LAPORAN TUGAS 2

Pemrograman Berbasis Framework

PERTEMUAN 2

Oleh:

ALAN PERDHANA TIMOR

1841720187 TI-3E

PROGRAM STUDI D-IV TEKNIK INFORMATIKA JURUSAN TEKNOLOGI INFORMASI POLITEKNIK NEGERI MALANG

2020

MODUL 1: Modern JavaScript

A. TUJUAN

1. Mahasiswa dapat mengetahui dan mengimplemantasikan kegunaan variables using const.
2. Mahasiswa dapat mengetahui dan mengimplemantasikan kegunaan variables using let.
3. Mahasiswa dapat mengetahui dan mengimplemantasikan kegunaan template string.
4. Mahasiswa dapat mengetahui dan mengimplemantasikan kegunaan arrow function.
5. Mahasiswa dapat mengetahui dan mengimplemantasikan kegunaan destructuring object.
6. Mahasiswa dapat mengetahui dan mengimplemantasikan kegunaan destructuring an array.
7. Mahasiswa dapat mengetahui dan mengimplemantasikan kegunaan spread and rest operator.
8. Mahasiswa dapat mengetahui dan mengimplemantasikan kegunaan classes constructor and super.

B. MATERI

1. Referensi
    a. https://reactjs.org/tutorial/tutorial.html
    b. https://reactjs.org/tutorial/tutorial.html#setup-for-the-tutorial

2. Modern JavaScript
    a. JavaScript didasarkan pada spesifikasi EcmaScript
    b. Dimana merupakan standar yang ditetapkan oleh Asosiasi Produsen Komputer Eropa (ECMA)
    c. Kita akan mulai dengan mempelajari JavaScript dengan fitur baru yang ditambahkan oleh ES6
    d. Kita akan menulis ES6 dengan React
    e. Memahami ES6 sangat penting untuk memahami React

C. PRAKTIKUM

a. Membuat variabel menggunakan const

1. Buatlah halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya.

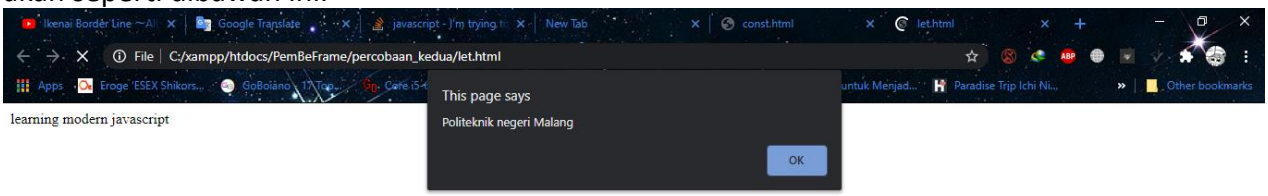
   const.html

```
!<DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <tittle>learning modern javascript</tittle>
</head>
<body>
    <script src="const.js"></script>
</body>
</html>
```

const.js

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

2. Selanjutnya buka file const.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.



Penjelasaan:
Setelah membuat program di atas, jika kita run, maka akan muncul alert sesuai yang sudah kita buat sebelumnya

