**MODUL PEMROGRAMAN WEB**

**(CCP220**

**MODUL SESI 7**

**JAVASCRIPT LANJUTAN**

**DISUSUN OLEH**

**YUNITA FAUZIA ACHMAD, S.KOM, M.KOM**

**UNIVERSITAS ESA UNGGUL**

**2019**

**Looping dan Percabangan pada Javascript**

1. **Kemampuan Akhir Yang Diharapkan**

Setelah mempelajari modul ini, diharapkan mahasiswa mampu :

* 1. Menjelaskan perintah Looping dan Percabangan pada Javascript
  2. Menjelaskan penerapan perintah Looping dan Percabangan pada javascript
  3. Menerapkan perintah Looping dan Percabangan di javascript pada web

1. **Uraian dan Contoh** 
   1. **Pengertian Looping**

Looping atau perulangan adalah suatu blok perintah atau kode yang diulang selama syarat tersebut bernilai TRUE.

Perulangan di dalam Bahasa pemrograman digunakan untuk mengulang perintah program. Terdapat beberapa struktur perulangan yang di dukung oleh javascript.

Pada javascript terdapat 5 macam bentuk perulangan di javascript dan secara umum dibagi menjadi dua, yaitu :

* + - * 1. **Counted loop**

merupakan perulangan yang jelas dan sudah tentu banyak perulangannya

Perulangan yang termasuk ke dalam counted loop, diantaranya adalah :

1. Perulangan for
2. Perulangan foreach
3. Perulangan repeat
4. **Perulangan For di javascript**

Perulangan For sering digunakan untuk membuat perulangan dengan jumlah perulangan yang telah diketahui secara pasti.

Berikut sintaks dari perulangan For:

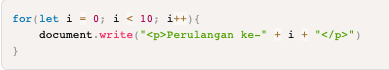
For (Inisialisasi Awal ; kondisi perulangan ; increment / descrement)

{

Kode yang dieksekusi apabila syarat terpenuhi

}

Contoh :

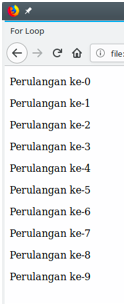


Keterangan :

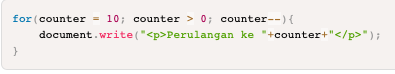
Kondisi diatas akan menentukan:

* Hitungan yang diawali dari 0 => i = 0;
* Kondisi Pengulangan yang di lakukan mencapai i < 10;
* Setiap perulangan i akan bertambah +1 => i++;

Hasil dari perulangan diatas:



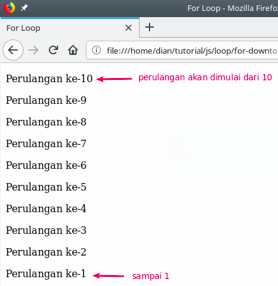
Contoh 2 :



Keterangan :

Nama variabel pada perulangan for tidak hanya menggunakan karakter i, j, atau karakter lain. Tetapi juga dapat berupa string seperti contoh 2 menggunakan variabel counter. Pada insialisasi awal yang akan di lakukan perulangan adalah 10 . kondisi perulangan disini counter > 0. Berbeda dengan contoh 1 dimana kondisi perulangan akan dilakukan penambahan pada nilai i => +1. Untuk contoh 2 kondisi perulangan yang di tampilkan adalah pengurangan nilai awal pada variabel counter => -1.

Hasil :



1. **Perulangan Foreach di javascript**

Perulangan foreach biasa digunakan untuk mencetak item di dalam array. Perulangan ini termasuk perulangan *counted loop*, dikarenakan jumlah perulangan akan ditentukan oleh Panjang dari array.

Berikut penggunakan perulangan foreach pada javascript, diantaranya :

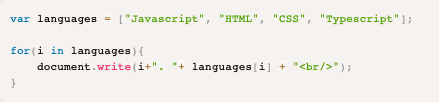
* Menggunakan for dengan operator in

Perulangan foreach tanpa menggunakan operator in

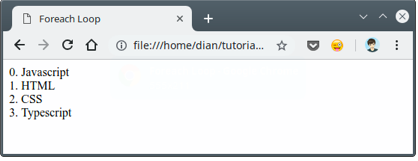
Contoh :



Perulangan ini dapat dibuat lebih sederhana dengan menggunakan operator in, seperti dibawah ini:



Hasil :



* Menggunakan method forEach()

Contoh:

/ kita punya array seperti berikut

var days = ["Senin", "Selasa", "Rabu", "Kamis", "Jum'at", "Sabtu", "Minggu"];

// Kemudian kita tampilkan semua hari

// dengan menggunakan method foreach

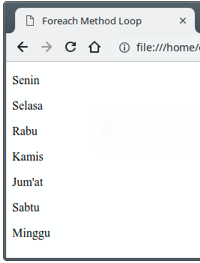
days.forEach(function(day){

document.write("<p>" + day + "</p>");

});

Metode forEach() memiliki parameter berupa fungsi *callback.*

Hasil :

