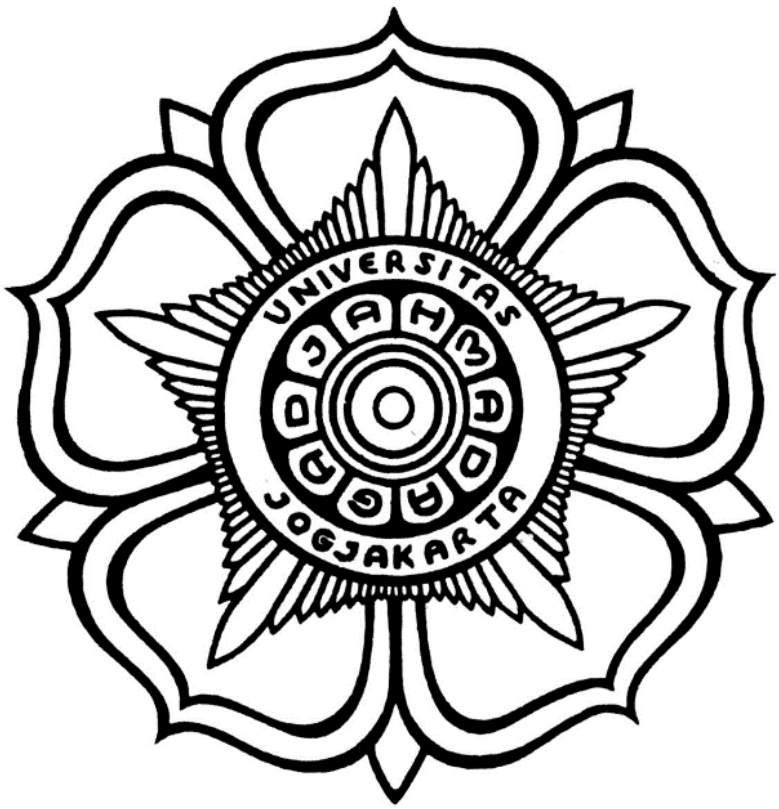
LAPORAN PRAKTIKUM
PEMROGRAMAN WEB III
PERTEMUAN KE-4



Oleh:
Maulana Adam Sahid (18/431735/SV/15706)

D3-KOMPUTER DAN SISTEM INFORMASI
DEPARTEMEN TEKNIK ELEKTRO DAN INFORMATIKA
SEKOLAH VOKASI – UNIVERSITAS GADJAH MADA
2021

Laporan Praktikum PWEB3A 2021

A. Mengenali Tipe Objek Variabel

Dalam Javascript terdapat kelas-kelas objek yang sudah disediakan secara default yaitu Number, String, dan Array. Dengan menggunakan kelas-kelas tersebut, maka suatu variabel tsb. akan dianggap sebagai variabel bertipe data object. Karena pada dasarnya kelas-kelas tersebut dalam Javascript adalah sebuah objek. Objek sendiri adalah suatu tipe data abstrak yang dibuat sendiri oleh developer yang didalamnya terdapat satu atau lebih properties maupun methods.

```
10  var a = new Number(9);
11  console.log(typeof a);
12  let b = 4;
13  console.log(typeof b);
14  let c = new String('Javascript');
15  console.log(typeof c);
16  let d = 'Javascript';
17  console.log(typeof d);
18  let e = new Array(10,20,40);
19  console.log(typeof e);
20  let f = [11,22,33];
21  console.log(typeof f);
22
23  console.log(a instanceof Number);
24  console.log(b instanceof Object);
25  console.log(c instanceof String);
26  console.log(c instanceof Object);
```

```
object                  Awal.html:11
number                  Awal.html:13
object                  Awal.html:15
string                  Awal.html:17
object                  Awal.html:19
object                  Awal.html:21
true                    Awal.html:23
false                   Awal.html:24
true                    Awal.html:25
true                    Awal.html:26
```

