MAKALAH PENGKONDISIAN DALAM BAHASA

PEMROGRAMAN C



Oleh :

Diki Candra

NIM 2022903430010

D4 TEKNOLOGI REKAYASA KOMPUTER JARINGAN

POLITEKNIK NEGERI LHOKSEMAWE

2022/2023

**Daftar Isi**

1. BAB I (2)

PENDAHULUAN

**1.1 Latar Belakang** (2)

Rumusan Masalah (2)

Manfaat (2)

Tujuan (3)

1. BAB II (4)

PEMBAHASAN

**1.2 Penggenalan** (4)

A. Pengkondisian (4)

B. Kondisi Aksi (4)

C. Statement If-Else (6)

D. Jenis-jenis If (21)

1. BAB III (35)

PENUTUP

Daftar Pustaka (36)

BAB 1

PENDAHULUAN

**1.1 Latar Belakang**

Pengkondisian (atau dalam bahasa inggris disebut dengan condition) adalah kegiatan untuk bagaimana menentukan suatu program dijalankan atau tidak berdasarkan kondisi yang dicapai sebelumnya. Pengkondisian ini masih dalam lingkup materi struktur kendali pada bahasa pemrograman PHP.

Struktur kendali adalah Bagaimana alur program kita dibaca oleh interpreter PHP. Umumnya bahasa pemrograman dibaca dari atas ke bawah dari kiri ke kanan. Namun jika menggunakan pengkondisian alur program dapat bercabang seperti ini.

**Rumusan masalah**

Berdasarkan judul dari makalah ini, Saya merumuskan masalah yaitu pengkondisian dalam bahasa pemrograman bahasa C!

**Manfaat**

Manfaat yang diperoleh dari makalah tersebut adalah sebagai berikut:

1.**Pembaca**

A. Menjadikan pembaca makalah semata-mata agar pembaca memahami dasar-dasar pemrograman.

B. Untuk memotivasi pembaca untuk membuat makalah yang lebih baik dari Ini.

2.**Pengaramg**

C. Untuk memberikan semangat pada mata kuliah yang sedang diambil saat ini.

D. Sebagai titik awal untuk melakukan penelitian agar penelitian selanjutnya dapat dilakukan dengan lebih baik lagi.

**Tujuan**

1. Mengetahui Pengkondiasian

2. Dapat menerapkan Bahasa

BAB II

PEMBAHASAN

1. 2 **PENGENALAN**

Sangat sering ketika Anda menulis kode, Anda ingin melakukan tindakan berbeda untuk kondisi yang berbeda. Anda dapat menggunakan pernyataan kondisional dalam kode Anda untuk melakukan ini.

Dalam PHP kami memiliki pernyataan kondisional berikut:

1. if - mengeksekusi beberapa kode jika satu syarat benar.

2. if ... else - mengeksekusi beberapa kode jika suatu kondisi benar dan kode lain jika kondisi itu salah

3. if ... elseif ... else - mengeksekusi kode yang berbeda untuk lebih dari dua syarat

4. switch - memilih salah satu dari banyak blok kode yang akan dieksekusi

**A. pengkondisian**

Conditioning adalah `instruksi` dalam pemrograman untuk menjalankan suatu baris program menurut `kondisi tertentu`. Misal ada dua syarat, A dan B serta ada juga jalur program A dan B. Jika kondisi A terpenuhi maka program akan menjalankan program jalur A, sedangkan program jalur B tidak akan berjalan dan sebaliknya.

Dalam kehidupan sehari-hari kita sering menjumpai situasi atau kondisi seperti ini. Misalnya “jika saya terlambat hadir maka gaji saya akan dipotong, jika saya tidak terlambat maka gaji saya tidak akan dipotong”. Anbi akan membahas penggunaan pernyataan if dalam bahasa pemrograman C.