b. Membuat variabel menggunakan let

1. Buatlah halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya.

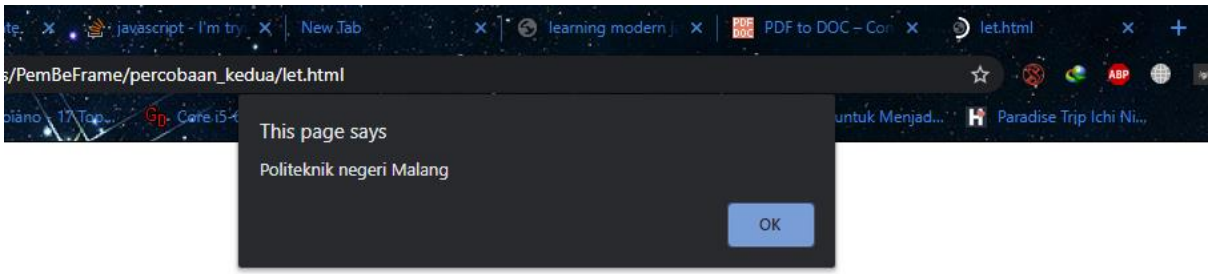
let.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <tittle>learning modern javascript</tittle>
</head>
<body>
    <script src="let.js"></script>
</body>
</html>
```

let.js

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

2. Selanjutnya buka file let.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.



Penjelasaan: let.js fungsinya sama seperti sebelumnya, namun pada pada let.js menggunakan decission if, jika (true) name 'polinema', maka alert akan muncul

c. Membuat Template Strings

1. Buatlah halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya.

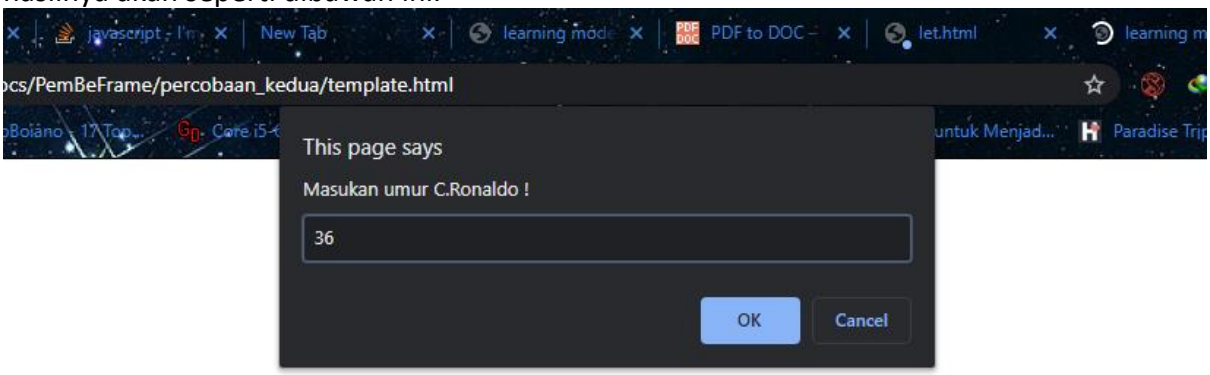
   template.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>learning modern javascript</title>
</head>
<body>
    <script src="template.js"></script>
</body>
</html>
```

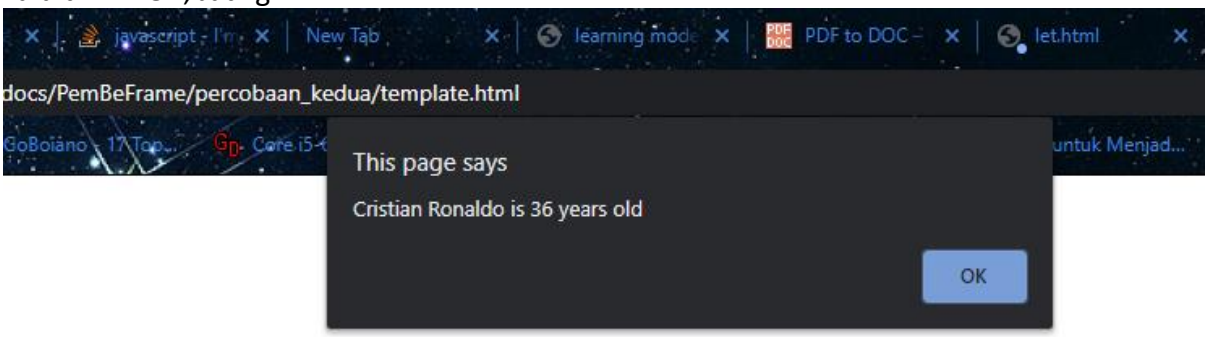
   template.js

```
let fname = 'Cristian';
let lname = 'Ronaldo';
let age = prompt("Masukan umur C.Ronaldo !");


//cara lama
// let result = fname + '' + lname + 'is' + age + 'years old';
// alert

// pake template strings

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

2. Selanjutnya buka file template.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.



3. Lalu di klik OK, Jadi gini



Penjelasaan: pada template.js, kita membuat program yang bisa bisa kita isi parameter yaitu angka/umur dan memunculkannya pada alert

d. Membuat default parameters

1. Buatlah halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya.

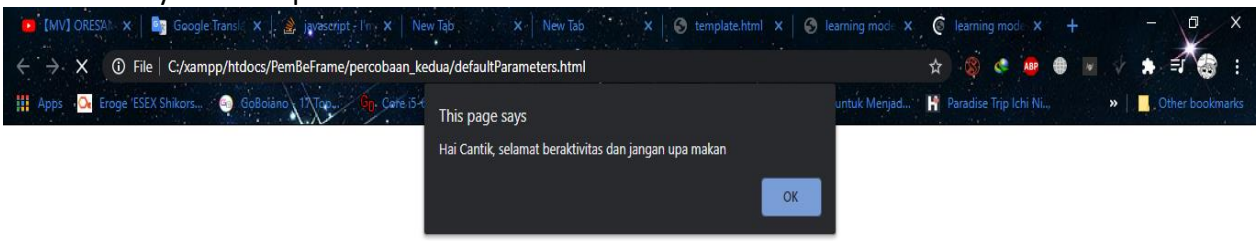
   defaultParameters.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>learning modern javascript</title>
</head>
<body>
    <script src="defaultParameters.js"></script>
</body>
</html>
```

