



# MODUL 1

PENGEMBANGAN APLIKASI PLATFORM KHUSUS  
DAVID SETIADI

## MODUL 2

JavaScript adalah bahasa pemrograman tingkat tinggi yang pada awalnya dikembangkan untuk membuat website menjadi lebih “hidup”. Bersama dengan HTML dan CSS, JavaScript menjadi bahasa pemrograman paling populer untuk mengembangkan aplikasi berbasis web. Bahasa ini mampu memberikan logic ke dalam website, sehingga website tersebut memiliki fungsionalitas tambahan dan lebih interaktif.

Awalnya JavaScript dibuat supaya dapat berjalan di lingkungan browser dan membuat website menjadi lebih interaktif. Namun, saat ini Anda sebagai developer dapat menggunakan bahasa pemrograman JavaScript di berbagai lingkungan pengembangan. Sehingga, tidak hanya sebatas browser/client, namun JavaScript juga bisa berjalan di server menggunakan Node.js.

JavaScript termasuk ke dalam kategori scripting language. Apa maksudnya? Salah satu ciri-ciri utama dari bahasa scripting adalah kode tidak perlu dikompilasi agar bisa dijalankan. Scripting language menggunakan interpreter untuk menerjemahkan kode atau perintah yang kita tulis supaya dimengerti oleh mesin.

Itulah kenapa bahasa scripting tidak membutuhkan banyak kode yang perlu ditulis agar sebuah program bisa dijalankan.

Saat ini javascript tidak hanya digunakan di sisi client (browser) saja. Javascript juga digunakan pada server, console, program desktop, mobile, IoT, game, dan lain-lain.

Hal ini membuat javascript semakin populer dan menjadi bahasa yang paling banyak digunakan di Github.



Javascript awalnya bernama Mocha, lalu berubah menjadi LiveScript saat browser Netscape Navigator 2.0 rilis versi beta (September 1995). Namun, setelah itu dinamai ulang menjadi Javascript. 1

Terinspirasi dari kesuksesan Javascript, Microsoft mengadopsi teknologi serupa. Microsoft membuat ‘Javascript’ versi mereka sendiri bernama JScript. Lalu di tanam pada Internet Explorer 3.0.

Hal ini mengakibatkan ‘perang browser’, karena JScript milik Microsoft berbeda dengan Javascript racikan Netscape.

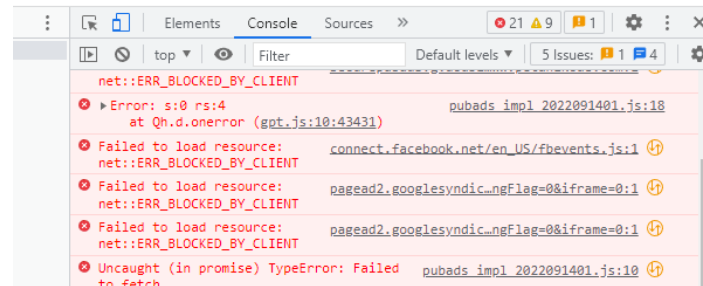
Akhirnya pada tahun 1996, Netscape mengirimkan standarisasi ECMA-262 ke Ecma International. Sehingga lahirlah standarisasi kode Javascript bernama ECMAScript atau ES.

Sejak 2016 versi javascript dinamai Dengan EcmaScript 2016,2017,2018,2019,dst  
perlatan yang harus disiapkan untuk belajar Javascript?

- Web Browser (Google Chrome, Firefox, Opera, dll)
- Teks Editor (rekomendasi: VS Code)

Untuk melihat kode error js kita bisa melihatnya melalui console di web browser

Console Javascript bisa di buka Inspect Element->Console di browser



Untuk memulai, silahkan buka vscode dan buat file dengan nama belajar.html

Ada beberapa cara untuk menuliskan kode js pada html :

### 1. Menulis Javascript pada Tag <script>

```
belajar.html x
belajar.html > html
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="utf-8">
5   <title>Penulisan Javascript</title>
6   <script>
7     console.log("Hi, ini kode Javascript");
8   </script>
9 </head>
10 <body>
11
12 <script>
13   document.write("Javascript itu keren!");
14 </script>
15 </body>
16 </html>
```

### 2. Menulis Javascript pada File Eksternal

```

<> belajar.html > html
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="utf-8">
5      <title>Penulisan Javascript</title>
6      <script>
7          console.log(1 + 2);
8      </script>
9  </head>
10 <body>
11
12 <script src="kode.js"></script>
13 </body>
14 </html>

```

File belajar.html

```

<> belajar.html  JS kode.js  X
JS kode.js
1  document.write("Kode Javascript dari File Eksternal");

```

File kode.js

```

v JS
<> belajar.html
JS kode.js

```

Jadi ada dua file terpisah yaitu file html dan file js.