**B. Kondisi-Aksi**

Setiap program komputer umumnya terdiri dari banyak tindakan, bayangkan apa yang dapat dilakukan aplikasi chatting di ponsel anda:

* mengirim pesan
* menghapus pesan
* menelfon
* membuat group baru
* masuk ke halaman setting

Meskipun ada banyak kemungkinan tindakan, biasanya tindakan tersebut tidak dilakukan dengan segera dan bersamaan. Pada suatu saat ada tindakan yang dieksekusi, dan ada tindakan yang tidak dieksekusi. Biasanya ada kondisi tertentu yang memicu satu atau lebih tindakan untuk dieksekusi.

**Untuk lebih jelasnya:**

* Kondisi: Ekspresi yang jika dievaluasi bernilai True atau False.
* Aksi: Instruksi yang dijalankan berdasarkan nilai Kondisi.

**Berikut adalah contoh kondisi dan aksi:**

* Misalnya, apakah menu setting disentuh, jika iya maka program akan menampilkan halaman setting.
* Apakah ada pesan masuk, jika True, maka bunyikan nada masuk dan getar.
* Apakah nilai akhir lebih dari 60, jika iya tampilkan pesan “Lulus”.

Contoh Program :

**#Input**

nilai\_akhir = int(input("Nilai Akhir: "))

kondisi = nilai\_akhir > 60 #ini adalah kondisi

if kondisi == True:

print("Lulus") #ini adalah aksi yang ingin dijalankan hanya ketika kondisi bernilai True

**#Output**

Nilai Akhir: 30

Lulus

Di kode di atas kita dapat menuliskan kondisi dan aksi, namun aksi tersebut belum **bergantung** terhadap nilai kondisi. Baik kondisi True ataupun False, aksi print("Lulus") tetap dijalankan. Berikut ini kita akan melihat bagaimana agar aksi dapat bergantung dengan kondisi.

**C. Statement if-else**

Pada statement if kita telah melihat bahwa jika **kondisi true** maka ada aksi yang akan dijalankan, namun bagaimana jika kita juga perlu menjalankan suatu aksi lain jika **kondisi false**?

Python menyediakan keyword **else else** yang bisa digunakan setelah if yang artinya **aksidefault** di bawah else akan dijalankan jika **kondisi bernilai false**.

If kondisi:

aksi

Else:

aksidefault

Perhatikan contoh berikut:

Pwd = input("masukkan password : ")

If pwd == "rahasia":

print("password benar")

Else:

print("password salah")

**Cara kerja if-else**

* Pertama, kondisi setelah if akan dievaluasi nilainya apakah true atau false,
* Jika kondisi bernilai true, maka aksi yang berada di bawah if akan dijalankan, lalu if-else statement selesai.
* Jika kondisi bernilai false, maka aksi di bawah if akan diskip, dan yang dijalankan adalah aksi di bawah else, lalu if-else statement selesai.

Perhatikan bahwa, pada statement if-else **pasti ada salah satu aksi** yang dijalankan. Sementara jika **tanpa else** ada kemungkinan **tidak ada aksi** yang dijalankan dari statement if.

### Statement elif

Adanya else telah membuat program mampu memilih antara **2 buah aksi**, namun di dunia nyata tak jarang ada lebih dari 2 pilihan aksi. Misalnya, contoh menu makanan kita sebelumnya terdapat 3 jenis menu. Contoh lain, kontrol game komputer sederhana yang bisa menggerakkan pemain ke 4 pilihan arah (i.e., maju, mundur, kiri, kanan).

Untuk mengakomodasi hal tersebut, python menyediakan keyword lainnya yaitu elif yang bisa diintegrasikan dengan statement if.

**Cara kerja elif**

Elif adalah singkatan dari else if, bedanya dengan else adalah elif tidak serta merta akan menjalankan aksinya jika kondisi if di atasnya bernilai false, melainkan ia akan melakukan pengecekan tambahan pada **kondisi kedua** yang berada setelah elif. Jika **kondisi kedua** bernilai true, barulan aksi elif dijalankan.

Bentuk paling simple dari elif adalah sebagai berikut:

If kondisi1:

aksi1

Elif kondisi2:

aksi2

Namun jika ada banyak kondisi, maka kita bisa menggunakan beberapa elif atau bisa ditambahkan else di akhir jika ada aksi default.

