

# **Laporan Praktikum Modul 6 Pemrograman Javascript Level Basic**



Disusun oleh :

Nama : Riandra Alkautsar

Email : riandra.alkautsar@gmail.com

# I. Tujuan Praktikum

Peserta didik diharapkan dapat memahami konsep javascript dasar diantaranya :

1. Pernyataan loncat
2. Fungsi
3. Tipe Objek
4. Kelas
5. Penanganan Eksepsi

## II. Hasil Praktikum dan Pembahasan

### 1. Pernyataan loncat

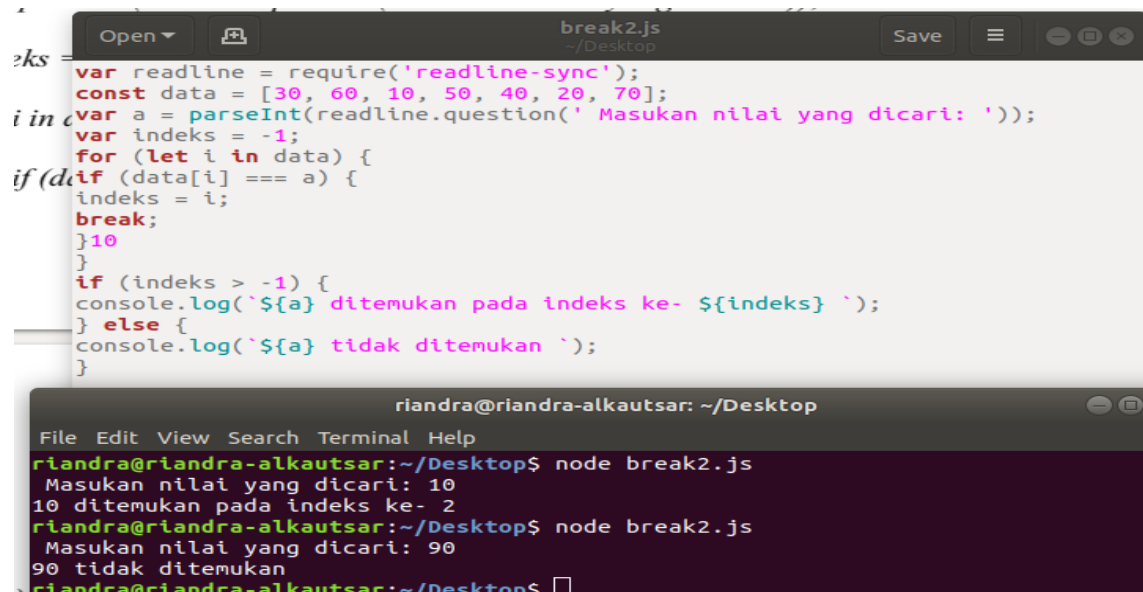
#### a. Perintah break



The screenshot shows a code editor window titled 'break.js' with the following JavaScript code:

```
for (let i = 0; i < 10; i++) {  
  if (i === 4) break;  
  process.stdout.write(i + " ");  
}
```

Below the code editor is a terminal window with the command prompt 'riandra@riandra-alkautsar: ~/Desktop'. The command 'node break.js' has been executed, resulting in the output '0123' followed by a cursor.



The screenshot shows a code editor window titled 'break2.js' with the following JavaScript code:

```
var readline = require('readline-sync');  
const data = [30, 60, 10, 50, 40, 20, 70];  
var a = parseInt(readline.question('Masukan nilai yang dicari: '));  
var indeks = -1;  
for (let i in data) {  
  if (data[i] === a) {  
    indeks = i;  
    break;  
  }  
}  
if (indeks > -1) {  
  console.log(`${a} ditemukan pada indeks ke- ${indeks} `);  
} else {  
  console.log(`${a} tidak ditemukan `);  
}
```

Below the code editor is a terminal window with the command prompt 'riandra@riandra-alkautsar: ~/Desktop'. The command 'node break2.js' has been executed twice. The first time, the user input was '10', and the output was '10 ditemukan pada indeks ke- 2'. The second time, the user input was '90', and the output was '90 tidak ditemukan'.

