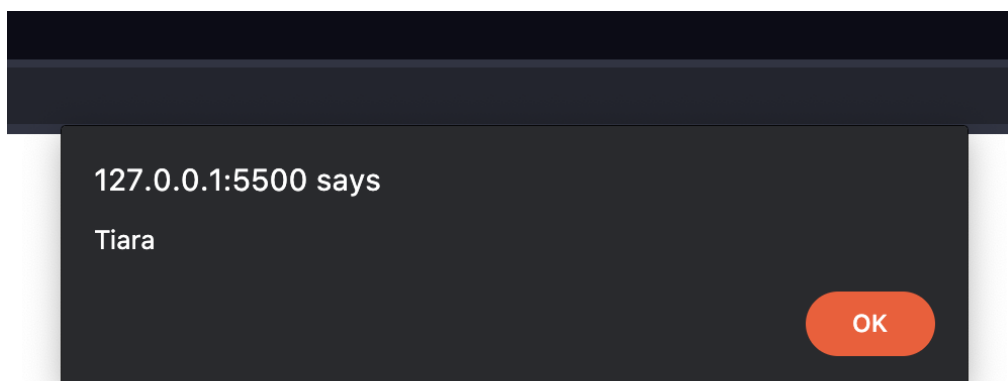
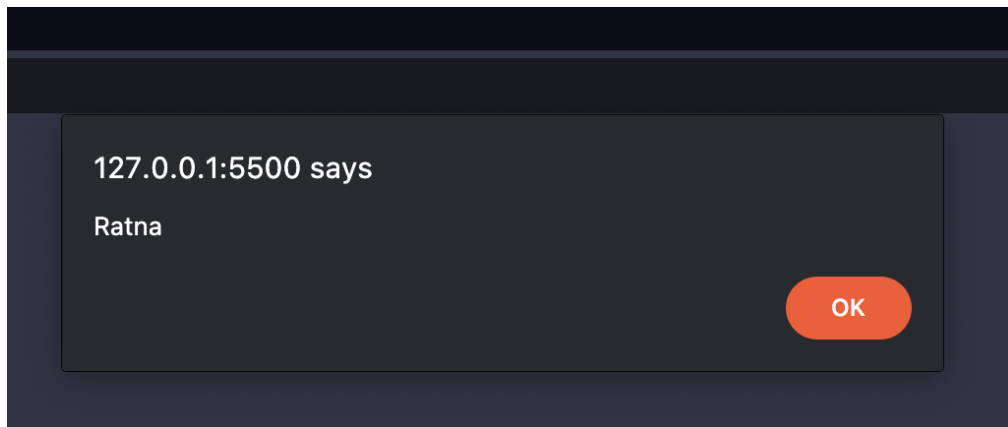
array.js

```
let [wife] = ['Ratna', 'Bunga', 'Tiara'];  
alert(wife);
```

array.js setelah diubah

```
let [ , , wife] = ['Ratna', 'Bunga', 'Tiara'];  
alert(wife);
```

○ Hasil



○ Penjelasan

- Script array.js pertama menginisialisasi variabel wife dengan menggunakan array destructuring pada array ['Ratna', 'Bunga', 'Tiara']. Hal ini akan mengambil nilai pada indeks pertama dari array tersebut, sehingga variabel wife akan memiliki nilai 'Ratna'.
- Sedangkan script array.js yang telah diubah menggunakan pengecualian (empty slot). Pada array destructuring, menggunakan pengecualian berarti melewati/mengabaikan indeks pada elemen array tertentu. Dalam hal ini, indeks pertama dan kedua diabaikan dan hanya nilai pada indeks ketiga ('Tiara') yang disimpan di dalam variabel wife.

