



Struktur Kontrol Percabangan

Hendri Winarto, S.T., M.Kom.

Mapel Pemrograman Dasar - X RPL
Kurikulum 2013 Revisi 2017
SMK Negeri 1 Pacitan

06

Materi Pokok

- Struktur percabangan
- Percabangan tidak bersyarat
- Percabangan sederhana
- Percabangan bertingkat
- Percabangan bersarang

Struktur Percabangan

- Digunakan untuk memilih alur program berdasarkan suatu kondisi yang diuji nilai kebenarannya
- Merupakan implementasi struktur algoritma pemilihan (*selection*), yang pada flowchart direpresentasikan dengan bangun belah ketupat



- Pada bahasa C/C++, percabangan diimplementasikan menggunakan syntax:
 - goto
 - if ...[else if] ... [else]
 - switch .. case

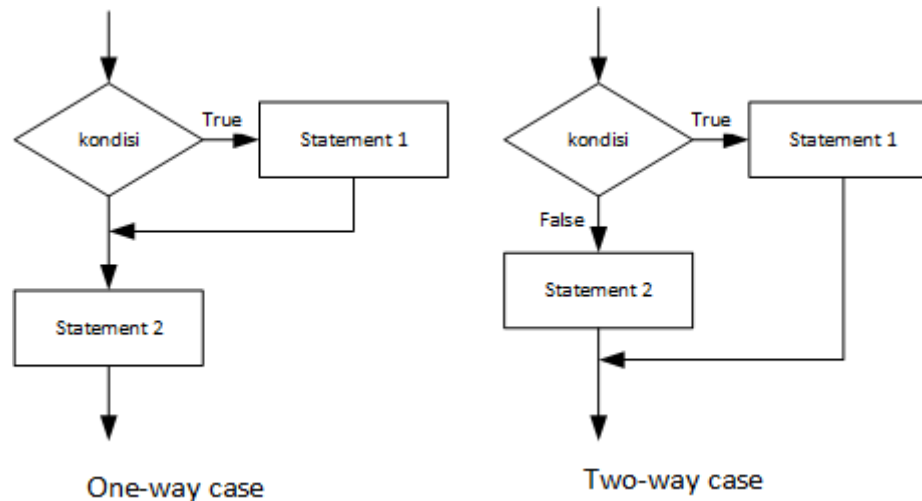
Percabangan Tidak Bersyarat

- Percabangan yang tidak membutuhkan pengecekan kondisi true/false untuk menuju ke bagian tertentu dari program
- Pada bahasa C++ percabangan tidak bersyarat bisa dilakukan menggunakan syntax `goto` diikuti dengan label yang kita tentukan sendiri di bagian lain dari program.
- Label diketikkan dengan nama bebas (memenuhi kaidah penamaan seperti variabel), diakhiri dengan karakter titik dua (:)

```
Percabangan > goto.cpp > ...
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      cout << "Hello RPL!" << endl;
6      goto akhir;
7      cout << "Apakah kode ini dieksekusi ?" << endl;
8  akhir:
9      cout << "Program diakhiri" << endl;
10     return 0;
11 }
```

Percabangan Sederhana

- C++ menggunakan `if ...` dan `if ... else` untuk percabangan sederhana
- Percabangan yang hanya menggunakan `if ...` disebut *one-way case*, sedangkan yang menggunakan `if ... else` disebut *two-way case*



- Pada one-way case, jika kondisi bernilai true, maka Statement 1 dan Statement 2 akan dijalankan. Sedangkan pada two-way case, Akan dipilih hanya salah satu statement saja

Percabangan Sederhana One-way

- Percabangan sederhana one-way di C++ berupa `if ..` tanpa `else`. Jika dibuat kode program, implementasi flowchart pada halaman sebelumnya kurang lebih akan menjadi seperti ini :

```
Percabangan > C+ percabangan-sederhana-one-way.cpp > ...
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      bool kondisi = false;
6      if (kondisi) {
7          cout << "Statement 1" << endl;
8      }
9      cout << "Statement 2" << endl;
10     return 0;
11 }
```

- Penjelasan: pada saat kondisi bernilai `true` maupun `false`, Statement 2 tetap akan dieksekusi. Namun jika kondisi bernilai `true`, Statement 1 akan dieksekusi terlebih dahulu, baru mengeksekusi Statement 2

Percabangan Sederhana Two-way

- Percabangan sederhana two-way di C++ menggunakan if ... else dengan format:
if (kondisi) {
 // blok statement yang dijalankan saat kondisi bernilai true
} else {
 // blok statement yang dijalankan saat kondisi bernilai false
}
- Contoh penggunaan pada penentuan bilangan ganjil atau genap :

```
#include <iostream>
using namespace std;

int main() {
    int n;
    cout << "Masukkan bilangan bulat: "; cin >> n;
    if ((n % 2) == 0)
        cout << n << " adalah bilangan genap" << endl;
    else
        cout << n << " adalah bilangan ganjil" << endl;
    return 0;
}
```

- Catatan: kurung kurawal dapat dihilangkan jika hanya menangani **satu statement**

Sintaks Alternatif

Percabangan if ... else

- Pada kebutuhan tertentu di mana sintaks `if .. else` hanya digunakan untuk memilih 2 nilai atau membuat keputusan sederhana, maka sintaks `if .. else` dapat disederhanakan menggunakan sintaks alternatif, dengan hasil yang sama.
- Sintaks berikut:

```
if (kondisi)  
    nilai = "A";  
else  
    nilai = "B";
```
- Dapat disederhanakan menjadi :

```
nilai = (kondisi) ? "A" : "B";
```
- Keduanya memiliki makna yang sama persis, meski ada keterbatasan tertentu pada implementasi sintaks alternatif.
- Sintaks alternatif tersebut disebut juga sebagai **ternary operator**.