Dari contoh kode di atas dapat diketahui bahwa ada perbedaan variabel “a” dengan variabel “b” dan variabel “c” dengan “d”. Variabel “b” ketika dicheck tipenya ternyata ia adalah number sedangkan “a” ketika dicheck bertipe object padahal keduanya sama-sama diinputkan sebuah angka/number. Hal ini karena pada variabel “a” yang dinputkan adalah sebuah objek yang mengambil template dari kelas bernama “Number” terdahulu baru kemudian di dalam objek tersebut didefinisikan sebuah properti berupa angka. Sehingga variabel “a” memiliki tipe data sebagai objek yang didalamnya diisikan sebuah properti angka primitif, yaitu 9. Sedangkan pada variabel “b” angka/number 4 diinputkan secara langsung pada “b” sehingga “b” bukanlah bertipe objek melainkan number secara langsung. Karena ia diberikan value berupa angka primitif secara langsung.

Begitupula yang terjadi pada “c” dan “d” yang satu diperlakukan sebagai objek sedangkan yang lain lagi diperlakukan sebagai value primitif. Pada “c” kata “Javascript” ketika masuk ke dalam konstruktor dari kelas “String” dipecah tiap-tiap charnya kemudian disebar sebagai value ke tiap-tiap properties yang berurutan sesuai dengan indeks masing-masing char. Sedangkan pada variabel “d” kata “Javascript” diperlakukan sebagai value string biasa yang terdiri atas array of chars yang diurutkan oleh indeks secara langsung.

```
28  console.log(a);
29  console.log(b);
30  console.log(c);
31  console.log(d);
32  console.log(e);
33  console.log(f);
```

```
▸ Number {9}                    Awal.html:28
4                               Awal.html:29
▸ String {"Javascript"}         Awal.html:30
Javascript                      Awal.html:31
▸ (3) [10, 20, 40]              Awal.html:32
▸ (3) [11, 22, 33]              Awal.html:33
```

Sedangkan untuk kasus “e” dengan “f” ternyata tidaklah berbeda. Karena pada dasarnya ketika kita mendeklarasikan array seperti cara “f” itu sejatinya jugalah sedang membuat sebuah objek berkelas Array seperti pada cara pendeklarasian variabel “e”.

B. Membuat Tipe Objek

Dalam Javascript kita dapat membuat objek dengan dua cara yaitu dengan membuat fungsi pendefinisi objek sebagai template/blueprint atau dengan langsung mendeklarasikan objek di dalam suatu variabel.

```
36  //Membuat Objek
37
38  //Membuat Objek dengan fungsi pendefinisi/tempalte/blueprint terlebih dahulu
39  //Mendefinisikan properties suatu objek
40  function segitiga(a,t) {
41    this.alas = a;
42    this.tinggi = t;
43  }
44
45  //Mendefinisikan methods soatu objek
46  segitiga.prototype.luas=function () {
47    return this.alas * this.tinggi /2;
48  }
49
50  //Membuat dan menggunakan objek
51  var obj = new segitiga(4,5);
52
53  console.log(`Alas : ${obj.alas}`);
54  console.log(`Tinggi : ${obj.tinggi}`);
55  console.log(obj.luas());
56  console.log(obj);
```

```
Alas : 4                                  Awal.html:53
Tinggi : 5                                Awal.html:54
10                                        Awal.html:55
▶ segitiga {alas: 4, tinggi: 5}           Awal.html:56
```

B.1 Contoh membuat dan menggunakan objek dengan mendefinisikan blueprintnya terlebih dahulu

```
58  //Membuat Objek dengan Langsung Memasukkan ke dalam Sebuah Variabel
59  var objek = {
60    alas:7,
61    tinggi:12,
62    luas:function () {
63      return this.alas*this.tinggi/2;
64    }
65  }
66
67  console.log(`alasnya : ${objek.alas}`);
68  console.log(`tingginya : ${objek.tinggi}`);
69  console.log(`luasnya : ${objek.luas()}`);
```

```
alasnya : 7                               Awal.html:67
tingginya : 12                            Awal.html:68
luasnya : 42                              Awal.html:69
```

B.2 Contoh membuat objek dengan langsung memasukkan dalam suatu variabel dan penggunaannya

C. Kelas

Dalam ES6 kini Javascript sudah mendukung adanya pembuatan suatu kelas sebagai fasilitas untuk mendukung OOP. Sehingga kita dapat mengimplementasikan OOP pada Javascript sebagaimana OOP dalam bahasa pemrograman lainnya.

