

Operations & If Else Statements

28 FEBRUARI 2024

OPERATOR ARITMATIKA:

Operator aritmatika digunakan untuk melakukan operasi matematika pada nilai numerik.

Contoh:

```
let x = 10;
let y = 5;

let tambah = x + y; // Penjumlahan
let kurang = x - y; // Pengurangan
let kali = x * y;    // Perkalian
let bagi = x / y;    // Pembagian
let modulo = x % y;  // Modulo (Sisa pembagian)
```

Operator String:

Operator string digunakan untuk melakukan operasi pada nilai string, seperti penggabungan (concatenation) atau pemotongan (substring).

Contoh:

```
let namaDepan = "Option";
let namaBelakang = "Programming";

let namaLengkap = namaDepan + " " + namaBelakang; // Penggabungan string
```

INCREMENTING DAN DECREMENTING:

Incrementing (penambahan) dan decrementing (pengurangan) digunakan untuk menambah atau mengurangi nilai variabel.

Contoh:

```
let x = 5;

x++; // Incrementing (x akan menjadi 6)
x--; // Decrementing (x akan kembali menjadi 5)
```

IF, ELSE IF, SWITCH CASE:

Digunakan untuk pengambilan keputusan berdasarkan kondisi tertentu.

If, Else If, Else:

```
let nilai = 80;

if (nilai >= 90) {
    console.log("Nilai Anda A");
} else if (nilai >= 80) {
    console.log("Nilai Anda B");
} else {
    console.log("Nilai Anda C");
}
```

Switch Case

```
let nilai = 75;
let grade;

switch(true) {
  case (nilai >= 90):
    grade = "A";
    break;
  case (nilai >= 80):
    grade = "B";
    break;
  default:
    grade = "C";
}

console.log("Nilai Anda: " + grade);
```

STUDI KASUS:

PROGRAM KALKULATOR SEDERHANA:

```
let angka1 = 10;
let angka2 = 5;
let operator = "+"; // Operator bisa diganti dengan +, -, /, *

let hasil;

switch(operator) {
  case "+":
    hasil = angka1 + angka2;
    break;
  case "-":
    hasil = angka1 - angka2;
    break;
  case "/":
    hasil = angka1 / angka2;
    break;
  case "*":
    hasil = angka1 * angka2;
    break;
  default:
    console.log("Operator tidak valid");
}

console.log("Hasil Dari " + angka1 + " " + operator + " " + angka2 + " adalah:");
console.log(hasil);
```

PSEUDOCODE:

Pseudocode adalah representasi berbasis teks dari algoritma pemrograman. Ini adalah cara untuk mengekspresikan langkah-langkah dalam sebuah algoritma tanpa harus mengikuti sintaksis dari bahasa pemrograman tertentu. Pseudocode biasanya digunakan untuk merencanakan dan menggambarkan algoritma secara informal sebelum diimplementasikan dalam bahasa pemrograman yang sebenarnya. Ini memungkinkan programmer untuk fokus pada logika dan struktur algoritma tanpa harus memikirkan sintaksis bahasa pemrograman tertentu.

Contoh Pseudocode untuk Program Kalkulator Sederhana:

```
ALGORITMA KalkulatorSederhana

MASUKAN angka1
MASUKAN angka2
MASUKAN operator

KASUS operator DARI
    KETIKA "+" THEN
        hasil <- angka1 + angka2
    KETIKA "-" THEN
        hasil <- angka1 - angka2
    KETIKA "/" THEN
        hasil <- angka1 / angka2
    KETIKA "*" THEN
        hasil <- angka1 * angka2
    SELAIN_ITU
        CETAK "Operator tidak valid"
END KASUS

CETAK "Hasil: " + hasil
```

FAQ :

1. Struktur Dasar:

- `if-else`: Menggunakan struktur kondisional dengan `if`, `else if` (opsional), dan `else`.
- `switch-case`: Menggunakan struktur pemilihan kasus dengan `switch` dan `case`, serta `default` (opsional).

2. Penggunaan:

- `if-else`: Cocok digunakan ketika terdapat beberapa kondisi yang saling bergantung atau berbeda-beda.
- `switch-case`: Lebih efisien ketika terdapat banyak opsi untuk dipilih dan kasus-kasus tersebut bersifat eksplisit.

3. Pengecekan Kondisi:

- ``if-else``: Dapat mengevaluasi ekspresi kebenaran (true/false) dari satu atau beberapa kondisi.
- ``switch-case``: Membandingkan nilai ekspresi dengan beberapa nilai kasus yang telah ditentukan.

4. Kondisi Default:

- ``if-else``: Tidak ada kondisi default yang tersedia secara langsung; harus menggunakan blok ``else`` terakhir untuk menangani kondisi yang tidak memenuhi syarat sebelumnya.
- ``switch-case``: Memiliki opsi default yang dapat dieksekusi jika tidak ada nilai kasus yang cocok.

5. Flexibilitas:

- ``if-else``: Lebih fleksibel karena dapat menangani kondisi yang kompleks dengan lebih mudah, seperti menggunakan operasi logika dan perbandingan yang rumit.
- ``switch-case``: Lebih terbatas karena hanya cocok untuk kasus-kasus sederhana di mana kita hanya membandingkan nilai dengan beberapa opsi yang sudah diketahui.

6. Penanganan Ekspresi:

- ``if-else``: Dapat menangani ekspresi yang kompleks dan tidak terbatas dalam hal jenis kondisi yang bisa dievaluasi.
- ``switch-case``: Hanya cocok untuk ekspresi yang dapat dievaluasi menjadi nilai tunggal (misalnya, angka atau string).

Pada dasarnya, baik ``if-else`` maupun ``switch-case`` adalah alat kontrol alur yang digunakan untuk membuat keputusan berdasarkan kondisi tertentu. Pemilihan antara keduanya tergantung pada kompleksitas kondisi dan kebutuhan spesifik dari kode yang sedang Anda tulis. Jika Anda memiliki beberapa kondisi yang kompleks atau tidak terbatas, maka ``if-else`` bisa menjadi pilihan yang lebih baik. Namun, jika Anda memiliki serangkaian opsi yang cukup jelas dan spesifik, ``switch-case`` mungkin lebih bersih dan mudah dipahami.

KESIMPULAN

Kesimpulannya, `if-else` digunakan untuk kondisi yang kompleks atau bervariasi, sementara `switch-case` lebih efisien ketika terdapat banyak opsi yang bersifat eksplisit. `if-else` lebih fleksibel dalam menangani kondisi yang kompleks, sementara `switch-case` cocok untuk kasus sederhana dan spesifik. Pemilihan antara keduanya tergantung pada kompleksitas kondisi dan kebutuhan spesifik dari kode yang sedang dibangun.