Laporan Praktikum Jobsheet 1
Mata Kuliah Pemrograman Berbasis Framework
"MODUL 1: Modern JavaScript"

Nama Penyusun:

Khosy Robbin Hood (1941720067) / TI3D

JURUSAN TEKNOLOGI INFORMASI
PROGRAM STUDI D-IV TEKNIK INFORMATIKA
FEBRUARI 2022

A. PRAKTIKUM

a. Membuat variabel menggunakan const

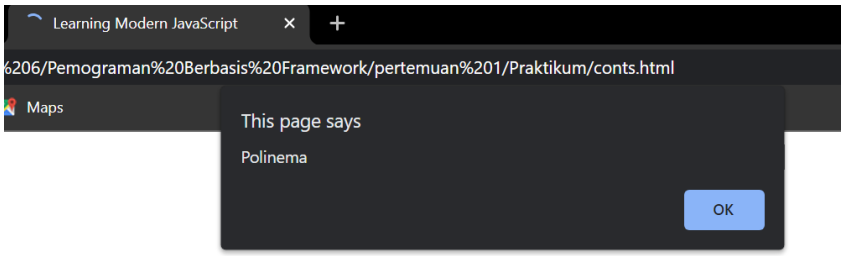
conts.html

```
<!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="conts.js"></script>
</body>
</html>
```

conts.js

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

output



- fungsi onst digunakan untuk mendeklarasikan variabel yang readonly.

b. Membuat variabel menggunakan let

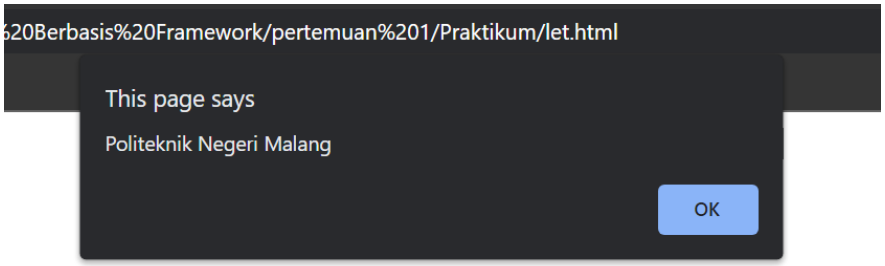
let.java

```
<!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);
  }
```

output



- let akan mendeklarasikan variabel dengan scope terbatas pada blok dan pernyataan dimana mereka digunakan.

c. Membuat Template Strings

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>
```

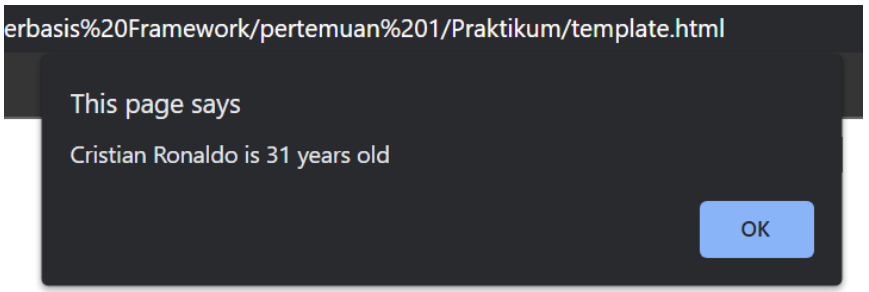
template.js

```
let fname = "Cristian"; // firstname
let lname = "Ronaldo"; // lastname
let age = prompt("Masukkan umur Cristian Ronaldo!"); //
input umur

// Cara lama
// let result = fname + ' ' + lname + 'is' + age + 'years
old';
// alert(result);
```

```
// Memakai template strings
let result = `${fname} ${lname} is ${age} years old`; //
output kalimat
alert(result);
```

output



- berfungsi untuk melakukan input angka yang nanti akan muncul alert dengan angka yang diinputkan.

d. Membuat default parameters

defaultParameters.html

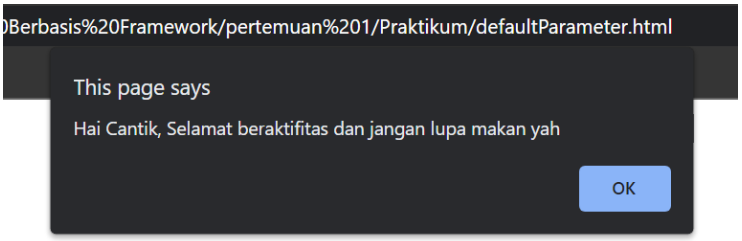
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Learning Modern JavaScript</title>
</head>
<body>
    <script src="defaultParameter.js">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

defaultParameters.js