Jika file html dan js berada di folder yang berbeda maka ikuti aturan penulisan PATH nya.

Missal : `<script src="js/kode-script.js"></script>`

Apabila ingin menggunakan file js dari web maka contoh penulisannya :

`<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>`

### 3. Menulis Javascript pada Atribut

Penulisan Javascript pada atribut bisanya dilakukan pada atribut event.

Contoh :

```

<> belajar.html > html
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <meta charset="utf-8">
5    <title>Penulisan Javascript</title>
6  </head>
7  <body>
8    <button onclick="alert('Ok Terima kasih!')">Klik donk!</button>
9  </body>
10 </html>

```

Contoh lain :

```

<> belajar.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5    <meta charset="UTF-8">
6    <meta name="viewport" content="width=device-width, initial-scale=1.0">
7    <title>Penulisan Javascript di Atribut</title>
8  </head>
9
10 <body>
11   <textarea onkeyup="document.getElementById('counter').innerText = this.value.length" cols="30" rows="10"></
    textarea>
12   <p>Panjang: <span id="counter">0</span> karakter</p>
13 </body>
14
15 </html>

```

Contoh event atribut lainnya :

	Elemen terkait: button, document, link, layer
onmouseover	Event ini terjadi saat user menggerakkan mouse ke atas elemen HTML Elemen terkait: button, document, link, layer
onmouseout	Event ini terjadi saat user menggerakkan mouse keluar elemen HTML Elemen terkait: button, document, link, layer
onmouseup	Event ini terjadi saat user melepaskan klik pada elemen HTML Elemen terkait: button, document, link, layer
onkeydown	Event ini terjadi saat user sedang menekan tombol keyboard pada elemen HTML Elemen terkait: document, image, link, input, textarea
onkeypress	Event ini terjadi saat user menekan tombol keyboard pada elemen HTML Elemen terkait: document, image, link, input, textarea
onkeyup	Event ini terjadi saat user melepas penekanan tombol keyboard pada elemen HTML Elemen terkait: document, image, link, input, textarea
onload	Event ini terjadi dokumen atau frame telah ditampilkan oleh browser. Elemen terkait: document, image, layer, window
onresize	Event ini terjadi saat ukuran dari dokumen berubah Elemen terkait: document, image, layer, window
onscroll	Event ini terjadi saat user melakukan scrolling pada dokumen Elemen terkait: document, window
onunload	Event ini terjadi saat user menutup dokumen Elemen terkait: document, window
onblur	Event ini terjadi saat fokus pada elemen tersebut dipindahkan ke elemen lainnya Elemen terkait: button, checkbox, file upload, layer, password, radio, reset, select, submit, text, textarea, window
onchange	Event ini terjadi saat nilai elemen form berubah Elemen terkait: file upload, select, submit, text, textarea

#### 4. Penulisan Javascript pada URL

Cara ini jarang digunakan dan kadang diblok karena bisa di kategorikan sebagai XSS dan di beberapa browser yang men set addressbar menjadi mesin pencari maka akan lgsg di arahkan ke halaman google.

Cara menampilkan Output pada JS :

Ada 4 cara menampilkan output pada Javascript:

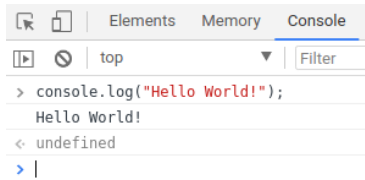
1. Menggunakan Fungsi **console.log()**;
2. Menggunakan Fungsi **alert()**;
3. Menggunakan Fungsi **document.write()**;
4. Menggunakan **innerHTML**.

##### 1. Menggunakan Fungsi console.log()

Fungsi console.log() adalah fungsi untuk menampilkan teks ke console Javascript.

Contoh : `console.log("Hello World!");`

Maka hasilnya :



Fungsi `console.log()` biasanya digunakan untuk debugging. Karena setiap pesan error di Javascript selalu ditampilkan di dalam Console.

Selain `console.log()`, terdapat juga beberapa fungsi untuk debugging seperti `console.debug()`, `console.info()`, `console.error()`, `console.dir()`, dsb.

## 2. Menggunakan Fungsi `alert()`

Fungsi `alert()` adalah fungsi untuk menampilkan jendela dialog. Fungsi sebenarnya berada pada objek `window`.

