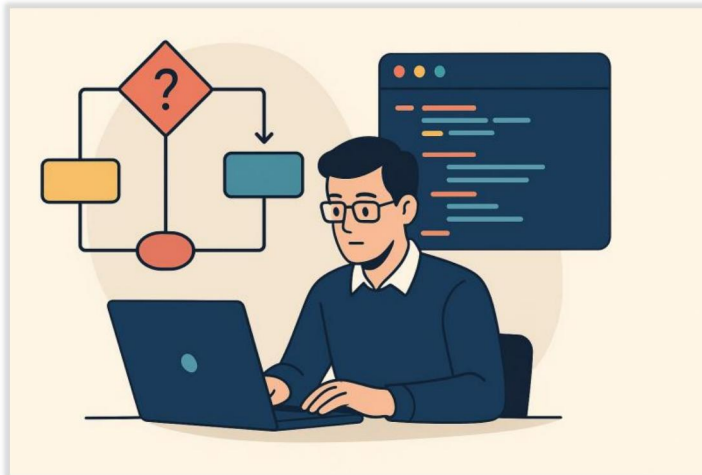




Percabangan



A. Pengantar Percabangan dalam Pemrograman

1) Apa itu Percabangan?

Dalam pemrograman, percabangan (*branching / conditional*) adalah cara membuat program memilih jalur eksekusi berbeda berdasarkan suatu kondisi/logika. Dengan kata lain, program dapat “bertanya” sesuatu (apakah syarat A terpenuhi?) dan jika ya, lakukan aksi X; jika tidak, lakukan aksi Y.

Ini sangat penting dalam pengembangan solusi karena banyak persoalan kehidupan nyata yang membutuhkan keputusan berdasarkan kondisi.

2) Hubungan dengan Algoritma & Pemrograman

Dalam elemen “Algoritma dan Pemrograman” dari Kurikulum Merdeka, dikatakan bahwa siswa harus mampu mengembangkan solusi dari berbagai persoalan dengan membaca bermakna dan menulis teks algoritmik terstruktur (logis, sistematis, bertahap, konvergen, dan linier).

Percabangan adalah bagian dari algoritma terstruktur: kita menulis langkah-langkah (instruksi) yang sistematis, satu setelah satu, tetapi dengan titik di mana alur bisa “bercabang” tergantung kondisi.

Contoh:

- Jika suhu $> 30^{\circ}\text{C}$ maka hidupkan AC, lainnya matikan AC.
- Jika nilai ≥ 70 maka siswa “lulus”, lainnya siswa “tidak lulus”.



3) Terminologi & Struktur Dasar

Beberapa istilah penting:

- a) Kondisi (condition): ekspresi yang bernilai benar (true) atau salah (false).
- b) Blok instruksi: kumpulan instruksi yang dijalankan jika kondisi terpenuhi (atau tidak).
- c) Struktur dasar:
 - if ... kemudian ...
 - if ... else ...
 - if ... else if ... else ...
 - (Beberapa bahasa: switch/case)
- d) Percabangan bersarang (nested): satu if/else di dalam if/else lain.

B. Struktur Percabangan dan Variannya

1) Percabangan “if” sederhana

Contoh algoritma langkah-tertulis:

```
Algoritma CekPositif
Input angka
Jika angka > 0 maka
    Tampilkan “Angka positif”
Selesai
```

Dalam kode (pseudo-kode) misalnya:

```
if (angka > 0) {
    print("Angka positif");
}
```

Di sini hanya satu jalur alternatif: jika kondisi terpenuhi, maka aksi; jika tidak, maka program melanjutkan ke langkah selanjutnya tanpa aksi khusus.



2) Percabangan “if ... else”

Menambahkan jalur alternatif bila kondisi tidak terpenuhi.