- Sehingga hasil yang ditampilkan pada kedua script adalah 'Ratna' untuk script pertama, dan 'Tiara' untuk script yang sudah diubah.

i. Membuat Restructuring

- o Kode Program

restructuring.html

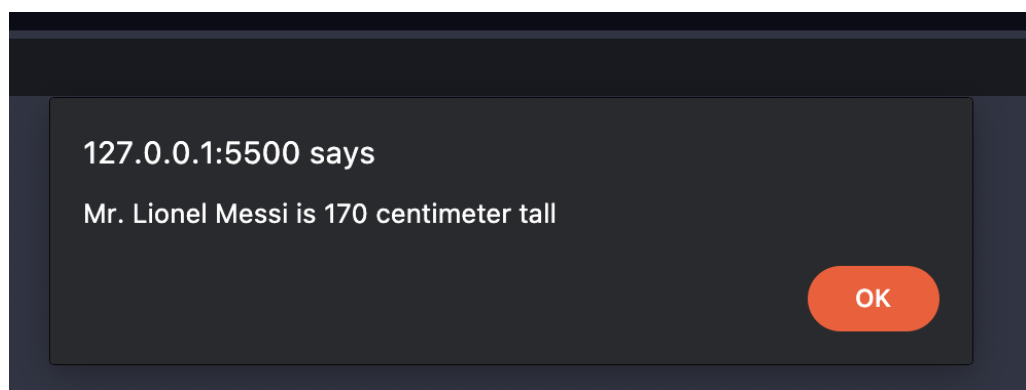
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>learning modern JavaScript</title>
</head>
<body>
  <script src="restructuring.js"></script>
</body>
</html>
```

restructuring.js

```
var pemainSepakBola = {
  name: 'Lionel Messi',
  height: 170,
  output () {
    alert(`Mr. ${this.name} is ${this.height} centimeter
tall`)
  }
}

pemainSepakBola.output();
```

- o Hasil



- o Penjelasan

- Kode di atas adalah contoh pembuatan objek pemainSepakBola dengan dua properti, yaitu name dan height, serta satu method output yang akan menampilkan pesan yang berisi nama dan tinggi badan pemain sepak bola.

j. Membuat Spread and Rest operator

o Kode Program

spread.html

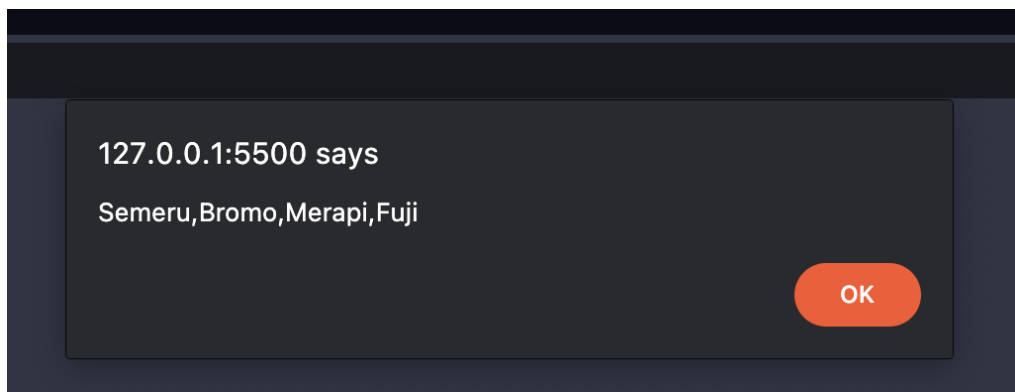
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>learning modern JavaScript</title>
</head>
<body>
  <script src="spread.js"></script>
</body>
</html>
```

spread.js

```
var mountains = ['Semeru', 'Bromo', 'Merapi'];
var mountainsFromJapan = ['Fuji'];

var allMountains = [...mountains, ...mountainsFromJapan];
alert(allMountains);
```

o Hasil



o Penjelasan

- Baris kode tersebut melakukan operasi penggabungan (concatenation) array menggunakan spread operator (...). Pertama-tama, array mountains yang berisi tiga elemen ('Semeru', 'Bromo', dan 'Merapi') didefinisikan. Kemudian, array mountainsFromJapan yang berisi satu elemen ('Fuji') juga didefinisikan.
- Kemudian, kedua array tersebut digabungkan menjadi satu array baru menggunakan spread operator (...) dan disimpan ke dalam variabel allMountains. Dalam hal ini, spread operator digunakan untuk mengeluarkan seluruh elemen dari setiap array dan menggabungkannya ke dalam array baru.