Secara lengkap bisa ditulis seperti ini:

```
window.alert("Hello World!");
```

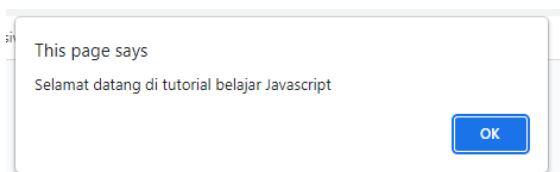
Bisa juga ditulis `alert()` saja seperti ini:

```
alert("Hello World!");
```

Contoh :

```
<> belajar.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <title>Belajar Javascript</title>
5    <script>
6      alert("Selamat datang di tutorial belajar Javascript");
7
8      function sayHello(){
9        alert("Hello!");
10     }
11   </script>
12 </head>
13 <body>
14   <button onclick="sayHello()">Klik Aku!</button>
15 </body>
16 </html>
```

Hasilnya :



### 3. Menggunakan Fungsi document.write()

Objek document adalah objek yang mewakili dokumen HTML di dalam Javascript.

Dalam objek document, terdapat fungsi write() untuk menulis sesuatu ke dokumen HTML.

Contoh :

```
<> belajar.html > html > head > script
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <title>Belajar Javascript</title>
5    <script>
6      document.write("<h1>Hello World!</h1>");
7      document.write("<hr>");
8      document.write("<p>Saya sedang belajar Javascript</p>");
9      document.write("<di <b>unsap.ac.id</b>");
10   </script>
11 </head>
12 <body>
13
14 </body>
15 </html>
```

Hasilnya :

# Hello World!

---

Saya sedang belajar Javascript

di **unsap.ac.id**

### 4. Menggunakan innerHTML

innerHTML adalah sebuah atribut di dalam (objek) elemen HTML yang berisi string HTML.

Dengan innerHTML, kita dapat menampilkan output ke elemen yang lebih spesifik.

Contoh:



```

<> belajar.html > html > body > div#hasil-output
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  |   <title>Belajar Javascript</title>
5  </head>
6  <body>
7  |
8  |   <h1>Tutorial Javascript untuk Pemula</h1>
9  |   <div id="hasil-output"></div>|
10 |
11 |   <script>
12 |       // membuat objek elemen
13 |       var hasil = document.getElementById("hasil-output");
14 |
15 |       // menampilkan output ke elemen hasil
16 |       hasil.innerHTML = "<p>Aku suka Javascript</p>";
17 |   </script>
18 |
19 </body>
20 </html>

```

## Cara Membuat Variabel di Javascript

Variabel adalah sebuah wadah yang digunakan untuk menampung nilai atau data pada bahasa pemrograman. Variabel yang bisa diberi nama ini bisa menampung angka, tulisan, boolean, dan lain-lain.

Ada tiga tahap dalam menggunakan variabel ketika belajar JavaScript, yaitu:

1. Declaration- Tahap mendaftarkan variabel ke dalam program yang ditulis.
2. Initialization- Tahap menyiapkan memori yang nantinya digunakan oleh variabel.
3. Assignment – Tahap menetapkan nilai spesifik untuk disimpan ke dalam variabel.

Contoh penerapannya seperti ini:

```

var x; //declaration & initialization
x = 10; //assignment

```

Nah, tiga Reserved Words yang dapat Anda gunakan untuk mendeklarasikan variabel yaitu **var**, **let**, dan **const**. Tapi untuk pemula, direkomendasikan untuk menggunakan var saja karena punya aturan yang lebih simpel.

Berikut beberapa aturan pembuatan variabel yang harus Anda perhatikan:

1. Hindari Penggunaan Spasi
2. Hindari Awalan Angka
3. Gunakan camelCase
4. Gunakan Shorthand

## Menampilkan Isi Variabel

Untuk menampilkan isi variabel, kita bisa memanfaatkan fungsi-fungsi untuk menampilkan output seperti:

- Fungsi `console.log()` menampilkan output ke console javascript;
- Fungsi `document.write()` menampilkan output ke dokumen HTML;
- dan Fungsi `alert()` menampilkan output ke jendela dialog.

```
<> belajar.html > html > body
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <title>Belajar Variabel dalam Javascript</title>
5  <script>
6      // membuat variabel
7      var name = "WEB UNSAP";
8      var visitorCount = 50322;
9      var isActive = true;
10     var url = "https://www.unsap.ac.id";
11
12     // menampilkan variabel ke jendela dialog (alert)
13     alert("Selamat datang di " + name);
14
15     // menampilkan variabel ke dalam HTML
16     document.write("Nama Situs: " + name + "<br>");
17     document.write("Jumlah Pengunjung: " + visitorCount + "<br>");
18     document.write("Status Aktif: " + isActive + "<br>");
19     document.write("Alamat URL: " + url + "<br>");
20 </script>
21 </head>
22 <body>
23
24 </body>
25 </html>
```

## Tipe Data JavaScript

### 1. Number

Tidak seperti bahasa pemrograman lain, JavaScript tidak memakai tipe data angka yang berbeda-beda. Misalnya, integer, float, dan long. Ia hanya punya satu tipe angka, yaitu number yang sanggup menyimpan data hingga 64 bit.