If kondisi1:

aksi1

Elif kondisi2:

aksi2

Elif kondisi3:

aksi3

Elif kondisi4:

aksi4

Else:

aksidefault

Perhatikan dan jalankan kode berikut ini, lalu ubahlah nilai variable menu ke 2, 3, dan 4 dan jalankan lagi “untuk memahami efek penggunaan elif:

Menu = int(input("pilih menu : "))

If menu == 1:

print("nasi goreng: rp. 15.000")

Elif menu == 2:

print("ayam bakar: rp. 20.000")

Elif menu == 3:

print("mie rebus: rp. 10.000")

Jika menu = 1, maka kode di atas akan menghasilkan nasi goreng: rp. 15.000. Jika menu = 2, ayam bakar: rp. 20.000 akan tampil di layar. Namun jika menu diassign dengan **selain 1, 2, atau 3**, maka tidak ada aksi yang dieksekusi.

Pada statement if, program selalu **mulai memeriksa dari kondisi paling atas**. Lalu jika false akan berpindah ke elif di bawahnya, dan sterusnya. Statement if akan **selesai** jika:

* Ada **kondisi yang bernilai true**, maka aksinya dijalankan, lalu keluar dari statement if.
* Atau **semua kondisi telah diperiksa**.

Perhatikan contoh kode berikut:

Menu = int(input("pilih menu : "))

If menu == 1:

print("nasi goreng: rp. 15.000")

Elif menu == 1: *#kondisi sama persis dengan kondisi pertama*

print("ayam bakar: rp. 20.000")

Elif menu == 3:

print("mie rebus: rp. 10.000")

Jika kode di atas dieksekusi, terlihat hanya **aksi1** saja yang dijalankan, padahal kondisi kedua juga bernilai true, namun **aksi2** tidak dijalankan. Hal ini dikarenakan, ketika kondisi pertama bernilai true, maka program akan mengeksekusi aksi1, lalu setelahnya langsung keluar dari statement if **tanpa memeriksa kondisi di bawahnya**.

**Aksi default**

Dalam merumuskan kondisi-kondisi di statement if, kita perlu memperhatikan **semua kemungkinan** nilai variable yang berada di kondisi. Hal ini untuk memastikan agar program kita selalu bisa memberikan respon yang tepat.

Seperti pada contoh kode menu di atas, ketika user menginputkan menu selain 1, 2, atau 3, maka program tidak memberikan respon apapun. Meskipun tidak membuat error, ada kalanya kita ingin agar program memberikan informasi ke user tentang nilai yang valid.

Untuk menangani nilai variable selain yang sudah ditangani oleh kondisi-kondisi pada if, kita bisa menambahkan else di akhir statement if.

Sebagai contoh, perhatikan kode berikut:

Menu = 4

If menu == 1:

print("nasi goreng: rp. 15.000")

Elif menu == 2:

print("ayam bakar: rp. 20.000")

Elif menu == 3:

print("mie rebus: rp. 10.000")

Else:

print("menu tidak valid")

Dengan penambahan else di akhir, maka kita bisa memberikan sebuah aksi default yang akan dijalankan ketika semua kondisi if di atasnya bernilai false.

Rangkaian if

Perlu diingat bahwa satu rangkaian statement if:

• Dimulai dari if

• Lalu secara opsional diikuti satu atau lebih elif

• Lalu secara opsional diakhiri oleh else.

Urutan ini tidak bisa diubah, jika diubah akan menyebabkan error.

Ubahlah urutan if-elif-else berikut dan perhatikan error yang muncul.

Kodemk = "a1"

Else:

print("kode tidak valid")

Elif kodemk == "a2":

print("kalkulus")

Elif kodemk == "a3":

print("pengenalan pemrograman")

If kodemk == "a1":

print("bahasa indonesia")

If independen¶

Jika ada statement if kemudian diikuti if yang lain yang memiliki indentasi yang sama, maka kedua if tersebut bukanlah satu rangkaian alias independen. Konsekuensi if independen adalah, apapun nilai kondisi (true atau false) pada if pertama, program tetap akan mengevaluasi kondisi pada if kedua, sehingga memungkinkan aksi pada masing-masing if akan dijalankan semuanya.

