

Nama : Fikri Maulana  
Kelas ABP : IF-44-01 / IF-44-10  
NIM : 1301200239

## MODUL 4

### 4.1 Pengenalan JavaScript

#### 4.1.1 Sejarah Singkat JavaScript

Javascript merupakan Bahasa pemrograman scripting. Tujuan awal javascript adalah untuk mengendalikan program Java tetapi pada akhirnya komunitas javascript menggunakan Bahasa ini untuk tujuan lain, memanipulasi gambar dan isi dari dokumen HTML. Pada akhirnya tujuan utama javascript adalah “menghidupkan” dokumen HTML dengan mengubah konten statis menjadi dinamis dan interaktif. Bersamaan dengan perkembangan Internet dan dunia web yang pesat, Javascript akhirnya menjadi bahasa utama dan satu-satunya untuk membuat HTML menjadi interaktif di dalam browser.

#### 4.1.2 Prinsip Dasar Javascript

Terdapat beberapa prinsip dasar Bahasa Pemrograman Javascript yaitu sebagai berikut.

1. Javascript menjalankan perintah program baris demi baris.
2. Javascript memiliki model pemrograman fungsional yang sangat ekspresif
3. Javascript adalah Bahasa Pemrograman Berorientasi Objek
4. Pada Javascript program kompleks dipandang sebagai program – program yang saling berinteraksi.

### 4.2 Sintaks Umum pada JavaScript

#### 4.2.1 Tipe Data Dasar

1. Number = Tipe data berupa angka
2. String = Tipe data berupa huruf
3. Boolean = Tipe data berupa true / false
4. Object = Tipe data yang berbentuk objek
5. Function = Tipe data yang berbentuk fungsi yang dapat mereturn hasil
6. Null = Tipe data null ( kosong )

Berikut adalah contoh penulisan pada Bahasa javascript:

```

JS script.js > ...
1  //Fikri Maulana 1301200239
2
3  //TIPE DATA NUMBER
4  var number = 1301200239;
5
6  //TIPE DATA STRING
7  var string = "Fikri Maulana";
8
9  //TIPE DATA BOOLEAN
10 var boolean = true;
11
12 //TIPE DATA OBJECT
13 var object = {
14     nama : "Fikri",
15     nim : "1301200239"
16 }
17
18 //TIPE DATA FUNCTION
19 const function_1 = (a, b) => {
20     return a*b;
21 }
22
23 //TIPE DATA NULL
24 const tipenull = null;

```

#### 4.2.2 Variable

Variabel sebuah tempat untuk menyimpan data. Pada Javascript untuk membuat variabel menggunakan keyword :

1. Var = variable yang bisa diubah isi nya
2. Let = sama seperti var
3. Const = variable yang tidak dapat diubah isinya

berikut adalah contoh penulisan pada Bahasa javascript:

```

26 //Fikri Maulana 1301200239
27
28 //VARIABLE var
29 var nama = "Fikri";
30
31 //VARIABLE let
32 let nim = 1301200239;
33
34 //VARIABLE const
35 const umur = 20;

```

#### 4.2.3 Array

Array adalah sebuah tipe data yang digunakan untuk menampung banyak tipe data lainnya. Pembuatan array pada JavaScript dilakukan dengan kurung siku [].

```

37 //Fikri Maulana 1301200239
38
39 //ARRAY
40 var array = ["Fikri Maulana", "Sapriudin", "Adam Malik"]
41
42 //PEMANGGILAN ARRAY
43 array[0];
44
45 //MENAMBAHKAN ARRAY PADA AKHIR INDEX
46 array.push("Pikri");
47
48 //MENGHAPUS ARRAY PADA AKHIR INDEX
49 array.pop();
50
51 //MENGHAPUS ARRAY PADA AWAL INDEX
52 array.shift();
53
54 //MENAMBAHKAN ARRAY PADA AWAL INDEX
55 array.unshift("Index awl");
56
57 //MELIHAT JUMLAH ARRAY
58 array.length;