- Bilangan Bulat – Misalnya 10, 1500, dan 12345.
- Bilangan Pecahan – Contohnya 3.14, 0.5, dan 100.00.
- Bilangan Eksponensial – Misalnya 123e5 atau 123e-5.
- Bilangan Negatif – Contohnya -0.25 atau -123.
- Bilangan Spesial – Misalnya Infinity (10/0), -Infinity (-10/0), dan NaN (0/0).

### 2. String

Pada JavaScript, String adalah tipe data yang digunakan untuk menuliskan data tekstual (plain text). Untuk membuat string, Anda perlu menambahkan tanda petik pada teks tersebut. Bisa petik dua (") atau petik satu (').

### 3. Boolean

Boolean adalah sebuah tipe data JavaScript yang digunakan untuk memberikan nilai logika true atau false. Nah, Boolean berguna untuk menentukan aksi berbeda dan mengatur alur kerja program.

### 4. Function

Function adalah tipe data dalam bentuk perintah yang dapat digunakan di bagian program yang memerlukan. Untuk menggunakannya, Anda perlu membuat Function terlebih dahulu, baru kemudian memanggilnya.

### 5. Array

Array adalah tipe data JavaScript yang digunakan untuk menampung lebih dari satu nilai dan memiliki index.

Dengan Array, Anda bisa memasukkan tipe data yang berbeda dalam satu Array. Selain itu, ia berguna untuk menghemat penggunaan variabel ketika coding JavaScript.

Anda bisa memakai Array dengan membuat kurung siku. Lalu, menulis nilai Array di dalamnya dan dipisahkan dengan koma.

Contoh Array pada bahasa pemrograman JavaScript adalah sebagai berikut:

```
var person = ["John", "Doe", 46];
```

### 6. Object

Mirip dengan Array, Object adalah tipe data yang memiliki lebih dari satu nilai. Bedanya, Object tidak punya index tapi nama.

Kenapa Anda harus menggunakan Object saat belajar JavaScript? Ternyata, Object dapat mengorganisir data lebih baik dari Array. Sebab, Anda bisa memasukkan Function dan Array pada Object.

Hasilnya, Anda bisa menghemat baris kode yang dihasilkan saat coding JavaScript.

Contoh :

```
const person = {  
  firstName: "John",  
  lastName : "Doe",  
  score    : [5,7,8,10,9],  
  fullName : function() {  
    return this.firstName + " " + this.lastName;  
  }  
};
```

## Operator pada javascript

Operator adalah simbol yang digunakan untuk melakukan operasi pada suatu nilai dan variabel.

Operator dalam pemrograman terbagi dalam 6 jenis:

1. Operator aritmatika;
2. Operator Penugasan (Assignment);
3. Opeartor relasi atau perbandingan;
4. Operator Logika;
5. Operator Bitwise;
6. Operator Ternary;

### 1. Opeartor Aritmatika pada Javascript

Operator aritmatika merupakan operator untuk melakukan operasi aritmatika seperti penjumlahan, pengurangan, pembagian, perkalian, dsb.

Nama Operator	Simbol
Penjumlahan	+
Pengurangan	-
Perkalian	*
Pemangkatan	**
Pembagian	/
Sisa Bagi	%

Contoh :

```
<> belajar.html •
<> belajar.html > html > body
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <title>Operator Aritmatika</title>
5 </head>
6 <body>
7   <script>
8     var a = 15;
9     var b = 4;
10    var c = 0;
11    // pengurangan
12    c = a - b;
13    document.write(`${a} - ${b} = ${c}<br/>`);
14    // Perkalian
15    c = a * b;
16    document.write(`${a} * ${b} = ${c}<br/>`);
17    // pemangkatan
18    c = a ** b;
19    document.write(`${a} ** ${b} = ${c}<br/>`);
20    // Pembagian
21    c = a / b;
22    document.write(`${a} / ${b} = ${c}<br/>`);
23    // Modulo
24    c = a % b;
25    document.write(`${a} % ${b} = ${c}<br/>`);
26  </script>
27 </body>
28 </html>
```

## 2. Opeartor Penugasan pada Javascript

Operator penugasan adalah operator yang digunakan untuk memberikan tugas kepada variabel. Biasanya digunakan untuk mengisi variabel.

