Pemrograman Dasar Javascript

ISH3D4 – Sistem Informasi

Rahmat Fauzi, S.T., M.T

AGENDA

Week	Topic	Week	Topic
1	Introduction HTML dan CSS	9	Model View Controller pada Framework Laravel
2	HTML, CSS dan Java Script	10	Model View Controller pada Framework Laravel
3	Server-Side berbasis web menggunakan PHP	11	•POST dan GET •Pengenalan Cookies dan Session
4	Operator-operator pada PHP	12	•POST dan GET •Pengenalan Cookies dan Session
5	Operator-operator pada PHP	13	TUBES
6	Percabangan dan Perulangan pada PHP	14	TUBES
7	mengintegrasikan halaman website dengan database	15	UAS
8	Mid Term Exam	16	UAS

Home

Saya Bee.. akan membantu Anda mempelajari modul ini.

Silahkan pilih materi yang ingin Anda pelajari

Pokok Bahasan

Capaian

Bahasan

Video Animasi

Kuis / Latihan

Link

Kesimpulan

Pustaka

Pokok Bahasan

01 Pengenalan javascript

02 Cara menulis javascript

03 Variabel dan tipe data di javascript

04 6 Operator dalam Javascript

05 6 Operator percabangan

06 6 Operator pengulangan

07 Struktur Data Javascript

Telkom University

Pokok Bahasan

08 DOM JavaScript

09

10

11

12

13

14

Telkom University

Pemrograman Web

Capaian Pembelajaran

P02

Kemampuan menganalisis permasalahan, melakukan identifikasi dan mendefinisikan kebutuhan komputasi Yang bersesuaian dengan solusi

P03

Kemampuan untuk merancang, melakukan implementasi dan mengevaluasi sistem berbasis komputer, proses, komponen, atau program untuk memenuhi kebutuhan yang diinginkan.

P07

Kemampuan untuk menganalisis dampak lokal dan global dari komputasi pada individu, organisasi dan masyarakat

Telkom University

Bab 1
Pengenalan Javascript

Dasar
PHP

Definisi JavaScript

JavaScript adalah bahasa pemrograman yang berbentuk kumpulan skrip yang biasanya digunakan untuk **menambahkan interaksi** antara halaman web dengan pengunjung halaman web. JavaScript dijalankan pada **sisi klien** yang akan memberikan kemampuan fitur-fitur tambahan **halaman web** yang **lebih baik** dibandingkan fitur-fitur yang terdapat pada HTML.

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.

Dasar
PHP

Definisi JavaScript

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**.

Javascript

Petanicode.com

Javascript

- Javascript adalah bahasa pemrograman yang awalnya dirancang untuk berjalan di atas browser. Namun, seiring perkembangan zaman, javascript tidak hanya berjalan di atas browser saja. Javascript juga dapat digunakan pada sisi Server, Game, IoT, Desktop, dsb.
- 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.

Prinsip Dasar Javascript

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

- Javascript mendukung paradigma pemrograman imparatif (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”).
- Javascript memiliki model pemrograman fungsional yang sangat ekspresif.
- Pemrograman berorientasi objek (PBO) pada Javascript memiliki perbedaan dari PBO pada umumnya.
- Program kompleks pada javascript umumnya dipandang sebagai program-program kecil yang saling berinteraksi.

Prinsip Dasar Javascript

JavaScript memiliki struktur sederhana, kodenya dapat disisipkan pada **dokumen HTML atau berdiri sendiri**.
Struktur penulisan JavaScript dalam dokumen atau internal berada dalam tag
script :

```
<script>
…
</script>
```

Telkom University

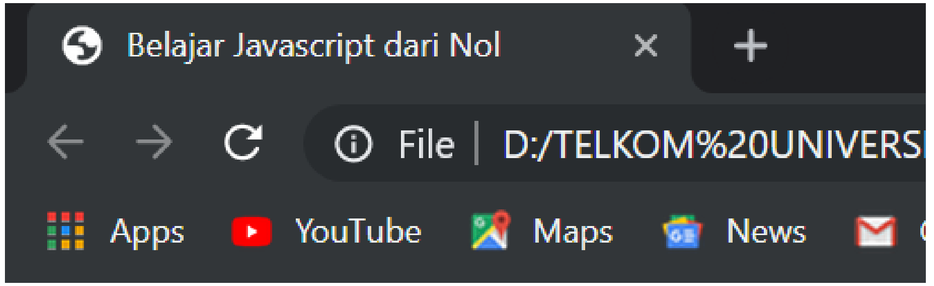
Bab 2
Cara Menulis Javascript

Bagaimana Cara Menulis Kode Javascript di HTML?

1. *Embed* (Kode Javascript ditempel langsung pada HTML. Contoh: slide setelah ini)
2. *Inline* (kode Javascript ditulis pada atribut HTML)
3. Eksternal (Kode Javascript ditulis terpisah dengan file HTML)

LETS CODE Penulisan Kode javascript dengan Embed

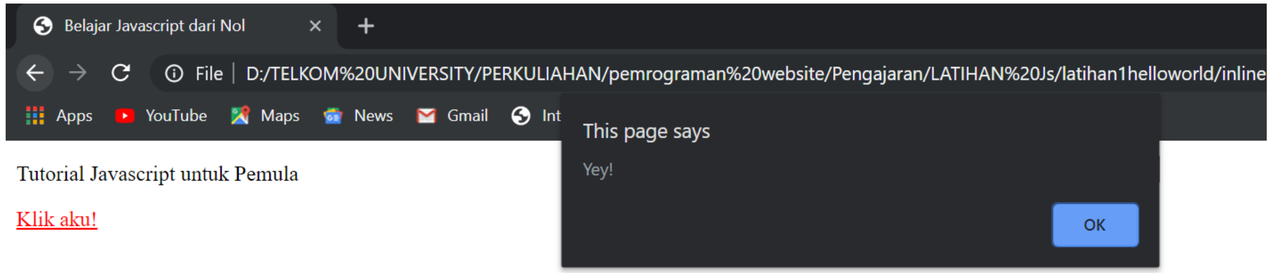
```html
<!DOCTYPE html>
<html>
    <head>
        <title>Belajar Javascript dari Nol</title>
        <script>
            // ini adalah penulisan kode javascript
            // di dalam tag <head>
            console.log("Hello JS dari Head");
        </script>
    </head>
    <body>
        <p>Tutorial Javascript untuk Pemula</p>
        <script>
            // ini adalah penulisan kode javascript
            // di dalam tag <body>
            console.log("Hello JS dari body");
        </script>
    </body>
</html>
```



Tutorial Javascript untuk Pemula

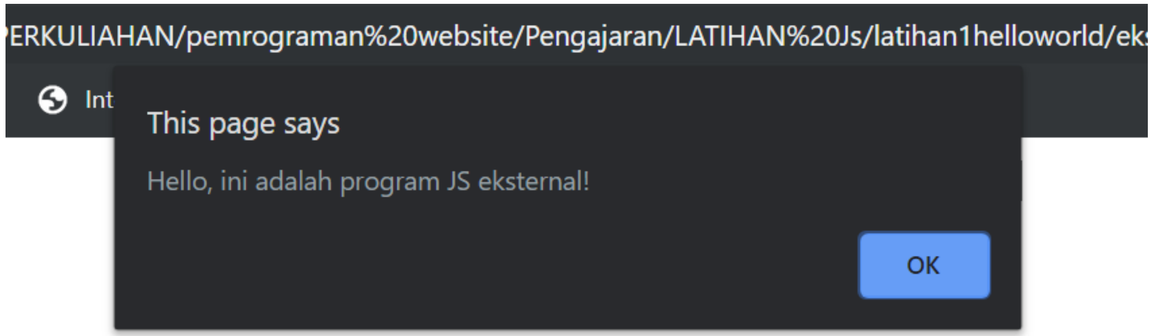
LETS CODE Penulisan Kode javascript Inline

```html
<!DOCTYPE html>
<html>
    <head>
        <title>Belajar Javascript dari Nol</title>
        <script>
            // ini adalah penulisan kode javascript
            // di dalam tag <head>
            console.log("Hello JS dari Head");
        </script>
    </head>
    <body>
        <p>Tutorial Javascript untuk Pemula</p>
        <a href="#" onclick="alert('Yey!')">Klik aku!</a>
        <script>
            // ini adalah penulisan kode javascript
            // di dalam tag <body>
            console.log("Hello JS dari body");
        </script>
    </body>
</html>
```



LETS CODE Penulisan Kode javascript Eksternal

```html
<!DOCTYPE html>
<html>
    <head>
        <title>Belajar Javascript dari Nol</title>
    </head>
    <body>
        <p>Tutorial Javascript untuk Pemula</p>

        <!-- Menyisipkan kode js eksternal -->
        <script src="kode-program.js"></script>
    </body>
</html>
```




```javascript
alert("Hello, ini adalah program JS eksternal!");
```

4 Cara Menampilkan Output pada Javascript

Ada 4 cara menampilkan output pada Javascript:

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

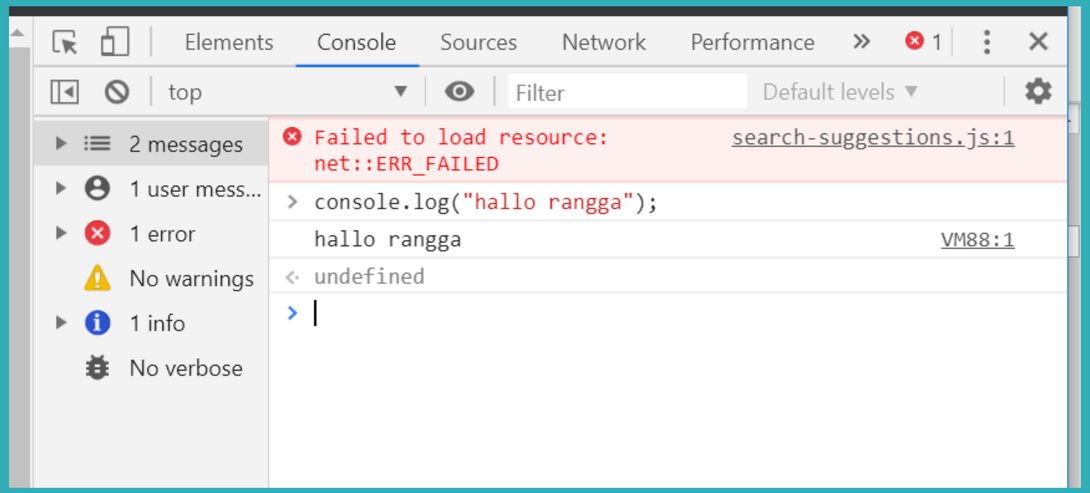
(petanicode.com)

1. Menggunakan Fungsi console.log();

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

Contoh penggunaan:

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



2. Menggunakan Fungsi **alert()**

Fungsi alert() adalah fungsi untuk menampilkan jendela dialog. Fungsi sebenarnya berada pada objek window. Secara lengkap bisa ditulis seperti ini:

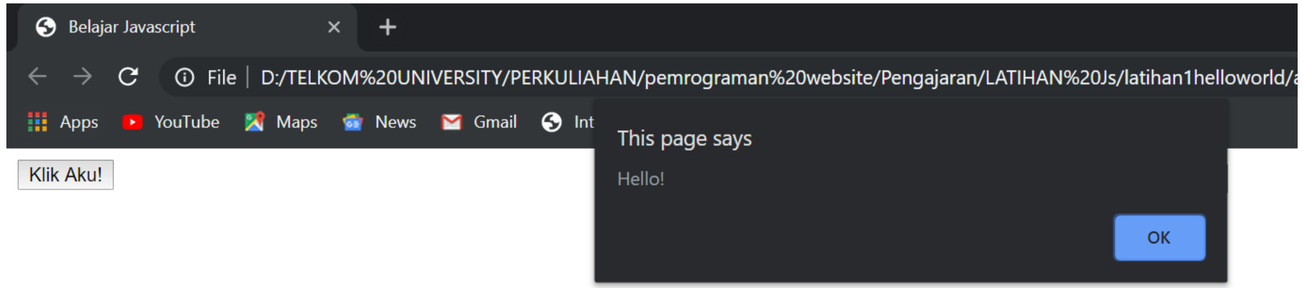
```
window.alert("Hello World!");
```

Bisa juga ditulis alert() saja seperti ini:

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

2. Menggunakan Fungsi alert()

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Belajar Javascript</title>
    <script>
        alert("Selamat datang di tutorial belajar Javascript");

        function sayHello(){
            alert("Hello!");
        }
    </script>
</head>
<body>
    <button onclick="sayHello()">Klik Aku!</button>
</body>
</html>
```



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.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Belajar Javascript</title>
    <script>
        document.write("<h1>Hello World!</h1>");
        document.write("<hr>");
        document.write("<p>Saya sedang belajar Javascript</p>");
        document.write("di <b>petanikode.com</b>")
    </script>
</head>
<body>

</body>
</html>
```

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.

