# Contoh Program Sederhana Percabangan C++

Salah satu strukur kontrol dalam bahasa pemrograman c++ yang akan kita bahas adalah konsep percabangan. Idenya, kita akan belajar membuat program pengambil keputusan.

Sebelum itu, mari kita lihat pengertian dari percabangan terlebih dahulu:

**Percabangan** merupakan pemilihan statemen yang akan dieksekusi dimana pilihan tersebut didasarkan atas kondisi tertentu untuk mengarahkan jalannya program.

*(Silahkan baca:*[*Algoritma percabangan*](https://kodedasar.com/algoritma-percabangan/)*)*

Dalam kehidupan sehari-hari terkadang kita dihadapkan pada kondisi dimana kita harus memilih suatu tindakan. Sebagai contoh ketika anda ingin menonton film di bioskop.

Konsepnya seperti ini:

* **Jika** anda memiliki tiket, **maka** anda boleh masuk untuk menonton film.

Nah, disana ada sebuah kondisi yang harus terpenuhi yaitu memiliki tiket.

Artinya jika anda tidak memiliki tiket (kondisi = FALSE) maka anda tidak diperbolehkan untuk menonton film di bioskop. Kurang lebih itulah contoh nyata konsep percabangan.

Intinya selama kondisi terpenuhi maka program akan dijalankan.

Begitu juga dalam membuat program, kita akan sedikit banyak berurusan dengan pilihan. Disini kita akan mengenal dua fungsi percabangan yaitu if else dan switch case.

**Table Of Contents**

* [#1 Contoh Program If Else pada C++](https://kodedasar.com/percabangan-cpp/#1-contoh-program-if-else-pada-c)
* [#2 Contoh Program Switch Case pada C++](https://kodedasar.com/percabangan-cpp/#2-contoh-program-switch-case-pada-c)
* [Menarik Kesimpulan](https://kodedasar.com/percabangan-cpp/#menarik-kesimpulan)

**#1 Contoh Program If Else pada C++**

Berikut ini merupakan contoh program C++ sederhana menggunakan fungsi if else:

#include <iostream>

using namespace std;

void pilihan (int x){

if (x==1) {

cout << "Anda Memesan Nasi Goreng";

}

else {

cout << "Anda Memesan Nasi Kucing";

}

}

int main(){

int pilih;

cout << "1. Nasi Goreng" << '\n';

cout << "2. Nasi Kucing" << '\n';

cout << "Masukan Pulihan Anda (1-2): "; cin >> pilih;

pilihan (pilih);

}

Mari kita bahas satu persatu:

Dalam program di atas, saya membuat prosedur bernama pilihan dengan parameter x bertipe integer. *(Silahkan baca:*[*Apa itu prosedur dan fungsi dalam Pemrograman*](https://kodedasar.com/prosedur-dan-fungsi/)*.)*

Apabila variabel X bernilai 1 maka program akan mencetak "Anda pesan nasi goreng", jika tidak (selain angka 1) maka program akan mencetak "anda memesan nasi kucing".

Nah, didalam program utama saya hanya menyiapkan varibel pilih bertipe integer untuk menyimpan nilai inputan yang akan digunakan pada prosedur pilihan.

Mari kita lihat struktur dari fungsi If-else pada C++.

Secara sederhana, fungsi if dapat ditulis seperti berikut:

if (kondisi) {

Statemen1;

//kode yang dijalankan jika bernilai benar

}

else {

Statemen2;

//kode yang dijalankan jika salah

}

Jadi cara kerjanya seperti ini: Jika kondisi bernilai benar (TRUE) maka statement1 akan dijakanlan, sedangkan jika bernilah salah (FALSE) maka statemen2 yang dieksekusi.