Contoh:

```
var a = 19;
```

Variabel a kita berikan tugas untuk menyimpan nilai 19.

Nama Operator	Sombol
Pengisian Nilai	=
Pengisian dan Penambahan	+=
Pengisian dan Pengurangan	-=
Pengisian dan Perkalian	*=
Pengisian dan Pemangkatan	**=
Pengisian dan Pembagian	/=
Pengisian dan Sisa bagi	%=

Operator penugasan sama seperti operator aritmatika. Ia juga digunakan untuk melakukan operasi aritmatika.

Contoh:

```
var jumlahView = 12;
```

```
// menggunakan operator penugasan penjumlahan
```

```
// untuk menambah nilai
```

```
jumlahView += 1;
```

```
belajar.html > html > body
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  |   <title>Operator Penugasan</title>
5  </head>
6  <body>
7  |   <script>
8  |       document.write("Mula-mula nilai score...<br>");
9  |       // pengisian nilai
10 |       var score = 100;
11 |       document.write("score = "+ score + "<br/>");
12 |
13 |       // pengisian dan menjumlahan dengan 5
14 |       score += 5;
15 |       document.write("score = "+ score + "<br/>");
16 |
17 |       // pengisian dan pengurangan dengan 2
18 |       score -= 2;
19 |       document.write("score = "+ score + "<br/>");
20 |
21 |       // pengisian dan perkalian dengan 2
22 |       score *= 2;
23 |       document.write("score = "+ score + "<br/>");
24 |
25 |       // pengisian dan pembagian dengan 4
26 |       score /= 4;
27 |       document.write("score = "+ score + "<br/>");
28 |
29 |       // pengisian dan pemangkatan dengan 2
30 |       score **= 2;
31 |       document.write("score = "+ score + "<br/>");
32 |
33 |       // pengisian dan modulo dengan 3;
34 |       score %= 3;
35 |       document.write("score = "+ score + "<br/>");
36 |   </script>
37 </body>
38 </html>
```

Hasilnya :

```
Mula-mula nilai score...
score = 100
score = 105
score = 103
score = 206
score = 51.5
score = 2652.25
score = 0.25
```

### 3. Operator Perbandingan pada Javascript

Operator relasi atau perbandingan adalah operator yang digunakan untuk membandingkan dua nilai.

Operator perbandingan akan menghasilkan sebuah nilai boolean true dan false.

Operator perbandingan terdiri dari:

Nama Operator	Simbol
Lebih Besar	>
Lebih Kecil	<
Sama Dengan	== atau ===
Tidak Sama dengan	!= atau !==
Lebih Besar Sama dengan	>=
Lebih Kecil Sama dengan	<=

Contoh :

```

<> belajar.html > html > body
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <title>Operator Perbandingan</title>
5  </head>
6  <body>
7  <script>
8    var aku = 20;
9    var kamu = 19;
10
11    // sama dengan
12    var hasil = aku == kamu;
13    document.write(`${aku} == ${kamu} = ${hasil}<br/>`);
14
15    // lebih besar
16    var hasil = aku > kamu;
17    document.write(`${aku} > ${kamu} = ${hasil}<br/>`);
18
19    // lebih besar samadengan
20    var hasil = aku >= kamu;
21    document.write(`${aku} >= ${kamu} = ${hasil}<br/>`);
22
23    // lebih kecil
24    var hasil = aku < kamu;
25    document.write(`${aku} < ${kamu} = ${hasil}<br/>`);
26
27    // lebih kecil samadengan
28    var hasil = aku <= kamu;
29    document.write(`${aku} <= ${kamu} = ${hasil}<br/>`);
30
31    // tidak samadengan
32    var hasil = aku != kamu;
33    document.write(`${aku} != ${kamu} = ${hasil}<br/>`);
34  </script>
35 </body>
36 </html>

```

Hasilnya :

```

20 == 19 = false
20 > 19 = true
20 >= 19 = true
20 < 19 = false
20 <= 19 = false
20 != 19 = true

```

#### 4. Opeartor Logika pada Javascript

Operator logika digunakan untuk melakukan operasi terhadap dua nilai boolean.