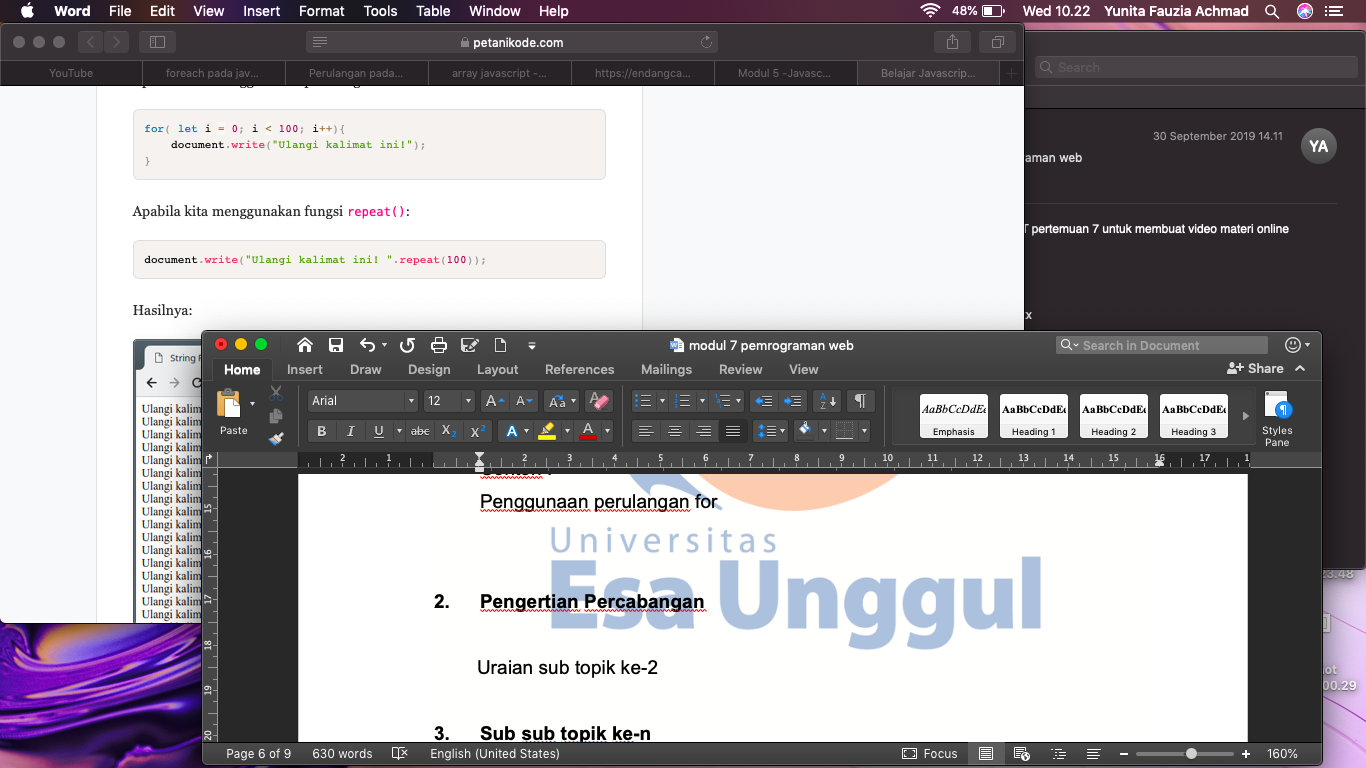


1. **Perulangan dengan Method repeat()**

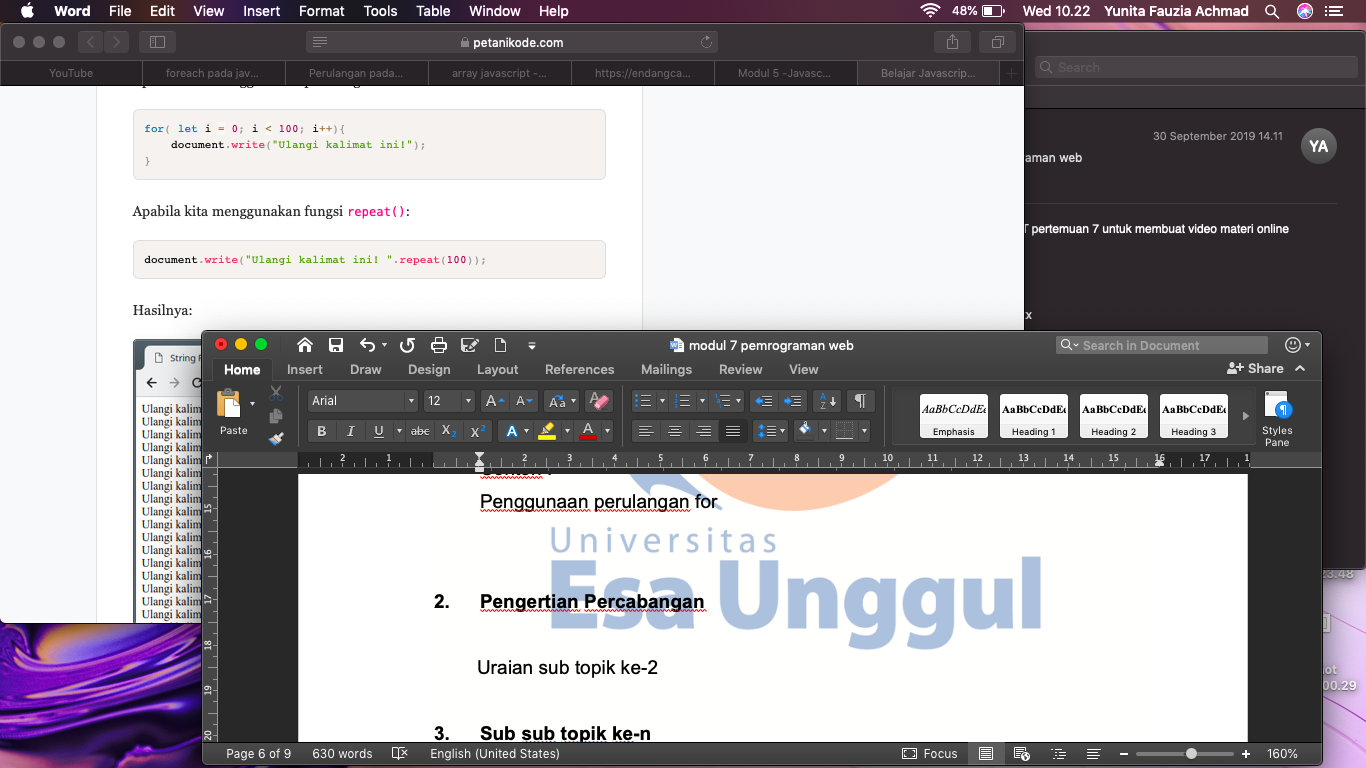
Perulangan dengan method dan fungsi repeat() termasuk ke dalam perulangan *counted loop.* Fungsi ini digunakan untuk mengulang sebuah teks (string). Dan method repeat() merupakan perulangan sederhana dari perulangan for.

Contoh :

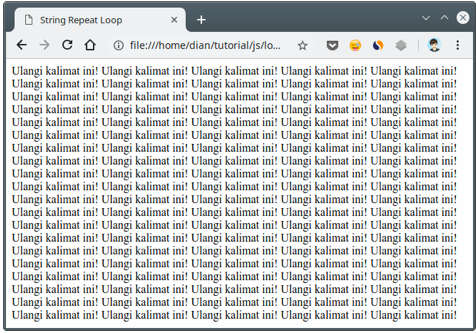
Penggunaan perulangan for

****

Dapat di gantikan dengan menggunakan method atau fungsi repeat()



Hasil:



* + - 1. ***uncounted loop*** merupakan perulangan yang tidak jelas berapa kali dilakukan perulangan

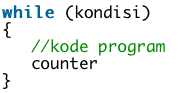
Perulangan yang termasuk kedalam *uncounted loop*, diantaranya :

1. **Perulangan while**

Perulangan while digunakan untuk melakukan pengulangan suatu kelompok perintah berdasarkan suatu kondisi, dengan pemeriksaan yang dilakukan pada diawal bagian perulangan. Pengulangan akan terus dilakukan selama pemeriksaan kondisi adalah benar. Dan akan berhenti jika kondisi salah.

**Cara penulisan struktur perulangan while**

Penulisan struktur pengulangan while mirip dengan struktur percabangan IF, berikut struktur penulisan perulangan while:



Keterangan :

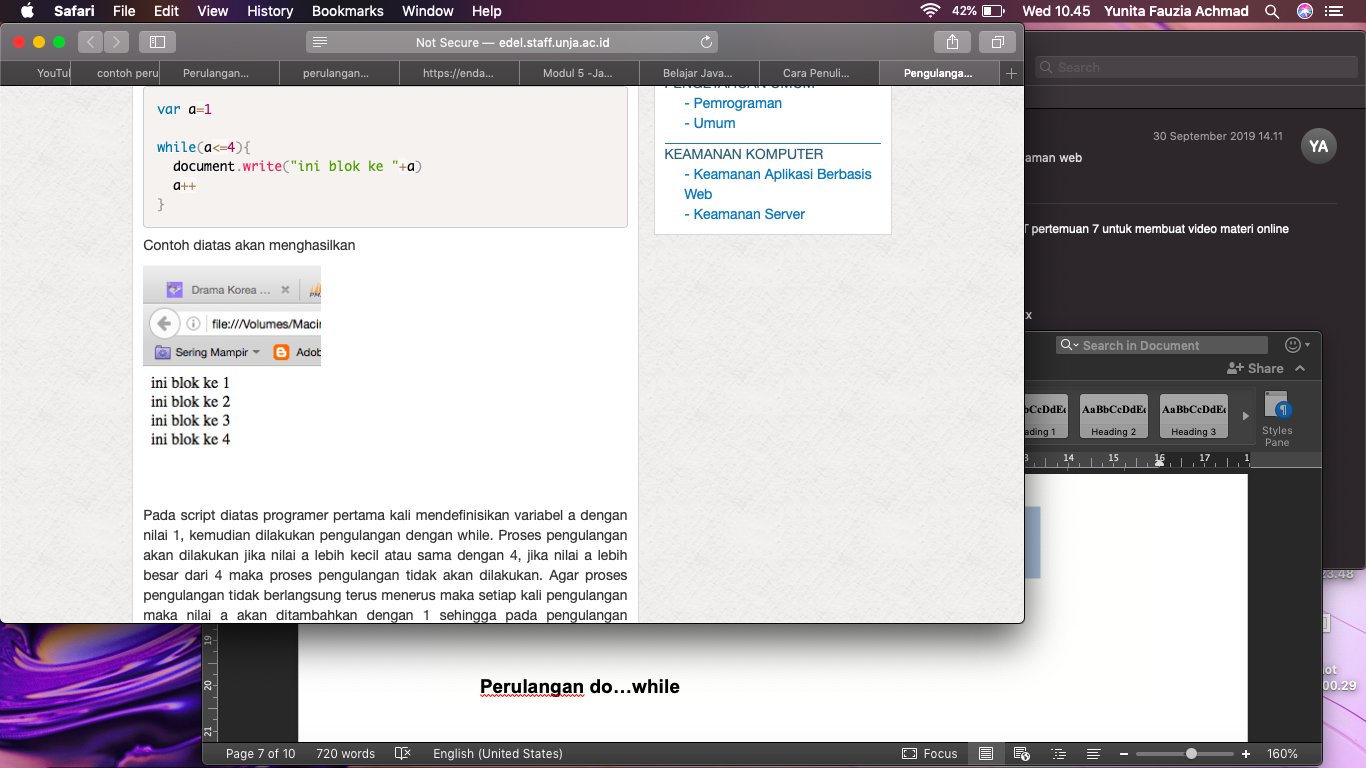
Kondisi akan selalu diperiksa pada setiap perulangan dan dikendalikan kondisi pada bagian counter di dalam perulangan

