

# Javascript Lanjutan

Berikut ini akan dibahas tentang lebih mendalam lagi bagaimana kita menggunakan javascript sebagai salah satu Bahasa pemrograman client side.

## Tipe Data

Di javascript terdapat beberapa tipe data yang dapat digunakan yaitu: String, Number, Boolean, Array, Object.

```
<!DOCTYPE html>
<html>
<body>

<p id="demo"></p>

<script>
var carName = "Volvo XC60";
var build = 2014;
var types = ["Sport", "City Car", "Coupe"];
var person = {
  firstName : "John",
  lastName  : "Doe",
  age       : 50
};
</script>

</body>
</html>
```

Dari script diatas cobalah untuk menampilkan data dari beberapa tipe variable tersebut ke dalam elemen <p> dengan id = "demo". Pada text bergaris bawah data diambil dari variable yang ada di javascript.

Cara menuliskan di elemen p menggunakan:

```
document.getElementById("demo").innerHTML
```

John Doe, orang yang berusia 50 th itu pemilik Mobil Sport Volvo XC60 buatan tahun 2014

## Javascript Function

Function di javascript merupakan suatu block code yang menjalankan suatu perintah/algorithm tertentu.

```
function toCelsius(f) {  
    return (5/9) * (f-32);  
}  
document.getElementById("demo").innerHTML = toCelsius(77);
```

Dengan function kita dapat menggunakan suatu code yang sama untuk digunakan beberapa kali, dan function dapat juga dialikasikan ke dalam suatu variable.

```
var x = toCelcius(77);
```

## Javascript Objects

Pada object di javascript mempunyai suatu properties dan methods. Penulisan object sama seperti var person diatas menggunakan format **name:value**.

```
var person = {  
    firstName: "John",  
    lastName : "Doe",  
    age      : 50,  
    fullName : function() {  
        return this.firstName + " " + this.lastName;  
    }  
};
```

Dari contoh object diatas terdapat tambahan method fullName cara pemanggilan menggunakan script **person.fullName( )**, sedikit berbeda dengan pemanggilan properties yang pemanggilannya **person.firstName**.

Dengan method tersebut kita bias ubah code sebelumnya dimana menampilkan firstName + lastName diganti dengan fullName( ).

## Javascript Scope

Scope di javascript sama seperti pemrograman yang lain yaitu local dan global, dimana saat bersifat local dia berada di dalam suatu function dan terdapat keyword var di sebelum nama variable tersebut. Dan Global variable dapat di definisikan di luar function atau di dalam function tanpa awalan keyword.

```
myFunction();  
document.getElementById("demo").innerHTML =  
"The type of carName is " + myCar;  
  
function myFunction() {  
    var myCar = "Volvo";  
}
```

Dari script diatas ubahlah supaya carName menjadi global variable dan dapat diakses dari luar function.

## Javascript Array

Array di javascript digunakan untuk menyimpan beberapa data dalam satu variable. Dan di javascript, dalam satu variable array dapat menyimpan beberapa tipe data.

```
<p id="demo"></p>

<script>
var person = ["Jhon", "Doe", 50];
document.getElementById("demo").innerHTML = person;
```

Pada javascript tidak mendukung assosiatif array, jadi untuk index di array menggunakan number sedangkan untuk object menggunakan index berupa text.

Dalam melakukan perulangan di array atau object dapat menggunakan for, while atau do.. while, seperti script berikut ini.

```
for (i = 0; i < person.length; i++) {
    text += "<li>" + person[i] + "</li>";
}
```

```
var i = 0;
while(person[i]){
    text += "<li>" + person[i] + "</li>";
    i++;
}
```

## JavaScript Debugging

Kesalahan yang ada di javascript, baik berupa syntax error atau logical error sebagian besar tidak tampil di browser oleh sebab itu debugging code program di javascript dapat menggunakan tools yang ada dari browser.

Pada beberapa browser modern yang sudah support untuk debugging, kita dapat menggunakan script console.log() untuk menampilkan nilai di debugger.

```
<script>
a = 5;
b = 6;
c = a + b;
console.log(c);
</script>
```

Secara umum untuk mengaktifkan debugging di browser dengan menekan tombol F12 dan pilih menu "Console" debugger.