

Nama : Akmal Muhamad Firdaus

NIM : 1301204188

Laporan Praktikum Modul 4

Pendahuluan

Javascript, seperti namanya, merupakan bahasa pemrograman scripting. Dan seperti bahasa scripting lainnya, Javascript umumnya digunakan hanya untuk program yang tidak terlalu besar, biasanya hanya beberapa ratus baris. Javascript pada umumnya mengontrol program yang berbasis Java. Jadi memang pada dasarnya Javascript tidak dirancang untuk digunakan dalam aplikasi skala besar.

Prinsip dasar yang terdapat pada bahasa pemrograman javascript adalah sebagai berikut.

1. Javascript mendukung paradigma pemrograman imperatif (Javascript dapat menjalankan perintah program baris demi baris, dengan masing-masing baris berisi satu atau lebih perintah), fungsional (struktur dan elemen-elemen dalam program sebagai fungsi matematis yang tidak memiliki keadaan (state) dan data yang dapat berubah (mutable data)), dan orientasi objek (segala sesuatu yang terlibat dalam program dapat disebut sebagai "objek").
2. Javascript memiliki model pemrograman fungsional yang sangat ekspresif.
3. Pemrograman berorientasi objek (PBO) pada Javascript memiliki perbedaan dari PBO pada umumnya.
4. Program kompleks pada Javascript umumnya dipandang sebagai program-program kecil yang saling berinteraksi.

Pada eksplorasi berikut, saya menggunakan node js sebagai runtimenya

Tipe data pada JavaScript

variabel dalam Javascript merupakan sebuah tempat untuk menyimpan data sementara. Variabel dibuat dengan kata kunci let, var, const pada Javascript.

- **Number (bilangan)**

Number adalah tipe data yang digunakan untuk merepresentasikan bilangan, baik bilangan bulat (integer) maupun bilangan pecahan (float). Number di JavaScript dapat digunakan untuk melakukan operasi matematika seperti penjumlahan, pengurangan, perkalian, dan pembagian.

```
console.log("=== Start of number ===")
let x = 10;
let y = 3.14;
console.log(x); // Output: 10
console.log(y); // Output: 3.14
console.log("=== End of number ===\n")
```

- **String (serangkaian karakter)**

String adalah tipe data yang digunakan untuk merepresentasikan serangkaian karakter atau teks. String di JavaScript dapat dibuat dengan menggunakan tanda kutip tunggal ('...') atau tanda kutip ganda ("...").

```
console.log("=== Start of string ===")
let firstName = "Akmal";
let lastName = 'Firdaus';
console.log(firstName); // Output: "Akmal"
console.log(lastName); // Output: "Firdaus"
console.log("=== End of string ===\n")
```

- **Boolean (benar / salah)**

Boolean adalah tipe data yang digunakan untuk merepresentasikan nilai kebenaran dari suatu ekspresi. Boolean hanya memiliki dua nilai, yaitu true dan false. Boolean sering digunakan dalam kondisi percabangan (if-else), perulangan, dan operasi logika

```
console.log("=== Start of boolean ===")
let isMarried = true;
let isSingle = false;
console.log(isMarried); // Output: true
console.log(isSingle); // Output: false
console.log("=== End of boolean ===\n")
```

- Object

Object adalah tipe data yang digunakan untuk merepresentasikan suatu objek atau entitas yang memiliki beberapa properti (properties) dan metode (methods). Properti adalah variabel yang merepresentasikan sifat dari objek, sedangkan metode adalah fungsi yang dapat digunakan untuk memanipulasi objek atau melakukan tindakan tertentu.

```
console.log("=== Start of object ===")
let person = { firstName: "Akmal", lastName: "Firdaus" };
console.log(person); // Output: { firstName: "Akmal", lastName: "Firdaus" }
console.log("=== End of object ===\n")
```

- Function (fungsi)

Function adalah blok kode yang dirancang untuk melakukan tugas tertentu tertentu dan dapat dipanggil di tempat yang berbeda dalam program. Fungsi dapat digunakan untuk mengorganisir kode program, menghindari pengulangan kode, membuat kode lebih mudah dibaca dan dimengerti, serta meningkatkan modularitas dan reusable dari kode program.

```
console.log("=== Start of function ===")
function greeting(name) {
  console.log("Hello " + name);
}
greeting("Akmal") // Output: Hello Akmal
console.log("=== End of function ===\n")
```

- Array

Array merupakan sebuah tipe data yang digunakan untuk menampung banyak tipe data lainnya. Berbeda dengan tipe data object, array pada Javascript merupakan sebuah tipe khusus. Walaupun memiliki method dan properti, array bukanlah objek, melainkan sebuah tipe yang "mirip objek". Pembuatan array dalam Javascript dilakukan dengan menggunakan kurung siku ([]):

```
console.log("=== Start of array ===")
let fruits = ["lemon", "pisang", "semangka"];
console.log(fruits); // Output: ["lemon", "pisang", "semangka"];
console.log("=== End of array ===\n")
```