Operator ini terdiri dari:

Nama Operator	Simbol
Logika AND	&&
Logika OR	
Negasi/kebalikan	!



```

belajar.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <title>Operator Logika</title>
6  </head>
7
8  <body>
9
10 <script>
11     var aku = 20;
12     var kamu = 19;
13
14     var benar = aku > kamu;
15     var salah = aku < kamu;
16
17     // operator && (and)
18     var hasil = benar && salah;
19     document.write(`${benar} && ${salah} = ${hasil}<br/>`);
20
21     // operator || (or)
22     var hasil = benar || salah;
23     document.write(`${benar} || ${salah} = ${hasil}<br/>`);
24
25     // operator ! (not)
26     var hasil = !benar;
27     document.write(`${!benar} = ${hasil}<br/>`);
28
29 </script>
30 </body>
31
32 </html>

```

Hasilnya :

```

true && false = false
true || false = true
!true = false

```

## 5. Opeartor Bitwise pada Javascript

Operator bitwise merupakan operator yang digunakan untuk operasi berdasarkan bit (biner).

Operator ini terdiri dari:

Nama	Simbol di Java
AND	&
OR	
XOR	^
Negasi/kebalikan	~
Left Shift	«
Right Shift	»
Left Shift (unsigned)	«<
Right Shift (unsigned)	»>

## 6. Opeartor Ternary pada Javascript

Operator ternary merupakan operator yang terdiri dari tiga bagian.

Operator-operator sebelumnya hanya dua bagian saja, yaitu: bagian kiri dan kanan. Ini disebut operator binary.

Sementara operator trinary ada bagian kiri, tengah, dan kanan.

Contoh :

```
<> belajar.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5  |   <title>Operator Ternary</title>
6  </head>
7
8  <body>
9
10 |   <script>
11 |       var pertanyaan = confirm("Apakah kamu berumur diatas 18 tahun?")
12 |
13 |       var hasil = pertanyaan ? "Selamat datang" : "Kamu tidak boleh di sini";
14 |       document.write(hasil);
15 |   </script>
16 </body>
17
18 </html>
```

Jika ditekan OK maka yg akan tampil “Selamat datang”

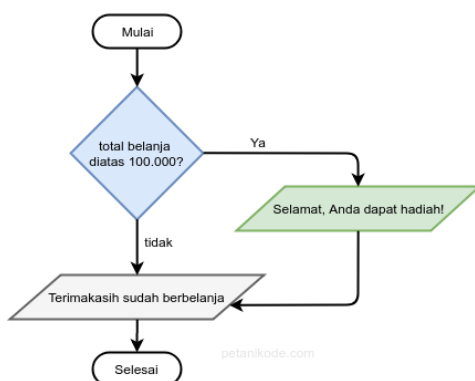
Jika tekan CANCEL maka yang akan tampil “Kamu tidak boleh disini”

## Bentuk Percabangan pada Javascript

### 1. Percabangan if

Percabangan if merupakan percabangan yang hanya memiliki satu blok pilihan saat kondisi bernilai benar.

Coba perhatikan flowchart berikut ini:



Flowchart tersebut dapat kita baca seperti ini:

“Jika total belanja lebih besar dari Rp 100.000, Maka tampilkan pesan Selamat, Anda dapat hadiah”

Kalau dibawah Rp 100.000 bagaimana?

Ya pesannya tidak ditampilkan.

```
belajar.html > html > body > script
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <title>Percabangan if</title>
5 </head>
6 <body>
7   <script>
8     var totalBelanja = prompt("Total belanja?", 0);
9
10    if(totalBelanja > 100000){
11      document.write("<h2>Selamat Anda dapat hadiah</h2>");
12    }
13    document.write("<p>Terimakasih sudah berbelanja di toko kami</p>");
14  </script>
15 </body>
16 </html>
```

Hasilnya:

**Selamat Anda dapat hadiah**

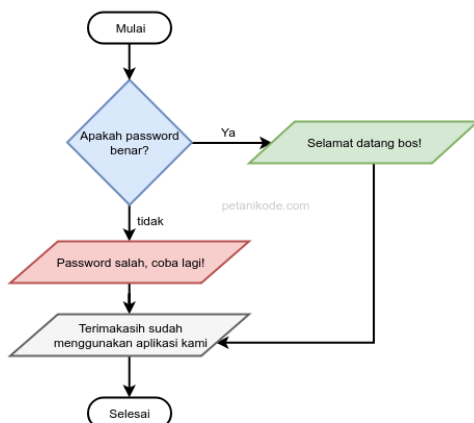
Terimakasih sudah berbelanja di toko kami

## 2. Percabangan if/else

Percabangan if/else merupakan percabangan yang memiliki dua blok pilihan.

Pilihan pertama untuk kondisi benar, dan pilihan kedua untuk kondisi salah (else).

Coba perhatikan flowchart ini:



Ini adalah flowchart untuk mengecek password.

Apabila password benar, pesan yang ada pada blok hijau akan ditampilkan: “Selamat datang bos!”

Tapi kalau salah, maka pesan yang ada di blok merah yang akan ditampilkan: “Password salah, coba lagi!”

