



Fundamental Javascript #3^o

+

Sesi 5



Kondisional Javascript

Boolean

Sebuah nilai benar atau salah, dan secara universal menggunakan bahasa inggris, yaitu **true** dan **false**.

Misalnya, kita membuat sebuah program pendataan diri dan kita membuat sebuah variable yang bernama *isMarried*, untuk menandakan apakah seseorang sudah menikah atau belum. Dengan **boolean**, kita dengan mudah bisa membuat isi dari variable *isMarried* tadi dengan nilai *true* atau *false*.

Contoh, jika sudah menikah, kita tinggal menggunakan *isMarried = true*. Ganti nilai dengan *false* jika sebaliknya.

```
<script>
var a = true;
var benar = true;
var salah = false;
</script>
```



Comparator

Simbol-simbol perbandingan

Kita sering sekali bertemu dengan simbol ini di matematika, misalnya simbol $>$ (lebih besar), $<$ (lebih kecil), atau $=$ (sama dengan). Tapi karena simbol $=$ sudah digunakan di assignment nilai di variable, maka kita menggunakan simbol $==$ (sama dengan dua) atau $===$ (sama dengan tiga) untuk membandingkan.

$5 > 3$, akan mengeluarkan true

$5 < 2$, akan mengeluarkan false

$5 == 5$, akan mengeluarkan true



Equal Operator

Untuk membandingkan dua nilai apakah sama atau tidak, kita bisa menggunakan equal operator. Nah sebelumnya sempat disebutkan bahwa ada simbol `==` dan `===`.

Apa bedanya dan lebih baik yang mana?

```
console.log("5" == 5); // akan menampilkan true
```

```
console.log("5" === 5); // akan menampilkan false
```



Kondisional Javascript

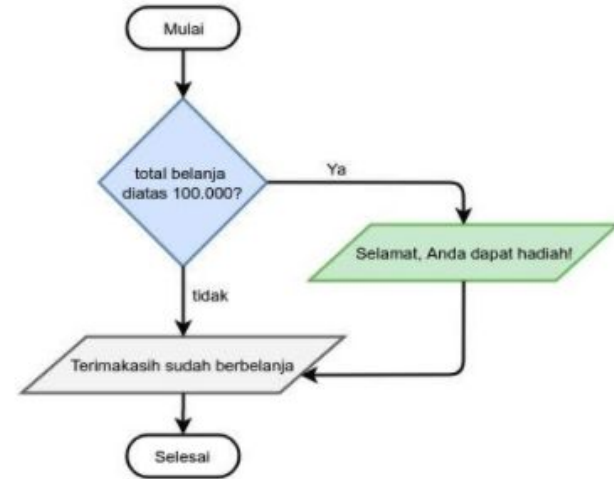
Memahami 5 Bentuk Percabangan pada Javascript

Percabangan if

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

Di sebuah program, atau bahkan di kehidupan sehari-hari, kita akan menemukan berbagai macam kondisi, dan respon kita terhadap kondisi tersebut akan berbeda. Misal, jika di kepala kita ada sebuah pernyataan atau statement "Jika saya lapar, maka saya akan makan".

Nah, ini adalah sebuah conditional statement yang berjalan di kepala kita. Komputer bisa membuat proses serupa.



```
if(sayaLapar) {
    //SAYA AKAN MAKAN
}
```



Kondisional Javascript

Memahami 5 Bentuk Percabangan pada Javascript

Percabangan if

Biasanya, conditional paling sederhana di programming menggunakan statement if. Kita akan coba gambarkan if dalam kasus tadi.

```
if(true) {  
    // 1 - JALANKAN SAYA  
}  
if(false) {  
    // 2 - JANGAN JALANKAN SAYA  
}  
// 1 - KEMUDIAN JALANKAN SAYA
```



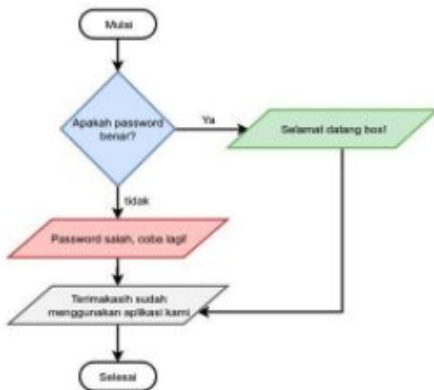
Kondisional Javascript

Memahami 5 Bentuk Percabangan pada Javascript

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).