```

#### 4.2.4 Pengendalian Struktur

Javascript memiliki perintah – perintah untuk pengendalian struktur. Untuk percabangan menggunakan if-else sedangkan untuk perulangan bisa menggunakan for, for-in, while, dan do-while. Pada kode dibawah terdapat contoh percangan dan perulangan.

```
60  √ //FIKRI MAULANA 1301200239
61
62  //IF ELSE
63  var x = 10;
64  √ if (x > 5) {
65      console.log("Nilai x lebih besar dari 5.");
66  √ } else {
67      console.log("Nilai x tidak lebih besar dari 5.");
68  }
69
70  //FOR LOOP
71  √ for (var i = 0; i < 5; i++) {
72      console.log(i);
73  }
74
75  //WHILE
76  var i = 0;
77  √ while (i < 5) {
78      console.log(i);
79      i++;
80  }
```

#### 4.3 OOP pada JavaScript

OOP (Object-Oriented Programming) pada JavaScript adalah sebuah paradigma pemrograman yang memungkinkan programmer untuk menggunakan konsep objek dalam pembuatan program. Dalam OOP, program dipecah menjadi bagian-bagian yang lebih kecil yang disebut objek, yang dapat berinteraksi satu sama lain dan memiliki properti dan metode sendiri.

##### 4.3.1 Pembuatan Object pada JavaScript

Berikut dibawah ini adalah contoh pembuatan objek yang bernama biodata dan memiliki beberapa property yaitu nama, dan nim. Pada

property terdapat juga masing – masing valuenya.

```
82 //FIKRI MAULANA 1301200239
83
84 //OBJECT
85 var objek1 = {
86     nama : "Fikri Maulana",
87     nim : "1301200239"
88 }
```

#### 4.3.2 Akses Nilai Property

Penggunaan kurung siku ( [ ] ) setelah nama objek. Kurung siku harus diisi dengan nama properti yang memiliki tipe data string. Penggunaan tanda titik ( . ) setelah nama objek dengan nama properti, notasi ini merupakan notasi yang biasa digunakan pada bahasa pemrograman lainnya. Contoh seperti pada bawah ini :

```
82 //FIKRI MAULANA 1301200239
83
84 //OBJECT
85 var objek1 = {
86     nama : "Fikri Maulana",
87     nim : "1301200239",
88     "alamat rumah" : "Jalan Telkom"
89 }
90
91 //PEMANGGILAN KETIKA KEY TIDAK ADA SPASI
92 console.log(objek1.nama);
93
94 //PEMANGGILAN KETIKA KEY ADA SPASI
95 console.log(objek1["alamat rumah"])
```

#### 4.3.3 Prototype pada JavaScript

Pada JavaScript yang mengimplementasikan PBO, kita bisa menurunkan objek dengan menggunakan Object.create seperti berikut:

```

84 //OBJECT
85 var objek1 = {
86     nama : "Fikri Maulana",
87     nim : "1301200239",
88     "alamat rumah" : "Jalan Telkom"
89 }
90
91 //PEMANGGILAN KETIKA KEY TIDAK ADA SPASI
92 console.log(objek1.nama);
93
94 //PEMANGGILAN KETIKA KEY ADA SPASI
95 console.log(objek1["alamat rumah"]);
96
97 //PROTOTYPE
98 var objek2 = Object.create(objek1);
99 objek2.nama;

```

#### 4.4 Function

Pada Bahasa Pemrograman Javascript tentunya memiliki fungsi seperti halnya Bahasa pemrograman lainnya. Fungsi berfungsi untuk membungkus perintah spesifik sehingga ketika perintah tersebut dibutuhkan untuk menggunakannya hanya tinggal memanggil fungsi tersebut.