Percabangan Bertingkat if ... else if .. else

- Pada penyelesaian kasus pemrograman, seringkali ada beberapa kondisi bertingkat yang harus ditangani, di mana pada operasi logika yang kita gunakan sebagai kondisi pada if tidak selalu bisa ditangani langsung pada blok untuk else. Untuk keperluan ini, kita dapat menggunakan sintaks percabangan bertingkat.

- Sintaks :

```
if (kondisi1) {  
    statement 1;  
} else if (kondisi2) {  
    statement 2;  
}  
.... // bisa ditambahkan blok else if sesuai kebutuhan  
else {  
    statement lain;  
}
```

Contoh Percabangan Bertingkat if ... else if .. else

- Program sederhana berikut ini mengecek bilangan yang diinputkan pengguna, dan memutuskan apakah bilangan tersebut positif, negatif, ataukah bilangan nol

```
Percabangan > percabangan-bertingkat-bilangan-bulat.cpp > ...
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int b;
6      cout << "Masukkan bilangan bulat: ";
7      cin >> b;
8      if (b > 0)
9          cout << b << " adalah bilangan positif" << endl;
10     else if (b < 0)
11         cout << b << " adalah bilangan negatif" << endl;
12     else
13         cout << b << " adalah bilangan nol" << endl;
14     return 0;
15 }
```

Percabangan dengan switch ... case

- Percabangan bertingkat dengan `if .. else if .. else` yang memiliki nilai kondisi berupa nilai pasti (bukan berupa operasi perbandingan atau rentang nilai), dapat direpresentasikan ke dalam sintaks `switch ... case`
- Sintaks:

```
switch (expression) {  
    case nilai1: statement1;  
                break;  
    case nilai2: statement2;  
                break;  
                ...  
    default: statement default;  
}
```
- Berbeda dengan `if` yang mengevaluasi kondisi, `switch` menggunakan perbandingan kecocokan antara `expression` (biasanya berupa variabel saja) dengan nilai yang ada pada bagian `case`. Nilai dapat berupa numerik, karakter, string, dsb.
- Setiap kecocokan nilai pada `case` dapat digunakan untuk mengeksekusi blok `statement` sebanyak yang dibutuhkan, hingga menemui sintaks `break`.
- Jika tidak ada kecocokan nilai dengan semua yang disediakan pada bagian `case`, maka blok `statement` pada bagian `default` akan dieksekusi. Blok `statement` pada bagian `default` tidak perlu diakhiri dengan `break`.

Contoh Percabangan dengan switch ... case

- Contoh percabangan berikut ini memilih untuk menampilkan pilihan menu oleh pengguna program, yang disimpan pada variabel p

```
Percabangan > G percabangan-switch-case.cpp > ...
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int p;
6      cout << "Masukkan pilihan menu (1 - 5):";
7      cin >> p;
8      switch (p) {
9          case 1: cout << "Anda memilih menu 1. Harga Rp. 7.000" << endl;
10             break;
11          case 2: cout << "Anda memilih menu 2. Harga Rp. 10.000" << endl;
12             break;
13          case 3: cout << "Anda memilih menu 3. Harga Rp. 12.000" << endl;
14             break;
15          case 4: cout << "Anda memilih menu 4. Harga Rp. 15.000" << endl;
16             break;
17          case 5: cout << "Anda memilih menu 5. Harga Rp. 20.000" << endl;
18             break;
19          default:
20             cout << "Anda memasukkan pilihan yang tidak tersedia" << endl;
21      }
22      return 0;
23 }
```

Percabangan If Bersarang (*Nested If*)

- Percabangan if bersarang pada dasarnya hanyalah percabangan if yang terletak di dalam blok percabangan if lainnya, baik pada bagian if, else if, maupun else.
- Struktur ini lazim digunakan pada alur program yang memang membutuhkan suatu percabangan if untuk dijalankan sebagai syarat bagi percabangan if lainnya
- Sintaks umum:

```
if (kondisi1) { // outer if
    // statements
    if (kondisi2) { // inner if
        // statements
    }
    ....
}
```
- Baik struktur if pada bagian *outer* maupun *inner*, dapat berupa if saja (*one-way*), if ... else (*two-way*), maupun if ... else if ... else (bertingkat)

Contoh Percabangan If Bersarang (*Nested If*)

- Diketahui persyaratan untuk seleksi anggota Paskibra menggunakan persyaratan tinggi badan yang berbeda untuk calon dengan jenis kelamin laki-laki dan perempuan.
- Meski pendekatan algoritma yang berbeda tentunya juga bisa digunakan, contoh berikut ini dapat memberikan gambaran mengenai penggunaan percabangan if bersarang secara sederhana.

```
Percabangan > G percabangan-bersarang.cpp > ...
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      char jk; int tinggi;
6      cout << "Apa jenis kelamin Anda? (L/P): ";
7      cin >> jk;
8      cout << "Berapa tinggi Anda (dalam cm) ? ";
9      cin >> tinggi;
10     if (jk == 'L' || jk == 'l') {
11         if (tinggi >= 170)
12             cout << "Memenuhi syarat";
13         else
14             cout << "Belum memenuhi syarat";
15     } else if (jk == 'P' || jk == 'p') {
16         if (tinggi >= 160)
17             cout << "Memenuhi syarat";
18         else
19             cout << "Belum memenuhi syarat";
20     } else {
21         cout << "Jenis kelamin tidak valid!";
22     }
23     return 0;
24 }
```



Fin.