Pada Javascript ES6 kita dapat membuat kelas dengan menuliskan tipe fungsinya terlebih dahulu yaitu “class” kemudian diikuti dengan nama kelas yang ingin kita buat, misalnya “Segitiga”. Kemudian dilanjutkan dengan dua kurung kurawal pembuka dan penutupnya.

Untuk mendefinisikan konstruktornya, kita dapat mendefiniskan dengan mendefinisikan method “contructor” dikuti dengan kurung buka dan tutup yang di dalamnya berisi selarik parameter-parameter konstruktor kelas kemudian dikuti dengan kurung kurawal yang di dalamnya kita bisa definisikan atribut-atribut kelas tsb.

Sedangkan dalam membuat suatu method dalam kelas di ES6, kita dapat membuatnya cukup dengan mendefinisikan fungsi method tsb. dalam scope kelas yang kita buat tersebut setingkat/selevel dengan “constructor” pada kelas tersebut. Untuk lebih jelasnya dpat melihat pada tangkapan layar berikut.

```
71  //Membuat class
72  class Segitiga{
73    constructor(a,t){
74      this.alas=a;
75      this.tinggi=t;
76    }
77
78    luas(){
79      return this.alas*this.tinggi/2;
80    }
81  }
82
83  var obj_1 = new Segitiga(4,5);
84  console.log(`Luas segitiga ini adalah ${obj.luas()}`);
```

```
Luas segitiga ini adalah 10                    Awal.html:84
```

Untuk membuat objek baru dengan kelas tersebut dan menggunakannya kita dapat melakukan seperti ketika kita membuat dan memanggil sebuah objek dengan cara pada screenshot B.1 sebelumnya.

D. Pewarisan

Pewarisan/inheritance adalah salah satu konsep dalam OOP yang mengizinkan kita untuk membuat suatu kelas dengan menwarisi kelas atribut dan method dari kelas lain. Kelas lain yang diwariskan atribut dan method disebut dengan super class (kelas induk). Sedangkan kelas baru yang mewarisi atribut dan method dari superclass itu disebut dengan subclass (kelas turunan).

Untuk menerapkan pewarisan dalam Javascript kita dapat dengan membuat kelas seperti biasa. Kemudian setelah menuliskan nama subclass mesti dilanjutkan dengan menggunakan kata kunci extends kemudian diikuti oleh nama superclass.

```
86  //Membuat Kelas Turunan (Subclass) dan penerapan Inheritance
87  class SegitigaWarna extends Segitiga{
88    constructor(a,t,w){
89      super(a,t);
90      this.warna=w;
91    }
92
93    cetakProperti(){
94      console.log(super.luas());
95
96      console.log(`warnanya adalah ${this.warna}`);
97    }
98  }
99
100 var obj_2 = new SegitigaWarna(6,5,'Hijau');
101 obj_2.cetakProperti();
```

```
15                                          Awal.html:94
warnanya adalah Hijau                       Awal.html:96
```

E. Tugas Prakikum

1. Membuat Kelas BangunDatar yang Mengakomodir Persegi Panjang dan Lingkaran

Dalam membuat kelas BangunDatar kita buat seperti kelas biasa pada umumnya dengan parameter konstruktornya adalah “tipe”, “a”, dan “b”. Kemudian di dalam konstruktornya diberikan terlebih dahulu percabangan untuk mengkategorikan argumen apakah yang dimasukkan dalam parameter “tipe” barulah kemudian di dalam kondisi-kondisi percabangan tersebutlah didefinisikan atribut-atribut suatu objek yang dibuat dengan kelas BangunDatar tersebut. Kemudian setelah menentukan pendefinisian atribut-atribut kelas dalam konstruktor, selanjutnya adalah mendefinisikan method getLuas() seperti contoh berikut.

```
//TUGAS
class BangunDatar{

  constructor(tipe, a, b){
    if(tipe=="[]"){
      this.tipe=tipe;
      this.panjang = a;
      this.lebar = b;
      this.keterangan ="Ini Persegi";
    }else if (tipe=="O") {//Radius O menggunakan rumus Elips yang lebih detail
      this.tipe=tipe;
      this.radius1 = a;
      if(b!=undefined){
        this.radius2 = b;
      }else{
        this.radius2 = this.radius1;
      }
      this.keterangan = "Ini Lingkaran";
    }else{
      this.tipe = tipe;
      this.keterangan = "Ini bukan lingkaran ataupun persegi";
    }
  }

  getLuas(){
    if(this.tipe=="[]"){
      return this.panjang*this.lebar;
    }else if(this.tipe=="O"){
      return this.radius1*this.radius2*2/7;
    }else{
      console.log(this.keterangan);
    }
  }
}
```