Kemudian, pesan yang berada di blok abu-abu akan tetap ditampilkan, karena dia bukan bagian dari blok percabangan if/else.

```
belajar.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  |   <title>Percabangan if/else</title>
5  </head>
6  <body>
7  <script>
8  |   var password = prompt("Password:");
9  |
10 |   if(password == "kopi"){
11 |       document.write("<h2>Selamat datang bos!</h2>");
12 |   } else {
13 |       document.write("<p>Password salah, coba lagi!</p>");
14 |   }
15 |
16 |   document.write("<p>Terima kasih sudah menggunakan aplikasi ini!</p>");
17 |
18 |   </script>
19 </body>
20 </html>
```

Jika benar :

**Selamat datang bos!**

Terima kasih sudah menggunakan aplikasi ini!

Jika salah :

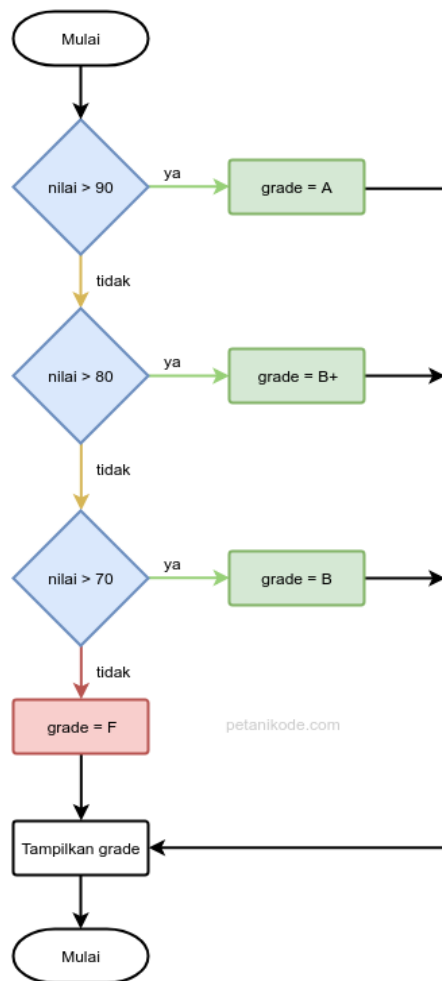
Password salah, coba lagi!

Terima kasih sudah menggunakan aplikasi ini!

### 3. Percabangan if/else/if

Percabangan if/else/if merupakan percabangan yang memiliki lebih dari dua blok pilihan.

Coba perhatikan flowchart berikut:



Perhatikan blok yang saya beri warna...

Ini adalah blok untuk percabangan if/else/if. Kita bisa menambahkan berapapun blok yang kita inginkan.

Contoh Program:

```

<> belajar.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  |   <title>Percabangan if/else/if</title>
5  </head>
6  <body>
7  <script>
8      var nilai = prompt("Inputkan nilai akhir:");
9      var grade = "";
10
11      if(nilai >= 90) grade = "A"
12      else if(nilai >= 80) grade = "B+"
13      else if(nilai >= 70) grade = "B"
14      else if(nilai >= 60) grade = "C+"
15      else if(nilai >= 50) grade = "C"
16      else if(nilai >= 40) grade = "D"
17      else if(nilai >= 30) grade = "E"
18      else grade = "F";
19
20      document.write(`<p>Grade anda: ${grade}</p>`);
21  </script>
22 </body>
23 </html>

```

Hasilnya :

Grade anda: A

Pada program di atas, kita tidak menggunakan kurung kurawal untuk membuat blok kode untuk if/else/if.

karena hanya terdapat satu baris perintah saja. Yaitu: grade = ....

Bila kita menggunakan kurung kurawal, maka program di atas akan menjadi seperti ini:

```

<> belajar.html > html > body > script
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  |   <title>Percabangan if/else/if</title>
5  </head>
6  <body>
7  |   <script>
8  var nilai = prompt("Inputkan nilai akhir:");
9  |   var grade = "";
10
11     if (nilai >= 90){
12     |   grade = "A"
13     } else if(nilai >= 80) {
14     |   grade = "B+"
15     } else if(nilai >= 70) {
16     |   grade = "B"
17     } else if(nilai >= 60) {
18     |   grade = "C+"
19     } else if(nilai >= 50) {
20     |   grade = "C"
21     } else if(nilai >= 40) {
22     |   grade = "D"
23     } else if(nilai >= 30) {
24     |   grade = "E"
25     } else {
26     |   grade = "F";
27     }
28     document.write(`<p>Grade anda: ${grade}</p>`);|
29     </script>
30 </body>
31 </html>

```

#### 4. Percabangan switch/case

Percabangan switch/case adalah bentuk lain dari percabangan if/else/if.