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Belajar Javascript</title>
</head>
<body>

    <h1>Tutorial Javascript untuk Pemula</h1>
    <div id="hasil-output"></div>


    <script>
        // membuat objek elemen
        var hasil = document.getElementById("hasil-output");

        // menampilkan output ke elemen hasil
        hasil.innerHTML = "<p>Aku suka Javascript</p>";
    </script>

</body>
</html>
```

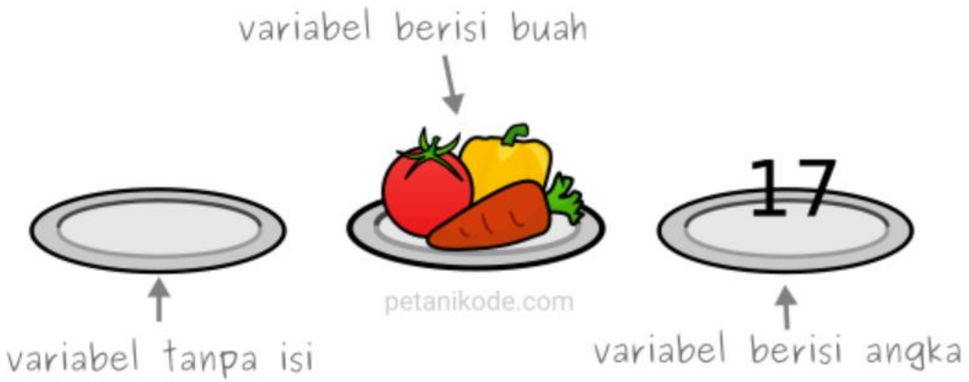
Telkom University

Bab 3

Variabel dan Tipe Data dalam Javascript

Variabel dalam Javascript

- **Variabel** adalah sebuah nama yang mewakili sebuah nilai. Variabel bisa diisi dengan berbagai **macam nilai** seperti string (teks), number (angka), objek, array, dan sebagainya.
- Kita bisa ibaratkan, **variabel** itu seperti **wadah** untuk menyimpan sesuatu.



- Sumber : petanikode.com

Cara Membuat Variabel di Javascript

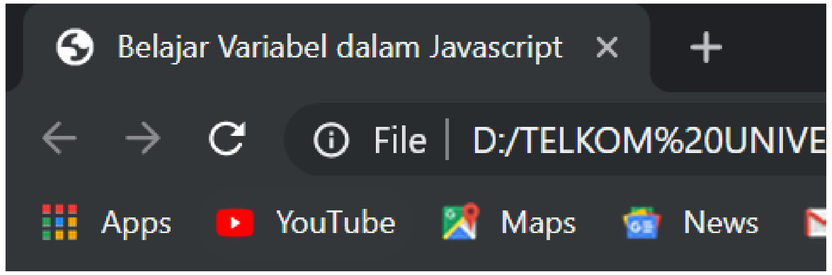
- Cara membuat variabel yang umum digunakan di javascript adalah menggunakan **kata kunci var** lalu diikuti dengan nama variabel dan nilainya. **Contoh :**
- var title = "Belajar Pemrograman Javascript";
- var siteName = “Blog Saya";
- var url = "https://www.blogku.com";
- var visitorCount = 90;

Cara Membuat Variabel di Javascript

- Perlu kamu ketahui juga, selain kata kunci var kita juga bisa membuat variabel dengan **kata kunci let** atau **tanpa awalan** apapun.

LETS CODE Membuat Variabel di Javascript

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Belajar Variabel dalam Javascript</title>
    <script>
        // membuat variabel
        var name = "Detik.com";
        var visitorCount = 34567;
        var isActive = true;
        var url = "https://www.detik.com";

        // menampilkan variabel ke jendela dialog (alert)
        alert("Selamat datang di " + name);

        // menampilkan variabel ke dalam HTML
        document.write("Nama Situs: " + name + "<br>");
        document.write("Jumlah Pengunjung: " + visitorCount + "<br>");
        document.write("Status Aktif: " + isActive + "<br>");
        document.write("Alamat URL: " + url + "<br>");
    </script>
</head>
<body>


</body>
</html>
```



Nama Situs: Detik.com
Jumlah Pengunjung: 34567
Status Aktif: true
Alamat URL: https://www.detik.com

Mengisi Ulang Variabel

- Variabel bersifat *mutable*, artinya nilai yang tersimpan di dalamnya dapat kita isi ulang (berubah).

```
// mula-mula kita buat variabel dengan isi seperti ini
var age = 18;

// lalu kita isi ulang
age = 21;
```

- Karena kata kunci **var** dibutuhkan saat membuat variabel saja. Sedangkan untuk mengisi ulang, kita cukup tulis seperti di atas.
- Apabila kita menggunakan **kata kunci var**, berarti jadinya kita membuat **variabel baru donk, bukan mengisi ulang**.

Menghapus Variabel

- Penghapusan variabel dalam Javascript memang jarang dilakukan. Namun, untuk program yang membutuhkan ketelitian dalam alokasi memori, penghapusan variabel perlu dilakukan agar penggunaan memori lebih optimal.
- Penghapusan variabel dapat dilakukan dengan katakunci **delete.**

```
bookTitle = "Belajar Pemrograman Javascript";
delete bookTitle;
```

Mengenal Tipe Data Js

Tipe data adalah jenis-jenis data yang bisa kita simpan di dalam variabel. Ada beberapa tipe data dalam pemrograman Javascript:

- **String (teks)**
- **Integer atau Number (bilangan bulat)**
- **Float (bilangan Pecahan)**
- **Boolean**
- **Object**

Mengenal Tipe Data Js

```
var name = "Dian";
var age = 22;
var single = true;
```

Javascript adalah bahasa yang bersifat *dynamic typing*, artinya kita tidak harus menuliskan tipe data pada saat pembuatan variabel seperti pada bahasa C, C++, Java, dsb. yang bersifat *static typing*.

Javascript **akan otomatis** mengenali tipe data yang kita berikan pada variabel.

Aturan Penulisan Nama Variabel di Javascript

- Penamaan variabel **tidak boleh** menggunakan angka di depannya contoh **var 123name = “hasan”;**
- Penamaan variabel **boleh** menggunakan awal underscore. Contoh **var _name = “hasan”** ;
- Penamaan variabel **dianjurkan** menggunakan camelCase apabila tediri dari dua suku kata. Contoh **var fullName = “hasan”;**
- Penamaan variabel **dianjurkan** menggunakan bahasa inggris. Contoh **var postTitle = "Tutorial Javascript untuk Pemula";**

Telkom University

Bab 4
OPERATOR DALAM JAVASCRIPT

OPERATOR DALAM JAVASCRIPT

Operator adalah simbol yang digunakan untuk melakukan operasi pada suatu nilai dan variabel. Operator dalam pemrograman terbagi dalam 6 jenis:

- Operator aritmatika;
- Operator Penugasan (Assignment);
- Opeartor relasi atau perbandingan;
- Operator Logika;
- Operator Bitwise;
- Operator Ternary;

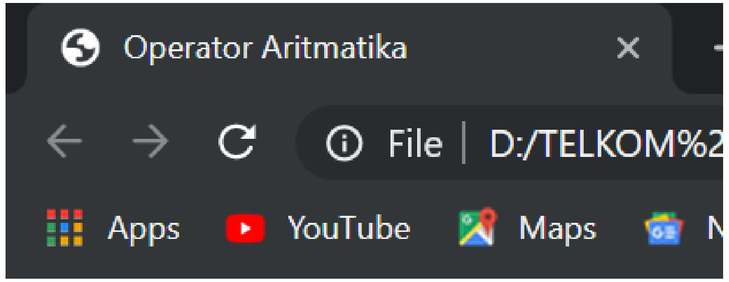
1.OPERATOR ARITMATIKA

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

LETS CODE : OPERATOR ARITMATIKA

aritmatika.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Operator Aritmatika</title>
</head>
<body>
    <script>
        var a = 20;
        var b = 4;
        var c = 0;
        // pengurangan
        c = a - b;
        document.write(`${a} - ${b} = ${c}<br/>`);
        // Perkalian
        c = a * b;
        document.write(`${a} * ${b} = ${c}<br/>`);
        // pemangkatan
        c = a ** b;
        document.write(`${a} ** ${b} = ${c}<br/>`);
        // Pembagian
        c = a / b;
        document.write(`${a} / ${b} = ${c}<br/>`);
        // Modulo
        c = a % b;
        document.write(`${a} % ${b} = ${c}<br/>`);
    </script>
</body>

</html>
```



20 - 4 = 16
20 * 4 = 80
20 ** 4 = 160000
20 / 4 = 5
20 % 4 = 0

2.Operator Penugasan

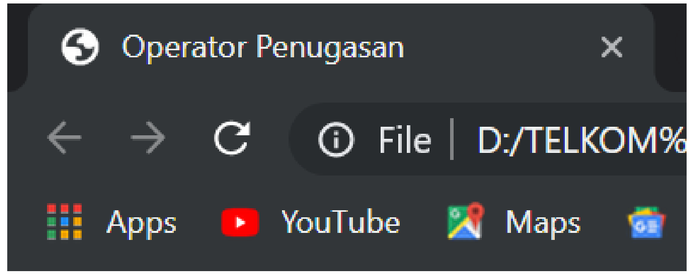
Operator penugasan adalah operator yang digunakan untuk memberikan tugas kepada variabel. Biasanya digunakan untuk mengisi variabel.

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

LETS CODE : OPERATOR PENUGASAN

penugasan.html

```
<!DOCTYPE html>
<html lang="en">
<head><title>Operator Penugasan</title></head>
<body>
    <script>
        document.write("Mula-mula nilai score...<br>");
        // pengisian nilai
        var score = 100;
        document.write("score = "+ score + "<br/>");
        // pengisian dan menjumlahan dengan 5
        score += 5;
        document.write("score = "+ score + "<br/>");
        // pengisian dan pengurangan dengan 2
        score -= 2;
        document.write("score = "+ score + "<br/>");
        // pengisian dan perkalian dengan 2
        score *= 2;
        document.write("score = "+ score + "<br/>");
        // pengisian dan pembagian dengan 4
        score /= 4;
        document.write("score = "+ score + "<br/>");
        // pengisian dan pemangkatan dengan 2
        score **= 2;
        document.write("score = "+ score + "<br/>");
        // pengisian dan modulo dengan 3;
        score %= 3;
        document.write("score = "+ score + "<br/>");
    </script>
</body>
</html>
```



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

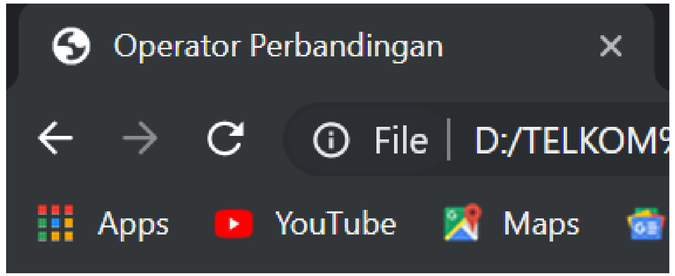
3.Operator Perbandingan

Operator relasi atau perbandingan adalah operator yang digunakan untuk membandingkan dua nilai. Operator perbandingan akan menghasilkan sebuah nilai **boolean true dan false**.

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

LETS CODE : OPERATOR PERBANDINGAN

```
<body>
    <script>
        var aku = 25;
        var kamu = 15;

        // sama dengan
        var hasil = aku == kamu;
        document.write(`${aku} == ${kamu} = ${hasil}<br/>`);

        // lebih besar
        var hasil = aku > kamu;
        document.write(`${aku} > ${kamu} = ${hasil}<br/>`);

        // lebih besar samadengan
        var hasil = aku >= kamu;
        document.write(`${aku} >= ${kamu} = ${hasil}<br/>`);

        // lebih kecil
        var hasil = aku < kamu;
        document.write(`${aku} < ${kamu} = ${hasil}<br/>`);

        // lebih kecil samadengan
        var hasil = aku <= kamu;
        document.write(`${aku} <= ${kamu} = ${hasil}<br/>`);

