

[Home](#) / [Tutorial](#) / [C++](#) / [Pernyataan Switch](#)

## Pernyataan Switch

---

Posted on Januari 23, 2018 | Last Modified Januari 23, 2018

---



Pernyataan switch adalah pernyataan penyeleksian untuk memanipulasi jalannya aliran program berdasarkan conditional expression, Mengendalikan aliran program agar dapat berjalan secara fleksibel atas keadaan program dan keinginan user. Pernyataan ini akan memilih solusi / pilihan yang sesuai dengan ekspresi kondisi.

Pada umumnya pernyataan switch tidak jauh berbeda seperti pernyataan penyeleksian IF, yang berbeda adalah switch dikhususkan untuk penyeleksian berdasarkan nilai dari ekspresi. Jika if mengevaluasi kebenaran dari sebuah ekspresi (conditional expression) seperti operasi logika, relasi dan perbandingan yang menghasilkan bilangan boolean 1 (benar) atau 0 (salah).

Dalam pernyataan switch, penyeleksian dilakukan dengan memeriksa kondisi (lebih tepatnya nilai) dari suatu variabel bertipe integer, char atau enum dan menemukan kesamaan data yang dicari oleh salah satu label case yang ada.

[Bentuk penulisan](#)

```
switch (kondisi )
{
    case konstanta-1 :
        ... perintah;
        ... perintah;
        break;
    case konstanta-2 :
        ... perintah;
        ... perintah;
        break;
    default :
        ... perintah;
        ... perintah;
        break;
}
```

Pernyataan switch dimulai dengan “switch” keyword dan setelah itu sertakan conditional expression yang diapit dengan sepasang tanda kurung “ () ”, conditional expression bisa berbentuk variabel dari integer, variabel dari char atau enum. dan kita juga bisa melakukan operasi aritmatika, hal itu diperbolehkan asal hasil dari operasi merupakan integer, char atau enum.

Dan di dalam pernyataan switch ada beberapa case label sebanyak yang dibutuhkan. Penulisan diawali dengan keyword case dan diikuti dengan konstanta nilai yang dicari, Mereka dimasukan di dalam sepasang tanda kurung kurawal “ {} ”, case di dalam pernyataan switch merupakan label yang berfungsi untuk menandai nilai yang dicari.

Keyword break merupakan pernyataan lompatan, dan merupakan bagian optional (dapat tidak digunakan) berfungsi untuk mengeluarkan CPU yang sedang berada dalam pernyataan switch, keluar dan melanjutkan baris setelah keseluruhan pernyataan switch.

Label default merupakan pilihan yang akan dieksekusi dimana CPU tidak menemukan nilai yang dia cari sesuai yang dinyatakan pada conditional expression. Label tersebut tidak wajib untuk digunakan dan tempatnya harus berada di bawah.

### Contoh Penulisan

```
switch ( nomer ) {  
    case 1 :  
        cout << " Anda Memilih Angka 1 " ;  
        break ;  
  
    case 2 :  
        cout << " Anda Memilih Angka 2 " ;  
        break ;  
  
    case 3 :  
        cout << " Anda Memilih Angka 3 " ;  
        break ;  
  
    default :  
        cout << " Pilih default Terpilih " ;  
        break ;  
}
```

Disaat CPU bertemu pernyataan switch, CPU akan memeriksa nilai yang berada sebagai conditional expression dari switch tersebut. dari situ CPU akan mencari konstanta pada label case yang sesuai dengan nilai yang di cari. Ketika telah ditemukan, maka CPU akan mengeksekusi perintah / pernyataan-pernyataan yang berada di bawah label dengan konstanta yang sesuai.

Keyword break merupakan optional, dapat tidak digunakan. Jika break tercantum, ketika CPU selesai mengeksekusi pernyataan di atas break dan bertemu break maka eksekutor akan dikeluarkan dari pernyataan switch dan melanjutkan baris setelah pernyataan switch. Dan jika break tidak disertakan, maka CPU akan terus berjalan untuk mengeksekusi semua pernyataan termasuk pernyataan yang berada pada label case lain hingga bertemu kurung kurawal penutup "}".

```
switch ( nomer ) { //jika nomer bernilai 2
    case 1 :
        cout << " Anda Memilih Angka 1 " ;

        case 2 : // CPU mulai mengeksekusi
            cout << " Anda Memilih Angka 2 " ; //Dieksekusi

            case 3 ://dan akan terus mengeksekusi semua pernyataan
                hingga betemu break atau akhir dari pernyataan switch
                    cout << " Anda Memilih Angka 3 " ;

            default :
                cout << " Pilih default Terpilih " ;
}
```

Jika CPU tidak menemukan nilai yang sesuai pada seperti ekspresi, hal itu berarti CPU akan mengeksekusi perintah di dalam label default, jika pernyataan switch tidak mempunyai label default maka pernyataan switch akan diabaikan.

