Laporan Praktikum Modul 6 Pemrograman Javascript Level Basic

Disusun oleh :

Nama : Riandra Alkautsar

Email : riandra.alkautsar@gmail.com

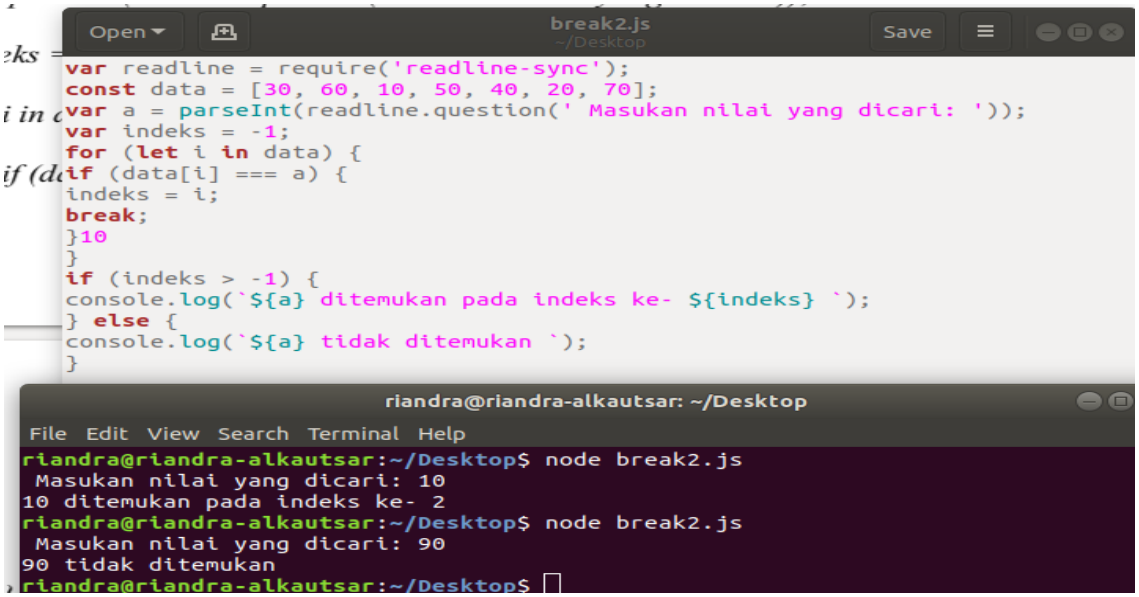
I. Tujuan Praktikum

Peserta didik diharapkan dapat memahami konsep javascript dasar diantaranya :
1.Pernyataan loncat
2.Fungsi
3.Tipe Objek
4.Kelas
5.Penanganan Eksepsi