## b. Perintah continue

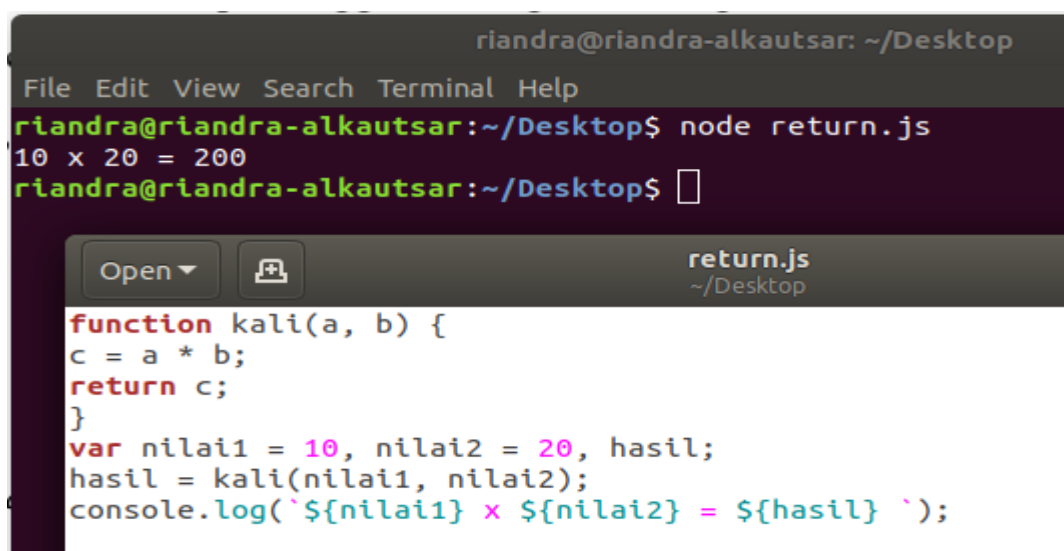


The screenshot shows a code editor window titled 'continue.js' with the following JavaScript code:

```
for (let i=0; i<10; i++) {  
  if (i % 2 === 0) continue;  
  process.stdout.write(i + " ");  
}
```

Below the editor is a terminal window with the command prompt 'riandra@riandra-alkautsar: ~/Desktop'. The command 'node continue.js' has been executed, resulting in the output '13579' followed by a new line.

## c. Perintah return



The screenshot shows a code editor window titled 'return.js' with the following JavaScript code:

```
function kali(a, b) {  
  c = a * b;  
  return c;  
}  
var nilai1 = 10, nilai2 = 20, hasil;  
hasil = kali(nilai1, nilai2);  
console.log(`${nilai1} x ${nilai2} = ${hasil} `);
```

Below the editor is a terminal window with the command prompt 'riandra@riandra-alkautsar: ~/Desktop'. The command 'node return.js' has been executed, resulting in the output '10 x 20 = 200' followed by a new line.

## 2. Fungsi

### Memanggil Fungsi

```
riandra@riandra-alkautsar: ~/Desktop
File Edit View Search Terminal Help
riandra@riandra-alkautsar:~/Desktop$ node callfunction.js
Masukan nilai a: 1.25
Masukan nilai b: 4.0
1.25 x 4 = 5
riandra@riandra-alkautsar:~/Desktop$
```

callfunction.js  
~/Desktop

```
var readline = require('readline-sync');
function kali(a, b) {
  return a * b;
}
var a = parseFloat(readline.question("Masukan nilai a: "))
var b = parseFloat(readline.question("Masukan nilai b: "))
var c;
c = kali(a, b);
console.log(`${a} x ${b} = ${c}`);
```

```
riandra@riandra-alkautsar: ~/Desktop
File Edit View Search Terminal Help
riandra@riandra-alkautsar:~/Desktop$ node callfunction.js
Masukan nilai a: 1.25
Masukan nilai b: 4.0
1.25 x 4 = 5
riandra@riandra-alkautsar:~/Desktop$ node callfunction1.js
Masukan nilai a: 1.25
Masukan nilai b: 4.0
1.25 x 4 = 5
riandra@riandra-alkautsar:~/Desktop$
```

callfunction1.js  
~/Desktop

```
var readline = require('readline-sync');
function kali(a, b) {
  return a * b;
}
var a = parseFloat(readline.question("Masukan nilai a: "))
var b = parseFloat(readline.question("Masukan nilai b: "))

console.log(`${a} x ${b} = ${kali(a, b)}`);
```