Contoh :



Pada script ini pertama kali didefinisikan variabel a dengan nilai 1, kemudian dilakukan pengulanagn dengan while. Proses pengulanagn akan dilakukan jika nilai a ≤ 4. Jika nilai a ≥ 4, maka proses pengulanagn tidak akan dilakukan. Jika ingin pengulangan tidak berlangsung terus menerus maka setiap kali pengulangan

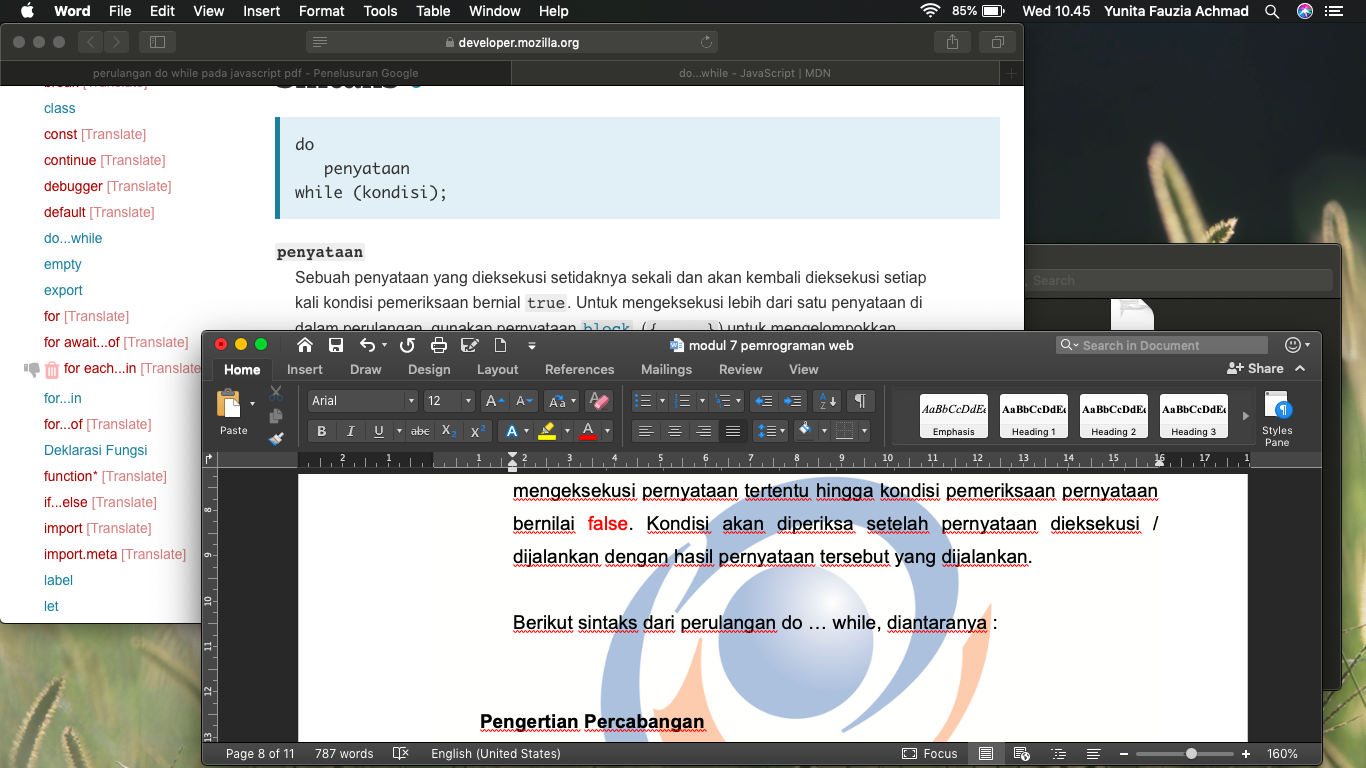
Hasil :



* 1. **Perulangan do…while**

Perulangan do…while digunakan untuk membuat perulangan yang mengeksekusi pernyataan tertentu hingga kondisi pemeriksaan pernyataan bernilai false. Kondisi akan diperiksa setelah pernyataan dieksekusi / dijalankan dengan hasil pernyataan tersebut yang dijalankan.

Berikut sintaks dari perulangan do … while, diantaranya :

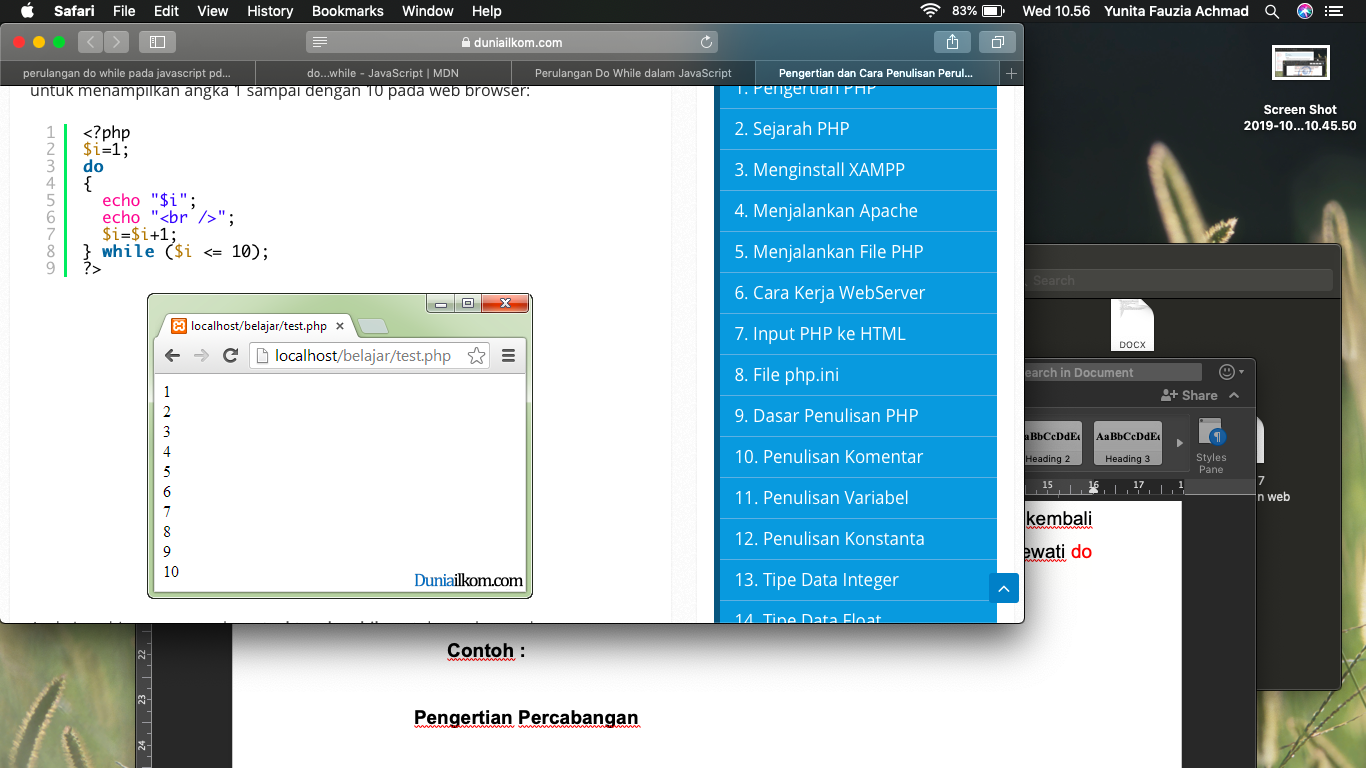
****

Keterangan :