Perhatikan contoh berikut:

Var = 15

If var % 3 == 0:

print("kelipatan 3")

If var % 5 == 0:

print("kelipatan 5")

Meskipun kondisi pertama sudah bernilai true dan aksi pertama dijalankan, program tetap saja mengevaluasi kondisi kedua dan menjalankan aksi kedua.

Block statements-indentasi¶

Dari contoh-contoh sebelumnya, kita lihat terdapat indentasi (beberapa spasi dari kiri yang membuat kode menjorok ke dalam) pada penulisan statement aksi. Indentasi tersebut menentukan bahwa aksi tersebut merupakan bagian dari statement if.

Seandainya indentasinya dihapus, apa yang terjadi?

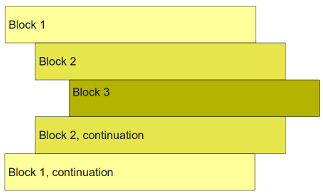
Bil = 4

If (bil > 0):

Print("bilangan positif")

Maka akan terjadi error indentationerror: expected an indented block. Artinya python mengharapkan ada block di bawah statement if yang terindentasi.

**Apa itu block (statement)?**



Block statement adalah **urutan satu atau lebih statement** yang bertindak seperti satu kesatuan, sehingga jika satu statement dijalankan, statement lainnya di block yang sama juga akan dijalankan. Block pada python ditandai dengan **indentasi**. Seperti pada gambar di atas, **block 1** merupakan keseluruhan statement pada program, sementara **block 2** merupakan bagian dari **block 1** dan seterusnya.

**Perhatikan contoh berikut untuk memahami lebih lanjut tentang block.**

Bil = 10 #1 block1

If bil >= 0: #2 block1

print("positif") #3 block2a

if bil % 2 == 0: #4 block2a

print("genap") #5 block3a

else: #6 block2a-lanjutan

print("ganjil") #7 block3b

Else: #8 block1-lanjutan

print("bilangan") #9 block2b

print("negatif") #10 block2b

Print("selesai") #11 block1-lanjutan

Suatu block dimulai dengan indentasi dan akan berakhir ketika menemukan statement yang lebih kurang indentasinya. Contohnya, block2a dimulai pada baris ke-3 dan berakhir pada baris ke-7, karena baris ke-8 memiliki indentasi yang lebih sedikit. Sementara itu, jika menemukan statement dengan indentasi lebih dalam (block baru), maka block tetap berlanjut.

**Statement yang mengharapkan block**

Perlu diperhatikan bahwa selain statement if, ada beberapa statement lagi (e.g., looping, function) yang mengharuskan ada block baru yang lebih dalam di bawahnya , dan semua statement yang membutuhkan block tersebut diakhiri dengan : (titik dua).

**Multi-aksi**

Pada contoh-contoh sebelumnya kita selalu melihat bahwa aksi yang ada pada statement if berjumlah 1 saja, namun setelah kita tau bahwa if mengharapkan sebuah block, dan block artinya satu atau lebih statement, maka tentu kita bisa menuliskan banyak aksi dengan indentasi yang sama di bawah statement if, sebagaimana contoh pada baris ke-9 dan ke-10 pada kode di atas.

**Block pada bahasa pemrograman lainnya**

Tidak semua bahasa pemrograman menggunakan indentasi sebagai penanda block, bahkan lebih banyak yang menggunakan { dan } (kurung kurawal) sebagai pembuka dan penutup block, seperti pada bahasa c, java, dan javascript. Meskipun sering terlihat ada indentasi di bahasa-bahasa tersebut, namun hal tersebut bukanlah sebuah keharusan melainkan style penulisan kode saja.

**Pep 8-indentation**

Python memperbolehkan anda menggunakan jumlah spasi atau tab berapapun untuk menandakan indentasi. Namun, berdasarkan pep 8 maka direkomendasikan untuk menggunakan 4 spasi untuk setiap level indentasi. Sedangkan penggunaan tab tidak disarankan.