        // tidak samadengan
        var hasil = aku != kamu;
        document.write(`${aku} != ${kamu} = ${hasil}<br/>`);
    </script>
</body>
```



25 == 15 = false
25 > 15 = true
25 >= 15 = true
25 < 15 = false
25 <= 15 = false
25 != 15 = true

4. Operator Logika

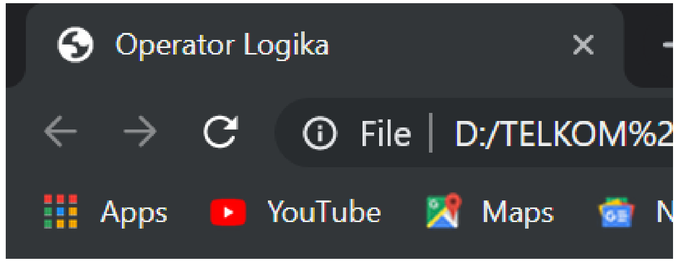
Operator logika digunakan untuk melakukan operasi terhadap dua nilai Boolean. true dan false

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

LETS CODE : OPERATOR PERBANDINGAN

logika.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Operator Logika</title>
</head>
<body>
    <script>
        var aku = 20;
        var kamu = 10;

        var benar = aku > kamu;
        var salah = aku < kamu;

        // operator && (and)
        var hasil = benar && salah;
        document.write(`${benar} && ${salah} = ${hasil}<br/>`);

        // operator || (or)
        var hasil = benar || salah;
        document.write(`${benar} || ${salah} = ${hasil}<br/>`);

        // operator ! (not)
        var hasil = !benar
        document.write(`!${benar} = ${hasil}<br/>`);

    </script>
</body>

</html>
```



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

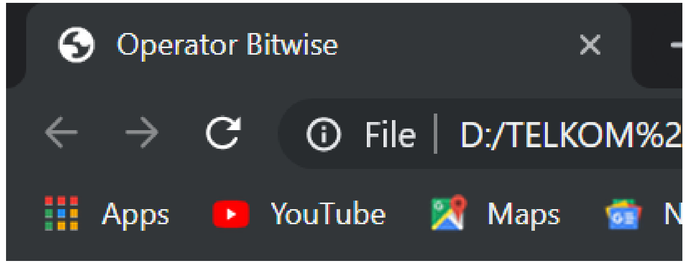
5. Operator Bitwise

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

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

LETS CODE : OPERATOR BITWISE

```
<head>
    <title>Operator Bitwise</title>
</head>
<body>
    <script>
        var x = 4;
        var y = 3;
        // operator bitwise and
        var hasil = x & y;
        document.write(`${x} & ${y} = ${hasil}<br/>`);
        // operator bitwise or
        var hasil = x | y;
        document.write(`${x} | ${y} = ${hasil}<br/>`);
        // operator bitwise xor
        var hasil = x ^ y;
        document.write(`${x} ^ ${y} = ${hasil}<br/>`);
        // operator negasi
        var document: Document
        document.write(`~${x} = ${hasil}<br/>`);
        // operator bitwise right shift >>
        var hasil = x >> y;
        document.write(`${x} >> ${y} = ${hasil}<br/>`);
        // operator bitwise right shift <<
        var hasil = x << y;
        document.write(`${x} << ${y} = ${hasil}<br/>`);
        // operator bitwise right shift (unsigned) >>>
        var hasil = x >>> y;
        document.write(`${x} >>> ${y} = ${hasil}<br/>`);
    </script>
</body>
```



4 & 3 = 0
4 | 3 = 7
4 ^ 3 = 7
~4 = -5
4 >> 3 = 0
4 << 3 = 32
4 >>> 3 = 0

6. Operator Ternary

Jika Operator-operator sebelumnya hanya dua bagian saja, yaitu: bagian kiri dan kanan. Atau biasa disebut operator binary.
Sedangkan Operator ternary merupakan operator yang teridiri dari tiga bagian.
Operator trinary terdiri dari bagian kiri, tengah, dan kanan

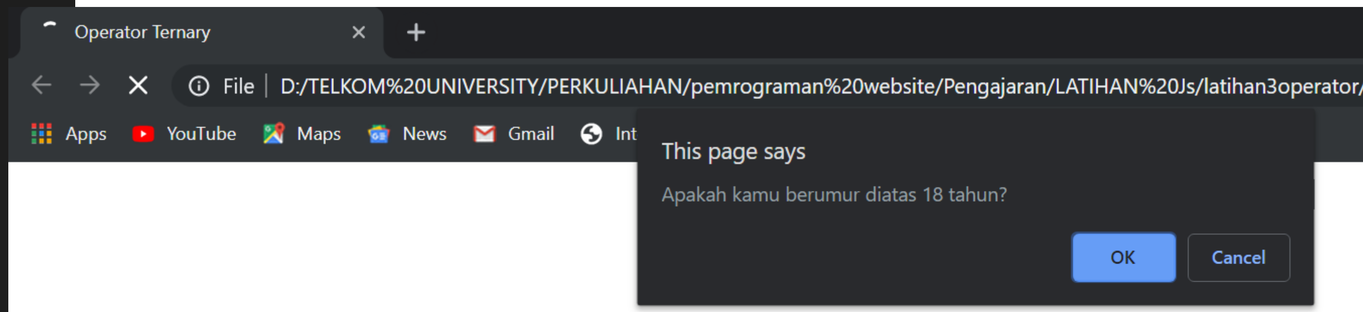
```
bagian kiri <operator> bagian tengah <operator> bagian kanan
```

```
<kodisi> ? "benar" : "salah"
```

LETS CODE : OPERATOR TERNARY

ternary.html

```
<!DOCTYPE html>
<html lang="en">

<head>
    <title>Operator Ternary</title>
</head>

<body>

    <script>
        var pertanyaan = confirm("Apakah kamu berumur diatas 18 tahun?")

        var hasil = pertanyaan ? "Selamat datang" : "Kamu tidak boleh di sini";
        document.write(hasil);
    </script>
</body>

</html>
```



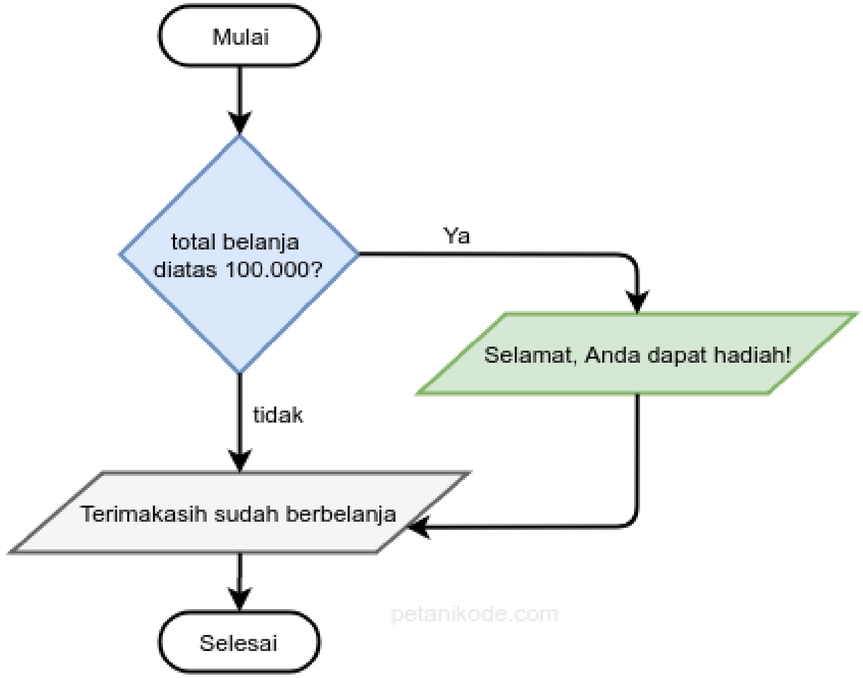
Telkom University

Bab 5
6 PERCABANGAN DALAM JAVASCRIPT

6 Bentuk Percabangan pada Javascript

1. Percabangan if

```
if (condition) {
    block of code to be executed if the condition is true
}
```



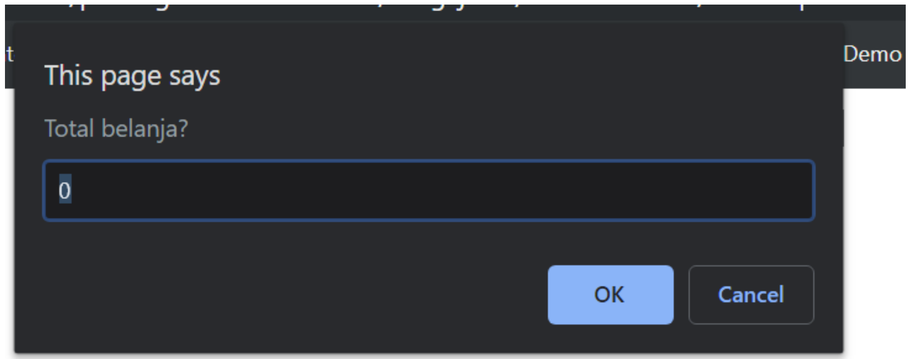
LETS CODE : PERCABANGAN IF

percabanganIF.html

latihan4percabangan ▸ percabanganIF.html ▸ html

```
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
14          document.write("<p>Terimakasih sudah berbelanja di toko kami</p>");
15      </script>
16  </body>
17  </html>
```

LETS CODE : HASIL PERCABANGAN IF



Terimakasih sudah berbelanja di toko kami

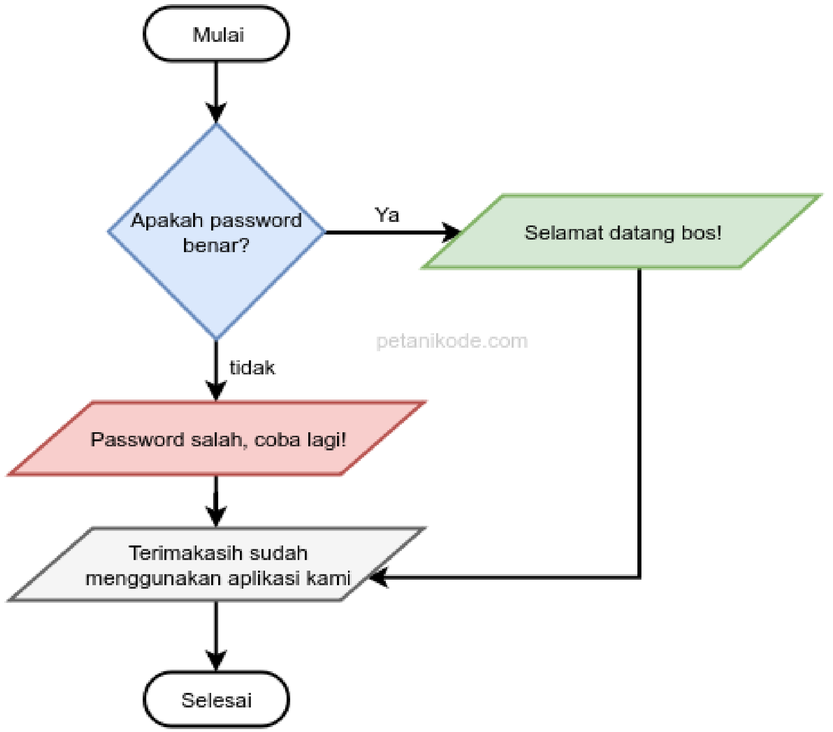
Selamat Anda dapat hadiah

Terimakasih sudah berbelanja di toko kami

6 Bentuk Percabangan pada Javascript

2. Percabangan if / else

```
if (condition) {
    block of code to be executed if the condition is true
} else {
    block of code to be executed if the condition is false
}
```



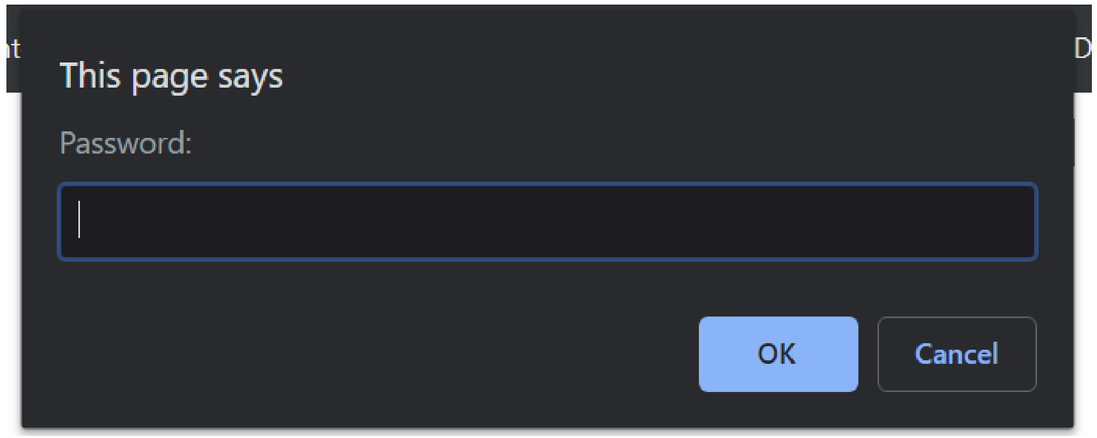
LETS CODE : PERCABANGAN IF ELSE

percabanganIFELSE.html

latihan4percabangan ▸ percabanganIFELSE.html ▸ html ▸ head

```
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>
```

LETS CODE : HASIL PERCABANGAN IF ELSE



Selamat datang bos!