**Macam-macam operator yang digunakan.**

**Operator Relasi** - Operator ini digunakan untuk membandingkan 2 buah data untuk mendepatkan kondisi benar (TRUE) atau salah (FALSE). Berikut daftarnya:

| **Operator** | **Keterangan** |
| --- | --- |
| == | Sama dengan |
| != | Tidak sama dengan |
| < | Kurang dari |
| > | Lebih dari |
| <= | Kurang dari atau sama dengan |
| >= | Lebih dari atau sama dengan |

**Operator Logika** - Faktanya, terkadang kita dihadapkan kondisi yang memiliki lebih dari satu syarat. Jika anda mengalami hal yang sama maka gunakan operator berikut:

| **Operator** | **Keterangan** |
| --- | --- |
| && | And (Semua kondisi harus bernilai benar) |
| || | Or (Salah satu harus bernilai benar) |
| ! | Not (Kebalikan dari kondisi sebenarnya) |

**Nested If** atau **If Bersarang** - Seperti yang telah saya singgung diatas, terkadang kita akan dihadapkan pada kondisi yang sedikit lebih rumit dalam menentukan pilihan.

Jika anda paham struktur **percabangan if else** maka anda hanya perlu menambahkan fungsi if dalam sebuah program if tersebut. contohnya seperti bari kode berikut:

#include <iostream>

using namespace std;

void Hasil (int x, int y){

if (x==1) {

if (y>=17){

cout << "Pria Dewasa";

}

else{

cout << "Perjaka";

}

}

else {

if (y>=17){

cout << "Wanita Dewasa";

}

else{

cout << "Gadis";

}

}

}

int main(){

int umur, kel;

cout << "Jenis Kelamin" << '\n';

cout << "1. Laki-Laki" << '\n';

cout << "2. Perempuan" << '\n';

cout << "Masukan Pilihan Anda (1-2): "; cin >> kel;

cout << "Masukan Umur Anda (1-100): "; cin >> umur;

Hasil(kel,umur);

}

Pada kasus ditas, kita ingin mengetahui bagaimana seseorang dikatakan gadis, perjaka, wanita dewasa, dan pria dewasa menggunakan dua syarat yaitu umur dan jenis kelamin.

Inti dari nested if adalah: Kita memiliki pecabangan didalam percabangan.

**#2 Contoh Program Switch Case pada C++**

Secara pengertian, switch dapat diartikan sebagai berikut:

**Switch** adalah konsep percabangan yang digunakan untuk menyelesaikan sebuah masalah dengan sejumlah kemungkinan dengan kondisi konstan.

Namun pada dasarnya memiliki fungsi yang sama dengan percabangan if else.

Lalu apa yang membedakan kedua sintaks tersebut dan kapan anda menggunakan fungsi switch atau sebaliknya? Mari kita lihat strukur peercabangan switch case.

Secara sederhana peryataan switch dapat ditulis sebagai berikut:

switch (kondisi)

{

case constant1:

statements-1;

break;

case constant2:

statements-2;

break;

default:

default-statements;

}

Mari kita pelajari bagaimana cara kerja program switch.

Peryataan switch mengevaluasi kondisi dan memeriksa apakah sama dengan constant1. Jika ya, maka statements-1 akan dieksekusi hingga kode break;

Jika tidak maka switch akan memeriksa terhadap constant2 dan seterusnya hingga kondisi terpenuhi dan mengeksekusi program sesuai konsisi tersebut.

Namun, jika nilai kondisi tidak sesuai dengan konstanta yang telah ditentukan sebalumnya, maka program akan mengeksekusi peryataan default.

**Catatan**: Saat menemukan pernyataan break, maka program akan beralih ke akhir keseluruhan pernyataan switch (tanda "}" ) tanpa mengeksekusi baris dibawahnya.

Sedangkan keyword default akan bekerja jika semua kondisi tidak terpenuhi.

Berikut contoh sederhana program percabangan pada C++ menggunakan switch case:

#include <iostream>

using namespace std;

int pilihan;

string text;

int main(){

cout << "Daftar bahasa pemrograman:\n";

cout << "1. Java \n2. C++ \n3. PHP\n";

ulangi:

cout << "Masukan pilihan anda (1-3): ";

cin >> pilihan;

switch(pilihan){

case 1:

text="Anda memilih Pemrograman Java";

break;

case 2:

text="Anda memilih Pemrograman C++";

break;

case 3:

text="Anda memilih Pemrograman PHP";

break;

default:

goto ulangi;

}

cout << text;

}

Disini saya menggunakan dua variabel, pertama variabel pilihan dengan tipe data integer yang akan kita gunakan sebagai kondisi pada peryataan switch.