defaultParameters.js

```
function welcome(user = 'Cantik', message = 'selamat beraktivitas dan jangan u
pa makan'){
    alert(`Hai ${user}, ${message}`);
}

welcome();
```

2. Selanjutnya buka file defaultParameters.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.



Penjelasaan: sebuah alert akan muncul yang berisikan parameter yang telah kita isi pada js, yang dimana telah dipanggil variable nya pada html

e. Membuat Arrow Function 1

1. Buatlah halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya.

arrow.html

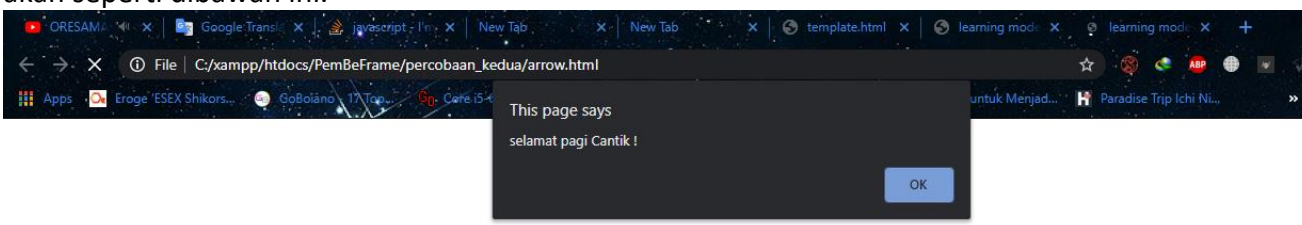
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>learning modern javascript</title>
</head>
<body>
    <script src="arrow.js"></script>
</body>
```

```
</html>
```

arrow.js

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

greeting('selamat pagi');
```

2. Selanjutnya buka file arrow.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.



Penjeleasan : Pada program yang kita terdapat nilai pada variable greeting bisa dilakukan diakhir, dan dijelaskan bahwa variable greeting sama dengan message sehingga hasilnya keluar alert seperti di atas ini

f. Membuat Arrow Function 2

1. Buatlah halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya.

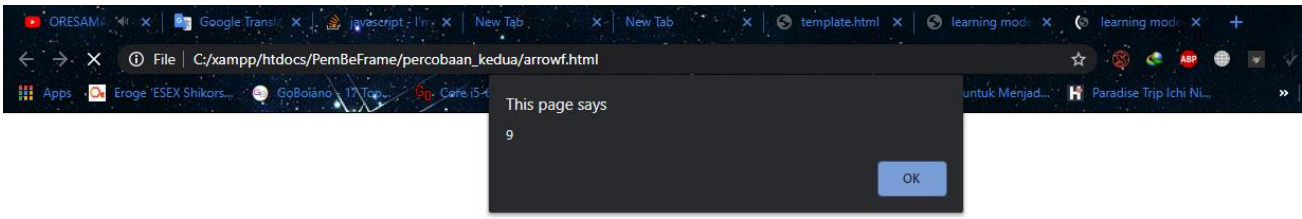
arrowf.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>learning modern javascript</title>
</head>
<body>
    <script src="arrowf.js"></script>
</body>
</html>
```

arrowf.js