### Parameter Fungsi

```
riandra@riandra-alkautsar: ~/Desktop
File Edit View Search Terminal Help
riandra@riandra-alkautsar:~/Desktop$ node parameter.js
200
riandra@riandra-alkautsar:~/Desktop$
```

parameter.js  
~/Desktop

```
function kali(a, b) {
  return a * b;
}
var x = 10, y = 20;
var hasil = kali(x, y);

console.log(hasil);
```

### Fungsi tanpa Nama

```
riandra@riandra-alkautsar: ~/Desktop
File Edit View Search Terminal Help
riandra@riandra-alkautsar:~/Desktop$ node noname.js
18
riandra@riandra-alkautsar:~/Desktop$
```

noname.js  
~/Desktop

```
var tambah = function (a, b) {
  return a + b;
}

var hasil = tambah(10, 8);

console.log(hasil)
```

```
riandra@riandra-alkautsar: ~/Desktop
File Edit View Search Terminal Help
riandra@riandra-alkautsar:~/Desktop$ node noname2.js
18
riandra@riandra-alkautsar:~/Desktop$
```

noname2.js  
~/Desktop

```
var tambah = (a, b) => a + b;

var hasil = tambah(10, 8);

console.log(hasil)
```

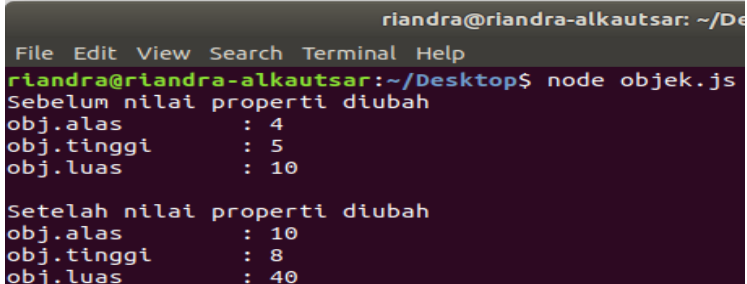
### 3. Tipe Objek

#### Membuat dan Menggunakan Objek

```
function Segitiga(a, t) {
  this.alas = a;
  this.tinggi = t;
}
Segitiga.prototype.luas = function () {
  return this.alas * this.tinggi / 2;
}

var obj = new Segitiga(4, 5);
console.log(`Sebelum nilai properti diubah`);
console.log(`obj.alas\t: ${obj.alas}`);
console.log(`obj.tinggi\t: ${obj.tinggi}`);
console.log(`obj.luas\t: ${obj.luas()}`);

obj.alas = 10;
obj.tinggi = 8;
console.log(`\nSetelah nilai properti diubah`);
console.log(`obj.alas\t: ${obj.alas}`);
console.log(`obj.tinggi\t: ${obj.tinggi}`);
console.log(`obj.luas\t: ${obj.luas()}`);
```



```
riandra@riandra-alkautsar: ~/De
File Edit View Search Terminal Help
riandra@riandra-alkautsar:~/Desktop$ node objek.js
Sebelum nilai properti diubah
obj.alas      : 4
obj.tinggi    : 5
obj.luas      : 10

Setelah nilai properti diubah
obj.alas      : 10
obj.tinggi    : 8
obj.luas      : 40
```

#### Sintaks Alternatif untuk Membuat Objek

```
function Segitiga(a, t) {
  this.alas = a;
  this.tinggi = t;
}

var obj = {alas: 4,tinggi: 5,luas: function(){
  return this.alas * this.tinggi / 2;
}
}

console.log(`Sebelum nilai properti diubah`);
console.log(`obj.alas\t: ${obj.alas}`);
console.log(`obj.tinggi\t: ${obj.tinggi}`);
console.log(`obj.luas\t: ${obj.luas()}`);

obj.alas = 10;
obj.tinggi = 8;
console.log(`\nSetelah nilai properti diubah`);
console.log(`obj.alas\t: ${obj.alas}`);
console.log(`obj.tinggi\t: ${obj.tinggi}`);
console.log(`obj.luas\t: ${obj.luas()}`);
```



```
riandra@riandra-alkautsar: ~
File Edit View Search Terminal Help
riandra@riandra-alkautsar:~/Desktop$ node objek1.js
Sebelum nilai properti diubah
obj.alas      : 4
obj.tinggi    : 5
obj.luas      : 10

Setelah nilai properti diubah
obj.alas      : 10
obj.tinggi    : 8
obj.luas      : 40
```