Terima kasih sudah menggunakan aplikasi ini!

Password salah, coba lagi!

Terima kasih sudah menggunakan aplikasi ini!

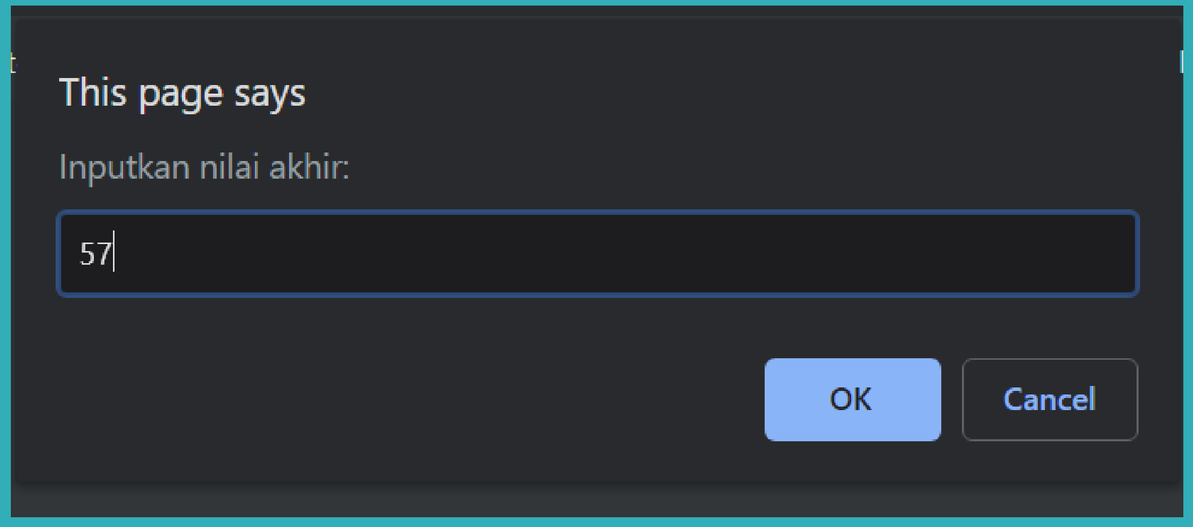
6 Bentuk Percabangan pada Javascript

3. Percabangan if / else / if

LETS CODE : PERCABANGAN IF ELSE 2

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Percabangan if/else/if</title>
</head>
<body>
    <script>
        var nilai = prompt("Inputkan nilai akhir:");
        var grade = "";

        if(nilai >= 90) grade = "A"
        else if(nilai >= 80) grade = "B+"
        else if(nilai >= 70) grade = "B"
        else if(nilai >= 60) grade = "C+"
        else if(nilai >= 50) grade = "C"
        else if(nilai >= 40) grade = "D"
        else if(nilai >= 30) grade = "E"
        else grade = "F";

        document.write(`<p>Grade anda: ${grade}</p>`);
    </script>
</body>
</html>
```

LETS CODE : HASIL PERCABANGAN IF ELSE - 2

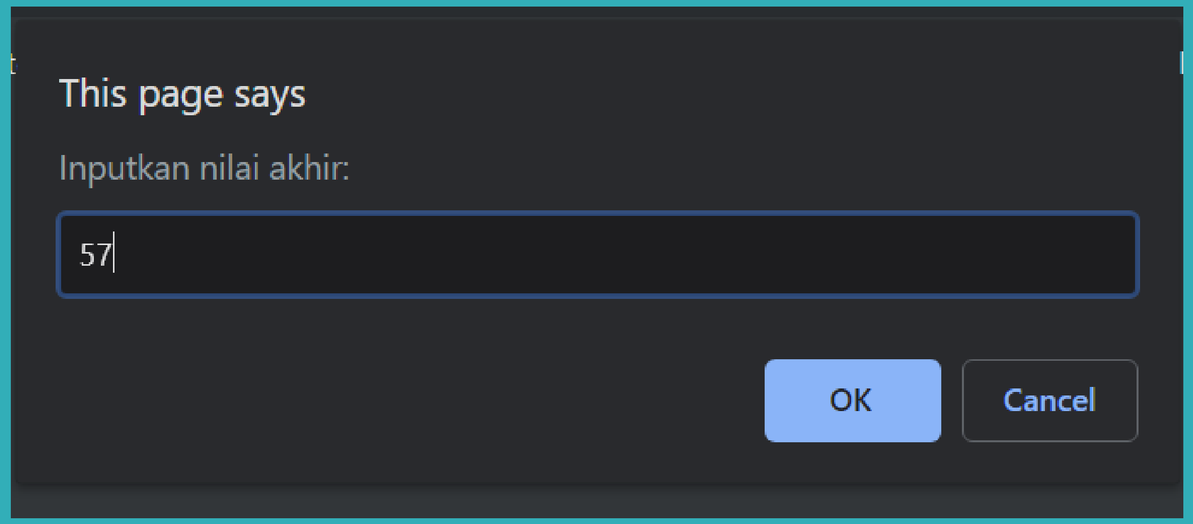


Grade anda: C

LETS CODE : PERCABANGAN IF ELSE 3

```
<script>
    var nilai = prompt("Inputkan nilai akhir:");
    var grade = "";

    if (nilai >= 90){
        grade = "A"
    } else if(nilai >= 80) {
        grade = "B+"
    } else if(nilai >= 70) {
        grade = "B"
    } else if(nilai >= 60) {
        grade = "C+"
    } else if(nilai >= 50) {
        grade = "C"
    } else if(nilai >= 40) {
        grade = "D"
    } else if(nilai >= 30) {
        grade = "E"
    } else {
        grade = "F";
    }
    document.write(`<p>Grade anda: ${grade}</p>`);
</script>
```

LETS CODE : HASIL PERCABANGAN IF ELSE - 3



Grade anda: C

6 Bentuk Percabangan pada Javascript

4. Percabangan switch

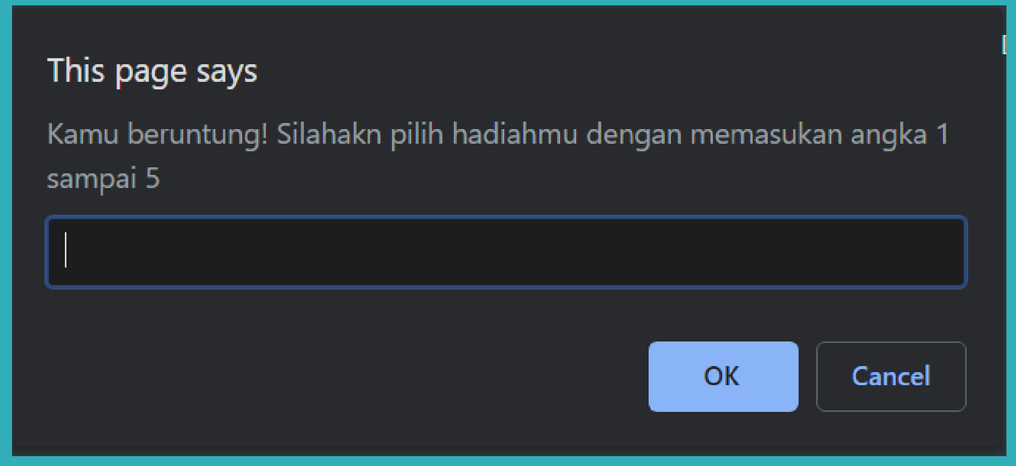
```
switch(expression) {
    case x:
        code block
        break;
    case y:
        code block
        break;
    default:
        code block
}
```

```
switch (new Date().getDay()) {
    case 0:
        day = "Sunday";
        break;
    case 1:
        day = "Monday";
        break;
    case 2:
        day = "Tuesday";
        break;
}
```

LETS CODE : PERCABANGAN SWITCH

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Percabangan switch/case</title>
</head>
<body>
    <script>

        var jawab = prompt("Kamu beruntung! Silahakn pilih hadiahmu dengan memasukan angka 1 sampai 5"
        var hadiah = "";

        switch(jawab){
            case "1":
                hadiah = "Tisu";
                break;
            case "2":
                hadiah = "1 Kotak Kopi";
                break;
            case "3":
                hadiah = "Sticker";
                break;
            case "4":
                hadiah = "Minyak Goreng";
                break;
            case "5":
                hadiah = "Uang Rp 50.000";
                break;
            default:
                document.write("<p>Opps! anda salah pilih</p>");
        }

        if(hadiah === ""){
            document.write("<p>Kamu gagal mendapat hadiah</p>");
        } else {
            document.write("<h2>Selamat kamu mendapatkan " + hadiah + "</h2>");
        }
    </script>
</body>
</html>
```

LETS CODE : HASIL PERCABANGAN SWITCH



Selamat kamu mendapatkan Sticker

6 Bentuk Percabangan pada Javascript

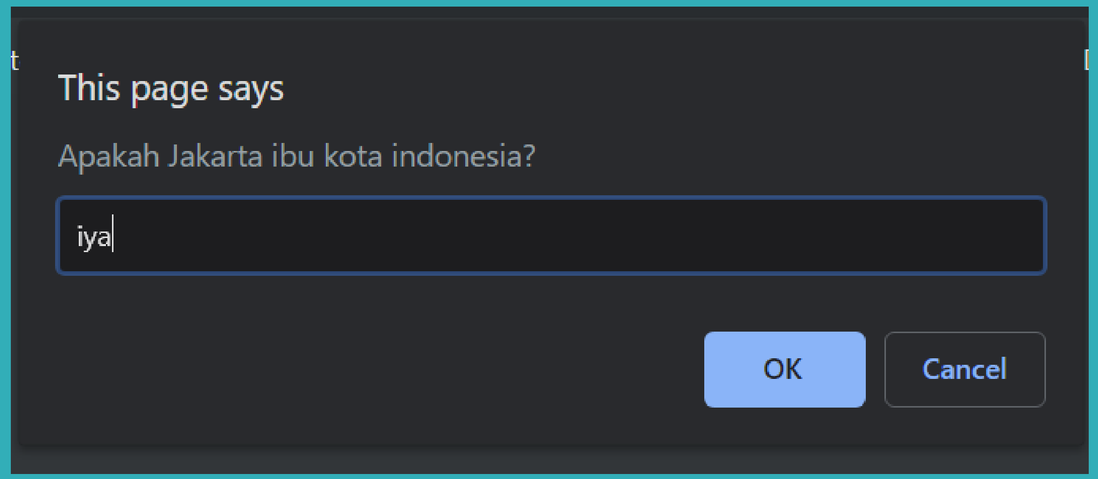
5. Percabangan Itenary

- Percabangan menggunakan opreator ternary merupakan **bentuk lain** dari percabangan **if/else.**

LETS CODE : PERCABANGAN ITENARY

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Percabangan Ternary</title>
</head>
<body>
    <script>
        var jwb = prompt("Apakah Jakarta ibu kota indonesia?");