- Date

Date adalah objek yang merepresentasikan tanggal dan waktu dalam bentuk string atau angka. Objek Date memungkinkan untuk melakukan manipulasi tanggal dan waktu, seperti menambah atau mengurangi waktu, memformat tanggal dan waktu, serta mengambil informasi seperti hari, bulan, tahun, jam, menit, detik, dan milidetik.

```
console.log("=== Start of date ===")
let date = new Date();
console.log(date); // Output: 2023-03-12T06:20:57.759Z
console.log("=== End of date ===\n")
```

- RegExp (regular expression)

Regex adalah sebuah objek yang merepresentasikan pola atau pattern tertentu yang digunakan untuk mencocokkan dan pengambilan teks dari sebuah string. Dengan regex, kita dapat melakukan pencarian atau manipulasi teks dengan lebih efisien dan fleksibel.

```
console.log("=== Start of regex ===")
let regex = /akmal/;
console.log("akmal muhamad f.".match(regex)); // Output: [ 'akmal', index: 0, input: 'akmal muhamad f.', groups: undefined ]
console.log("=== End of regex ===\n")
```

- Null (tidak berlaku / kosong)

Null adalah sebuah nilai atau data yang menunjukkan keadaan kosong atau tidak ada nilai atau referensi yang valid. Null biasanya digunakan sebagai nilai default pada variabel atau objek yang belum diinisialisasi atau tidak memiliki nilai.

```
console.log("=== Start of null ===")
let nul = null;
console.log(nul); // Output: null
console.log("=== End of null ===\n")
```

- Undefined (tidak didefinisikan)

Undefined adalah sebuah tipe data yang menunjukkan bahwa sebuah variabel belum didefinisikan atau belum diberikan nilai. Nilai undefined biasanya diberikan oleh JavaScript

secara otomatis pada variabel yang belum diinisialisasi, atau pada fungsi yang tidak mengembalikan nilai apapun.

```
console.log("=== Start of undefined ===")
let unkn;
console.log(unkn); // Output: undefined
console.log("=== End of undefined ===\n")
```

Result

```
=== Start of number ===
10
3.14
=== End of number ===

=== Start of string ===
Akmal
Firdaus
=== End of string ===

=== Start of boolean ===
true
false
=== End of boolean ===

=== Start of array ===
[ 'lemon', 'pisang', 'semangka' ]
=== End of array ===

=== Start of object ===
{ firstName: 'Akmal', lastName: 'Firdaus' }
=== End of object ===

=== Start of function ===
Hello Akmal
=== End of function ===

=== Start of undefined ===
undefined
=== End of undefined ===

=== Start of date ===
2023-03-12T06:25:55.262Z
=== End of date ===

=== Start of null ===
null
=== End of null ===

=== Start of regex ===
[ 'akmal', index: 0, input: 'akmal muhamad f.', groups: undefined ]
=== End of regex ===
```

Pengendalian Struktur / Control Flow

Javascript memiliki perintah-perintah pengendalian struktur (control stucture) yang sama dengan bahasa dalam keluarga C. Perintah if dan else digunakan untuk percabangan, sementara perintah for, for-in, while, dan do-while digunakan untuk perulangan. Percabangan pada Javascript bisa dikatakan sama persis dengan C atau Java.

If dan else

“if” dan “else” adalah pernyataan dalam bahasa pemrograman yang digunakan untuk mengeksekusi kode secara kondisional. if digunakan untuk mengeksekusi sebuah blok kode jika suatu kondisi terpenuhi, sedangkan else digunakan untuk mengeksekusi blok kode yang berbeda jika kondisi tidak terpenuhi.

```
let age = 18;

if (age >= 18) {
  console.log("Anda sudah dewasa");
} else {
  console.log("Anda masih di bawah umur");
}
```

Anda sudah dewasa

For-in

For-in adalah sebuah perulangan yang digunakan untuk mengiterasi properti-properti dari sebuah objek pada JavaScript. Dalam setiap iterasinya, for-in akan mengambil satu properti pada objek yang diiterasi dan menjalankan perintah-perintah yang diberikan.

```
for (const key in person) {  
  console.log(`${key}: ${person[key]}`);  
}
```

firstName: Akmal
lastName: Firdaus

While

While adalah salah satu jenis perulangan pada JavaScript yang digunakan untuk melakukan perulangan pada sebuah blok perintah selama kondisi yang diberikan bernilai true. Ketika kondisi bernilai false, maka perulangan akan berhenti dan kontrol akan diteruskan ke blok perintah setelah perulangan

```
let i = 1;  
  
while (i <= 5) {  
  console.log(i);  
  i++;  
}
```

1
2
3
4
5

Do-While

Do-While adalah jenis perulangan pada JavaScript yang mirip dengan while. Perbedaannya adalah pada do-while, blok perintah akan dijalankan minimal satu kali, bahkan jika kondisi tidak memenuhi syarat untuk dilakukan perulangan.

```
let j = 1;  
do {  
  console.log(j);  
  j++;  
} while (j <= 5);
```

1
2
3
4
5

Note: Soruce Code berikut masih dalam 1 code base yang sama pada modul 2 – 3.