```
const func = (a, b) => {
    return a + b;

};
alert(func(5,4));
```

2. Selanjutnya buka file arrowf.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.



Penjelasaan: pada keyword const berfungsi untuk declare kode program arrowf.js dimana const melakukan deklarasi variable func yang berisi variable a dan b untuk pertambahaan dan akan muncul laert seperti berikut

g. Membuat Destructuring Object

1. Buatlah halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya.

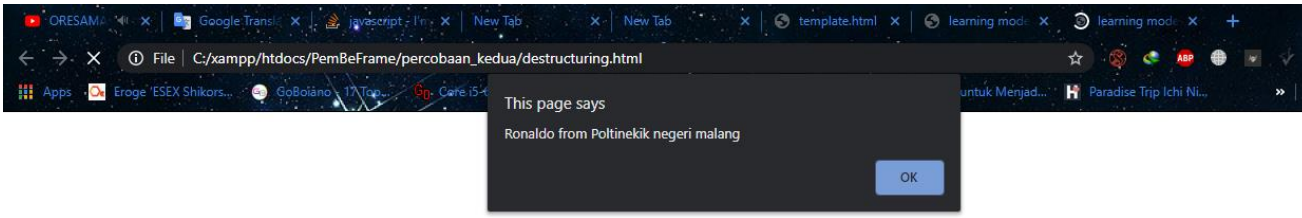
   destructuring.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>learning modern javascript</title>
</head>
<body>
    <script src="destructuring.js"></script>
</body>
</html>
```

   destructuring.js

```
let polStudent = ({name, polytechnic}) => {
    alert(`${name} from ${polytechnic}`);

};

polStudent({
    name: 'Ronaldo',
    polytechnic: 'Poltinekik negeri malang'
});
```

2. Selanjutnya buka file destructuring.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.



Penjelasan : Pada kode yang kita buat, kita mendefinisikan variable yang ada pada deklarasi tipe data let harus area yang sama, tidak terjadi error karena let tidak bisa mendefinisikan dan mendeklarasikan dua variable secara terpisah kecuali dalam satu lingkup yang sama seperti pada gambar diatas.

