

Tugas 01

**SISTEM INFORMASI GEOGRAFIS  
JAVASCRIPT**



Disusun oleh:

Nama : Muhammad Abdul Ghofar

NIM : L200140033

Kelas : A

**PRODI INFORMATIKA  
FAKULTAS KOMUNIKASI DAN INFORMATIKA  
UNIVERSITAS MUHAMMADIYAH SURAKARTA  
2017**

## **TUGAS 01 – Belajar Javascript**

### **1. Daftar Tipe Data**

JavaScript memiliki tipe dinamis. Ini berarti bahwa variabel yang sama dapat digunakan untuk menyimpan jenis data yang berbeda. Di dalam Javascript tipe data terbagi menjadi 2, yaitu tipe data dasar (primitif), tipe data objek.

#### **A. Tipe data dasar (Primitif),**

Tipe data primitif adalah tipe data sederhana tunggal tanpa sifat tambahan dan metode.

- a. Number (angka) s
- b. String
- c. Boolean
- d. Null
- e. Undefined

#### **B. Tipe data objek**

- a. Array
- b. Object

### **Pembahasan:**

#### **A. Tipe data dasar**

##### **a. NUMBER**

Pada dasarnya JavaScript hanya mengenal dua macam tipe numerik, yaitu bilangan bulat (integer) dan bilangan pecahan (real/float). Tetapi javascript itu tidak seperti bahasa pemrograman lain umumnya, karena tipe data number (angka) javascript tidak membedakan antara bilangan bulat dengan angka desimal, atau lebih tepatnya tidak membedakan antara bilangan integer dengan float.

Integer mewakili bilangan bulat sedangkan float mewakili nilai-nilai pecahan. Tipe ini dapat dimodifikasi dengan operator matematika.

Contoh:

```
<script type="text/javascript">
    var usia=20;
    var ratausia=25.5;
</script>
```

#### **b. STRING**

String adalah rangkaian teks yang dapat berupa sembarang karakter. Untuk menuliskan tipe data string ini menggunakan tanda kutip tunggal atau ganda.

Contoh:

```
<script type="text/javascript">
    var namaDepan='Muhammad';
    var namaBelakang='Ghofar';
    var usia='20';
</script>
```

#### **c. BOOLEAN**

Tipe data boolean hanya terdiri dari dua nilai yaitu *true* dan *false*. Untuk membuat tipe data Boolean cukup dengan menggunakan kata kunci *true* atau *false*.

Tipe data boolean juga bisa di bilang juga hasil yang didapat dari operasi perbandingan.

contoh:

```
<script>
    var x=true; // x bernilai benar
    var y=false; // y bernilai salah
    9>5 //kondisi ini menghasilkan nilai benar
    9<5 //kondisi ini menghasilkan nilai salah
</script>
```

#### **d. NULL**

Tipe Null digunakan untuk merepresentasikan variabel yang tidak diberi nilai awal (inisialisasi).

Contoh:

```
<script type="text/javascript">
    Var X = null;
</script>
```

#### e. UNDEFINED

Dalam JavaScript, variabel tanpa nilai, memiliki nilai terdefinisi. Undefined ini juga tidak terdefinisi.

Contoh:

```
<script type="text/javascript">
    var x;
    person = undefined;
</script>
```

### B. Tipe data objek

#### a. ARRAY

Array adalah kumpulan dari nilai-nilai. Masing-masing nilai dapat berupa variabel atau suatu nilai. Untuk membuat sebuah array menggunakan statement *new Array()*.

Contoh:

```
<script type="text/javascript">
    var orang=new Array('Javas','Ghofar',20,185,80);
</script>
```

Untuk mengakses setiap nilai array menggunakan indeks array dengan sintaks *namaarray[indeks]*. Indeks array dimulai dari 0 bukan 1. Jadi untuk mengakses nilai pertama array menggunakan statement *orang[0]* yang akan menghasilkan nilai 'Javas'. Array secara otomatis akan memberikan informasi khusus yaitu panjang dari array. Informasi ini dapat diakses menggunakan properti *length*. Jadi, jika diberikan perintah *hewan.length*, maka akan dihasilkan nilai 5. Array sangat tepat jika digunakan untuk merepresentasikan kumpulan nilai-nilai yang memiliki suatu kesamaan dan Javascript menyediakan berbagai fungsi untuk memanipulasi array seperti *sorting*, *filtering*, *searching*, dan sebagainya.

## b. OBJECT

Object adalah suatu konstruksi didalam pemrograman yang menggambarkan atau memodelkan sesuatu hal yang nyata, misalnya manusia. Object memiliki properti dan metode. Properti menjelaskan karakteristik object sedangkan metode merupakan aksi yang dapat dilakukan object. Untuk membuat suatu object menggunakan statement *new Object()*. Contoh berikut ini memberikan gambaran tentang tipe data object dalam javascript.