Strukturnya seperti ini:

```

switch(variabel){
    case <value>:
        // blok kode
        break;
    case <value>:
        // blok kode
        break;
    default:
        // blok kode
}

```

Kita dapat mermbuat blok kode (case) sebanyak yang diinginkan di dalam blok switch.

Pada <value>, kita bisa isi dengan nilai yang nanti akan dibandingkan dengan varabel.

Setiap case harus diakhiri dengan break. Khusus untuk default, tidak perlu diakhiri dengan break karena dia terletak di bagian akhir.

Pemberian break bertujuan agar program berhenti mengecek case berikutnya saat sebuah case terpenuhi.

Contoh:

```
belajar.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  |   <title>Percabangan switch/case</title>
5  </head>
6  <body>
7  |   <script>
8  |
9  |       var jawab = prompt("Kamu beruntung! Silahkan pilih hadiahmu dengan memasukan angka 1 sampai 5");
10 |       var hadiah = "";
11 |
12 |       switch(jawab){
13 |           case "1":
14 |               hadiah = "Tisu";
15 |               break;
16 |           case "2":
17 |               hadiah = "1 Kotak Kopi";
18 |               break;
19 |           case "3":
20 |               hadiah = "Sticker";
21 |               break;
22 |           case "4":
23 |               hadiah = "Minyak Goreng";
24 |               break;
25 |           case "5":
26 |               hadiah = "Uang Rp 50.000";
27 |               break;
28 |           default:
29 |               document.write("<p>Opps! anda salah pilih</p>");
30 |       }
31 |
32 |       if(hadiah === ""){
33 |           document.write("<p>Kamu gagal mendapat hadiah</p>");
34 |       } else {
35 |           document.write("<h2>Selamat kamu mendapatkan " + hadiah + "</h2>");
36 |       }
37 |   </script>
38 </body>
39 </html>
```



Percabangan switch/case juga dapat dibuat seperti ini:

```
var nilai = prompt("input nilai");
var grade = "";

switch(true){
  case nilai < 90:
    grade = "A";
    break;
  case nilai < 80:
    grade = "B+";
    break;
  case nilai < 70:
    grade = "B";
    break;
  case nilai < 60:
    grade = "C+";
    break;
  case nilai < 50:
    grade = "C";
    break;
  case nilai < 40:
    grade = "D";
    break;
  case nilai < 30:
    grade = "E";
    break;
  default:
    grade = "F";
}
```

## 5. Percabangan dengan Operator Ternary

Percabangan menggunakan operator ternary merupakan bentuk lain dari percabangan if/else.

Bisa dibilang:

Bentuk singkatnya dari if/else.

Contoh:

```
<> belajar.html > html
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <title>Percabangan Ternary</title>
5 </head>
6 <body>
7   <script>
8     var jwb = prompt("Apakah Jakarta ibu kota indonesia?");
9
10    var jawaban = (jwb.toUpperCase() == "IYA") ? "Benar": "Salah";
11
12    document.write(`Jawaban anda: <b>${jawaban}</b>`);
13  </script>
14 </body>
15 </html>
```

Fungsi dari method **toUpperCase()** untuk mengubah teks yang diinputkan menjadi huruf kapital semua.

Opertor ternary berperan sebagai percabangan if/else:

```
var jawaban = (jwb.toUpperCase() == "IYA") ? "Benar": "Salah";
```

Apabila kondisi yang ada di dalam kurung—(jwb.toUpperCase() == "IYA")—bernilai true, maka nanti isi dari variabel jawaban akan sama dengan "Benar".

Tapi kalau bernilai false, maka variabel jawaban akan berisi "Salah".

## 6. Percabangan Bersarang (Nested)

Kita juga dapat membuat blok percabangan di dalam percabangan. Ini disebut percabangan bersarng atau nested if.

```
<> belajar.html > html > body > script
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  |   <title>Percabangan Ternary</title>
5  </head>
6  <body>
7  |   <script>
8  |       var username = prompt("Username:");
9  |       var password = prompt("Password:");
10 |
11 |       if(username == "unsap"){
12 |           if(password == "kopi"){
13 |               document.write("<h2>Selamat datang pak bos!</h2>");
14 |           } else {
15 |               document.write("<p>Password salah, coba lagi!</p>");
16 |           }
17 |       } else {
18 |           document.write("<p>Anda tidak terdaftar!</p>");
19 |       }
20 |   </script>
21 </body>
22 </html>
```

Referensi :

- Petanikode.com
- Dicoding.com
- W3schools.com