## 4. Kelas

### Membuat Objek

```
var readline = require('readline-sync');
class Segitiga {
  constructor(a, t) {
    // mendefinisikan alas dan tinggi
    this.alas = a;
    this.tinggi = t;
  }
  luas() {
    return this.alas * this.tinggi / 2;
  }
}
// membaca data alas dan tinggi
var a = parseFloat(readline.question('Masukan alas: '));
var t = parseFloat(readline.question('Masukan tinggi: '));
// membuat objek dari kelas segitiga
var obj = new Segitiga(a, t);
console.log(`\nSetelah nilai properti diubah`);
console.log(`obj.alas\t: ${obj.alas}`);
console.log(`obj.tinggi\t: ${obj.tinggi}`);
console.log(`obj.luas\t: ${obj.luas()}`);
```

riandra@riandra-alkautsar: ~/Desktop

File Edit View Search Terminal Help

riandra@riandra-alkautsar:~/Desktop\$ node class.js

Masukan alas: 3.0

Masukan tinggi: 4.5

Setelah nilai properti diubah

obj.alas : 3

obj.tinggi : 4.5

obj.luas : 6.75

### Metode Statis

```
var readline = require('readline-sync');
class Segitiga {
  constructor(a, t) {
    // mendefinisikan properti
    this.alas = a;
    this.tinggi = t;
  }
  luas() {
    return this.alas * this.tinggi / 2;
  }
  // metode statis
  static buatObjek(a, t) {
    return new Segitiga(a, t);
  }
}
// membaca data alas dan tinggi
var a = parseFloat(readline.question('Masukan alas: '));
var t = parseFloat(readline.question('Masukan tinggi: '));
// memanggil metode statis
var obj = Segitiga.buatObjek(a, t);
console.log(`\nSetelah nilai properti diubah`);
console.log(`obj.alas\t: ${obj.alas}`);
console.log(`obj.tinggi\t: ${obj.tinggi}`);
console.log(`obj.luas\t: ${obj.luas()}`);
```

riandra@riandra-alkautsar: ~/Desktop

File Edit View Search Terminal Help

riandra@riandra-alkautsar:~/Desktop\$ node class1.js

Masukan alas: 3.0

Masukan tinggi: 4.5

Setelah nilai properti diubah

obj.alas : 3

obj.tinggi : 4.5

obj.luas : 6.75

## 5. Penanganan Eksepsi

Hasil program ke-1 (tidak terjadi eksepsi):

```
var readline = require ('readline-sync');

class DivisionByZeroError extends Error {
  constructor (msg) {
    super(msg);
    this.name = this.constructor.name
    this.message = msg;
  }
  getMessage() {
    return this.message;
  }
}

var a = parseFloat(readline.question("Masukan nilai a: "));
var b = parseFloat(readline.question("Masukan nilai b: "));

try {
  if (b == 0) {
    throw new DivisionByZeroError("KESALAHAN: Terjadi pembagian dengan nol");
  }

  var c = a / b;
  console.log (`${a} / ${b} = ${c}`);
} catch (e) {
  console.log (e.getMessage());
}

riandra@riandra-alkautsar: ~/Desktop
File Edit View Search Terminal Help
riandra@riandra-alkautsar:~/Desktop$ node eksepsi.js
Masukan nilai a: 1
Masukan nilai b: 2
1 / 2 = 0.5
```

Hasil program ke-2 (terjadi eksepsi):

```
var readline = require ('readline-sync');

class DivisionByZeroError extends Error {
  constructor (msg) {
    super(msg);
    this.name = this.constructor.name
    this.message = msg;
  }
  getMessage() {
    return this.message;
  }
}

var a = parseFloat(readline.question("Masukan nilai a: "));
var b = parseFloat(readline.question("Masukan nilai b: "));

try {
  if (b == 0) {
    throw new DivisionByZeroError("KESALAHAN: Terjadi pembagian dengan nol");
  }

  var c = a / b;
  console.log (`${a} / ${b} = ${c}`);
} catch (e) {
  console.log (e.getMessage());
}

riandra@riandra-alkautsar: ~/Desktop
File Edit View Search Terminal Help
riandra@riandra-alkautsar:~/Desktop$ node eksepsi.js
Masukan nilai a: 1
Masukan nilai b: 0
KESALAHAN: Terjadi pembagian dengan nol
```