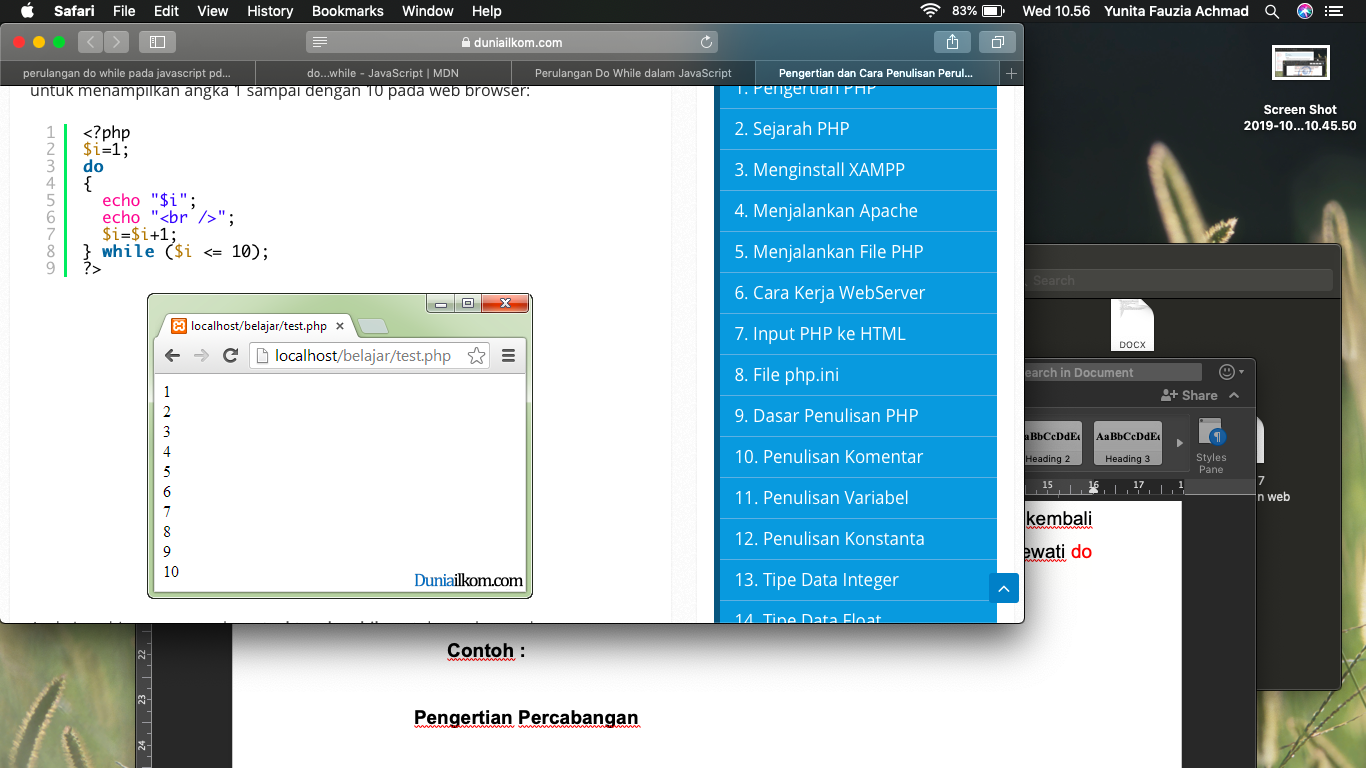
**Pernyataan** = sebuah pernyataan yang dieksekusi setidaknya sekali dan akan kembali dieksekusi setiap kondisi pemeriksaan bernilai true. Untuk mengeksekusi lebih dari satu pernyataan dalam perulangan ini, menggunakan pernyataan BLOCK => ({ ….. }) yang berguna untuk menyatakan pengelompokkan pernyataan.

**Kondisi** = sebuah ekspresi yang diperiksa setelah perulangan. Bila kondisi yang diperiksa bernilai true, maka pernyataan tersebut akan kembali dieksekusi. Tetapi jika kondisi bernilai false, maka kendali akan melewati do … while dan meneruskan ke pernyataan selanjutnya.

**Contoh :**

****

**Hasil :**

****

1. **Pengertian Percabangan**

Dalam merancang sebuah halaman web yang dinamis dan interaktif diperlukan perintah – perintah yang dapat mengatur aliran informasi pada halaman web. Dengan menggunakan javascript dapat melakukan perhitungan dan membuat keputusan jalur mana yang akan dieksekusi berdasarkan hasil komputasi.

Pada kondisi percabangan terdapat beberapa perintah, diantaranya :

* 1. **Perintah If …**

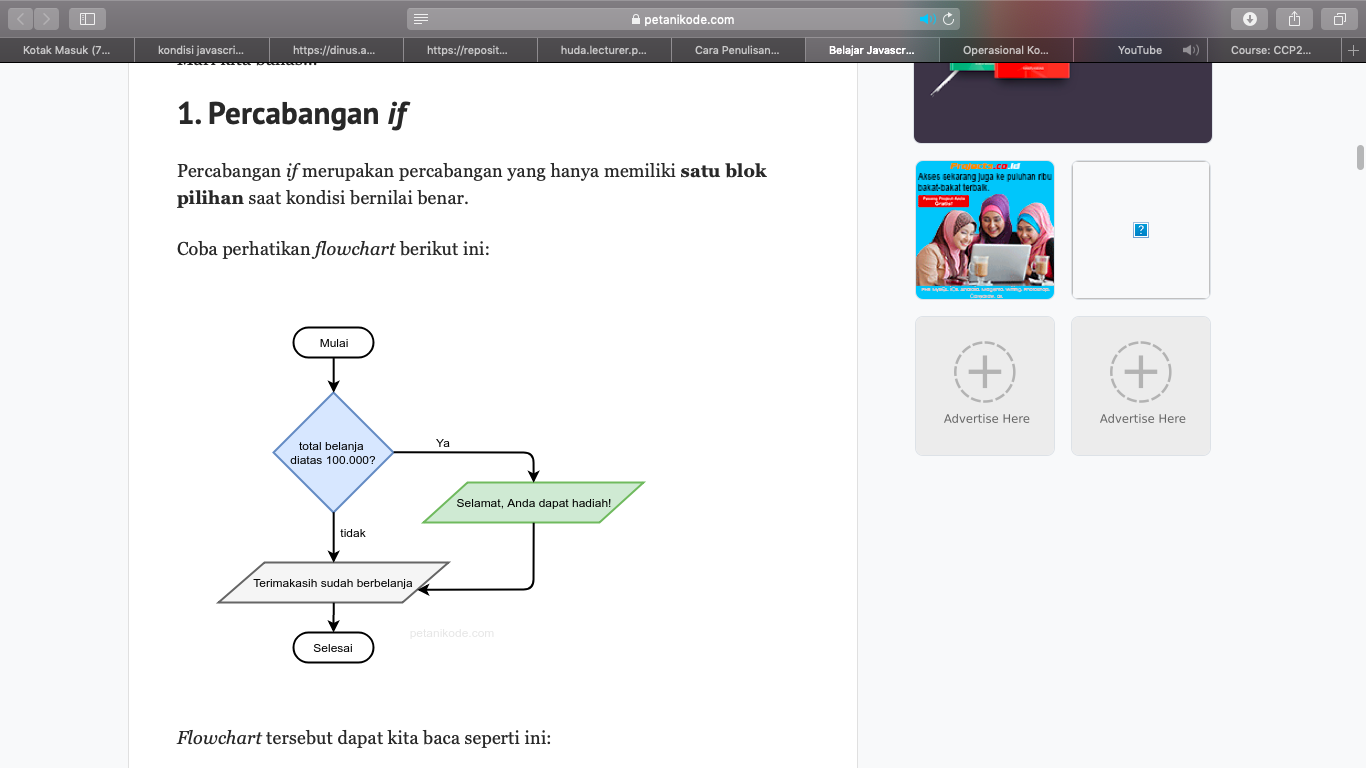
Kondisi percabangan yang hanya memiliki satu nilai yaitu true. Jika statemen dalam kondisi bernilai true maka statemen di dalamnya akan di eksekusi. Tetapi jika kondisi statemen bernilai false, maka statemen tersebut tidak akan dieksekusi.