2. Membuat Kelas Turunan dari Kelas BangunDatar dengan Nama BangunRuang Berisikan Volume Balok dan Tabung

Untuk membuat kelas BangunRuang yang mewarisi kelas BangunDatar kita dapat menuliskan kelas sebagaimana membuat kelas pada umumnya. Tetapi setelah pendefinisian nama kelas tersebut diikuti dengan keyword “extends” dan nama superclassnya, yaitu BangunDatar. Kemudian pada konstruktornya kita berikan tiga parameter seperti pada BangunDatar kemudian ditambah lagi satu parameter, yaitu “c”, untuk mendefinisikan atribut “tinggi” bangun kelas BangunRuang. Setelah itu di dalam operasi konstruktornya mesti dilakukan “super” terhadap variabel-variabel parameter yang ada dalam kelas BangunDatar dan juga pada kelas BangunRuang. Fungsi “super” adalah untuk melakukan overriding pada variabel-variabel parameter tersebut.

```
139  class BangunRuang extends BangunDatar{
140
141    constructor(tipe,a,b,c){
142      super(tipe,a,b);
143      this.tinggi = c;
144
145      this.keterangan2D=this.keterangan;
146    }
147
148    getVolume(){
149      if(this.tipe=="[]"||this.tipe=="O"){
150        return this.getLuas()*this.tinggi;
151      }else{
152        this.showKet2D();
153        return `tidak diketahui.`
154      }
155    }
156
157    showKet2D(){
158      console.log(`${this.keterangan2D}`);
159    }
160  }
```

Sebagai testing kita bisa melakukan koding seperti berikut:

```
162  var bangunDatar1 = new BangunDatar("[]", 4, 6)
163  var bangunDatar2 = new BangunDatar("O", 4)
164
165  console.log(bangunDatar1);
166  console.log(bangunDatar1.getLuas());
167  console.log(bangunDatar2);
168  console.log(bangunDatar2.getLuas());
169
170  var bangunRuang1 = new BangunRuang("[]",2,2,2)
171  var bangunRuang2 = new BangunRuang("O",4,null,5)
172  var bangunRuang3 = new BangunRuang("sak",2,1,2)
173
174  console.log(bangunRuang1);
175  console.log(bangunRuang2);
176
177  console.log(`Volume bangun ruang pertama adalah ${bangunRuang1.getVolume()}`);
178  console.log(`Volume bangun ruang kedua adalah ${bangunRuang2.getVolume()}`);
179  console.log(`Volume bangun ruang ketiga adalah ${bangunRuang3.getVolume()}`);
```

```
▶ BangunDatar {tipe: "[]", panjang: 4, lebar: 6, keterangan: "Ini Persegi"}                Awal.html:165
24                                                                                          Awal.html:166
▶ BangunDatar {tipe: "O", radius1: 4, radius2: 4, keterangan: "Ini Lingkaran"}              Awal.html:167
4.571428571428571                                                                           Awal.html:168
▶ BangunRuang {tipe: "[]", panjang: 2, lebar: 2, keterangan: "Ini Persegi", tinggi: 2, …}   Awal.html:174
▶ BangunRuang {tipe: "O", radius1: 4, radius2: 4, keterangan: "Ini Lingkaran", tinggi: 5, …} Awal.html:175
Volume bangun ruang pertama adalah 8                                                        Awal.html:177
Volume bangun ruang kedua adalah 22.857142857142854                                         Awal.html:178
Ini bukan lingkaran ataupun persegi                                                         Awal.html:158
Volume bangun ruang ketiga adalah tidak diketahui.                                          Awal.html:179
```