Kondisional Javascript

Memahami 5 Bentuk Percabangan pada Javascript

Percabangan if/else

Conditional akan menjadi sebuah proses yang sering kita gunakan di programming di bahasa manapun. Belum cukup sampai disini, saat belajar tentang if, kita akan selalu dipasangkan dengan else. Apa itu else? Misalnya, kita punya sebuah kondisi berikut: "Jika tinggi saya diatas 160 cm, maka saya boleh naik roller coaster. Jika tidak, maka saya lebih baik naik komedi putar." Disini kita akan ada dua proses berbeda yang akan dijalankan saat salah satu kondisi terpenuhi. Kondisi tersebut adalah apakah tinggi diatas 160. Kita dapat menuliskannya dengan kode dibawah ini:

```
var myHeight; //isi dengan angka berapapun

if( myHeight > 160 ) {
  console.log('hore, saya bisa roller coaster!!!');
}else {
  console.log('yasudah, saya naik komedi putar saja');
}

console.log('Sudah sore. Pulang yuk!');
```

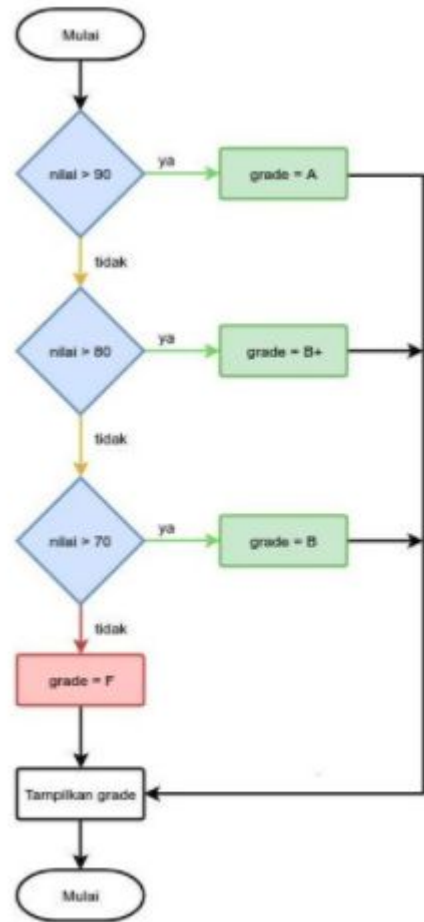


Kondisional Javascript

Memahami 5 Bentuk Percabangan pada Javascript

Percabangan if/elseif/else

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



Kondisional Javascript

Memahami 5 Bentuk Percabangan pada Javascript

Logical Operator

Seringkali, saat kita membuat sebuah kondisi, kita tidak cukup membuat satu kemungkinan. Misal kita punya statement kondisi "Jika saya punya KTP atau punya SIM, maka saya bisa membeli tiket".

Cara paling sederhana namun merepotkan adalah dengan membuat kode seperti pada gambar pertama di sebelah kanan.

tentunya kode ini terlalu panjang untuk kasus yang sangat sederhana. Disini kita bisa menggunakan logical operator || (or / atau) untuk mengecek salah satu kondisi, dan jika salah satu terpenuhi, maka kita bisa membeli tiket.

```
var ownKTP;  
var ownSIM;  
  
if(ownKTP) {  
    console.log('saya bisa beli tiket');  
}else if(ownSIM) {  
    console.log('saya bisa beli tiket');  
}
```

```
var ownKTP;  
var ownSIM;  
  
if(ownKTP || ownSIM) {  
    console.log('saya bisa beli tiket');  
}
```



Kondisional Javascript

Memahami 5 Bentuk Percabangan pada Javascript

Logical Operator

Logical operator tidak hanya OR (atau). Ada juga AND (dan). AND (menggunakan simbol **&&**), sesuai namanya, semua kondisi harus terpenuhi. Misalnya pada kasus **"Jika umur saya 17 tahun ke atas dan tinggi saya diatas 150, maka saya boleh memiliki SIM."**

```
if(myAge === 17 && myHeight > 150) {  
    console.log('Saya bisa memiliki SIM!!');  
}
```



Kondisional Javascript

Memahami 5 Bentuk Percabangan pada Javascript

Percabangan dengan Operator Ternary

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

```
myAge >= 17 && myHeight > 150 ? console.log('saya bisa punya SIM!!!') : console.log('saya tidak bisa punya SIM!!!')
```



HACKTIV8

Kondisional Javascript

Memahami 5 Bentuk Percabangan pada Javascript

Percabangan Bersarang (Nested)

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

```
if(myAge >= 17 && myHeight > 150) {  
  if(mobil) {  
    console.log('Anda memerlukan SIM A');  
  }else if(truck) {  
    console.log('Anda memerlukan SIM B');  
  }else if(motor) {  
    console.log('Anda memerlukan SIM C');  
  }else {  
    console.log('Kendaraan tidak ditemukan!!');  
  }  
}else {  
  console.log('Anda belum memenuhi persyaratan untuk membuat SIM!!');  
}
```



Challenge

1. Buatlah sebuah script untuk meminta user untuk meminta password, bisa menggunakan prompt.

Buat juga sebuah variabel yang menampung password yang seharusnya.