#### Contoh Program

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

int main() {
    int nomer ;

    cout << " Masukkan [1/2/3] : " ; cin >> nomer ;
    switch ( nomer ) {
        case 1 :
            cout << " Anda Memilih Angka 1 " ;
            break ;

        case 2 :
            cout << " Anda Memilih Angka 2 " ;
            break ;

        case 3 :
            cout << " Anda Memilih Angka 3 " ;
            break ;

        default :
            cout << " Pilihan default Terpilih " ;
            break ;
    }
    return 0 ;
}
```

Mendirikan variabel di dalam pernyataan switch.

Ini adalah sedikit informasi tambahan mengenai switch. Pada bahasa pemrogramana C++ kita dimungkinkan untuk mendirikan tapi tidak dapat inisialisasi. mungkin ini terdngar normal karena kita juga bisa mendirikan variabel pada pernyataan penyeleksian if. bukan hal itu yang akan penulis beritahu, ada sesuatu yang unik pada pernyataan switch.

Pendeklarasian beserta inisialisasi variabel kita dimungkinkan melakukan hal itu di dalam suatu ruang lingkup. Ruang lingkup function, pernyataan majemuk, bahkan pernaytaan if. dalam pernyataan switch kita dimungkinkan untuk mendeklarasi sebuah variabel tapi tidak untuk di inisialisasi.

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

int main() {
    int nomer ;

    cout<<" Masukkan [1/2/3] : ";cin>>nomer;
    switch (nomer){
        case 1:
            char huruf ;
            huruf = 'a';
            cout<<huruf;
            break;

        case 2:
            huruf = 'b';
            cout<<huruf;
            break;
        case 3:
            huruf = 'c' ;
            cout<<huruf;
            break;

        default:
            huruf = 'z' ;
            cout<<huruf;
            break;
    }
    return 0 ;
}
```

Program di atas akan berjalan tanpa error. dan akan mencetak huruf a, b, c atau z dengan menggunakan variabel yang didirikan di dalam label case 1. Meskipun aliran eksekusi CPU memilih label case selain label 1 tapi variable itu tetap dapat digunakan.

Apa yang kita tau sebelumnya adalah, aliran pengekekusian CPU jika menumukan pernyataan penyeleksian dia akan mengevaluasi conditional expression. Untuk program di atas conditional expressionnya adalah variabel 'nomer'. Dan akan mengeksekusi perintah di

bawah label case yang memiliki nilai yang sama.

Jika anda berpiki bahwa itu aneh, itu berarti anda benar. Kenapa variabel huruf tetap dapat digunakan meskipun berbeda case ?

Jawaban yang pertama adalah mereka masih didalam satu ruang lingkup. Dan kenapa variabel tersebut dapat digunakan meskipun CPU tidak terlihat menyentuhnya.

Disaat anda mendirikan local variabel, sebenarnya variabel tidak dibuat saat CPU mengeksekusi variabel tersebut tapi variabel dibuat dan ada ketika CPU bertemu dengan block pembuka ruang lingkup. Tapi variabel itu hanya tidak terlihat, dan terlihat setelah compiler bertemu deklarasi tersebut. Sebenarnya variabel tidak butuh untuk dieksekusi, keberadaan dari deklarasi variabel pada C++ sebenarnya hanya untuk memberitau compiler bahwa variabel akan terlihat setelah baris tersebut.

Dan hal yang lain, kita sebenarnya bisa deklarasi dan inisialisasi secara bersamaan di dalam pernyataan switch. Untuk membuat deklarasi variabel dengan inisialisasi di dalam pernyataan switch kita butuh sepasang block ruang lingkup yaitu sepasang tanda kurung kurawal “ {} ” tapi hal itu akan membatasi keberadaan dari variabel, yang bisa membuat tidak dapat digunakan di luar scope dan juga dalam label case lain.


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

int main() {
    int nomer ;

    cout<<" Masukkan [1/2/3] : ";cin>>nomer;
    switch (nomer){
        case 1:{
            char huruf = 'a' ;
            cout<<huruf;
            break;
        }

        case 2:
            huruf = 'b';
            cout<<huruf;
            break;
        case 3:
            huruf = 'c' ;
            cout<<huruf;
            break;

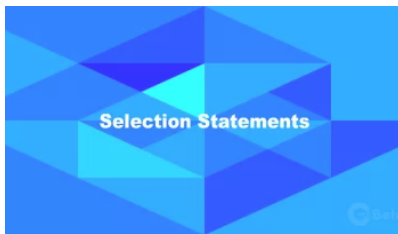
        default:
            huruf = 'z' ;
            cout<<huruf;
            break;
    }
    return 0 ;
}
```

 Share Tweet Share Pin Share

---

**Terkait**

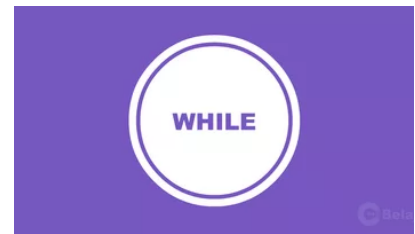




Pernyataan Penyeleksian  
Januari 23, 2018  
dalam "C++"



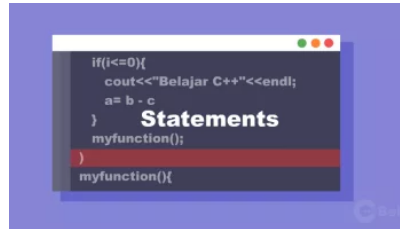
Pernyataan IF  
Januari 23, 2018  
dalam "C++"



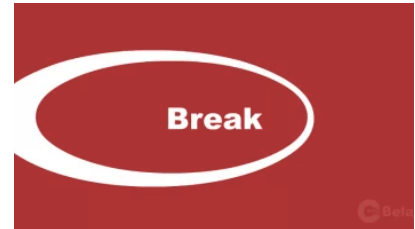
Pernyataan While  
Januari 23, 2018  
dalam "C++"



Pernyataan Do-while  
Januari 23, 2018  
dalam "C++"



Pernyataan dan Kontrol Aliran  
Januari 23, 2018  
dalam "C++"



Pernyataan Break  
Januari 23, 2018  
dalam "C++"