- Kode Program

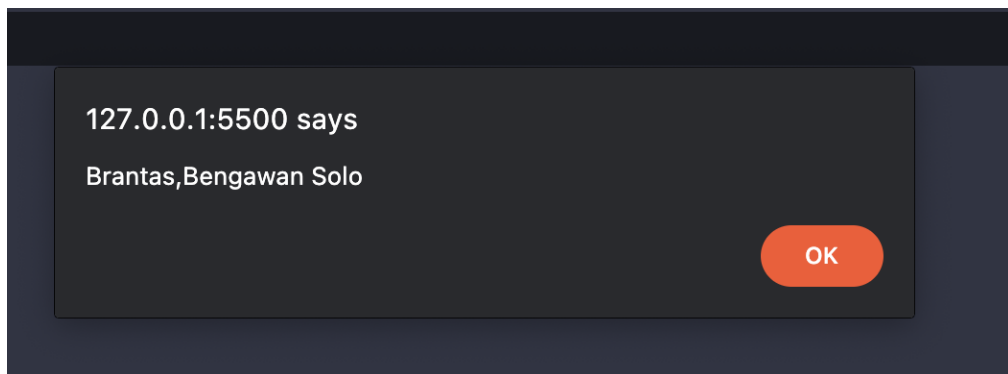
restO.html

```
<!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>learning modern JavaScript</title>
</head>
<body>
  <script src="restO.js"></script>
</body>
</html>
```

restO.js

```
var rivers = ['Ciliwung', 'Brantas', 'Bengawan Solo'];
var [first, ...rest] = rivers;
alert(rest);
```

- Hasil



- Penjelasan

- Baris kode tersebut melakukan deklarasi sebuah array rivers yang berisi tiga elemen, kemudian pada baris selanjutnya dilakukan destrukurisasi array menggunakan teknik rest operator, yaitu dengan memasukkan seluruh elemen yang tidak dipilih ke dalam variabel rest.
- Ketika variabel rest dipanggil dengan alert(), maka akan ditampilkan semua elemen dalam array rivers kecuali elemen pertama, yaitu ['Brantas', 'Bengawan Solo'].

k. Membuat Classes Constructor and Super

- Kode Program

class.html

```
<!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>learning modern JavaScript</title>
</head>
<body>
  <script src="class.js"></script>
</body>
</html>
```

class.js

```
class Holiday {
  constructor(destination, days){
    this.destination = destination;
    this.days = days;
  }

  info(){
    alert(`${this.destination} will take ${this.days}
days.`);
  }
}

class Expedition extends Holiday {
  constructor(destination, days, gear){
    super(destination, days);
    this.gear = gear;
  }

  info(){
    super.info();
    alert(`Bring your ${this.gear.join(' and your ')}`);
  }
}

const tripWithGear = new Expedition('Semeru', 10,
['Sunglasses', 'Flags', 'Camera']);
tripWithGear.info();
```

- Hasil

127.0.0.1:5500 says

Semeru will take 10 days.

OK

127.0.0.1:5500 says

Bring your Sunglasses and your Flags and your Camera

OK

- Penjelasan

- Di sini terdapat dua kelas, yaitu kelas Holiday dan Expedition. Kelas Holiday memiliki dua properti, yaitu destination dan days. Kelas ini juga memiliki metode info() yang menampilkan informasi tentang tujuan dan jumlah hari perjalanan.
- Kelas Expedition merupakan turunan dari kelas Holiday. Kelas ini memiliki tiga properti, yaitu destination, days, dan gear. Kelas ini juga memiliki metode info() yang menampilkan informasi tentang tujuan dan jumlah hari perjalanan. Namun, metode ini juga menampilkan daftar perlengkapan yang harus dibawa dalam perjalanan.
- Selanjutnya, sebuah objek tripWithGear dibuat dari kelas Expedition dengan tujuan Semeru, jumlah hari perjalanan selama 10 hari, dan daftar perlengkapan yang harus dibawa. Kemudian, metode info() dipanggil pada objek tripWithGear untuk menampilkan informasi tentang perjalanan dan daftar perlengkapan yang harus dibawa.