Algoritma:

```
Input angka
Jika angka  $\geq 0$  maka
    Tampilkan “Angka non-negatif”
lainnya
    Tampilkan “Angka negatif”
Selesai
```

Pseudo-kode:

```
if (angka  $\geq 0$ ) {
    print("Angka non-negatif");
} else {
    print("Angka negatif");
}
```

Struktur ini membuat alur menjadi konvergen — setelah if/else, alur kembali tunggal.

3) Percabangan “if ... else if ... else”

Untuk memeriksa beberapa kondisi secara berjenjang.

Contoh algoritma:

```
Input nilai
Jika nilai  $\geq 85$  maka
    Tampilkan “Predikat A”
lainnya jika nilai  $\geq 70$  maka
    Tampilkan “Predikat B”
lainnya jika nilai  $\geq 55$  maka
    Tampilkan “Predikat C”
lainnya
    Tampilkan “Predikat D”
Selesai
```



Pseudo-kode:

```
if (nilai >= 85) {  
    print("Predikat A");  
} else if (nilai >= 70) {  
    print("Predikat B");  
} else if (nilai >= 55) {  
    print("Predikat C");  
} else {  
    print("Predikat D");  
}
```

Struktur ini menjaga algoritma tetap linier (langkah demi langkah) namun memungkinkan jalur alternatif lebih dari dua.

4) Percabangan bersarang / kompleks

Kadang, di dalam salah satu blok if/else perlu ada percabangan tambahan. Hal ini disebut nested if.

Contoh nyata:

```
Input skor  
Jika skor ≥ 60 maka  
    Jika kehadiran ≥ 75% maka  
        Tampilkan "Lulus"  
    lain-lain  
        Tampilkan "Tidak lulus - hadir kurang"  
    lain-lain  
        Tampilkan "Tidak lulus - skor kurang"  
Selesai
```



Dalam kode:

```
if (skor >= 60) {  
    if (kehadiran >= 75) {  
        print("Lulus");  
    } else {  
        print("Tidak lulus - kehadiran kurang");  
    }  
} else {  
    print("Tidak lulus - skor kurang");  
}
```

Pemahaman terhadap percabangan bersarang penting agar siswa bisa membaca dan menulis teks algoritmik yang bertahap dan sistematis.

C. Contoh Nyata dan Penerapan Percabangan

1) Contoh nyata kasus sederhana

- Kasus tugas sekolah:

“Jika siswa mengumpulkan tugas tepat waktu maka dapat bonus nilai 5 point, lainnya tidak.”

Algoritma:

```
Input tepatWaktu (ya/tidak)  
Jika tepatWaktu = "ya" maka  
    nilaiAkhir = nilaiAkhir + 5  
Selesai
```

- Kasus sistem absensi:

“Jika jumlah kehadiran siswa $\geq 90\%$ maka berhak mendapatkan sertifikat, lainnya jika $\geq 75\%$ maka mendapatkan piagam, lainnya hanya mendapatkan penghargaan kelas.”



Algoritma:

```
Input kehadiran%
Jika kehadiran >= 90 maka
    tampil "Sertifikat"
else if kehadiran >= 75 then
    tampil "Piagam"
else
    tampil "Penghargaan Kelas"
Selesai
```

2) Penerapan dalam pemrograman (contoh dalam pseudo-kode / kode)

Misalnya dalam bahasa pemrograman umum (PHP / Java / Python / C++):

```
// contoh dalam bahasa PHP (pseudo
$nilai = ...;
if ($nilai >= 85) {
    echo "Predikat A";
} else if ($nilai >= 70) {
    echo "Predikat B";
} else {
    echo "Predikat C atau kurang";
}
```

atau dalam Python:

```
nilai = int(input("Masukkan nilai: "))
if nilai >= 85:
    print("Predikat A")
elif nilai >= 70:
    print("Predikat B")
else:
    print("Predikat C atau kurang")
```

Dengan memahami percabangan, siswa dapat menulis algoritma terstruktur untuk berbagai persoalan, dan kemudian menerjemahkannya ke kode program.