```
function welcome(
    user = "Cantik",
    message = "Selamat beraktifitas dan jangan lupa makan
yah"
  ) {
    alert(`Hai ${user}, ${message}`);
  }

  welcome();
```

output



- alert akan memanggil nilai dari $user dilaanjutkan $message

e. Membuat Arrow Function 1

arrow.html

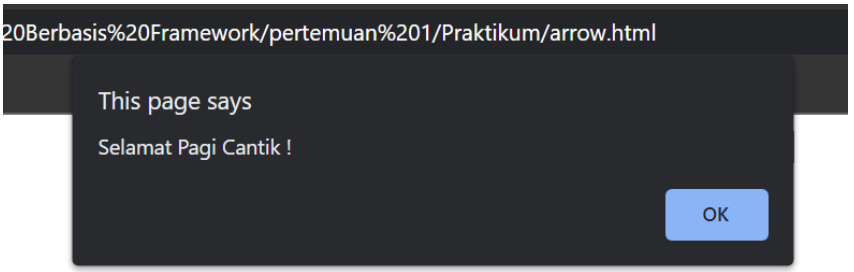
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Learning Modern JavaScript</title>
</head>
<body>
    <script src="arrow.js"></script>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

arrow.js

```
// Tanpa arrow
// function gretting(message) {
//   return alert(`${message} Cantik !`);
// }

// Menggunakan arrow
let gretting = message => alert(`${message} Cantik !`);
gretting("Selamat Pagi");
```

output



- Arrow function adalah sintak penulisan fungsi yang bisa dibilang lebih singkat, menggunakan token baru yaitu “=>”

f. Membuat Arrow Function 2

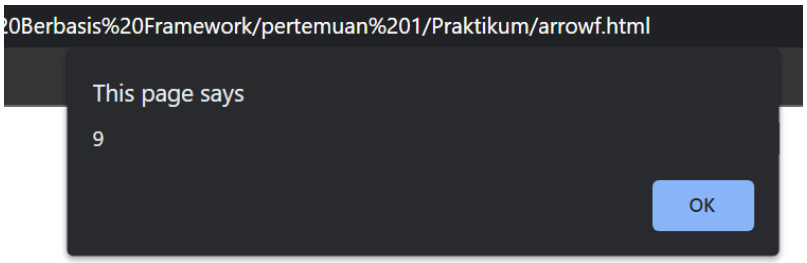
arrowf.html

```
<!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="arrowf.js"></script>
</body>
</html>
```

arrowf.js

```
// Tanpa arrow
// const func = function (a, b) {
//   return a + b;
// };
// alert(func(3, 3));
// Menggunakan arrow
const func = (a, b) => {
    return a + b;
  };
  alert(func(5, 4));
```

output



- Arrow function adalah sintak penulisan fungsi yang bisa dibilang lebih singkat, menggunakan token baru yaitu “=>”

g. Membuat Destructuring Object

destructuring.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
```

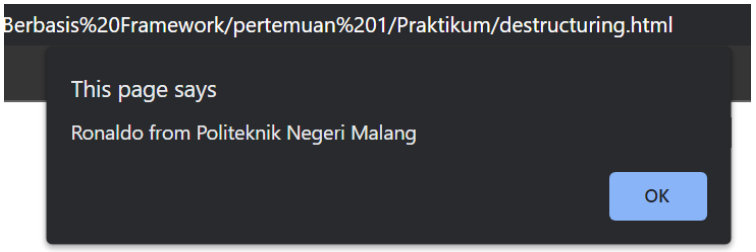
```
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Learning Modern JavaScript</title>
</head>
<body>
    <script src="destructuring.js"></script>
</body>
</html>
```

destructuring.js

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

  polStudent({
    name: "Ronaldo",
    polytechnic: "Politeknik Negeri Malang",
  });
```

output



- Destructuring mempermudahkan untuk mengolah nilai dari array maupun properti dari object pada javascript.

h. Membuat Destructuring an Array

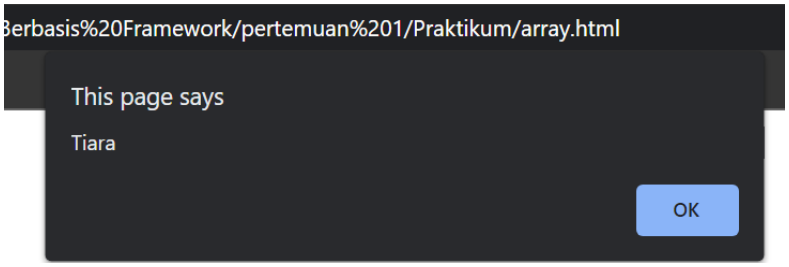
array.html