h. Membuat Destructuring an Array

1. Buatlah halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya.

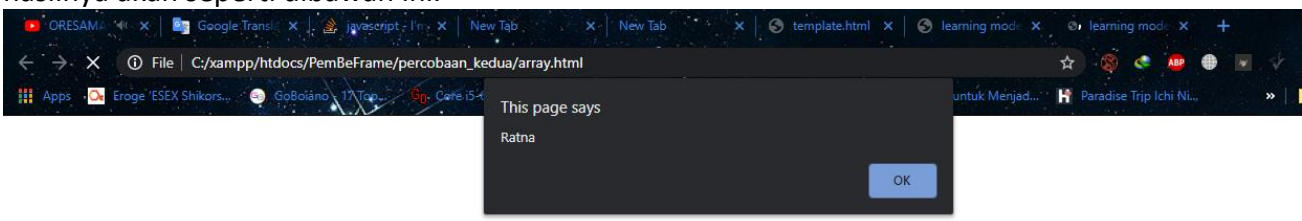
   array.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>learning modern javascript</title>
</head>
<body>
    <script src="array.js"></script>
</body>
</html>
```

   array.js

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

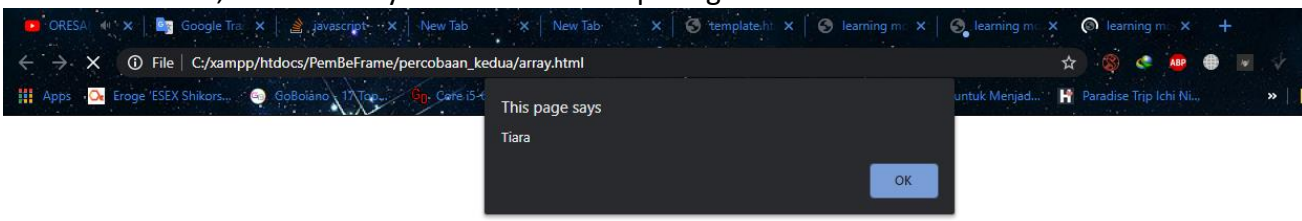
2. Selanjutnya buka file array.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.



3. Kemudian ganti sedikit code pada array.js seperti dibawah ini

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

6. Bukalah kembali, maka hasilnya akan berbeda seperti gambar dibawah ini



Penjelasan : mirip seperti kode sebelumnya, di mana kita hanya akan menampilkan 1 nilai variable saja dan yang kedua kita memilih untuk mengeluarkan nilai ke 2

i. Membuat Restructuring

1. Buatlah halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya.

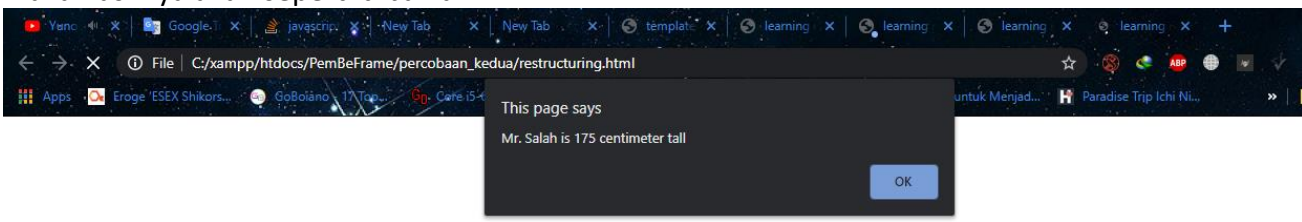
restructuring.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>learning modern javascript</title>
</head>
<body>
    <script src="restructuring.js"></script>
</body>
</html>
```

restructuring.js

```
var pemainSepakbola = {
    name: 'Salah',
    height: '175',
    output(){
        alert(`Mr. ${this.name} is ${this.height} centimeter tall`);
    }

};
pemainSepakbola.output();
```

2. Selanjutnya buka file restructuring.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.



Penjelasan : kode berikut melakukan restrukturisasi untuk susunan pada variable yang dideklrasikan menggunakan keyword var yang deklarasi berulang atau redeclare yang nantinya setiap variable tersebut akan disusun ulang pada bagian alert.

j. Membuat Spread and Rest operator

1. Buatlah halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya.

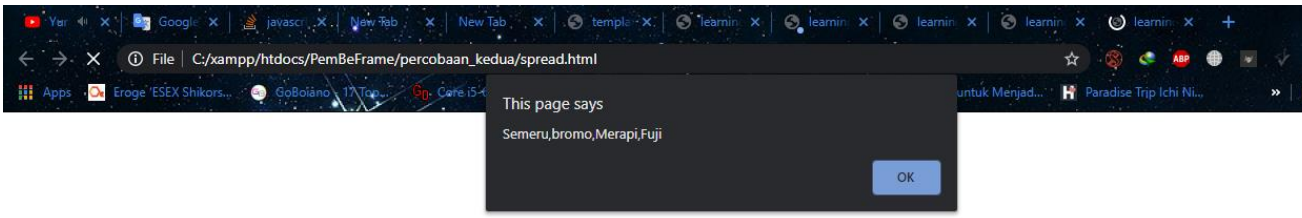
spread.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <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);
```

2. Selanjutnya buka file spread.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.



Penjelasan : kode berikut berguna untuk melakukan penyebaran kode program JavaScript, yang dimana ada 2 variable yang berbeda nialrnya pada suatu variable baru. Lalu muncul menggunakan fungsi alert.

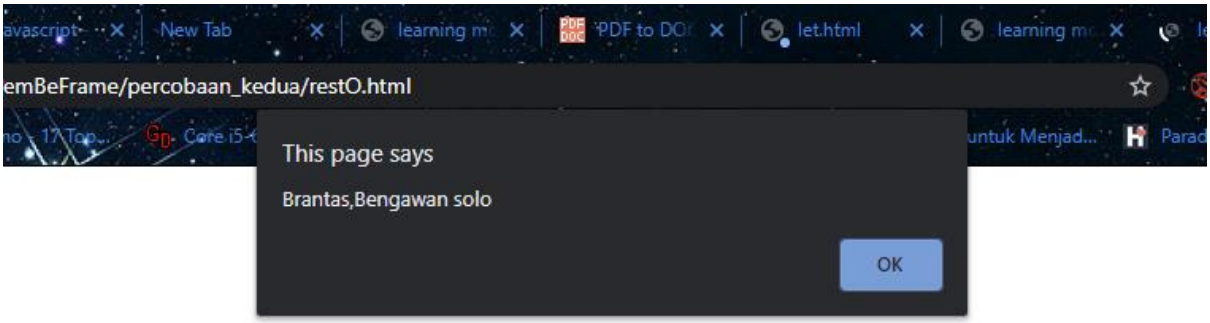
3. Untuk melakukan praktek rest operation, buatlah dua file .html dan .js seperti dibawah ini
   restO.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <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);
```