Sintaks perintah if … :

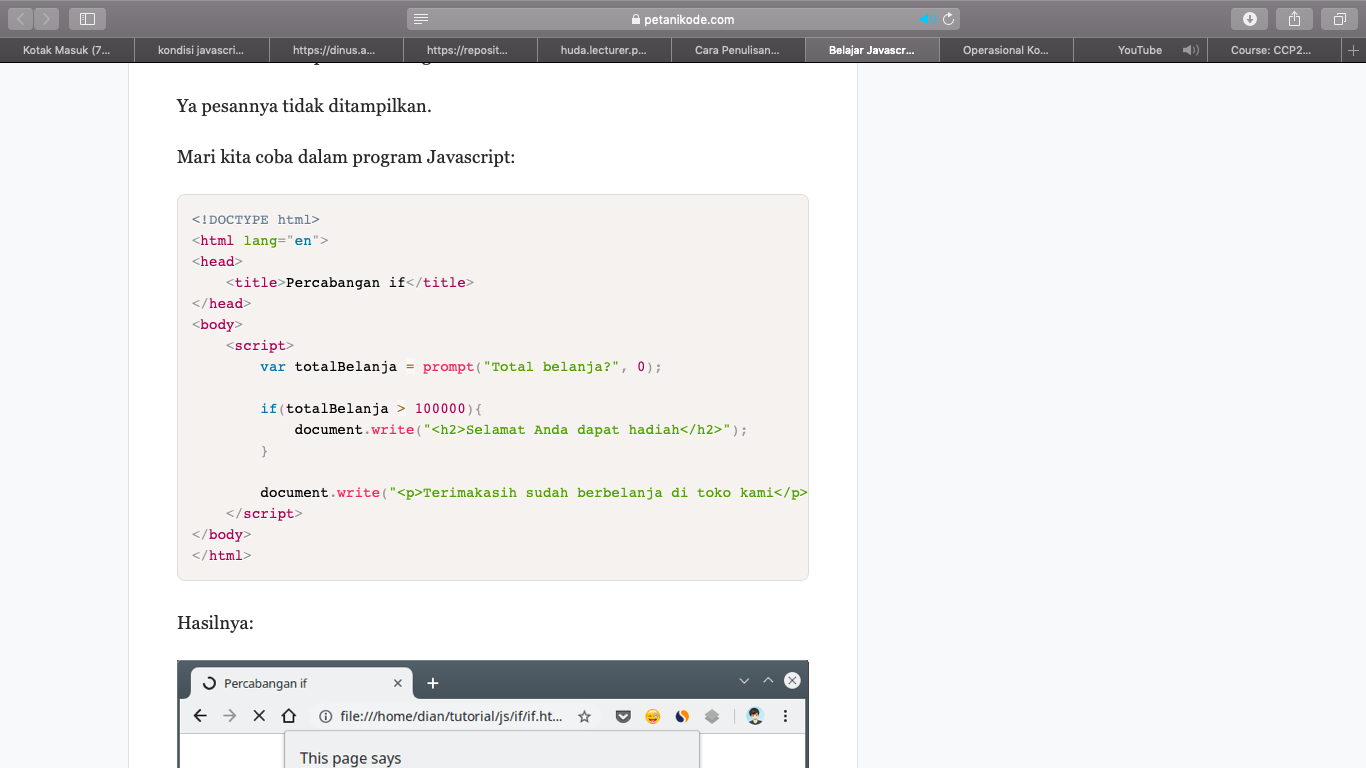
****

**Contoh :**

* Ilustrasi kondisi percabangan IF ….



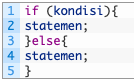
* Kode if…. Dalam javascript



* 1. **Perintah If …. else. …**

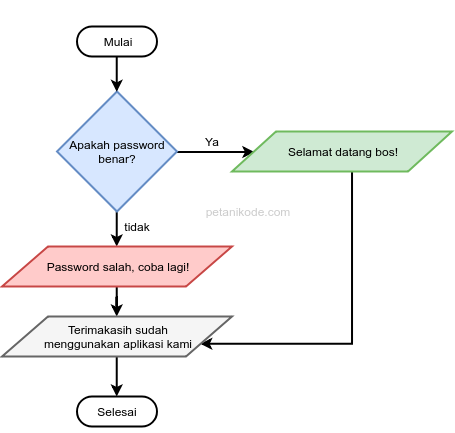
Perintah if … else … merupakan perintah percabangan yang memiliki dua nilai yaitu nilai true dan nilai false. Jika statemen yang terdapat pada perintah percabangan bernilai true, maka statemen akan dieksekusi. Dan jika kondisi percabangan bernilai false, maka statemen yang berada di else akan dieksekusi.

berikut sintaks dari perintah percabangan if …. Else :

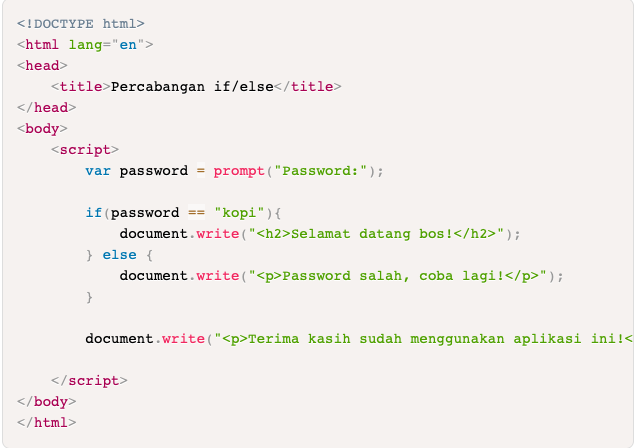


Contoh :

* Berikut ilustrasi percabangan if … else ….



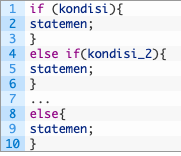
* Kondisi percabangan if … else dalam javascript



* 1. **Perintah If…elseif …. else**

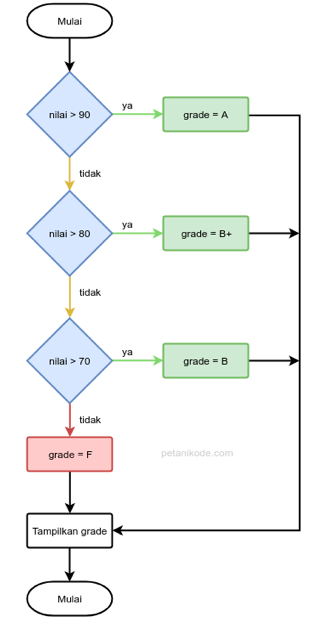
Perintah percabangan if … elseif….else merupakan kondisi percabangan yang memiliki lebih dari dua statemen. Jika kondisi pertama bernilai true, maka statemen di dalamnya akan di eksekusi. Jika kondisi pertama bernilai false, maka program akan melakukan pengecekan ke kondisi selanjutnya (kondisi ke 2, 3, 4, ……, n). dan jika semua kondisi bernilai false, maka statemen yang berada di else yang akan dieksekusi.

Sintaks percabangan if …elseif….else :



Contoh :

* Ilustrasi percabangan if … elseif … else



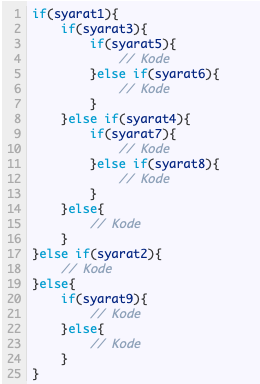
* Kondisi percabangan if … elseif… else dalam javascript



* 1. **Perintah Nested if…**

nested if digunakan apabila sudah menguji satu kondisi dan ingin menguji kembali setelah melewati kondisi sebelumnya atau juga terdapat if di dalam if

berikut sintak dari nested if :



Contoh :

<!DOCTYPE html>

<html>

<head>

<title>Cara Kode - Belajar JavaScript Dasar</title>

</head>

<body>

<p></p>

<script>

var hargaBarang = 21000;

var jmlUang = 50000;

if(hargaBarang <= jmlUang){

if(hargaBarang == jmlUang){

document.getElementsByTagName("p")[0].innerHTML = "Uang anda pass untuk membeli

barang ini";

}else if(hargaBarang < jmlUang){

document.getElementsByTagName("p")[0].innerHTML = "Uang masih sisa untuk membeli

barang ini";

}

}else{

document.getElementsByTagName("p")[0].innerHTML = "Uang anda tidak cukup membeli barang ini";