##### 4.4.1 Pembuatan Function JS

```

101 //FIKRI MAULANA 1301200239
102
103 //FUNGSI
104 function panggilNama(nama){
105     return `Halo selamat datang ${nama}`
106 }
107 console.log(panggilNama("Fikri"));

```

##### 4.4.2 Pemanggilan Function

```
101 //FIKRI MAULANA 1301200239
102
103 //FUNGSI
104 function panggilNama(nama){
105     return `Halo selamat datang ${nama}`
106 }
107
108 console.log(panggilNama("Fikri"));
109 var namaSaya = panggilNama("Fikri Maulana");
110 console.log(namaSaya);
```

#### 4.5 DOM Manipulation

DOM Manipulation pada JavaScript adalah teknik untuk mengubah atau memanipulasi struktur, tampilan, atau konten dari sebuah halaman web melalui JavaScript. DOM (Document Object Model) sendiri adalah representasi struktur halaman web dalam bentuk objek-objek JavaScript, yang dapat diakses dan dimanipulasi melalui kode JavaScript.

Sebelum manipulasi DOM :

**Fikri Maulana - 1301200239**

Tekan Tombol dibawah ini untuk mengubah warna:

Ubah

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, in
7      <title>Document</title>
8      <style>
9          .selected {
10             background-color: yellow;
11          }
12      </style>
13  </head>
14  <body>
15      <h1 id="title">Fikri Maulana - 1301200239</h1>
16      <p>Tekan Tombol dibawah ini untuk mengubah warna:</p>
17      <button id="btn">Ubah</button>
18      <script>
19          const btn = document.getElementById('btn');
20          const title = document.getElementById('title');
21
22          btn.addEventListener('click', () => {
23              title.style.color = 'aqua';
24              title.classList.add('selected');
25              btn.disabled = true;
26              btn.innerText = 'Sudah Berubah';
27          });
28      </script>
29  </body>
30  </html>

```

Setelah pencet button untuk manipulasi :



# Fikri Maulana - 1301200239

Tekan Tombol dibawah ini untuk mengubah warna:

Sudah Berubah

## 4.6 JQuery

jQuery adalah sebuah library Javascript yang dibuat oleh John Resig pada tahun 2006. jQuery memungkinkan manipulasi dokumen HTML dilakukan hanya dalam beberapa baris code

### 4.6.1 Efek hide/show

```
<!Doctype html>
<html>
  <head>
    <title>The jQuery Example</title>
    <script src = "https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>
    <script>
      $(document).ready(function() {
        $("#hide").click(function(){
          $("p").hide()
        });
        $("#show").click(function(){
          $("p").show()
        });
      });
    </script>
  </head>
  <body>
    <p>If you click on the "Hide" button, I will disappear</p>
    <button id="hide">Hide</button>
    <button id="show">Show</button>
  </body>
</html>
```

If you click on the "Hide" button, I will disappear

Hide

Show

Ketika button di tekan maka akan seperti ini

Hide

Show

### 4.6.2 Efek animasi

```

<!Doctype html>
<html>
  <head>
    <title>The jQuery Example</title>
    <script src = "https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>
    <script>
      $(document).ready(function() {
        $("button").click(function(){
          $("div").animate({
            height: 'toggle'
          });
        });
      });
    </script>
  </head>
  <body>
    <p>Click the button multiple times to toggle the animation.</p>
    <button>Start animation</button>
    <p>By default, all HTML elements have a static position, and cannot be moved.
      To manipulate the position, remember to first set the CSS position property of
      the element to relative, fixed, or absolute!</p>
    <div
      style="background: #98bf21; height: 100px; width: 100px; position: absolute;"></div>
  </body>
</html>

```

Click the button multiple times to toggle the animation.

Start animation

By default, all HTML elements have a static position, and cannot be moved. To manipulate the position, remember to first set the CSS position property of the element to relative, fixed, or absolute!



Click the button multiple times to toggle the animation.

Start animation

By default, all HTML elements have a static position, and cannot be moved. To manipulate the position, remember to first set the CSS position property of the element to relative, fixed, or absolute!

Ketika button animasi di tekan , maka kotak akan berubah