**Mixing spasi dan tab**

Python tidak memperbolehkan anda menggabungkan penggunaan spasi dan tab untuk identasi, sehingga konsistensi sangat diperlukan. Banyak editor sudah secara otomatis mengkonversi tab menjadi spasi, sehingga seolah-olah error tidak terjadi. Namun jika anda coba ketikkan ulang kode berikut ini di python shell, anda akan menemukan error: syntaxerror: inconsistent use of tabs and spaces in indentation.

If true:

print("4 spasi") #indentasi dengan 4 spasi

print("tab") #indentasi dengan tab

**Nested if-else**

Terkadang kita membutuhkan beberapa kali pengecekan kondisi sebelum bisa menentukan suatu aksi yang akan dieksekusi oleh program. Sebagai contoh, misalkan sebuah perusahaan akan memberikan bonus kepada karyawan dengan kriteria sebagai berikut:

• Bonus diberikan hanya untuk karyawan dengan masa kerja >=5 tahun (karyawan baru tidak mendapatkan bonus)

• Karyawan dengan umur > 50 tahun mendapat bonus 20%, selainnya akan mendapat bonus 10%

Untuk menghitung bonus yang didapatkan setiap karyawan, maka setelah kita mendapatkan input berupa masa\_kerja dan umur, maka kita perlu melakukan 2 kali pengecekan kondisi, 1) apakah masa\_kerja >= 5 tahun? Jika ya, maka kita akan lanjutkan pengecekan 2) apakah umur > 50 tahun?.

Pengecekan kondisi bertingkat tersebut disebut nested if-else. Perhatikan implementasinya berikut ini:

Masa\_kerja = int(input("masa kerja : "))

Umur = int(input("umur : "))

**If** masa\_kerja >= 5:

**if** umur >50:

print("anda mendapatkan bonus 20%")

**else:**

print("anda mendapatkan bonus 10%")

**Else:**

print("maaf anda belum mendapatkan bonus")

Tentu saja jumlah nested if-else tersebut tidak terbatas, sebagai contoh kita bisa mempunyai 3 tingkatan yaitu if di dalam if di dalam if seperti kode di bawah ini:

Bil1 = int(input("bilangan ke-1 : "))

Bil2 = int(input("bilangan ke-2 : "))

Bil3 = int(input("bilangan ke-3 : "))

**If** bil1 > bil2:

**if** bil2 > bil3:

print(bil1, bil2, bil3)

**else:**

**if** bil3 > bil1:

print(bil3, bil1, bil2)

**else:**

print(bil1, bil3, bil2)

**Else:**

**if** bil1 > bil3:

print(bil2, bil1, bil3)

**else:**

**if** bil3 > bil2:

print(bil3, bil2, bil1)

**else:**

print(bil2, bil3, bil1)

Bilangan ke-1 : 1

Bilangan ke-2 : 2

Bilangan ke-3 : 3

3 2 1

**Struktur**

**1. Penyataan if**

Pernyataan if mengeksekusi beberapa kode jika satu syarat benar.

Struktur Sintaks:

<?php

if (kondisi) {

//kode yang akan dieksekusi jika kondisinya true;

}

?>

Contoh :

Menghasilkan output "Semoga harimu menyenangkan!" jika waktu saat ini (JAM) kurang dari 18:

<?php

$t = date("H");

if ($t < "20") {

echo "Semoga harimu menyenangkan!";

}

?>

**2. Pernyataan if ... else**

Pernyataan if ... else akan mengeksekusi beberapa kode jika suatu kondisi benar dan kode lain jika kondisi itu salah.

Struktur Sintaks:

<?php

if (kondisi) {

//kode yang akan dieksekusi jika kondisi is true;

} else {

//kode yang akan dieksekusi jika kondisi is false;

}

?>

Contoh :

Menghasilkan output "Semoga harimu menyenangkan!" jika waktu saat ini kurang dari 18, dan "Selamat malam!" jika tidak:

<?php

$t = date("H");

if ($t < "18") {

echo "Semoga harimu menyenangkan!";

} else {

echo "Selamat malam!";

}

?>

**3. Pernyataan if ... elseif ... else**

Pernyataan if ... elseif ... else mengeksekusi kode yang berbeda untuk lebih dari dua syarat.