}

</script>

</body>

</html>

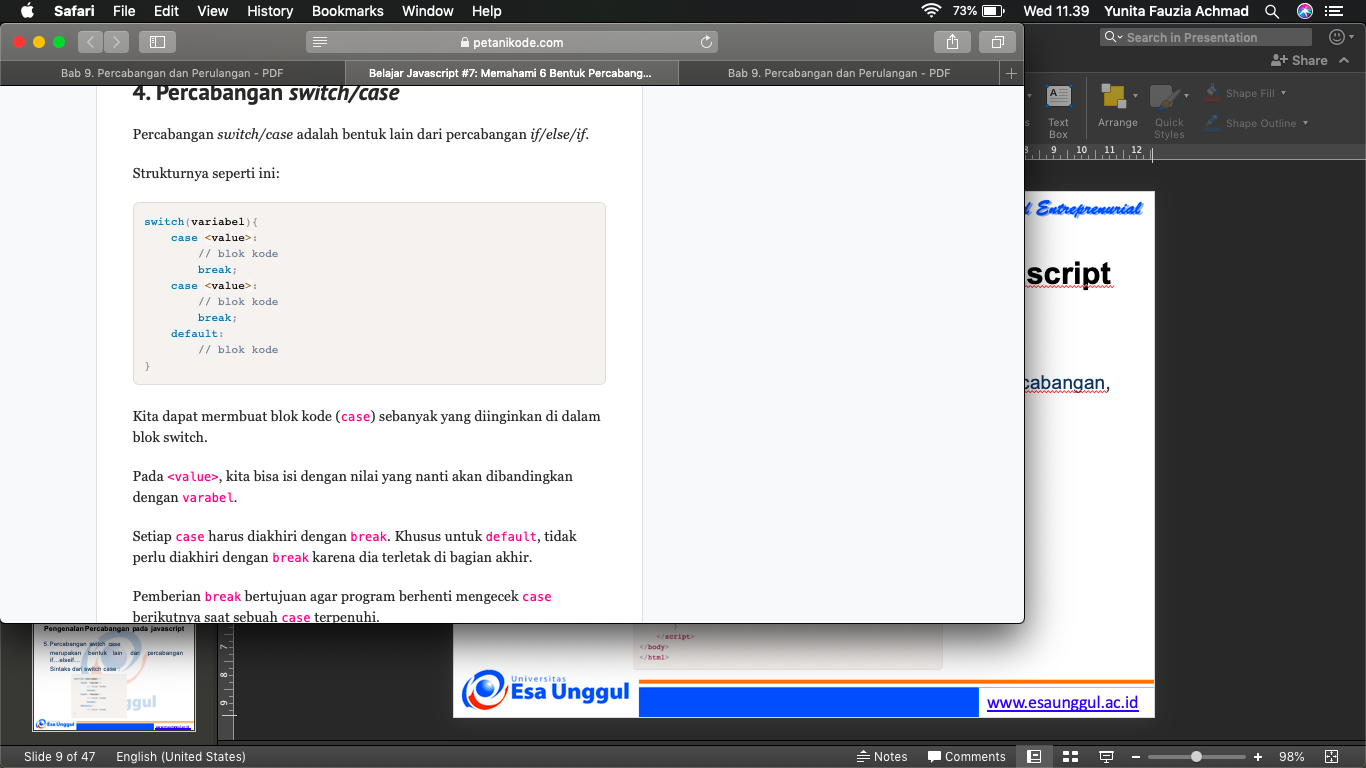
Hasil :



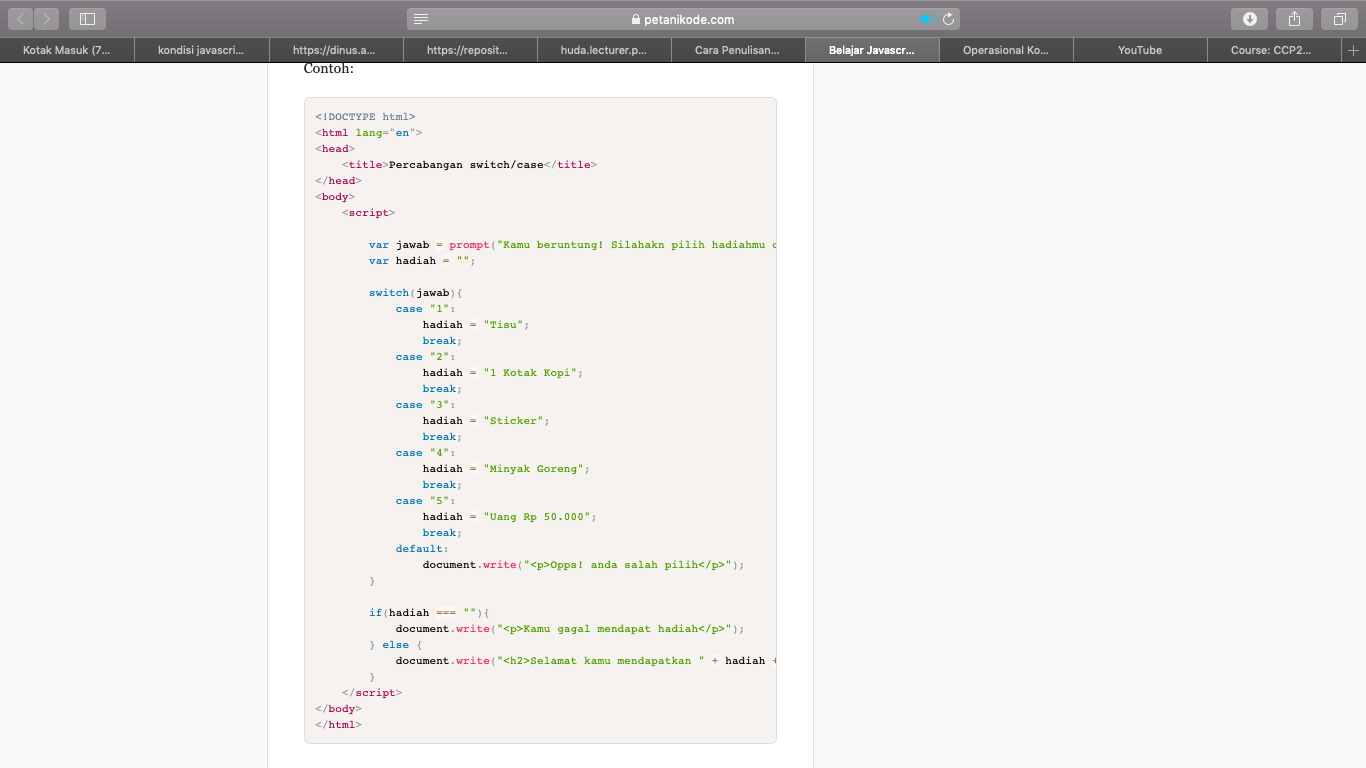
* 1. **Perintah Switch …. Case**

Perintah percabangan switch … case merupakan bentuk sederhana dari perintah percabangan if …elseif …. else. Tetapi pada perintah switch …. case hanya menggunakan dua tipe data yaitu integer dan char.

* Sintaks switch … case:



Contoh :



1. **Latihan**
   1. Sebutkan pengertian dari perulangan ?
   2. Sebutkan pengertian dari percabangan ?
   3. Sebutkan perintah – perintah yang terdapat pada perulangan ?
   4. Sebutkan perintah – perintah yang terdapat pada percabangan ?
   5. Sebutkan perbedaan antara switch case dan if …elseif…else?
2. **Kunci Jawaban**
3. Perulangan adalah proses mengulang-ulang eksekusi blok kode tanpa henti, selama kondisi yang dijadikan acuan terpenuhi.
4. Percabangan adalah cara yang digunakan dalam program untuk mengambil keputusan ke satu kemungkinan True atau False dari beberapa kondisi
5. Perintah yang terdapat pada perulangan, diantaranya adalah :
6. Perintah for
7. Perintah while
8. Perintah do … while
9. Perintah yang terdapat pada percabangan, diantaranya adalah :
10. Perintah if …
11. Perintah if … else
12. Perintah if … elseif … else
13. Perintah nested if
14. Perintah switch … case
15. Perbedaan antara switch case dan if … elseif … else adalah sebagai berikut:

**Fungsi dan Array pada Javascript**

1. **Kemampuan Akhir Yang Diharapkan**

Setelah mempelajari modul ini, diharapkan mahasiswa mampu :

* 1. Menjelaskan perintah array dan fungsi pada Javascript
  2. Menjelaskan penerapan perintah array dan fungsi pada javascript
  3. Menerapkan perintah array dan fungsi di javascript pada pemrograman web

1. **Uraian dan Contoh**
2. **Pengertian Array**

Array adalah tipe data yang berisi kumpulan dari nilai atau tipe data lain. Nilai di dalam array disebut dengan elemen, dan setiap elemen memiliki nomor urut yang dikenal dengan istilah index. Penomoran index di dalam array dimulai dari angka 0, sehingga index pertama berada di index 0, elemen kedua berada di index 2 dan seterusnya. Index maksimum yang dapat di tampung array dalam javascript adalah 4.294.967.294 dengan jumlah elemen maksimum adalah 4.294.967.295.