Contoh:

```
<script type="text/javascript">
    var person = {firstName:"JMuhammad", lastName:"ghofar", age:20};
</script>
```

## 2. Fungsi Dan Kelompoknya

Pada Javascript terdapat 2 jenis fungsi, yaitu fungsi built-in atau fungsi bawaan dari Javascript dimana tinggal dipergunakan saja dan fungsi buatan sendiri. Fungsi buatan sendiri dipergunakan untuk mempermudah pengerjaan perintah yang berulang-ulang, sehingga fungsi buatan sendiri harus didefinisikan terlebih dahulu agar dapat dikenali oleh Javascript.

### 1. Function Declaration

merupakan cara penulisan fungsi yang umumnya kita gunakan pada bahasa pemrograman imperatif dan berorientasi objek. Pada deklarasi fungsi, kita langsung memberikan nama fungsi sesuai dengan sintaks yang disediakan Javascript. fungsi dinyatakan tidak dieksekusi segera. Mereka "disimpan untuk digunakan nanti", dan akan dieksekusi kemudian, ketika mereka dipanggil (dipanggil).

Sintax:

```
function functionName(parameters) {
    code to be executed
}
```

## 2. Function Expression

merupakan cara pembuatan fungsi yang memperbolehkan kita melewati nama fungsi. Fungsi yang dibuat tanpa nama dikenal dengan sebutan fungsi anonim atau fungsi lambda. Ekspresi fungsi dapat disimpan dalam variabel:

```
var x = function (a, b) {return a * b};
```

## 3. Function Parameters dan Arguments

Pendefinisian function yang menggunakan parameter. Nilai parameter akan dimasukkan ketika function dipanggil.

```
functionName (parameter1, parameter2, parameter3) {  
    kode yang akan dieksekusi  
}
```

Fungsi parameter adalah nama-nama yang tercantum dalam definisi fungsi.

Fungsi argumen adalah nilai-nilai nyata dilewatkan ke (dan diterima oleh) fungsi.

## 4. Function Invocation

Meminjam fungsi sebagai fungsi global, menyebabkan nilai ini menjadi obyek global. Menggunakan objek jendela sebagai variabel dapat dengan mudah crash program Anda.

```
function myFunction(a, b) {  
    return a * b;  
}  
myFunction(10, 2);
```

## 5. Function Return

Mengembalikan fungsi kapanpun, dan fungsi akan segera berhenti ketika kata kunci return ditemukan. Kita dapat mengembalikan fungsi kapanpun, dan fungsi akan segera berhenti ketika kata kunci return ditemukan.

## 6. Closure

Salah satu kelebihan utama *function scope* adalah fungsi yang ada di dalam fungsi lainnya juga memiliki akses terhadap semua nilai-nilai yang dimiliki fungsi penampungnya. Ketika sebuah fungsi menggunakan dan bergantung kepada nilai yang ada di luar dirinya, fungsi tersebut dikenal dengan nama *closure*.

```
var a = 4;

function myFunction() {
  return a * a;
}
```

## 7. Function Hoisting

Function declaration dan variable declaration otomatis berpindah (“hoisted”) secara tak kasat mata ke posisi paling atas di dalam scope yang melakukan deklarasi oleh interpreter Javascript.

## 3. Bagaimana cara membuat sebuah fungsi dalam javascript

Untuk mendefinisikan sebuah fungsi, sintaks yang digunakan:

```
function functionname(var1,var2,...,varX)
{
  pernyataan-pernyataan fungsi;
}
```

Parameter *var1, var2*, adalah variabel atau nilai yang dilewatkan kedalam fungsi. Simbol kurung kurawal buka dan tutup menandakan awal dan akhir dari fungsi.

Cara penulisan fungsi seperti ini dikenal dengan nama *function declaration*, atau deklarasi fungsi. Terdapat empat komponen yang membangun fungsi yang baru kita definisikan di atas, yaitu:

1. Kata kunci ***function***, yang memberitahu Javascript bahwa kita akan membuat fungsi.

2. Nama fungsi, dalam contoh di atas adalah ***functionname***. Dengan memberikan sebuah fungsi nama maka kita dapat merujuk ke fungsi tersebut dengan nama yang diberikan. Harus diingat bawa nama fungsi bersifat *opsional*, yang berarti fungsi pada Javascript tidak harus diberi nama. Kita akan membahas tentang hal ini lebih dalam nanti.
3. Daftar parameter fungsi, yaitu ***var1, var2,...,varx*** pada contoh di atas. Daftar parameter ini selalu dikelilingi oleh tanda kurung (**()**). Parameter boleh kosong, tetapi tanda kurung wajib tetap dituliskan. Parameter fungsi akan secara otomatis didefinisikan menjadi variabel yang hanya bisa dipakai di dalam fungsi. Variabel pada parameter ini diisi dengan nilai yang dikirimkan kepada fungsi secara otomatis.
4. Sekumpulan perintah yang ada di dalam kurung kurawal (**{}**). Perintah-perintah ini dikenal dengan nama badan fungsi. Badan fungsi dieksekusi secara berurut ketika fungsi dijalankan.