Struktur Sintaks:

<?php

if (kondisi) {

kode yang akan dieksekusi jika kondisi bernilai true;

} elseif (kondisi) {

//kode yang akan dieksekusi jika kondisi pertama bernilai false dan kondisi ini

bernilai true;

} else {

//kode yang akan dieksekusi jika semua kondisis bernilai false;

}

?>

Contoh:

Menghasilkan output "Selamat pagi!" jika waktu saat ini kurang dari 10, dan "Semoga harimu menyenangkan!" jika waktu saat ini kurang dari 20. Jika tidak maka akan menampilkan "Selamat malam!":

<?php

$t = date("H");

if ($t < "10") {

echo "Have a good morning!";

} elseif ($t < "20") {

echo "Have a good day!";

} else {

echo "Have a good night!";

}

?>

**4. Pernyataan switch**

Pernyataan switch digunakan untuk melakukan tindakan yang berbeda berdasarkan kondisi yang berbeda.

Pernyataan switch PHP Gunakan untuk memilih salah satu dari banyak blok kode yang akan dieksekusi.

**Struktur Sintaks:**

<?php

switch (n) {

case label1:

//kode yang akan dieksekusi jika n=label1;

break;

case label2

//kode yang akan dieksekusi jika n=label2;

break;

case label3:

//kode yang akan dieksekusi jika n=label3;

break;

...

default:

//kode yang akan dieksekusi jika n berbeda dari semua labels;

}

?>

**Cara kerjanya seperti ini:**

Pertama kita memiliki ekspresi tunggal n (paling sering variabel), yang dievaluasi sekali. Nilai ekspresi kemudian dibandingkan dengan nilai untuk setiap kasus dalam struktur. Jika ada kecocokan, blok kode yang terkait dengan kasing dieksekusi.

Gunakan istirahat untuk mencegah kode dari berjalan ke kasus berikutnya secara otomatis. Pernyataan default digunakan jika tidak ada kecocokan yang ditemukan.

**Contoh:**

<?php

$favcolor = "merah";

switch ($favcolor) {

case "merah":

echo "Warna favorit kamu adalah merah!";

break;

case "biru";

echo "Warna favorit kamu adalah biru!";

break;

case "hijau":

echo "Warna favorit kamu adalah hijau!";

break;

default:

echo "Warna favorit kamu adalah neither merah, biru, atau hijau!";

}

?>

**D. Jenis-Jenis IF, yaitu :**

***1. If Satu Buah Kondisi***

IF satu buah kondisi digunakan ketika anda akan membuat program yang didalamnya terdapat pemilihan/pengkondisian/penyeleksian untuk satu buah kondisi. Berikut gambar IF untuk satu buah kondisi,bentuk umum :

if (kondisi) {

statement ";

} else {

Statement\_ketika kondisi tidak terpenuhi;

}

**Contoh aplikasi untuk menentukan bilangan positif :**

# include <iostream>

using namespace std;

main() {

//Deklarasikan Variabel

int bil;

//Inputkan Bilangan yang ingin diperiksa

cout << "input Bilangan :";

// Merespon proses inputan (mengambil/membaca nilai bil)

cin >> bil;

if (bil > 0) {

cout << "Bilangan Positif";

} else {

cout << "Bukan Bilangan Positif";

}

}

***2. IF Dua Buah Kondisi***

IF dua buah kondisi digunakan ketika anda akan membuat program yang didalamnya terdapat pemilihan/ pengkondisian/penyeleksian untuk lebih dari satu buah kondisi. kondisi. Berikut gambar IF untuk dua buah kondisi :

if (kondisi1) {

Statement1 & #x22;; //jalankan statemen jika kondisi 1

terpenuhi

} else if (kondisi2) {

Statement2 & #x22;; //jalankan statemen jika kondisi 2

terpenuhi

} else {

statement\_kondisi\_tidak\_terpenuhi;

}

Contoh program untuk menentukan bilangan positif dan negatif :

# include <iostream>

using namespace std;

main() {

//Deklarasikan Variabel

int bil;

//Inputkan Bilangan yang ingin diperiksa

cout << "Input Bilangan :";