Array di dalam javascripy tidak bertipe (*untyped array).* elemen dari array bisa bertipe data string, number dan boolean dalam sebuah array yang sma, bahkan elemen dari array bisa berupa objek atau array yang lain. Array di dalam javascript juga bersifat dinamis, dan tidak perlu mendefinisikan berapa ukuran array pada saat membuat variabel. Jumlah elemen dapat ditambah dan dikurang setiap saat. Index array di dalam javascript juga tidak harus berurutan . javascript memperbolehkan elemen dari array tidak terurut. Misalkan index yang diisi hanya index 0, 5 dan 10 saja di dalam array.

Sintak dari array:



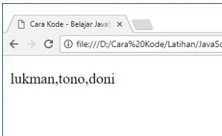
Contoh :



Keterangan :

Berdasarkan contoh di atas array yang terdapat data yang berisi tentang nama – nama dan kemudian dengan perintah selanjutnya dilakukan print nama arraynya sehingga hasil dari contoh di atas, adalah sebagai berikut :

Hasil :

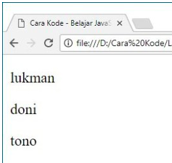


**Cara akses data array menggunakan nomor index**

Untuk mengakses atau menggunakan data tertentu pada array dapa menggunakan nomor index. Berikut contoh dari cara mengakses data array menggunakan nomor index :



Hasil :



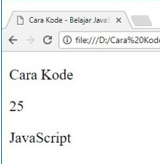
**Membuat array dengan tipe data yang berbeda**

Nilai array dapat isi dengan tipe data yang berbeda – beda misalkan string dengan integer, number dengan Boolean, dan lain – lain. Berikut ini contoh dari pembuatan array dengan tipe data yang berbeda.



Pada contoh diatas dapat dilihat terdapat sebuah array yang menampung data yang bertipe string dan number. Untuk menampung data number pada array tidak perlu menggunakan tanda kutip sedangkan data yang bertipe string menggunakan tanda kutip.

Hasil :



**Cara mengubah data atau nilai array**

Di dalam javascrip dapat mengganti atau mengubah nilai dari data array, untuk mengubahnya cukup menuliskan nama array beserta nomor index yang ingin diubah kemudian di tuliskan tanda sama dengan dan diikuti dengan value (nilai baru).

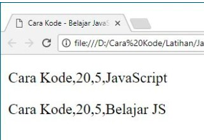
Sintaks :



Contoh :



Hasil :



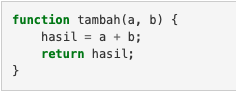
1. **Pengertian Function**

Fungsi / function adalah salah satu bagian yang paling indah dari Javascript. Sebagai bahasa fungsional Javascript mengimplementasikan fungsi kelas pertama (first class function). Fungsi dapat disimpan dalam variabel, dikembalikan oleh fungsi lain, dan dikirimkan sebagai argumen untuk fungsi lainnya. Implementasi fungsi yang sangat fleksibel seperti ini membuka banyak kesempatan kepada pengembang untuk menuliskan kode yang bukan hanya berjalan dengan baik, tetapi juga sangat elegan dan indah.

Sebuah fungsi membungkus satu atau banyak perintah. Setiap kali kita memanggil fungsi, maka perintah-perintah yang ada di dalam fungsi tersebut dijalankan. Secara umum fungsi digunakan untuk penggunaan kembali kode (code reuse) dan penyimpanan informasi (information hiding). Implementasi fungsi kelas pertama juga memungkinkan kita menggunakan fungsi sebagai unit-unit yang dapat dikombinasikan, seperti layaknya sebuah lego. Dukungan terhadap pemrograman berorientasi objek juga berarti fungsi dapat kita gunakan untuk memberikan perilaku tertentu dari sebuah objek.

**Cara menulis Function di javascript**

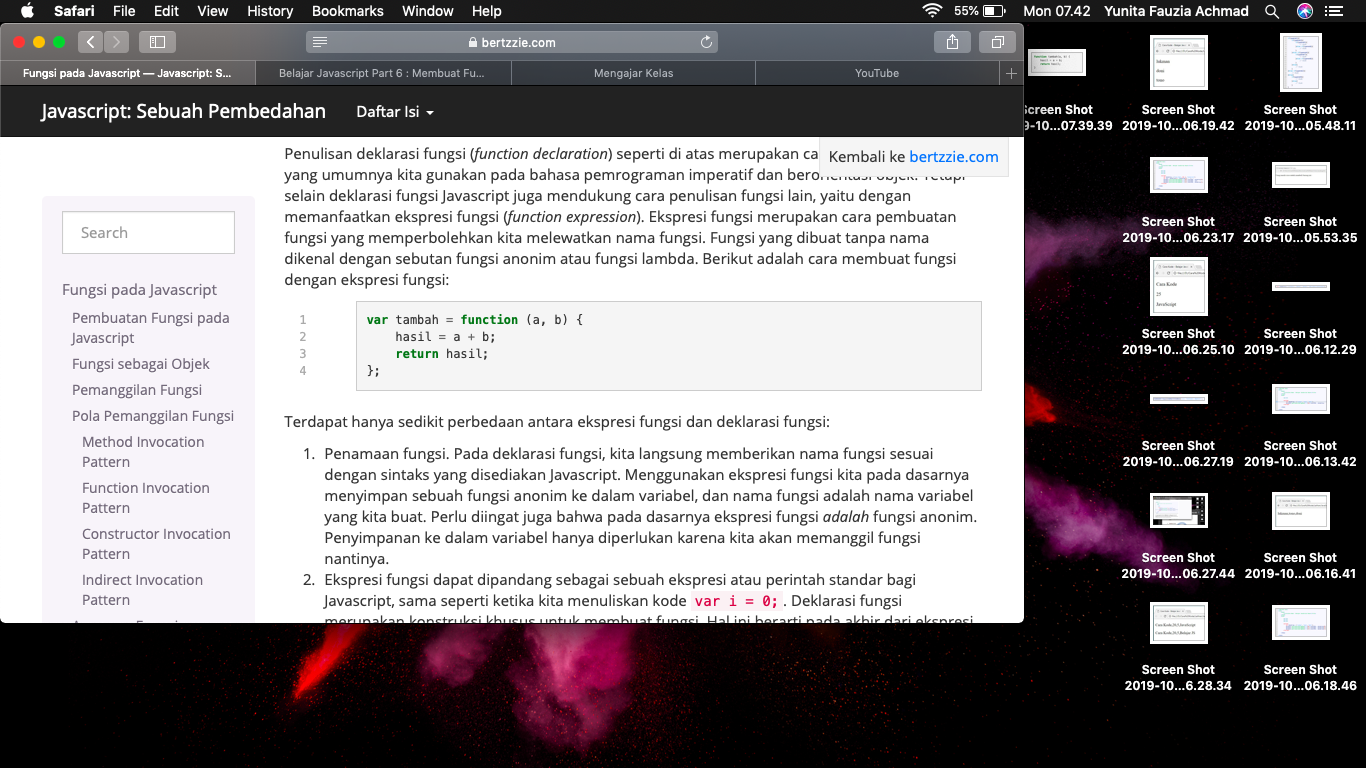
Berikut adalah sintaks dari function:

****

Terdapat 4 bagian di dalam function diantaranya adalah :

1. Kata kunci function, yang memberitahu Javascript bahwa kita akan membuat fungsi.
2. Nama fungsi, dalam contoh di atas adalah tambah. Dengan memberikan sebuah fungsi nama maka kita dapat merujuk ke fungsi tersebut dengan nama yang diberikan. Harus diingat bawa nama fungsi bersifat *opsional*, yang berarti **fungsi pada Javascript tidak harus diberi nama**. Kita akan membahas tentang hal ini lebih dalam nanti.
3. Daftar parameter fungsi, yaitu a, b pada contoh di atas. Daftar parameter ini selalu dikelilingi oleh tanda kurung (()). Parameter boleh kosong, tetapi tanda kurung wajib tetap dituliskan. Parameter fungsi akan secara otomatis didefinisikan menjadi variabel yang hanya bisa dipakai di dalam fungsi. Variabel pada parameter ini diisi dengan nilai yang dikirimkan kepada fungsi secara otomatis.
4. Sekumpulan perintah yang ada di dalam kurung kurawal ({}). Perintah-perintah ini dikenal dengan nama badan fungsi. Badan fungsi dieksekusi secara berurut ketika fungsi dijalankan.