4. Selanjutnya buka file restO.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.



Penjelasan : kode restoO menampilkan nilai yang terpada pada variable tidak keseluruhan, namun, terdapat jeda dalam menampilkan nilai yang ada pada variable.

k. Membuat Classes Constructor and Super

1. Buatlah halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya.

   class.html

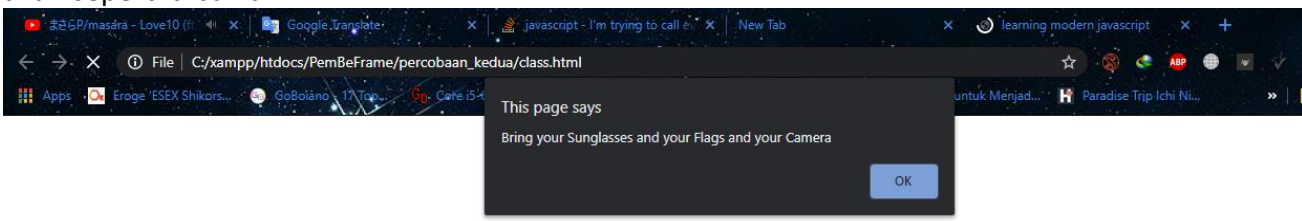
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>learning modern javascript</title>
</head>
<body>
    <script src="class.js"></script>
</body>
</html>
```

   class.js

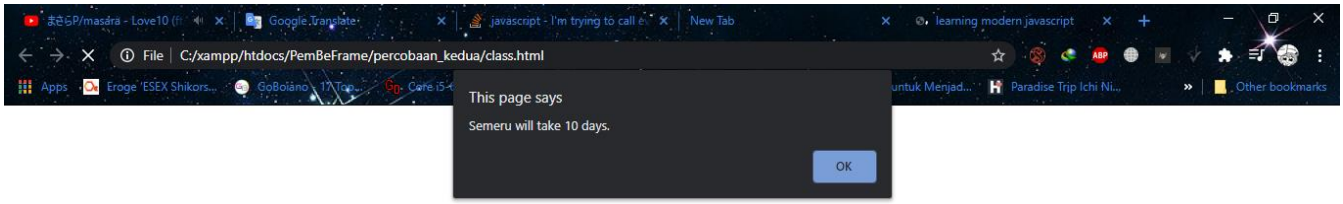
```
class Holiday {
    constructor(destination, days){
        this.destination = destination;
        this.days = days;
    }

    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', 'Cam
era']);
tripWithGear.info();
```

2. Selanjutnya buka file class.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.



3. Klik OK, maka akan muncul page selanjutnya dengan pesan sebagai berikut.



Penjelasan : pada kode program class.js ini kita membuat dua class yaitu class Holiday yang berperan sebagai parent dan class Expedition sebagai child, dimana ada tipe data const yang cukup sama seperti child, hanya bisa mendeklarasikan satu variable dengan banyak nilai. Terlihat variable tripWithGear yang menarik function info yang pada class Expedition dan ditampilkan pada gambar kedua