// Merespon proses inputan (mengambil/membaca nilai bil)

cin >> bil;

if (bil > 0) {

cout << "Bilangan Positif";

} else if (bil < 0) {

cout << "Bilangan Negatif";

} else {

cout << "Bukan Bilangan Positif & Negatif";

}

}

***3. If  Lebih Dari Dua Buah Kondisi***

IF lebih dari dua buah kondisi digunakan ketika anda akan membuat program yang didalamnya terdapat pemilihan/ pengkondisian/penyeleksian untuk lebih dari dua buah kondisi. Berikut gambar IF untuk lebih dari dua buah kondisi :

if (kondisi1) {

statement1 "; //jalankan statemen jika kondisi 1 terpenuhi

} else if (kondisi2) {

statement2 "; //jalankan statemen jika kondisi 2 terpenuhi

} else if (kondisi3) {

statement3 "; //jalankan statemen jika kondisi 3 terpenuhi

} else {

//jalankan statemen jika semua kondisi tidak terpenuhi

statement\_n ";

}

**Contoh :**

#include <iostream>

using namespace std;

main() {

//Deklarasikan Variabel

int bil;

//Inputkan Bilangan

cout << "Input Angka :";

// Merespon proses inputan (mengambil/membaca nilai bil)

cin >> bil;

if (bil == 1) {

cout << "Anda Menginput Angka Satu";  
 } else if (bil == 2) {

cout << "Anda Menginput Angka Dua";  
 } else if (bil == 3) {

cout << "Anda Menginput Angka Tiga";  
 } else {

cout << "Hanya menghasilkan output 1..3 saja ";

}

}

***4. If Bersarang***

IF Bersarang merupakan IF didalam IF, digunakan ketika anda akan membuat program yang didalamnya terdapat pemilihan/pengkondisian/penyeleksian dan didalam pemilihan/pengkondisian/penyeleksian tersebut terdapat pemilihan/ pengkondisian/penyeleksian lagi. Berikut algoritma untuk If Bersarang :

if (kondisi1) {

if (kondisi1) {

statement1;

} else if (kondisi2) {

statement2;

} else {

statement\_n; //ketika semua kondisi tidak terpenuhi

}

} else if (kondisi2) {

if (kondisi1) {

statement1;

} else if (kondisi2) {

statement2;

} else {

statement\_n; //ketika semua kondisi tidak terpenuhi

} else {

//jalankan statemen jika semua kondisi tidak terpenuhi

statement\_n ";

}

}

Contoh program IF Bersarang :

#include <iostream>

using namespace std;

main() {

int pil;

cout << "Input Pilihan [1..2] :";

cin >> pil;

if (pil == 1) {

cout << "Makanan" << endl;

cout << "1. Mie Goreng " << endl;

cout << "2. Mie Rebus " << endl;

cout << "3. Nasi Goreng" << endl;

cout << "Masukan Inputan [1..4] :";

cin >> pil;

if (pil == 1) {

cout << "Anda Memilih Mie Goreng";

} else if (pil == 2) {

cout << "Anda Memilih Mie Rebus";

} else if (pil == 3) {

cout << "Anda Memilih Nasi Goreng";

} else {

cout << "Pilihan Hanya Ada Sampai 4 Saja...";

}

} else if (pil == 2) {

cout << "Minuman" << endl;

} else {

cout << "Piihan Anda Salah!!!";

}

}

Hasil :

**Switch Case**

Selain menggunakan IF, bahasa C++ juga menyediakan penggunaan perintah lainnya, yaitu switch case. Hampir sama dengan penggunaan if pada umumnya karena digunakan ketika anda ingin membuat program yang mengandung pemilihan/pengkondisian/pemilihan. Perbedaan yang mendasar pada penggunaan switch case adalah cangkupan tipe data yang digunakan relatife sempit karena hanya dapat memeriksa tipe data yang bertipe integer dan char saja. Berikut algoritma penggunaan switch case :

switch(ekspresi) {

case nilai\_1: statement\_jika\_nilai1\_terpilih;

break;

case nilai\_ 2: statement\_jika\_nilai2\_terpilih;

break;

case nilai\_N: statement\_jika\_nilaiN\_terpilih;

break;

default: statement\_jika\_semua\_tidak\_terpenuhi;

break;