1. Modal Bootstrap : Studi case show & hide animation dan object pada js Kerangka Modal (HTML-nya)

```
<div class="modal fade" id="exampleModal" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="exampleModalLabel">Example Object in JS</h5>
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
      </div>
      <div class="modal-body">
        <div class="mb-3">
          <label for="name" class="form-label">Nama Lengkap</label>
          <input type="text" class="form-control" id="name" placeholder="Nama Lengkap" required>
        </div>
        <div class="mb-3">
          <label for="nim" class="form-label">NIM</label>
          <input type="text" class="form-control" id="nim" placeholder="NIM" required>
        </div>
        <div class="mb-3">
          <label for="kelas" class="form-label">Kelas</label>
          <input type="text" class="form-control" id="kelas" placeholder="Kelas" required>
        </div>
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Close</button>
        <button type="button" class="btn btn-primary" id="printObject">Print Object To Console</button>
      </div>
    </div>
  </div>
</div>
```

Cara mengakses modal

```
<a class="btn btn-teal" href="#" type="button" class="btn btn-primary" data-bs-toggle="modal" data-bs-target="#exampleModal">
  Click to open modal
```

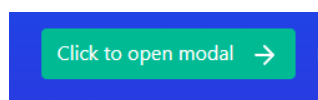
Menambahkan *data-bs-toggle* dengan value “modal” dan *data-bs-target* dengan value id modalnya

JS

```
$("#printObject").click(function(e) {
  console.log(
    {
      fullname: $("#name").val(),
      nim: $("#nim").val(),
      kelas: $("#kelas").val()
    }
  );
});
```

Melakukan print object ke console

Output



```
{fullname: 'Akmal Muhamad Firdaus', nim: '1301204188', kelas: 'IF-44-01'}
  fullname: "Akmal Muhamad Firdaus"
  kelas: "IF-44-01"
  nim: "1301204188"
  [[Prototype]]: Object
```

2. DOM Manipulation using jQuery : Studi case aritmatika dan control flow (if else)

Kerangka HTML : Form

```
<div class="container px-5">
  <div class="text-center mb-5">
    <h2>Example use form</h2>
  </div>
  <div class="row gx-5 z-1">
    <div id="result">

    </div>
    <div class="col-md-12">
      <div class="mb-3">
        <label for="num1" class="form-label">Angka pertama</label>
        <input type="number" class="form-control" id="num1" placeholder="2">
      </div>

      <select class="form-select" aria-label="Operasi" id="operation">
        <option value="" selected>Pilih Operasi</option>
        <option value="*">* (Perkalian)</option>
        <option value="+">+ (Pertambahan)</option>
        <option value="-">- (Pengurangan)</option>
        <option value="/">/ (Pembagian)</option>
      </select>

      <div class="mt-3">
        <label for="num2" class="form-label">Angka Kedua</label>
        <input type="number" class="form-control" id="num2" placeholder="5">
      </div>

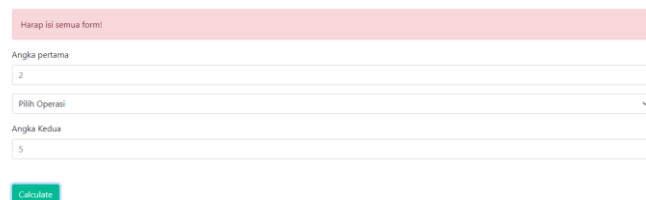
      <button class="btn btn-teal mt-5" type="button" id="calculate">Calculate</button>
    </div>
  </div>
</div>
```

JS

```
$("#calculate").click(function(e) {
  var num1 = $("#num1").val();
  var num2 = $("#num2").val();
  var operation = $("#operation").val();
  if(num1 == "" || num2 == "" || operation == ""){
    $("#result").empty();
    $("#result").html("<div class='alert alert-danger' role='alert'>Harap isi semua form!</div>");
  } else {
    if(operation == "+") {
      result = parseInt(num1) + parseInt(num2);
    } else if(operation == "-") {
      result = parseInt(num1) - parseInt(num2);
    } else if(operation == "*") {
      result = parseInt(num1) * parseInt(num2);
    } else if(operation == "/") {
      result = parseInt(num1) / parseInt(num2);
    } else {
      console.log("Invalid operator");
    }
    $("#result").empty();
    $("#result").html("<div class='alert alert-primary' role='alert'>Hasil dari <num1> <operation> <num2> adalah <result> </div>");
  }
});
```

Output

Example use form



Example use form



Pada js, diberikan onClick listener menggunakan jQuery pada id *calculate* (*Button*), lalu dilakukan pengecekan pada form tersebut, jika form kosong maka sistem akan menampilkan alert form harus diisi dan jika form sudah sesuai maka sistem akan menampilkan alert sesuai aritmatika dalam form.

Full Source code : <https://github.com/codernewbie04/CI3H4-ABP-Praktikum>