Kemudian, buat sebuah logika jika password yang dimasukkan user benar, berikan informasi benar (bisa menggunakan alert). Jika salah, direct ke halaman lain, misalnya Google, menggunakan code di bawah ini:

2. Gunakan Prompt untuk meminta user memasukkan angka antara 1 sampai 100.

ketika score di antara 80 and 100, alert 'your grade is A' ketika score di antara 80 and 60, alert 'your grade is B' ketika score di antara 60 and 40, alert 'your grade is C' ketika score di bawah 40, alert 'your grade is D'. Ketika value is di luar dari 1 sampai 100, alert 'cannot calculate your grade'





Perulangan Di⁺ Javascript

Perulangan Di Javascript

Perulangan yang termasuk dalam Counted Loop:

1. Perulangan For
2. Perulangan Foreach
3. Perulangan Repeat

Perulangan yang termasuk dalam Uncounted Loop:

1. Perulangan While
2. Perulangan Do/While

Perulangan

For

(counted loop)



Push up 10x

```
for(i=0; i<10; i++){  
  pushUp();  
}
```

VS

While

(uncounted loop)



Push up
Sampai bosan

```
while(not bosan){  
  pushUp();  
}
```

Perulangan Di Javascript

While Loop

Perulangan while merupakan perulangan yang termasuk dalam perulangan uncounted loop.

Perulangan while juga dapat menjadi perulangan yang counted loop dengan memberikan counter di dalamnya.

```
var index = 0;

while(index < 10) {
  console.log("Perulangan ke:", index);
  index++
}
```

Perulangan Di Javascript

Do/while Loop

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.

```
var index = 0;

do {
  console.log('Perulangan ke:', index);
}while(index < 10);
```

Perulangan Di Javascript

Foreach loop

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.

```
var students = ["John", "Mark", "Elisa"];

students.forEach((student), function(){
  console.log(student);
});
```



Perulangan Di Javascript

Repeat loop

Perulangan dengan 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.

```
var name = "John Doe";  
console.log(name.repeat(10));
```



Perulangan Di Javascript

For loop

Bentuk lain dari iterasi, dimana statement for menjadi kontrol atas loop yang dilakukan.

Statement for akan menampung tiga parameter, yaitu sebut saja inisialisasi, kondisi, dan incremental/decremental.

```
for(var i = 0; i < 10; i++) {  
    console.log("Perulangan ke:", i);  
}
```

Perulangan Di Javascript

Nested loop

Di dalam blok perulangan, kita juga dapat membuat perulangan.

Ini disebut dengan nested loop atau perulangan bersarang atau perulangan di dalam perulangan.

```
for(var i = 0; i < 10; i++) {  
    console.log("Perulangan dari Loop Pertama:", i);  
  
    for(var j = 0; j <= i; j++) {  
        console.log("Perulangan dari Loop Kedua:", j);  
    }  
}
```



Perulangan Di Javascript

Infinite loop

⚠ Waspada infinite Loop!

Dengan sengaja atau sengaja, kode kamu mungkin dapat menghasilkan forever looping, atau looping yang tidak akan pernah berhenti. Bila ini terjadi, segera periksa statement kondisi kamu.

```
var index = 0;

//Perulangan akan terus berjalan karena index tidak ditambahkan
while(index < 10) {

    console.log('Perulangan ke:', i);
}
```



Perulangan Di Javascript

Infinite loop

⚠ Waspada infinite Loop!

Dengan sengaja atau sengaja, kode kamu mungkin dapat menghasilkan forever looping, atau looping yang tidak akan pernah berhenti. Bila ini terjadi, segera periksa statement kondisi kamu.

```
var index = 0;

//Perulangan akan terus berjalan karena index tidak ditambahkan
while(index < 10) {

    console.log('Perulangan ke:', i);
}
```

Perulangan Di Javascript

Challenge

challenge

1. Buatlah sebuah perulangan dengan while untuk menampilkan list orang dari 120. Di setiap perulangan, tampilkan di console saja format berikut: 'Halo, saya orang ke URUTAN ORANG'

Ubah tugas 1 sebelumnya menjadi menggunakan for !

2. Sekarang, kembali ke kode tugas 1, kita modifikasi sedikit kode kita. Gunakan percabangan/kondisional untuk hanya menampilkan teks dari orang urutan ganjil saja.

Bebas menggunakan while atau for, buatlah sebuah perulangan yang akan menampilkan huruf O seperti di bawah ini: OOOO

OOOO OOOO OOOO

Tapi dengan syarat, kamu harus menggunakan looping untuk menampilkan bentuk di bawah ini! Be creative! Pikirkan bagaimana kalau kita harus menampilkan 100 "O" ke samping dengan 100 baris?