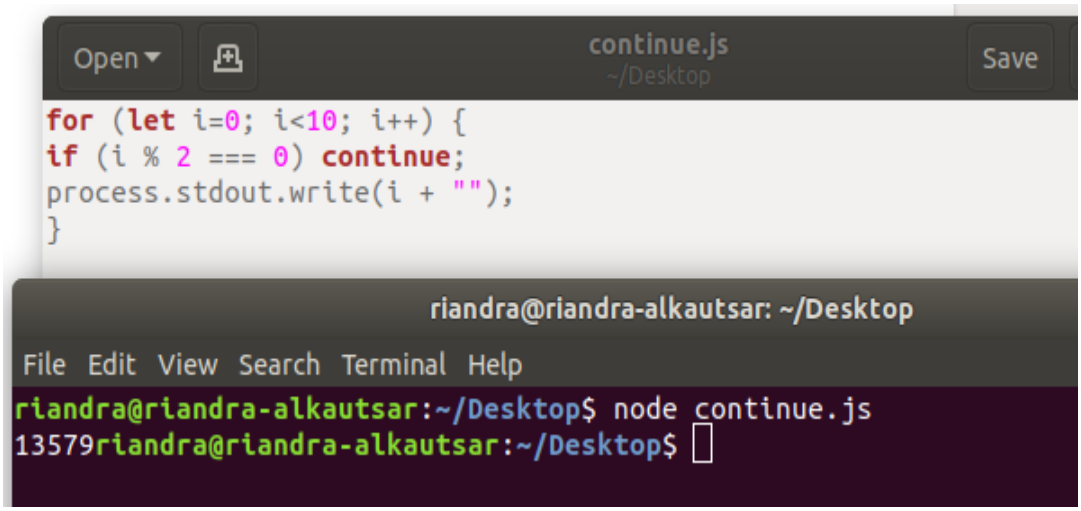
II. Hasil Praktikum dan Pembahasan

1. Pernyataan loncat

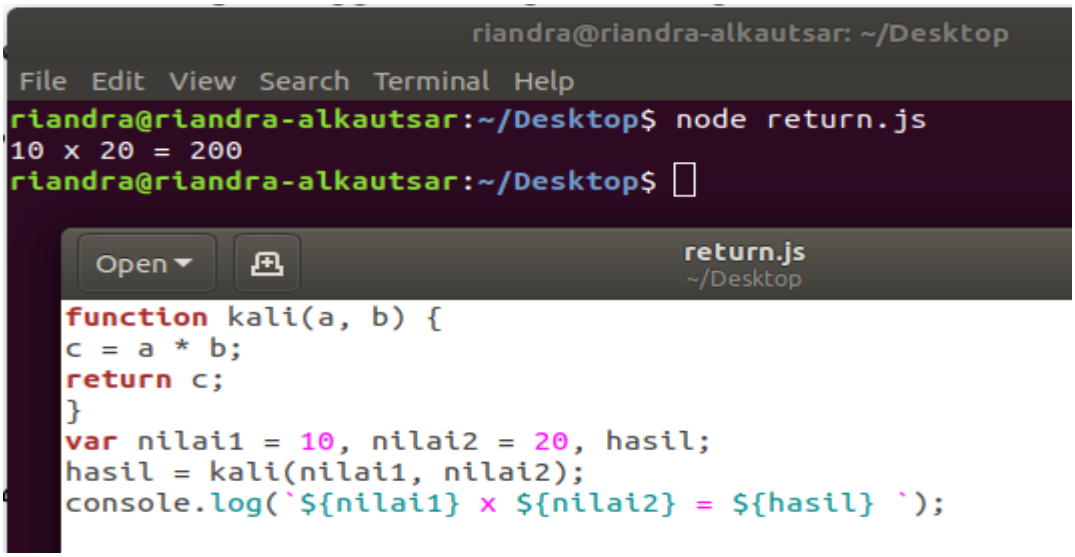
a. Perintah break



b. Perintah continue

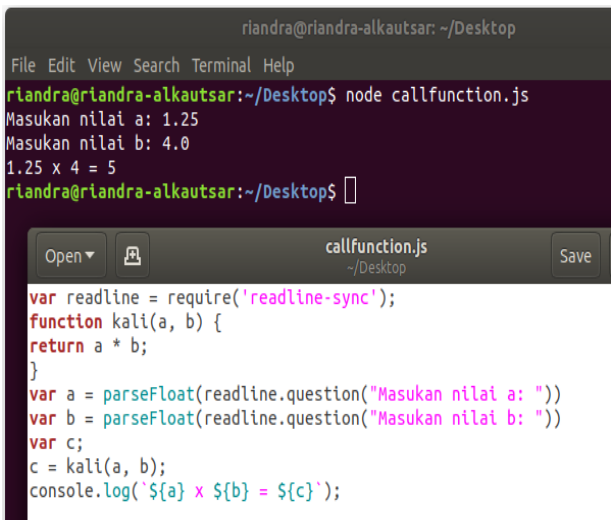


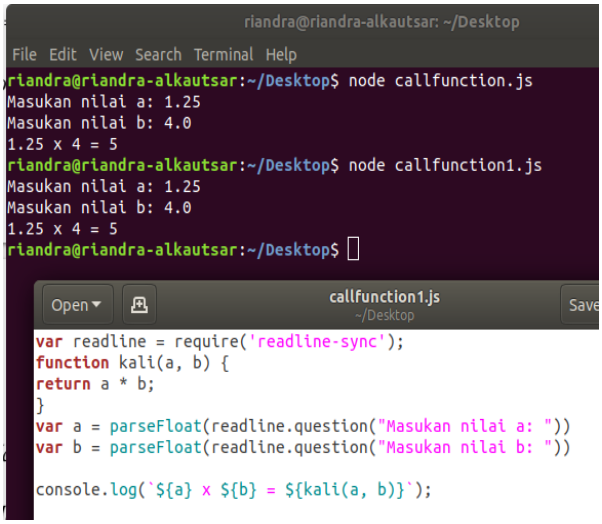
c. Perintah return



2. Fungsi

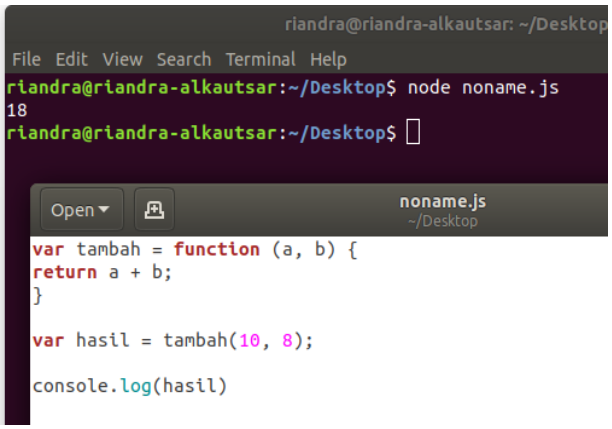
Memanggil Fungsi

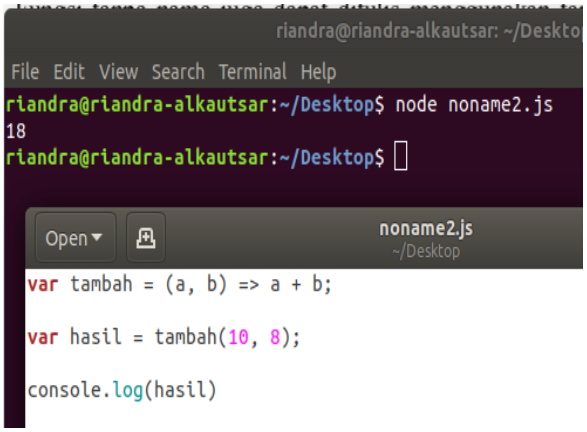




Parameter Fungsi

Fungsi tanpa Nama





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
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
```