Yang kedua variabel text dengan tipe data string yang akan kita gunakan untuk menyimpan tulisan sesuai pilihan kita sebelumnya.

Kemudian saya menggunakan input output seperti biasa, masih bingung silahkan baca:

* [Dasar Input & Output (I/O) pada C++ dengan Fungsi cin dan cout](https://kodedasar.com/io-cpp/)

Masuk ke pokok pembahasan kita kali ini, disini saya akan mengevalusi kondisi (pilihan) dengan konstanta 1-3 menggunakan fungsi switch case.

Namun apabila kondisi tidak terpenuhi maka fungsi default akan dieksekusi yaitu menuju ke indetifier *ulangi:* dan kita akan memasukan pilihan lagi.

Sebagai contoh: apabila anda memasukan angka 2, maka tulisan yang akan muncul adalah "Anda memilih pemrograman C++".

Namun jika angka yang anda masukan bukan 1, 2, dan 3 maka program akan menuju identifier ulangi: karena kondisi default diberi fungsi *goto ulangi*.

**Studi Kasus** - Sekarang anda sudah mengetahui apa itu switch case dan bagaimana stuktur penulisannya, anda juga telah mengetahui apa fungsi break dan default.

Sebagai studi kasus, disini saya akan memberikan contoh program "NPM (Nomor Pokok Mahasiswa) Generator" menggunakan peryataan switch dalam bahasa c++.

#include <iostream>

using namespace std;

string tahun;

int fakultas,prodi;

string idtahun,idfakultas,idprodi,nomor,npm;

int main(){

cout << "Masukan tahun: "; cin >> tahun;

idtahun = tahun.substr(2,3);

cout << "Fakultas:\n 1. Hukum\n 2. Pendidikan\n 3. Ilmu Kesehatan\n 4. Teknik\n";

cout << "Masukan fakultas anda: "; cin >> fakultas;

switch (fakultas){

case 1:

idfakultas="01";

cout << "Program Studi:\n 1. Ilmu Humum S1\n";

break;

case 2:

idfakultas="02";

cout << "Program Studi:\n 1. Bimbingan Konseling (S-1)\n 2. Pendidikan Guru PAUD (S-1) 3. Pendidikan Guru SD (S-1)\n";

break;

case 3:

idfakultas="03";

cout << "Program Studi:\n 1. Ilmu Keperawatan (S-1)\n 2. Keperawatan (D-3)\n 3. Farmasi (D-3)\n 4. Profesi Ners\n";

break;

case 4:

idfakultas="04";

cout << "Program Studi:\n 1. Teknik Industri (S-1)\n 2. Teknik Informatika (S-1)\n";

break;

}

cout << "Masukan pilihan anda: "; cin >> prodi;

switch (prodi){

case 1:

idprodi = "01";

break;

case 2:

idprodi = "02";

break;

case 3:

idprodi = "03";

break;

case 4:

idprodi = "04";

break;

}

cout << "Masukan Nomor Urut: "; cin >> nomor;

npm=idtahun+"."+idfakultas+"."+idprodi+"."+nomor;

cout << "NPM: " << npm;

}

Dalam program diatas saya sengaja tidak menggunakan fungsi default sehingga bisa saja anda memasukan angka diluar nilai yang tersedia.

Tugas anda adalah memperbaiki program tersebut, sebagai petunjuk silahkan gunakan peryataan switch didalam switch (nested switch case).

**Menarik Kesimpulan**

Meskipun kedua peryataan tersebut digunakan untuk menyelesaikan program percabangan atau pilihan, namun terdapat perbedaan yang mendasar.

Secara penulisan, peryataan switch lebih rapi dan mudah dipahami dibanding peryataan if else bertingkat terlebih untuk kondisi dengan banyak kemungkinan.

Intinya seperti ini: Peryataan switch biasa digunakan untuk kondisi konstan (contoh: 200, 301), sedangkan if else untuk konsisi dengan logika (contoh: >10, !=0).

Sekian artikel contoh program percabngan c++ dari saya, semoga jelas.