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

        document.write(`Jawaban anda: <b>${jawaban}</b>`);
    </script>
</body>
</html>
```

LETS CODE : HASIL PERCABANGAN ITENARY



Jawaban anda: **Benar**

6 Bentuk Percabangan pada Javascript

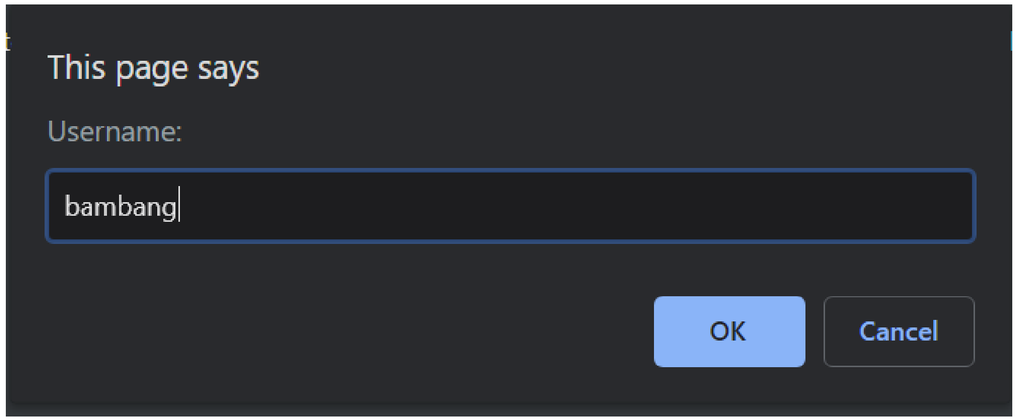
6. Percabangan Bersarang (Nested)

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

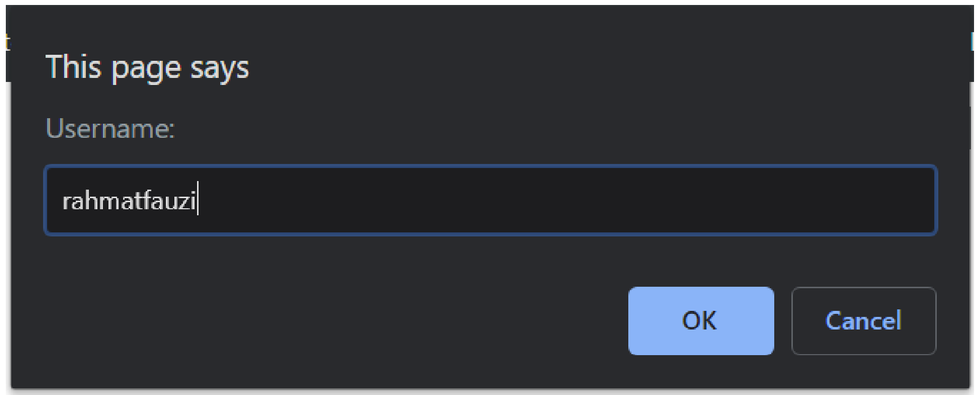
LETS CODE : PERCABANGAN NESTED

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Percabangan NESTED</title>
</head>
<body>
    <script>
        var username = prompt("Username:");
        var password = prompt("Password:");

        if(username == "rahmatfauzi"){
            if(password == "kopi"){
                document.write("<h2>Selamat datang pak bos!</h2>");
            } else {
                document.write("<p>Password salah, coba lagi!</p>");
            }
        } else {
            document.write("<p>Bos, Anda tidak terdaftar!</p>");
        }
    </script>
</body>
</html>
```

LETS CODE : HASIL PERCABANGAN NESTED



Bos, Anda tidak terdaftar!



Selamat datang pak bos!

Telkom University

Bab 6
PERULANGAN / LOOPING DALAM JAVASCRIPT

PERULANGAN / LOOPING

- Perulangan akan membantu kita mengeksekusi kode yang berulang-ulang, berapapun yang kita mau. Ada lima macam bentuk perulangan di Javascript. Secara umum, perulangan ini dibagi dua.
- Yaitu: **counted loop dan uncounted loop**.

Counted loop dan Uncounted loop

- **Counted Loop** merupakan perulangan yang jelas dan sudah jelas jumlah banyak perulangannya.
- Sedangkan **Uncounted Loop**, merupakan perulangan yang tidak jelas berapa kali ia harus mengulang.

Counted loop dan Uncounted loop

Perulangan yang termasuk dalam Counted Loop:

- ✓ Perulangan For
- ✓ Perulangan Foreach
- ✓ Perulangan Repeat

Perulangan yang termasuk dalam Uncounted Loop:

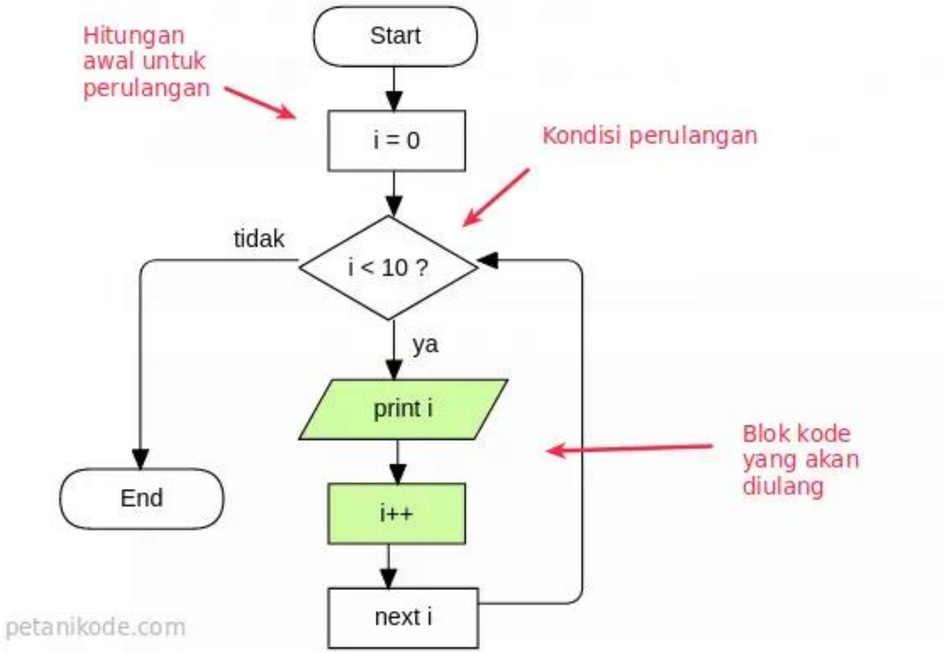
- ✓ Perulangan While
- ✓ Perulangan Do/While

Bentuk Perulangan pada Javascript

1. Percabangan For

```
for (statement 1; statement 2; statement 3) {
    code block to be executed
}
```

```
for (i = 0; i < 5; i++) {
    text += "The number is " + i + "<br>";
}
```



LETS CODE : PERULANGAN IF

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Perulangan / LOOPING</title>
</head>
<body>
    <script>
        for(let i = 0; i < 50; i++){
            document.write("<p>Perulangan ke-" + i + "</p>")
        }
    </script>
</body>
</html>
```

Perulangan ke-0

Perulangan ke-1

Perulangan ke-2

Perulangan ke-3

Perulangan ke-4

Perulangan ke-5

Perulangan ke-6

Perulangan ke-7

Perulangan ke-8

Perulangan ke-9

Perulangan ke-10

Perulangan ke-11

Perulangan ke-12

Bentuk Perulangan pada Javascript

2. Percabangan While

```
while (condition) {
    code block to be executed
}
```

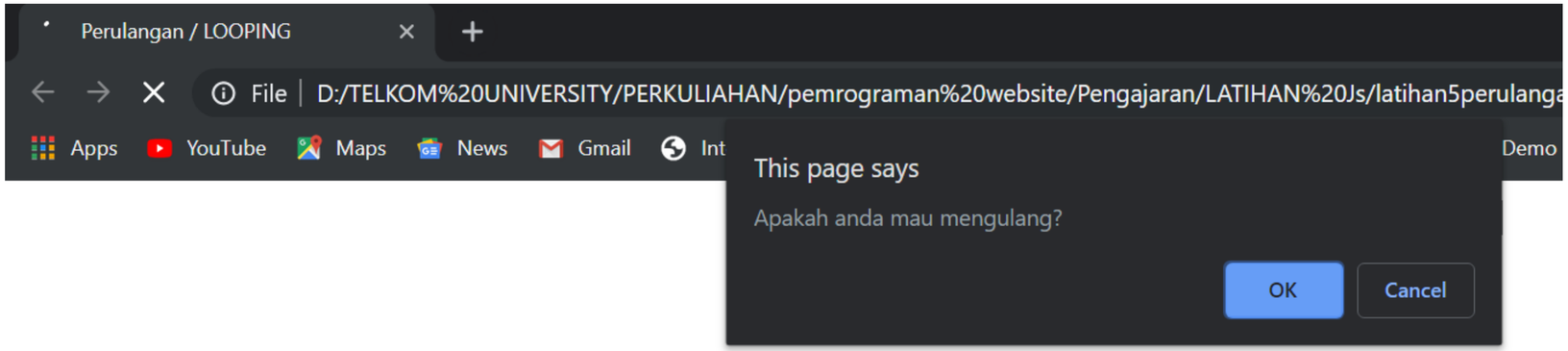
```
while (i < 10) {
    text += "The number is " + i;
    i++;
}
```

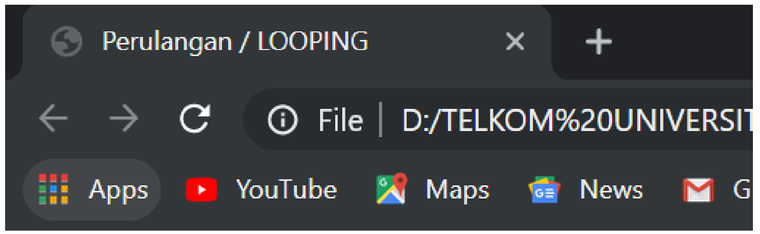
- Perulangan while merupakan perulangan yang termasuk dalam perulangan **uncounted loop**.
- Perulangan while juga dapat menjadi perulangan **yang counted loop** dengan memberikan counter di dalamnya.

LETS CODE : PERULANGAN WHILE

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Perulangan / LOOPING</title>
</head>
<body>
    <script>
        var ulangi = confirm("Apakah anda mau mengulang?");
        var counter = 0;

        while(ulangi){
            var jawab = confirm("Apakah anda mau mengulang?")
            counter++;
            if(jawab == false){
                ulangi = false;
            }
        }

        document.write("Perulangan sudah dilakuakn sebanyak "+ counter +" kali");

    </script>
</body>
</html>
```

LETS CODE : HASIL PERULANGAN IF





Perulangan sudah dilakuakn sebanyak 9 kali

Bentuk Perulangan pada Javascript

2. Percabangan Do-While

```
do {
    // blok kode yang akan diulang
} while (<kondisi>);
```

Perulangan do/while sama seperti perulangan while.

Perbedaanya:

Perulangan do/while akan melakukan perulangan sebanyak 1 kali terlebih dahulu, lalu mengecek kondisi yang ada di dalam kurung while.

Bentuk Perulangan pada Javascript

2. Percabangan Do-While

```
do {
    // blok kode yang akan diulang
} while (<kondisi>);
```

```
while (condition) {
    code block to be executed
}
```

```
while (i < 10) {
    text += "The number is " + i;
    i++;
}
```

Perulangan **do/while** akan mengecek kondisi di belakang (sesudah mengulang),

sedangkan **while** akan mencek kondisi di depan atau awal (sbelum mengulang).

LETS CODE : PERULANGAN DO-WHILE

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Perulangan / LOOPING</title>
</head>
<body>
    <script>

      var ulangi = confirm("Apakah anda mau mengulang?");;
       var counter = 0;

       do {
           counter++;
           ulangi = confirm("Apakah anda mau mengulang?");
       } while(ulangi)

       document.write("Perulangan sudah dilakuakn sebanyak "+ counter +" kali");

    </script>
</body>
</html>
```

LETS CODE : HASIL PERULANGAN DO-WHILE

Perulangan sudah dilakuakn sebanyak 1 kali

DO-WHILE

Perulangan sudah dilakuakn sebanyak 0 kali

WHILE

Bentuk Perulangan pada Javascript

4. Percabangan Foreach

Perulangan foreach biasanya digunakan untuk mencetak item di dalam array.

Perulangan ini termasuk dalam perulangan **counted loop**, karena jumlah perulangannya akan ditentukan oleh panjang dari array.

Ada **dua cara** menggunakan perulangan foreach di Javascript:

1. Menggunakan for dengan operator in;
2. Mengguunakan method forEach().

LETS CODE : Cara 1. PERULANGAN FOREACH

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Perulangan / LOOPING</title>
</head>
<body>
    <script>

      var languages = ["Javascript","HTML","CSS", "Typescript"];

      for(i=0;i<languages.length;i++)

      {
          document.write(i+"."+ languages[i]+"<br/>");
      }


    </script>
</body>
</html>
```

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Perulangan / LOOPING</title>
</head>
<body>
    <script>

       var languages = ["Javascript", "HTML", "CSS", "Typescript"];

        for(i in languages){
            document.write(i+". "+ languages[i] + "<br/>");
        }

    </script>
</body>
</html>
```

0.Javascript
1.HTML
2.CSS
3.Typescript

LETS CODE : Cara ke-2 PERULANGAN FOREACH

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Perulangan / LOOPING</title>
</head>
<body>
    <script>

       // kita punya array seperti berikut
        var days = ["Senin", "Selasa", "Rabu", "Kamis", "Jum'at", "Sabtu", "Minggu"];

        // Kemudian kita tampilkan semua hari
        // dengan menggunakan method foreach
        days.forEach(function(day){
            document.write("<p>" + day + "</p>");
        });

    </script>
</body>
</html>
```

LETS CODE : Cara ke-2 PERULANGAN FOREACH

Method **forEach()** memiliki parameter berupa fungsi **callback**. Sebenarnya kita juga bisa menggunakan arrow function seperti ini:

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Perulangan / LOOPING</title>
</head>
<body>
    <script>

        // kita punya array seperti berikut
        var days = ["Senin", "Selasa", "Rabu", "Kamis", "Jum'at", "Sabtu", "Minggu"];

        // Kemudian kita tampilkan semua hari
        // dengan menggunakan method foreach
        days.forEach((day) => {
            document.write("<p>" + day + "</p>");
        });

    </script>
</body>
</html>
```

LETS CODE : HASIL PERULANGAN FOREACH

Senin

Selasa

Rabu

Kamis

Jum'at

Sabtu

Minggu

Bentuk Perulangan pada Javascript

5. Percabangan Repeat()

- Perulangan dengen method atau fungsi repeat() termasuk dalam perulangan **counted loop**.

- Fungsi ini **khusus** digunakan untuk mengulang **sebuah teks (string).**

- Bisa dibilang: Ini merupakan singkat dari perulangan **for.**

LETS CODE : PERULANGAN Repeat()

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Perulangan / LOOPING</title>
</head>
<body>
    <script>

        for( let i = 0; i < 10; i++)
        {
            document.write("Ulangi kalimat ini!");
         }


    </script>
</body>
</html>
```

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Perulangan / LOOPING</title>
</head>
<body>
    <script>

        document.write("Ulangi kalimat ini! ".repeat(10));


    </script>
</body>
</html>
```

LETS CODE : HASIL PERULANGAN REPEAT()

Ulangi kalimat ini! Ulangi kalimat ini! Ulangi kalimat ini! Ulangi kalimat ini! Ulangi kalimat ini! Ulangi kalimat ini! Ulangi kalimat ini! Ulangi kalimat ini! Ulangi kalimat ini! Ulangi kalimat ini!

Bentuk Perulangan pada Javascript

6. Bonus: Perulangan Bersarang (Nested)

- Di dalam blok perulangan, kita juga dapat membuat perulangan.
- Ini disebut dengan *nested loop* atau perulangan bersarang atau perulangan di dalam perualangan.

LETS CODE : PERULANGAN NESTED

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Perulangan / LOOPING</title>
</head>
<body>
    <script>
      for(let i = 0; i < 10; i++)
      {
          for(let j = 0; j < 10; j++){
              document.write("<p>Perulangan ke " + i + "," + j + "</p>");
          }
      }


    </script>
</body>
</html>
```

- Pada perulangan tersebut, kita menggunakan **dua perulangan for**.
- Perulangan pertama menggunakan **variabel i** sebagai counter, sedangkan perulangan kedua menggunakan **variabel j** sebagai counter.

LETS CODE : HASIL PERULANGAN NESTED

Perulangan ke 0,0

Perulangan ke 0,1

Perulangan ke 0,2

Perulangan ke 0,3

Perulangan ke 0,4

Perulangan ke 0,5

Perulangan ke 0,6

Perulangan ke 0,7

Perulangan ke 0,8

Perulangan ke 0,9

Perulangan ke 1,0

Perulangan ke 1,1

Perulangan ke 1,2

Perulangan ke 1,3

Perulangan ke 1,4

Perulangan ke 1,5

LETS CODE : PERULANGAN NESTED

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Perulangan / LOOPING</title>
</head>
<body>
    <script>
     var ulangi = confirm("apakah anda ingin mengulang?");
     var counter = 0;

    while (ulangi) {
        counter++;
        var bintang = "*".repeat(counter) + "<br>";
        document.write(counter + ": " + bintang);
        ulangi = confirm("apakah anda ingin mengulang?");
    }


    </script>
</body>
</html>
```

1: *
2: **
3: ***
4: ****
5: *****
6: ******
7: *******
8: ********
9: *********

Telkom University

Bab 7

Struktur Data Array pada Javascript

LATAR BELAKANG Struktur Data Array Javascript

Bayangkan sekarang kita sedang membuat aplikasi web, lalu ingin menampilkan daftar nama-nama produk.

```
var produk1 = "Modem";
var produk2 = "Hardisk";
var produk3 = "Flashdisk";

document.write(`${produk1}<br>`);
document.write(`${produk2}<br>`);
document.write(`${produk3}<br>`);
```

Boleh-boleh saja. Tapi kurang efektif.

Struktur Data Array pada Javascript

- **Struktur data** merupakan cara-cara atau metode yang digunakan untuk menyimpan data di dalam memori komputer.
- Salah satu struktur data yang sering digunakan dalam pemrograman adalah **Array**.
- Array merupakan struktur data yang digunakan untuk **menyimpan sekumpulan data** dalam satu tempat.

Indeks [0]	Indeks [1]	Indeks [2]

LETS CODE Struktur Data Array pada Javascript

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Mengambil data dari array</title>
</head>
<body>
    <script>
        // membuat array
        var products = ["Senter", "Radio", "Antena", "Obeng"];

        // mengambil radio
        document.write(products[1]);

    </script>
</body>
</html>
```

LETS CODE Struktur Data Array pada Javascript

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Array dan perulangan</title>
</head>
<body>
    <script>
        // membuat array
        var products = ["Senter", "Radio", "Antena", "Obeng"];

        document.write("<h3>Daftar Produk:</h3>");
        document.write("<ol>");
        // menggunakan perulangan untuk mencetak semua isi array
        for(let i = 0; i < products.length; i++){
            document.write(`<li>${ products[i] }</li>`);
        }
        document.write("</ol>");
    </script>
</body>
</html>
```

Daftar Produk:

1. Senter
2. Radio
3. Antena
4. Obeng

LETS CODE Struktur Data Array pada Javascript

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Array dan perulangan</title>
</head>
<body>
    <script>
        // membuat array
        var products = ["Senter", "Radio", "Antena", "Obeng"];

        document.write("<h3>Daftar Produk:</h3>");
        document.write("<ol>");
        // menggunakan perulangan untuk mencetak semua isi array
        products.forEach((data) => {
            document.write(`<li>${data}</li>`);
        });
        document.write("</ol>");
    </script>
</body>
</html>
```

Daftar Produk:

1. Senter
2. Radio
3. Antena
4. Obeng

Cara Menambahkan Data ke Dalam Array

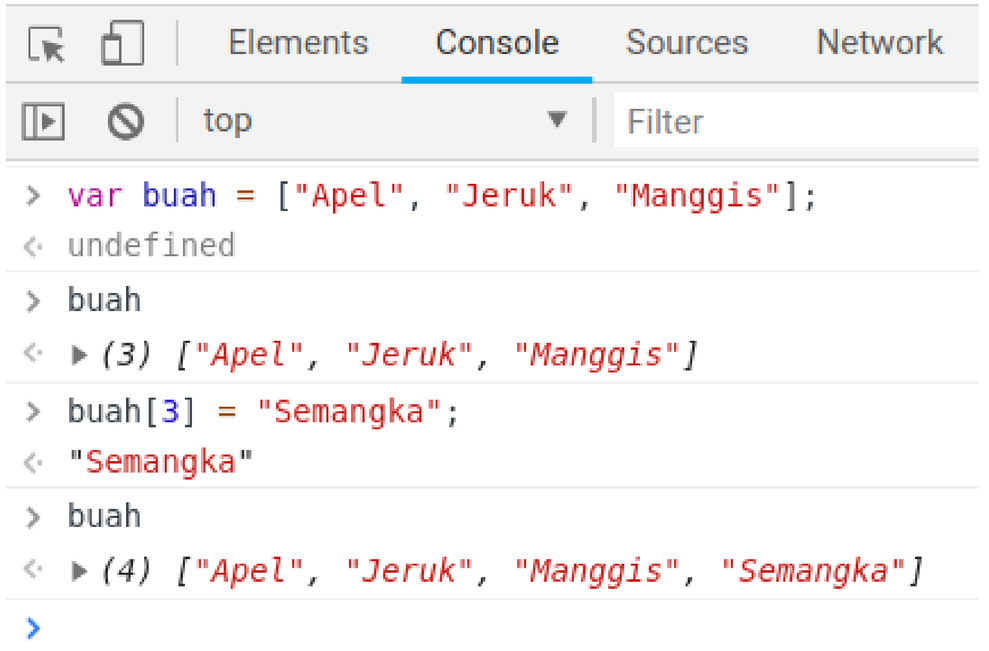
- Ada dua cara yang bisa dilakukan untuk menambah data ke dalam array:

- Mengisi menggunakan **indeks**;
- Mengisi menggunakan **method push().**

Cara 1 Menambahkan Data ke Dalam Array

```
var buah = ["Apel", "Jeruk", "Manggis"];
```

```
buah[3] = "Semangka";
```



LETS CODE Cara 2 Menambahkan Data ke Dalam Array

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Array dan perulangan</title>
</head>
<body>
        <script>
            // membuat array
            var products = ["Senter", "Radio", "Antena", "Obeng"];

            // menambahkan tv ke dalam array products
            products.push("Televisi");

            // menapilkan isi array
            document.write(products);
        </script>
</body>
</html>
```

Senter,Radio,Antena,Obeng,Televisi

Cara Menghapus Data dalam Array

- seperti menambahkan data ke array, menghapus data juga memiliki dua cara:

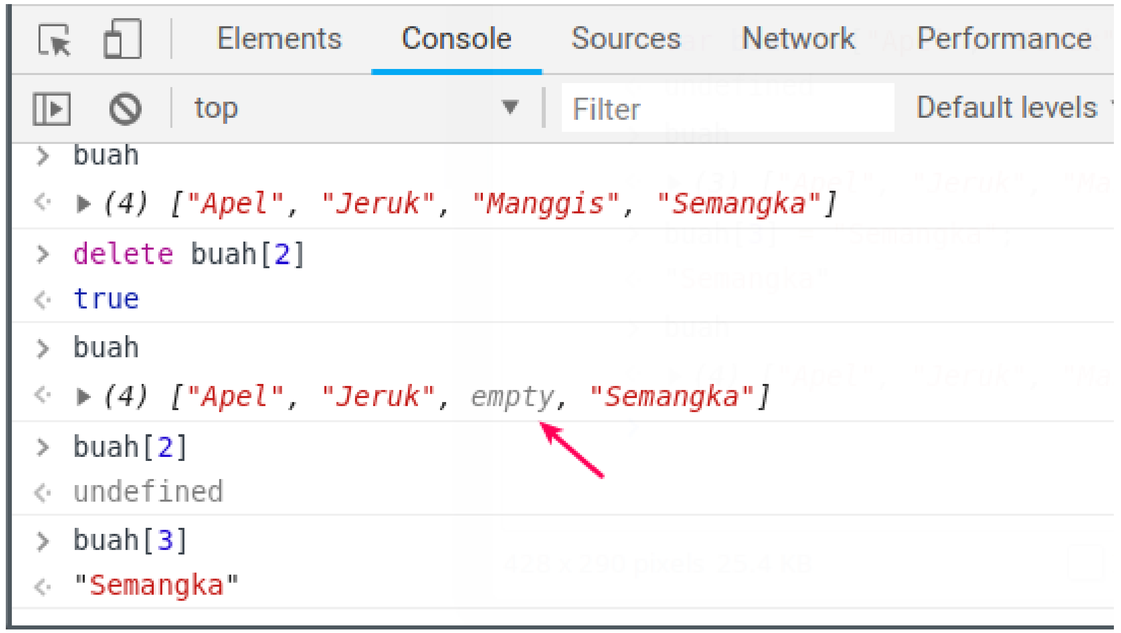
- Menggunakan **delete;**
- Menggunakan **method pop().**

Cara 1 Menghapus Data Dari Array

```
var buah = ["Apel", "Jeruk", "Manggis"];
```

```
buah[3] = "Semangka";
```

```
delete buah[2];
```



LETS CODE Cara 2 Menghapus Data Dari Array

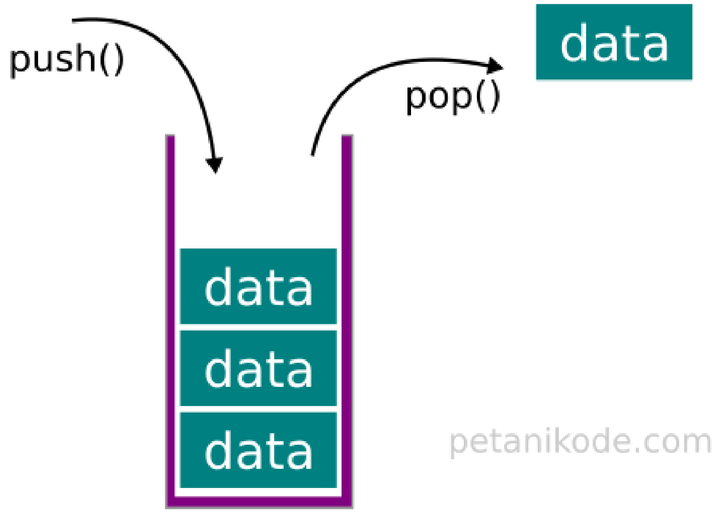
```html
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Mengisi data ke array</title>
</head>
<body>
    <script>
        // membuat array
        var products = ["Senter", "Radio", "Antena", "Obeng","Kursi"];

        // menghapus isi array
        products.pop();
        products.pop();
        products.pop();

        // menapilkan isi array
        document.write(products);


    </script>
</body>
</html>
```

Senter,Radio

Cara 2 Menghapus Data Dari Array

- Method pop() akan menghapus array yang ada di paling belakang.

- Array pada javascript dapat kita pandang sebagai sebuah stack (tumpukan), yang mana memiliki sifat LILO (Last in Last out).



LETS CODE Cara 2 Menghapus Data Dari Array (Dari depan)

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Mengisi data ke array</title>
</head>
<body>
    <script>
        // membuat array
        var products = ["Senter", "Radio", "Antena", "Obeng","Kursi"];

        // menghapus isi array
        products.shift();
        products.shift();


        // menapilkan isi array
        document.write(products);


    </script>
</body>
</html>
```

Antena,Obeng,Kursi

Menghapus Data pada Indeks Tertentu

- Apabila kita ingin menghapus data pada inteks tertentu, maka fungsi atau method yang digunakan adalah splice().
- Fungsi ini memiliki dua parameter yang harus diberikan:

```
array.splice(<indeks>, <total>);
```

- Keterangan:
- <indeks> adalah indeks dari data di dalam array yang akan dihapus;
- <total> adalah jumlah data yang akan dihapus dari indeks tersebut.