}

**Perlu Di Ingat**:

*Break digunakan untuk keluar dari kondisional pada pilihan case jika nilai konstan terpenuhi. Default dijalankan apabila seluruh kondisi pada pilihan case tidak ada yang terpenuhi. Default merupakan  kondisi optional.*

***Contoh 1 program penggunaan case :***

Menentukan nama hari dalam satu minggu berdasarkan inputan angka hari

#include <iostream>

using namespace std;

main() {

//Deklarasikan Variabel

int AngkaHari;

//Inputkan Angka Hari

cout << "Input Angka Hari :";

cin >> AngkaHari;

switch (AngkaHari) {

case 1:

cout <<"Angka Hari 1 merupakan hari SENIN";

break;

case 2:

cout <<"Angka Hari 2 merupakan hari SELASA";

break;

case 3:

cout <<"Angka Hari 3 merupakan hari RABU";

break;

case 4:

cout <<"Angka Hari 4 merupakan hari KAMIS";

break;

case 5:

cout <<"Angka Hari 5 merupakan hari JUMAT";

break;

case 6:

cout <<"Angka Hari 6 merupakan hari SABTU";

break;

case 7:

cout <<"Angka Hari 7 merupakan hari MINGGU";

break;

default:

cout <<"Angka Hari yang anda Input SALAH";

break;

}

}

Hasil :

***Contoh 2 Menentukan Harga Berdasarkan Paket yang dipilih :***

#include <iostream>

using namespace std;

main() {

//Deklarasikan Variabel

char pil;

cout << "Pilih Kode Pahe berdasarkan pilihan berikut :" << endl;

cout << "A : Ayam + Nasi + Salad + Coca-Cola " << endl;

cout << "B : Ayam + Nasi + Coca-Cola " << endl;

cout << "C : Ayam + Nasi + Frestea " << endl;

cout << "Input Pilihan Paket [A/B/C] :";

cin >> pil;

switch (pil) {

case 'A':

cout << "Anda Memilih Paket A : Ayam + Nasi + Salad + Coca-Cola" << endl;

cout << "Harga Paket A adalah : Rp. 25.000,-";

break;

case 'B':

cout << "Anda Memilih Paket A : Ayam + Nasi + Coca-Cola" << endl;

cout << "Harga Paket A adalah : Rp. 20.000,-";

break;

case 'C':

cout << "Anda Memilih Paket C : Ayam + Nasi + Frestea" << endl;

cout << "Harga Paket A adalah : Rp. 18.000,-";

break;

default:

cout << "PILIHAN SALAH";

break;

}

**BAB III**

**PENUTUP**

Demikianlah Makalah ini saya buat dengan sebaik-baiknya. Terima kasih buat Orang tua yang selalu mendukung saya baik materil maupun moril,dan terima kasih untuk abang dan kakak assisten laboratorium paket applikasi, juga terima kasih buat kawan-kawan yang telah membantu dalam mengerjakan makalah ini.semoga makalah ini dapat berguna buat kita semua dan buat angkatan ke depannya kelak.

lebih dan kurangnya saya mohon maaf, kepada abang dan kakak saya ucapkan terima kasih

Wassalammu’alaikum Wr.Wb.

DAFTAR PUSTAKA

1. <http://belajarhardwaredipesat.blogspot.com/2019/08/hai-hai-guys-welcome-to-my-blog-2-i.html>
2. <https://id.wikipedia.org/wiki/C_(bahasa_pemrograman)#:~:text=Bahasa%20pemrograman%20C%20adalah%20bahasa,mencerminkan%20kemampuan%20CPU%20yang%20ditargetkan>
3. <https://www.novriadi.com/apa-itu-pengertian-bahasa-c/>
4. <https://saidalfaraby.github.io/pengenalan-pemrograman/3-Pengkondisian/Modul-3.html>
5. <https://www.anbidev.com/c-if-statements/>
6. https://www.duniailkom.com/tutorial-belajar-c-percabangan-kondisi-if-else-if-bahasa-c/