

04 Percabangan

Ringkasan Materi

Struktur percabangan dalam pembelajaran algoritma adalah mekanisme yang memungkinkan program memilih jalur eksekusi berbeda berdasarkan kondisi tertentu. Percabangan membantu menyelesaikan masalah yang memerlukan keputusan logis—misalnya, menentukan apakah suatu angka genap atau ganjil, atau memilih tindakan berdasarkan input pengguna. Struktur ini menjadi dasar penting dalam membangun logika pemrograman yang dinamis dan responsif.

Beberapa struktur percabangan yang biasa digunakan dalam pemrograman adalah sebagai berikut:

1. IF
Struktur ini akan mengeksekusi *statement* (perintah) jika dan hanya jika kondisi terpenuhi.
2. IF ELSE
Struktur ini akan mengeksekusi *statement* jika kondisi terpenuhi, dan akan mengeksekusi *statement* lain bila kondisi tidak terpenuhi.
3. IF ELSE IF
Struktur ini adalah struktur IF sebagai cabang kondisi ELSE dari struktur IF ELSE sebelumnya.
4. SWITCH
Struktur ini akan mengeksekusi *statement* bila pilihan atau kondisi terpenuhi. Struktur SWITCH ini pada dasarnya mirip dengan IF ELSE IF.

Untuk penjelasan lebih lengkap bisa dilihat pada slide [04 Percabangan.pptx](#).

Percobaan

Dice

Project ini akan mensimulasikan permainan dadu, di mana user diminta untuk menebak angka yang keluar. Angka dadu 1 – 6 akan terpilih secara acak kemudian user akan memberikan input angka tebakan. Bila tebakan benar maka akan muncul notifikasi.

Untuk menghasilkan angka acak dibutuhkan *random generator*. Beberapa fungsi yang harus digunakan adalah:

1. `rand()`
Fungsi **`rand()`** berada di *header <cstdlib>*, digunakan untuk menghasilkan bilangan bulat acak. Untuk menghasilkan bilangan acak dengan rentang tertentu harus dilakukan operasi modulo. Contoh berikut ini adalah untuk menghasilkan bilangan acak antara 1 – 6, harus dilakukan modulo 6 + 1:
- ```
int diceNumber = rand() % 6 + 1;
```
2. `srand()`  
Fungsi **`srand()`** juga berada di *header <cstdlib>*, digunakan sebagai umpan *random generator* agar bilangan acak yang dihasilkan selalu berubah – ubah. Fungsi ini harus dipanggil sebelum fungsi **`rand()`**.
  3. `time()`  
Fungsi **`time()`** berada di *header <ctime>*, digunakan untuk membaca waktu sistem sekarang. Dalam kaitannya dengan bilangan acak, fungsi ini biasa digunakan sebagai besaran umpan

yang dimasukkan melalui fungsi `srand()`. Contoh penggunaannya bersama fungsi `srand()` adalah sebagai berikut:

```
srand(time(NULL));
```

Potongan kode selengkapnya adalah sebagai berikut:

```
srand(time(NULL));
int diceRoll = rand() % 6 + 1;
int guess;
cout << "Guess a number between 1 and 6: ";
cin >> guess;
if(guess == diceRoll)
{
 cout << "Coorrect!" << endl;
}
else
{
 cout << "Sorry, the correct number was " << diceRoll << "." << endl;
}
system("pause");
```

## ShapeArea

Pembuatan menu sederhana menggunakan struktur `switch`. Pilihan menu berupa input angka. Bila pilihan tidak tersedia maka akan diabaikan. Contoh kasus membuat *project* perhitungan beberapa luas bangun datar yaitu lingkaran, segi empat, dan segi tiga. Potongan kode selengkapnya adalah sebagai berikut:

```
cout << "Shape Area Calculator" << endl;
cout << "1. Triangle" << endl;
cout << "2. Rectangle" << endl;
cout << "3. Circle" << endl;
cout << "Choose a shape (1-3): ";

int menuChoice;
cin >> menuChoice;

system("cls");
switch(menuChoice)
{
 case 1:
 double base, height;
 cout << "Base: ";
 cin >> base;
 cout << "Height: ";
 cin >> height;
 double area = 0.5 * base * height;
 cout << "Area : " << area << endl;
 break;
 case 2:
 double length, width;
 cout << "Length: ";
 cin >> length;
 cout << "Width : ";
 cin >> width;
 double area = length * width;
 cout << "Area : " << area << endl;
 break;
 case 3:
 const double PI = 3.14159;
 double radius;
 cout << "Radius: ";
 cin >> radius;
```

```
 double area = PI * radius * radius;
 cout << "Area : " << area << endl;
 break;
 default:
 cout << "Invalid choice." << endl;
}
system("pause");
```

## Latihan

1. Salinlah *project* **BMICalc** pada materi **03 Input Output Standar** ke *solution* yang sekarang. Selain menampilkan nilai BMI, tambahkan juga informasi apakah termasuk kategori **underweight**, **normal**, **overweight**, atau **obesity**. Pembagiannya adalah sebagai berikut
  - underweight: BMI < 18.5
  - normal: 18.5 – 22.99
  - overweight: 23 – 24.99
  - obesity: >= 25

Contoh tampilannya adalah sebagai berikut:

```
BMI Calculator

Weight: 70 kg
Height: 1.65 m
Result:
- BMI 25.71
- You are 'obesity'
```