Menghapus Data pada Indeks Tertentu

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Mengisi data ke array</title>
</head>
<body>
    <script>
        var bunga = ["Mawar", "Melati", "Anggrek", "Sakura","Dahlia","Bonsai","Kamboja"];

    // hapus Anggrek
    bunga.splice(3, 1);

    // menapilkan isi array
    document.write(bunga);

    </script>
</body>
</html>
```

Mawar,Melati,Anggrek,Dahlia,Bonsai,Kamboja

Menghapus Data pada Indeks Tertentu

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Mengisi data ke array</title>
</head>
<body>
    <script>
        var bunga = ["Mawar", "Melati", "Anggrek", "Sakura","Dahlia","Bonsai","Kamboja"];

    // hapus Anggrek
    bunga.splice(3, 3);

    // menapilkan isi array
    document.write(bunga);

    </script>
</body>
</html>
```

Mawar,Melati,Anggrek,Kamboja

Mengubah Isi Data Pada Array

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Mengisi data ke array</title>
</head>
<body>
    <script>
       var bunga = ["Mawar", "Melati", "Anggrek", "Sakura","Dahlia","Bonsai","Kamboja"];

        bunga[1] = "Asoka";

    // menapilkan isi array
    document.write(bunga);

    </script>
</body>
</html>
```

Mawar,Asoka,Anggrek,Sakura,Dahlia,Bonsai,Kamboja

Method-mothod Array Lain

- 1. Method filter()
- Method filter() berfungsi untuk menyaring data dari array.
- Parameter yang harus diberikan pada method filter() sama seperti method forEach(), yaitu: sebuah fungsi callback.

Method-mothod Array Lain

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Mengisi data ke array</title>
</head>
<body>
    <script>

    const angka = [1, 2, 3, 4, 5, 6, 7, 8, 9];

    // Kita ambil data yang hanya habis dibagi dua saja
    const filteredArray = angka.filter((item) => {return item % 2 === 0});

    document.write(filteredArray);




    </script>
</body>
</html>
```

2,4,6,8

Method-mothod Array Lain

2. Method includes()

- Method ini berfungsi untuk mengecek apakah sebuah data ada di dalam array atau tidak. Biasanya digunakan untuk melakukan pencarian untuk memastikan data sudah ada di dalam array.

Method-mothod Array Lain

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Mengisi data ke array</title>
</head>
<body>
    <script>

    var tanaman = ["Padi", "Kacang", "Jagung", "Kedelai"];

    // apakah kacang sudah ada di dalam array tanaman?
    var adaKacang = tanaman.includes("Kacang");

    document.write(adaKacang); //true

    // apakah bayam ada?
    var adaBayam = tanaman.includes("Bayam");

    document.write(adaBayam); //false



    </script>
</body>
</html>
```

truefalse

Method-mothod Array Lain

- 3. Method sort()

Method sort() berfungsi untuk mengurutkan data pada array

Method-mothod Array Lain

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Mengisi data ke array</title>
</head>
<body>
    <script>

        var alfabet = ['a','f','z','e','r','g'];
        var angka = [3,1,2,6,8,5];

        document.write(alfabet.sort());

        document.write(angka.sort());
        // console.log(alfabet.sort()); //->  ["a", "e", "f", "g", "r", "z"]
        // console.log(angka.sort()); // -> [1, 2, 3, 4, 5, 6, 7, 8, 9]

    </script>
</body>
</html>
```

a,e,f,g,r,z1,2,3,5,6,8

Telkom University

Bab 8

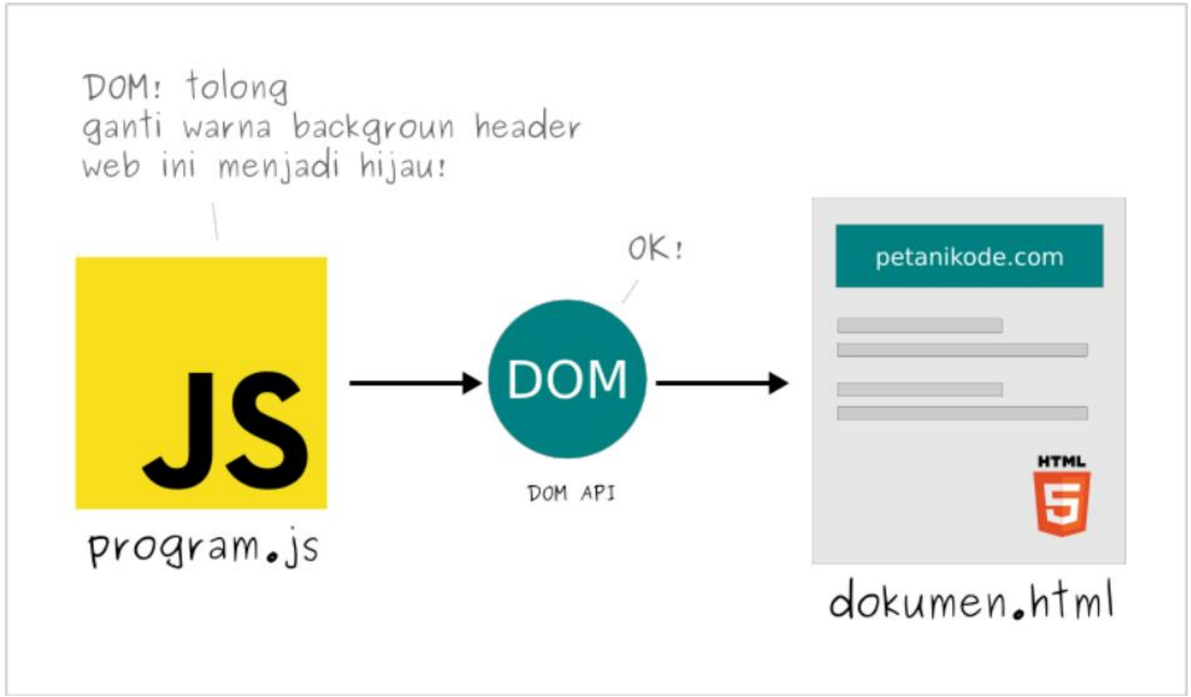
DOM JavaScript

DOM API Js

- Saat kamu memutuskan akan belajar Javascript, maka **wajib** hukumnya memahami tentang DOM.
- Mengapa Wajib ? Karena...
- Salah satu tugas utama Javascript di dalam web adalah membuat halaman web agar terlihat dinamis.
- Hal ini bisa dilakukan oleh Javascript dengan bantuan DOM.

Apa itu DOM API?

- DOM merupakan singkatan dari ***Document Object Model***.
- Artinya, dokumen (HTML) yang dimodelkan dalam sebuah objek.
- Objek dari dokumen ini menyediakan sekumpulan fungsi dan atribut/data yang bisa kita manfaatkan dalam membuat program Javascript. Inilah yang disebut API *(Application Programming Interface)*.



Bagaimana Cara Menggunakan DOM?

- Seperti yang kita sudah katahui, DOM adalah sebuah objek untuk memodelkan dokumen HTML.

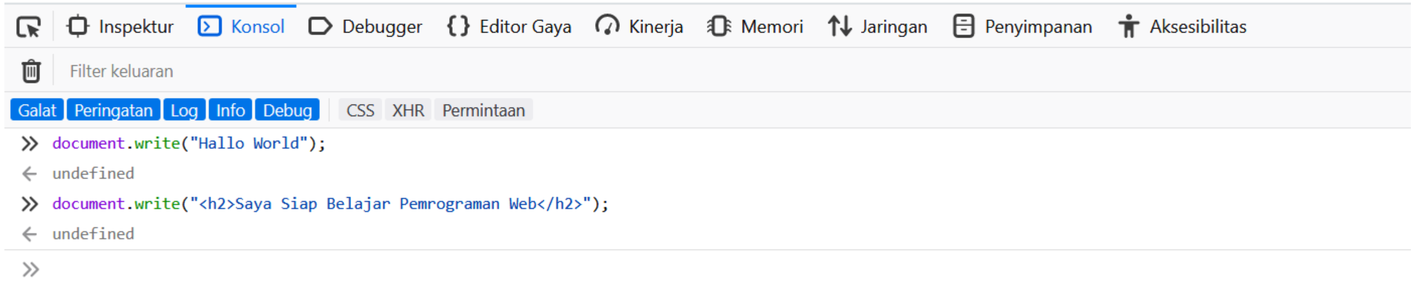
- Objek DOM di javascript bernama document. Objek ini berisi segala hal yang kita butuhkan untuk memanipulasi HTML.

- Jika kita coba ketik document pada console Javascript, maka yang akan tampil adalah kode HTML.

Bagaimana Cara Menggunakan DOM?

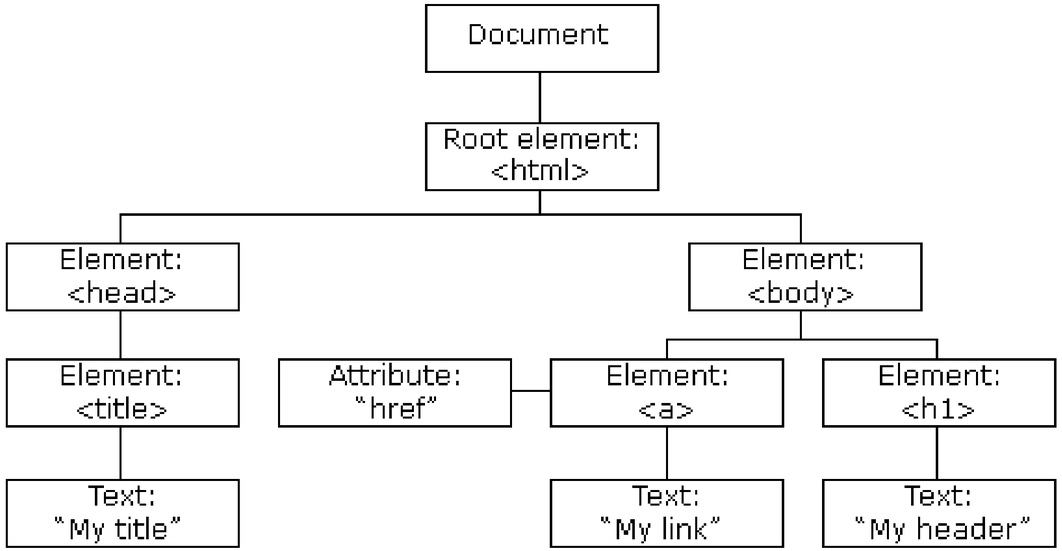
Hallo World

Saya Siap Belajar Pemrograman Web



Mengkases Elemen Tertentu dengan DOM

- Objek document adalah model dari dokumen HTML. Objek ini berisi kumpulan fungsi dan atribut berupa objek dari elemen HTML yang bisa digambarkan dalam bentuk pohon seperti ini:



LETS CODE : Contoh DOM Sederhana

```html
<!DOCTYPE html>
<html>
<head>
    <title>Memilih Elemen Berdasarkan ID</title>
</head>
<body>

    <!-- Elemen div yang akan kita pilih dari JS -->
    <div id="tutorial"></div>


    <script type="text/javascript">
        // mengakses elemen tutorial
        var tutorial = document.getElementById("tutorial");

        // mengisi teks ke dalam elemen
        tutorial.innerText = "Tutorial Javascript";

        // memberikan CSS ke elemen
        tutorial.style.backgroundColor = "gold";
        tutorial.style.padding = "10px";

    </script>

</body>
</html>
```

LETS CODE : Contoh DOM Sederhana

```
!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>DOM API Javascript</title>
</head>
<body>
    <p class="paragraf">Lorem ipsum dolor sit amet consectetur adipisicing elit.
        Quo quaerat recusandae qui ullam eaque cumque ea fugit,
        debitis commodi ipsum illo dolorum consequatur sed laudantium suscipit quis magni,
        maiores in?</p>

    <p class="paragraf">Lorem ipsum dolor sit amet consectetur adipisicing elit.
        Quo quaerat recusandae qui ullam eaque cumque ea fugit,
        debitis commodi ipsum illo dolorum consequatur sed laudantium suscipit quis magni,
        maiores in?</p>

    <p class="paragraf">Lorem ipsum dolor sit amet consectetur adipisicing elit.
        Quo quaerat recusandae qui ullam eaque cumque ea fugit,
        debitis commodi ipsum illo dolorum consequatur sed laudantium suscipit quis magni,
        maiores in?</p>

    <script>
        var paragraf = document.getElementsByClassName("paragraf");
        console.log(paragraf);
    </script>

</body>
```

!DOCTYPE html>

Lorem ipsum dolor sit amet consectetur adipisicing elit. Quo quaerat recusandae qui ullam eaque cumque ea fugit, debitis commodi ipsum illo dolorum consequatur sed laudantium suscipit quis magni, maiores in?

