

Javascript

Algorithm, Decision Making



Javascript Algorithm, Decision Making

- ☐ Algorithm
- ☐ Decision making flow (If-else, switch case)
- ☐ Recursion Flow
- ☐ Looping


Objektif sesi

- Mampu menjelaskan algoritma.
- Memahami representasi algoritma yang tepat dan sesuai fungsinya.
- Mampu membuat sebuah dokumen alur suatu algoritma dengan flowchart.
- Memahami cara membuat representasi suatu algoritma dengan pseudocode.
- Memahami dan mampu menerapkan alur pengambilan keputusan.

Algoritma

Algoritma adalah urutan langkah-langkah yang sistematis dan terstruktur untuk menyelesaikan masalah atau menjalankan suatu tugas. Dalam konteks JavaScript, algoritma seringkali digunakan untuk menggambarkan proses atau logika yang diterapkan dalam pengembangan perangkat lunak.

javascript

 Copy code

```
// Algoritma: Menentukan bilangan genap atau ganjil

// Langkah 1: Masukkan bilangan
var bilangan = 10;

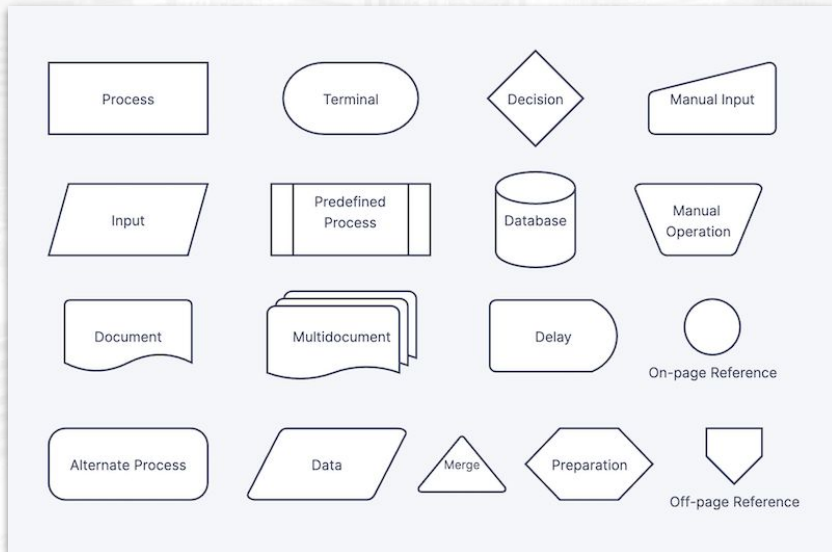
// Langkah 2: Periksa apakah bilangan itu genap atau ganjil
if (bilangan % 2 === 0) {
    // Langkah 3a: Jika bilangan genap, cetak "Bilangan genap"
    console.log("Bilangan genap");
} else {
    // Langkah 3b: Jika bilangan ganjil, cetak "Bilangan ganjil"
    console.log("Bilangan ganjil");
}
```


Algoritma

Ada 2 opsi dalam menyusun alur algoritma:

Flowchart : representasi visual dari control flow suatu algoritma.

Pseudocode : algoritma dengan menggunakan teks atau kata-kata untuk menunjukkan suatu alur.



plaintext

1. Start
2. Input bilangan
3. if bilangan modulo 2 equals 0
4. Print "Bilangan genap"
5. else
6. Print "Bilangan ganjil"
7. End

Decision Making

Alur pengambilan keputusan dapat menggunakan beberapa cara, yaitu:

- Kondisional if-else
- Kondisional operator ('?')
- Switch case

javascript

```
// Contoh dengan if-else
var nilai = 75;

if (nilai >= 70) {
  console.log("Lulus");
} else {
  console.log("Tidak Lulus");
}
```


javascript

```
// Contoh dengan kondisional operator (?)
var nilai = 75;
var hasil = (nilai >= 70) ? "Lulus" : "Tidak Lulus";
console.log(hasil);
```

Decision Making

Alur pengambilan keputusan
dengan Switch case

javascript

 Copy code

```
// Contoh dengan switch case
var hari = "Senin";
var aktivitas;

switch (hari) {
  case "Senin":
    aktivitas = "Meeting";
    break;
  case "Selasa":
    aktivitas = "Pelatihan";
    break;
  case "Rabu":
    aktivitas = "Kantor";
    break;
  default:
    aktivitas = "Libur";
}


console.log("Hari " + hari + ", aktivitas: " + aktivitas);
```



Recursion Flow

Rekursi adalah sebuah konsep di mana suatu fungsi memanggil dirinya sendiri selama eksekusi. Ini adalah teknik yang umum digunakan untuk memecahkan masalah dengan cara yang lebih elegan.

javascript

 Copy code

```
function hitungFaktorial(n) {
  // Basis atau kondisi terminasi
  if (n === 0 || n === 1) {
    return 1;
  } else {
    // Rekursif: Faktorial dari n = n * faktorial dari (n-1)
    return n * hitungFaktorial(n - 1);
  }
}

// Contoh penggunaan
var hasil = hitungFaktorial(5);
console.log("5 faktorial adalah:", hasil);
```


Looping

Looping (perulangan) adalah konsep dalam pemrograman yang memungkinkan eksekusi blok kode yang sama berulang kali selama kondisi tertentu terpenuhi. Looping sangat berguna untuk menangani situasi di mana kita perlu melakukan tugas yang sama secara berulang tanpa harus menulis kode yang sama berulang kali.

Terdapat tiga jenis alur perulangan yang perlu kita ketahui, yaitu:

- “for” loop
- “while” loop
- “do...while” loop

For Loop

for loop digunakan ketika kita tahu berapa kali kita ingin mengulang suatu blok kode.

Contoh: Menampilkan angka 1 hingga 5 dengan for loop:

javascript

```
for (var i = 1; i <= 5; i++) {  
    console.log(i);  
}
```

"while" Loop

while loop digunakan ketika kita tidak tahu berapa kali kita ingin mengulang suatu blok kode, tetapi kita tahu kondisinya.

Contoh: Menampilkan angka 1 hingga 5 dengan while loop:

javascript

```
var i = 1;

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

"do...while" Loop

do...while loop mirip dengan while loop, tetapi blok kode di dalamnya akan dieksekusi setidaknya satu kali, bahkan jika kondisi tidak terpenuhi.

Contoh: Menampilkan angka 1 hingga 5 dengan do...while loop:

javascript

```
var i = 1;

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



Thank you