```
<!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] = ["Tiara", "Bunga", "Ratna"];
// let [, , wife] = ["Tiara", "Bunga", "Ratna"];
alert(wife);
```

output



- fungsi array akan dilakukan untuk pemanggilan alert di bagian awal array.

i. Membuat Restructuring

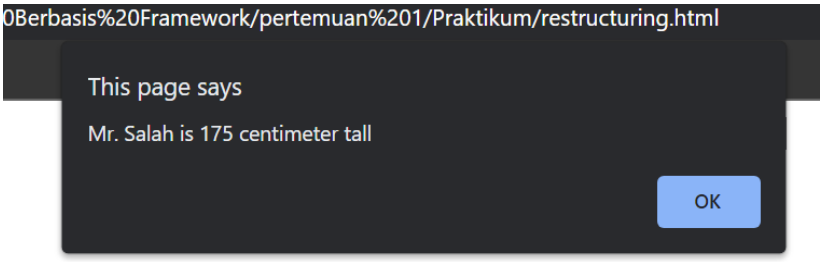
restructuring.html

```
<!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: "Salah",
    height: "175",
    output() {
      alert(`Mr. ${this.name} is ${this.height} centimeter
tall`);
    },
  };
  pemainSepakbola.output();
```

output



- dilakukan pemanggilan variable var yang disesuainakn dengan alert.

j. Membuat Spread and Rest operator

spread.html

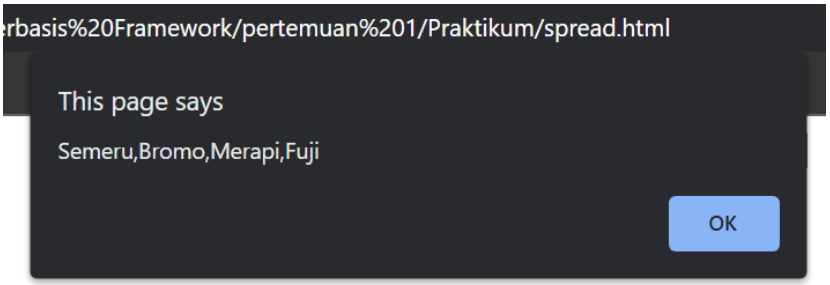
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Learning Modern JavaScript</title>
</head>
<body>
    <script src="spread.js"></script>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

spread.js

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

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

output



- Spread operator berfungsi untuk memasukan beberapa argument yang berbentuk data array kedalam fungsi

restO.html

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

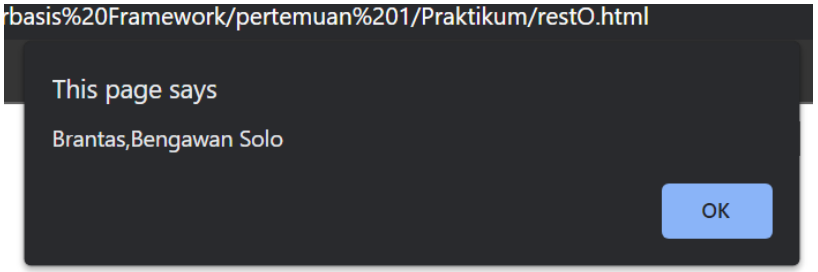
```
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Learning Modern JavaScript</title>
</head>
<body>
    <script src="restO.js"></script>
</body>
</html>
```

restO.js

```
var rivers = ["Ciliwung", "Brantas", "Bengawan Solo"];
var [first, ...rest] = rivers;

alert(rest);
```

output



- array pada var dilakukan pembagian dimana nilai first memiliki nilai array awal yaitu ciliwung.

k. Membuat Classes Constructor and Super
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>
```

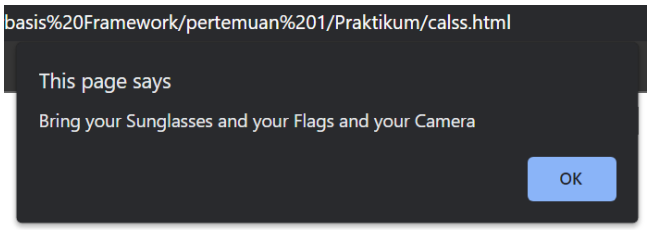
class.js

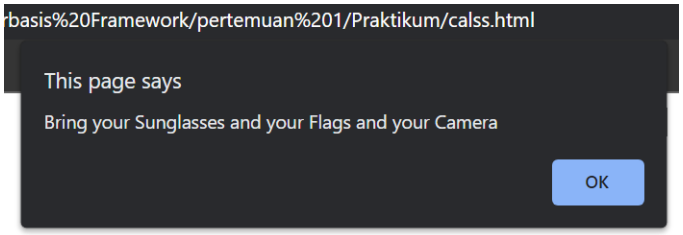
```
// super class
class Holiday {
    constructor(destination, days) {
      this.destination = destination;
      this.days = days;
    }
    info() {
      alert(`${this.destination} will take ${this.days}
days.`);
    }
  }

  // sub class
  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();
```

output





- Satu class mempunyai satu atau lebih properti atau fungsi. Satu class dapat menghasilkan banyak object. Semua object mempunyai properti dan fungsi yang sama. Setiap object bisa mempunyai nilai properti yang berbeda-beda.