Lorem ipsum dolor sit amet consectetur adipisicing elit. Quo quaerat recusandae qui ullam eaque cumque ea fugit, debitis commodi ipsum illo dolorum consequatur sed laudantium suscipit quis magni, maiores in?

Lorem ipsum dolor sit amet consectetur adipisicing elit. Quo quaerat recusandae qui ullam eaque cumque ea fugit, debitis commodi ipsum illo dolorum consequatur sed laudantium suscipit quis magni, maiores in?

LETS CODE : Contoh DOM Sederhana

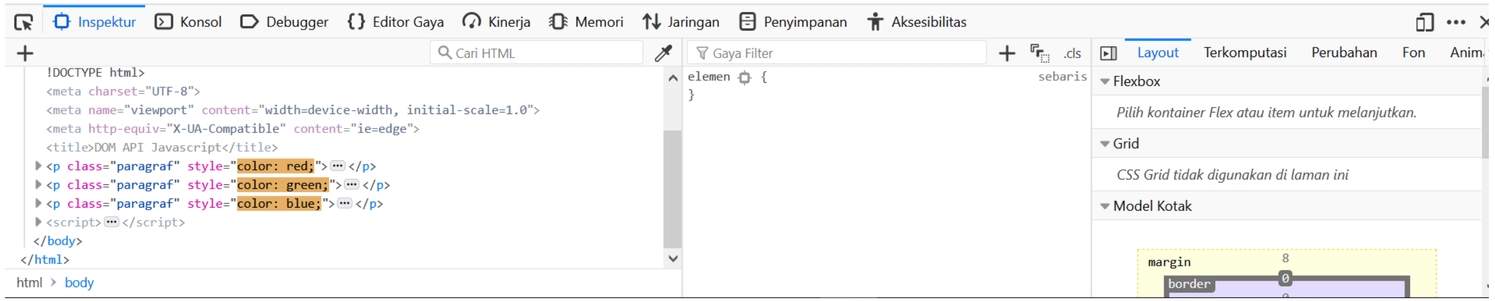
```
<script>
    var paragraf = document.getElementsByClassName("paragraf");
    setInterval(function () {
        paragraf[0].style.color = "red";
        paragraf[1].style.color = "green";
        paragraf[2].style.color = "blue";

        setTimeout(function () {
            paragraf[0].style.color = "black";
            paragraf[1].style.color = "black";
            paragraf[2].style.color = "black";
        }, 500)
    }, 1000);
</script>
```

!DOCTYPE html>

Lorem ipsum dolor sit amet consectetur adipisicing elit. Quo quaerat recusandae qui ullam eaque cumque ea fugit, debitis commodi ipsum illo dolorum consequatur sed laudantium suscipit quis magni, maiores in?

Lorem ipsum dolor sit amet consectetur adipisicing elit. Quo quaerat recusandae qui ullam eaque cumque ea fugit, debitis commodi ipsum illo dolorum consequatur sed laudantium suscipit quis magni, maiores in?

Lorem ipsum dolor sit amet consectetur adipisicing elit. Quo quaerat recusandae qui ullam eaque cumque ea fugit, debitis commodi ipsum illo dolorum consequatur sed laudantium suscipit quis magni, maiores in?



Membuat Elemen dengan DOM API

- DOM juga menyediakan fungsi untuk membuat elemen HTML.
- Salah satunya adalah fungsi createElement().

Contoh:

```
document.createElement('p');
```

- Maka, akan tercipta elemen <p> baru. Namun tidak akan ditampilkan ke dalam halaman web.
- Mengapa tidak ditampilakn?

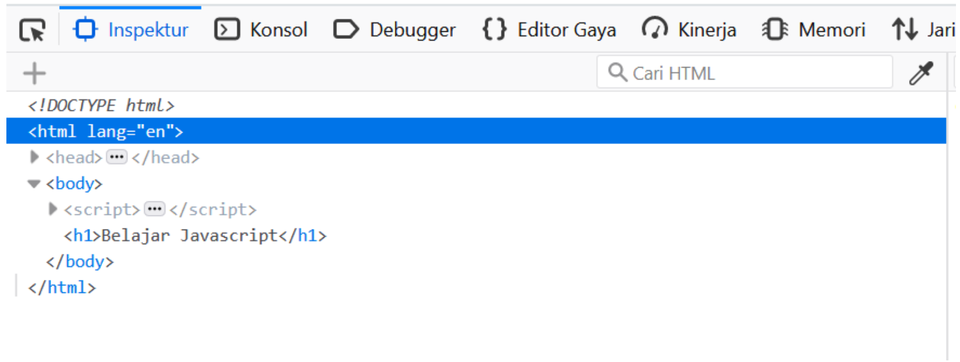
- Karena kita belum menambahkannya ke dalam body dokumen.

- Cara menambahkannya ke body dokumen, kita bisa gunakan fungsi **append().**

Membuat Elemen dengan DOM API

```html
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>DOM API Javascript</title>
</head>

<body>

    <script>
        // membuat element h1
        var judul = document.createElement("h1");

        // mengisi kontent elemen
        judul.textContent = "Belajar Javascript";

        // menambahkan elemen ke dalam tag body
        document.body.append(judul);
    </script>

</body>
</html>
```

Belajar Javascript

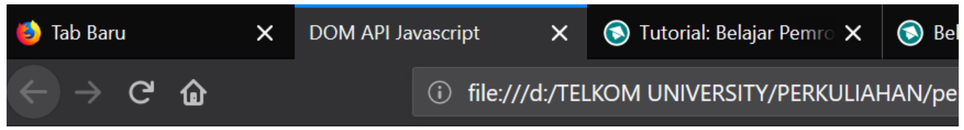


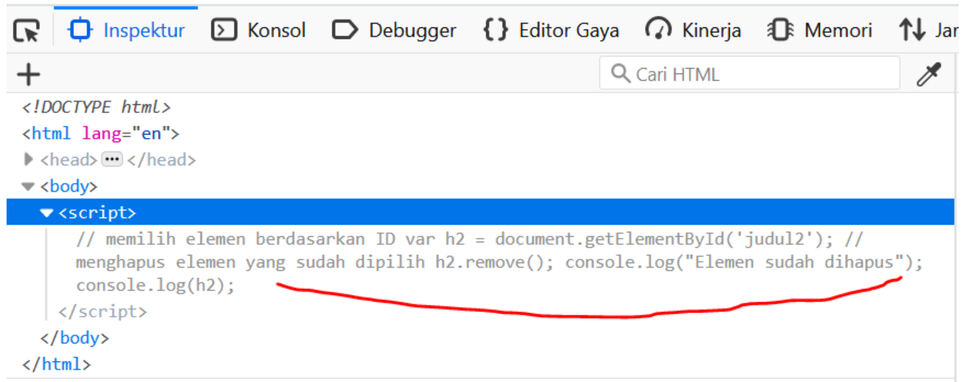
Menghapus Elemen dengan DOM API

- Jika kita menggunakan fungsi **append()** untuk menambahkan elemen, maka untuk menghapusnya kita menggunakan fungsi **remove().**

Menghapus Elemen dengan DOM API

```html
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>DOM API Javascript</title>
</head>

<body>

    <h2 id="judul2">Delete Saya!</h2>

    <script>
        // memilih elemen berdasarkan ID
        var h2 = document.getElementById('judul2');

        // menghapus elemen yang sudah dipilih
        h2.remove();

        console.log("Elemen sudah dihapus");
        console.log(h2);
    </script>

</body>
</html>
```





Method lain dalam DOM

Property / Method	Description
activeElement	Returns the currently focused element in the document
addEventListener()	Attaches an event handler to the document
adoptNode()	Adopts a node from another document
anchors	Returns a collection of all <a> elements in the document that have a name attribute
applets	Returns a collection of all <applet> elements in the document
baseURI	Returns the absolute base URI of a document
body	Sets or returns the document's body (the <body> element)
close()	Closes the output stream previously opened with document.open()
cookie	Returns all name/value pairs of cookies in the document
charset	Deprecated. Use characterSet instead. Returns the character encoding for the document
characterSet	Returns the character encoding for the document
createAttribute()	Creates an attribute node
createComment()	Creates a Comment node with the specified text
createDocumentFragment()	Creates an empty DocumentFragment node
createElement()	Creates an Element node
createEvent()	Creates a new event
createTextNode()	Creates a Text node
defaultView	Returns the window object associated with a document, or null if none is available.
designMode	Controls whether the entire document should be editable or not.
doctype	Returns the Document Type Declaration associated with the document
documentElement	Returns the Document Element of the document (the <html> element)
documentMode	Returns the mode used by the browser to render the document
documentURI	Sets or returns the location of the document
domain	Returns the domain name of the server that loaded the document
domConfig	Obsolete. Returns the DOM configuration of the document
embeds	Returns a collection of all <embed> elements the document
execCommand()	Invokes the specified clipboard operation on the element currently having focus.
forms	Returns a collection of all <form> elements in the document
fullscreenElement	Returns the current element that is displayed in fullscreen mode
fullscreenEnabled()	Returns a Boolean value indicating whether the document can be viewed in fullscreen mode
getElementById()	Returns the element that has the ID attribute with the specified value
getElementsByClassName()	Returns a NodeList containing all elements with the specified class name
getElementsByName()	Returns a NodeList containing all elements with a specified name
getElementsByTagName()	Returns a NodeList containing all elements with the specified tag name
hasFocus()	Returns a Boolean value indicating whether the document has focus
head	Returns the <head> element of the document
images	Returns a collection of all <img> elements in the document
implementation	Returns the DOMImplementation object that handles this document
importNode()	Imports a node from another document
inputEncoding	Returns the encoding, character set, used for the document
lastModified	Returns the date and time the document was last modified
links	Returns a collection of all <a> and <area> elements in the document that have a href attribute
normalize()	Removes empty Text nodes, and joins adjacent nodes
normalizeDocument()	Removes empty Text nodes, and joins adjacent nodes
open()	Opens an HTML output stream to collect output from document.write()
querySelector()	Returns the first element that matches a specified CSS selector(s) in the document
querySelectorAll()	Returns a static NodeList containing all elements that matches a specified CSS selector(s) in the document
readyState	Returns the (loading) status of the document
referrer	Returns the URL of the document that loaded the current document
removeEventListener()	Removes an event handler from the document (that has been attached with the addEventListener() method)
renameNode()	Renames the specified node
scripts	Returns a collection of <script> elements in the document
strictErrorChecking	Sets or returns whether error-checking is enforced or not

Telkom University

Thank You
Any Question ?

Tugas di Rumah

Semua latihan di slide ini dikumpulkan ke email saya :

Dengan subjek : NamaKelas_NamaMahasiswa_TugasJs

Lampirkan sintaks coding dan screenshoot hasilnya.

Daftar Referensi

1. Modul Praktikum WAD SI FRI
2. Modul Praktikum WAD Teknik Informatika FIF
3. Petanikode.com
4. Youtube.com/webprogrammingUNPAS