Di dalam function terdapat juga bentuk dari ekspresi function, berikut adalah bentuk ekspresi function:



Perbedaan deklarasi function dengan ekspresi function, diantaranya adalah :

1. Penamaan fungsi. Pada deklarasi fungsi, kita langsung memberikan nama fungsi sesuai dengan sintaks yang disediakan Javascript. Menggunakan ekspresi fungsi kita pada dasarnya menyimpan sebuah fungsi anonim ke dalam variabel, dan nama fungsi adalah nama variabel yang kita buat. Perlu diingat juga bahwa pada dasarnya ekspresi fungsi *adalah* fungsi anonim. Penyimpanan ke dalam variabel hanya diperlukan karena kita akan memanggil fungsi nantinya.
2. Ekspresi fungsi dapat dipandang sebagai sebuah ekspresi atau perintah standar bagi Javascript, sama seperti ketika kita menuliskan kode var i = 0;. Deklarasi fungsi merupakan konstruksi khusus untuk membuat fungsi. Hal ini berarti pada akhir dari ekspresi fungsi kita harus menambahkan ; , sementara pada deklarasi fungsi hal tersbut tidak penting.

**Function sebagai objek**

Fungsi pada javascript adalah sebuah objek. Sebagai sebuah objek, semua fungsi dalam Javascript merupakan turunan dari Function.prototype. Function.prototype juga adalah merupakan turunan dari Object.prototype, sama seperti semua objek-objek lain dalam Javascript. Perbedaan utama fungsi dengan objek lain pada umumnya adalah fungsi dapat dipanggil, dan memiliki dua buah properti khusus, yaitu konteks pemanggilan fungsi dan kode pada isi badan fungsi. Kegunaan dari dua buah properti khusus ini akan kita lihat pada bagian selanjutnya.

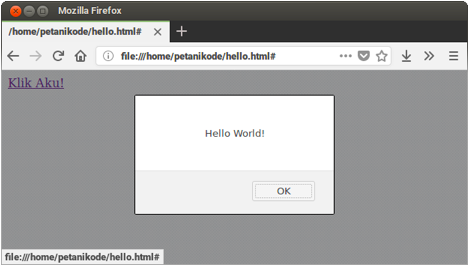
Sebagai sebuah objek, fungsi juga dapat kita perlakukan sama dengan objek lainnya. Pada bagian sebelumnya kita telah melihat bahwa fungsi dapat disimpan di dalam variabel. Fungsi juga dapat kita simpan di dalam array atau objek lain, dikirimkan sebagai argumen dari fungsi lain, atau dikembalikan dari sebuah fungsi. Sama seperti objek, kita juga dapat mengaitkan fungsi (method) kepada fungsi.

**Pemanggilan function di javascript**

**Contoh :**

****

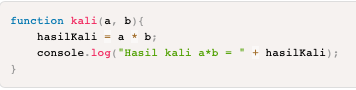
**Hasil :**

****

**Function dengan parameter**

Parameter adalah variabel yang menyimpan nilai untuk diproses di dalam fungsi.

**Contoh :**

****

Pada contoh di atas terdapat variabel parameter a dan b , dimana parameter a dan b akan di isi dengan nilai seperti berikut ini :



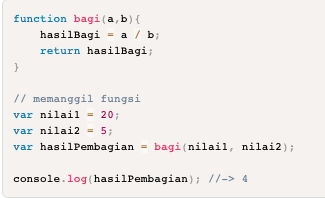
Sehingga, hasil dari perkalian antara parameter a dan parameter b adalah 6

**Function mengembalikan nilai**

Agar hasil pengolahan nilai di dalam fungsi dapat digunakan untuk proses berikutnya, maka fungsi harus mengembalikan nilai.

Pengembalian nilai pada fungsi menggunakan kata kunci return kemudian diikuti dengan nilai atau variabel yang akan dikembalikan.

Contoh :



Contoh program javascript dengan menggunakan function

Pada contoh ini terdapat 2 file yang akan dibuat yaitu file indeks.html dan file function.js. dan program ini berisi CRUD yaitu *create, read, update dan delete*.

Berikut file indeks.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>Belajar Fungsi di Javascript</title>

</head>

<body>

<fieldset>

<legend>Input Form</legend>

<input type="text" name="barang" placeholder="input nama barang..." />

<input type="button" onclick="addBarang()" value="Tambah" />

</fieldset>

<div>

<ul id="list-barang">

</ul>

</div>

<script src="fungsi.js"></script>

</body>

</html>

Kemudian file kedua yaitu function.js

var dataBarang = [

"Buku Tulis",

"Pensil",

"Spidol"

];

function showBarang(){

var listBarang = document.getElementById("list-barang");

// clear list barang

listBarang.innerHTML = "";

// cetak semua barang

for(let i = 0; i < dataBarang.length; i++){

var btnEdit = "<a href='#' onclick='editBarang("+i+")'>Edit</a>";

var btnHapus = "<a href='#' onclick='deleteBarang("+i+")'>Hapus</a>";

listBarang.innerHTML += "<li>" + dataBarang[i] + " ["+btnEdit + " | "+ btnHapus +"]</li>";

}

}

function addBarang(){

var input = document.querySelector("input[name=barang]");

dataBarang.push(input.value);

showBarang();

}

function editBarang(id){

var newBarang = prompt("Nama baru", dataBarang[id]);

dataBarang[id] = newBarang;

showBarang();

}

function deleteBarang(id){

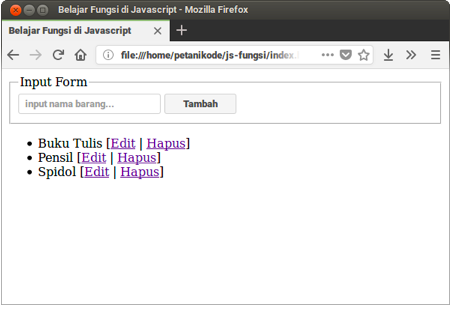
dataBarang.splice(id, 1);

showBarang();

}

showBarang();

**Hasil :**

****

1. **Latihan**
2. Sebutkan pengertian array ?
3. Sebutkan pengertian function ?
4. Sebutkan perbedaan deklarasi function dengan ekspresi function ?
5. **Kunci Jawaban**
6. Array adalah tipe data yang berisi kumpulan dari nilai atau tipe data lain.
7. Fungsi adalah sub-program yang bisa digunakan kembali baik di dalam program itu sendiri, maupun di program yang lain. Fungsi di dalam Javascript adalah sebuah objek. Karena memiliki properti dan juga method.
8. Berikut ini perbedaan antara deklarasi function dengan ekspresi function :
9. Penamaan fungsi. Pada deklarasi fungsi, kita langsung memberikan nama fungsi sesuai dengan sintaks yang disediakan Javascript. Menggunakan ekspresi fungsi kita pada dasarnya menyimpan sebuah fungsi anonim ke dalam variabel, dan nama fungsi adalah nama variabel yang kita buat. Perlu diingat juga bahwa pada dasarnya ekspresi fungsi *adalah* fungsi anonim. Penyimpanan ke dalam variabel hanya diperlukan karena kita akan memanggil fungsi nantinya.
10. Ekspresi fungsi dapat dipandang sebagai sebuah ekspresi atau perintah standar bagi Javascript, sama seperti ketika kita menuliskan kode var i = 0;. Deklarasi fungsi merupakan konstruksi khusus untuk membuat fungsi. Hal ini berarti pada akhir dari ekspresi fungsi kita harus menambahkan ; , sementara pada deklarasi fungsi hal tersbut tidak penting.
11. **Daftar Pustaka**
12. <https://www.petanikode.com/javascript-fungsi/>
13. <https://bertzzie.com/knowledge/javascript/Fungsi-pada-Javascript.html>