

**LAPORAN PRAKTIKUM**

**ALGORITMA DAN PEMROGRAMAN**

**DALAM BAHASA C++ DENGAN CODEBLOCKS**

**Materi Praktikum : Struktur Pemilihan**

Disusun oleh:

Nama = Merlinda Lavenia

NIM = 4111421019

Program studi = Matematika

**JURUSAN MATEMATIKA**

**UNIVERSITAS NEGERI SEMARANG**

**2021**

**Materi Praktikum : Struktur Pemilihan**

**(1 Pertemuan : 1 x 2 SKS)**

Setelah mempelajari materi pada modul 5, silahkan menjawab pertanyaan berikut ini sesuai dengan petunjuk jawaban:

1. Apakah kondisi dalam **IF** harus ditulis dalam tanda kurung? Bagaimana kalau tidak diberi tanda kurung apa yang akan terjadi?

**Jawab:** Ya, Penulisan kondisi dalam **IF** harus ditulis dalam tanda kurung pada CodeBlocks sebab **IF** merupakansuatu fungsi dengan bentuk umum **IF(kondisi)**  jika kondisi dalam **IF** tidak ditulis dalam tanda kurung makaprogram tidak dapat dijalankan/dirun.

**Contoh Program Dengan Menggunakan IF(CodeBlocks):**

|  |  |
| --- | --- |
| **Tampilan Program** | **Hasil Run** |
|  |  |
| **Penjelasan:** Pada program di atas, kita menuliskan fungsi IF dengan menggunakan tanda kurung setelah **IF** kemudian tambahkan kondisi atau syarat di dalam tanda kurung tersebut dengan menggunakan operasi seperti <,>,<=,>=, dll. Kemudian, ketik cout<< “pernyataan” untuk mengoutput keluaran berupa pernyataan jika kondisi dipenuhi. Sehingga saat di run, program akan memproses keluaran berupa pernyataan sesuai dengan kondisi yang diminta. Seperti pada program di atas, jika kita menginputkan nilai>=85 maka didapatkan keluaran berupa pernyataan “SELAMAT, ANDA DINYATAKAN LULUS” dan jika kita menginput nilai <85 maka tidak ada hasil keluaran atau output. | |

**Contoh Program Dengan Menggunakan IF (Flowchart):**

|  |  |
| --- | --- |
| **Tampilan Program** | **Hasil Run** |
|  |  |
| **Penjelasan:** Pada program di atas, sebenarnya sama saja hanya di atas menggunakan Flowchart. Jika pada flowchart, **IF** tidak perlu ditulis dalam tanda kurung dan tidak perlu menulis kembali kata IF, sehingga kita cukup langsung menuliskan kondisi pada tabel **IF** pada flowchart. Kemudian tambahkan output berupa pernyataan jika hasil bernilai True dan output jika hasil bernilai False. Sehingga saat di run program akan memproses dengan mengeluarkan pernyataan sesuai dengan syarat/kondisi yang diminta. Seperti pada program di atas, jika kita menginputkan nilai>=85 maka didapatkan output berupa pernyataan “SELAMAT, ANDA DINYATAKAN LULUS” sedangkan jika kita menginput nilai <85 maka tidak ada hasil keluaran atau output karena tidak ditambahkan pernyataan pada hasil keluaran jika nilai<85. | |

**Jika kondisi pada IF tidak ditulis dalam tanda kurung:**

|  |  |
| --- | --- |
| **Tampilan Program** | **Hasil Run** |
|  |  |
| **Penjelasan:** Program di atas tidak dapat dijalankan sebab IF tidak diketik menggunakan tanda kurung. | |

1. Dapatkah pernyataan yang dijalankan oleh **IF** lebih dari satu pernyataan?.

**Jawab:** Bisa, karena pernyataan pada fungsi **IF** dapat berupa pernyataan tunggal, majemuk, maupun pernyataan kosong. Sehingga setelah kita menuliskan **if(kondisi)** maka kita bebas menentukan pernyataan yang akan dikeluarkan jika kondisi **IF** terpenuhi ataupun pernyataan jika kondisi pada **IF** tidak terpenuhi. Selain itu, untuk penulisan **IF** kita dapat menggunakan percabangan struktur **IF** baik dalam satu kondisi, dua kondisi/if-else, maupun IF dengan banyak kondisi majemuk pada C++. Atau dengan menggabungkan fungsi **IF** dengan dua tes logika sehingga kita dapat menambahkan fungsi **AND**  pada fungsi  **IF.**

**Contoh IF Satu Kondisi Dua Pernyataan (if):**

|  |  |
| --- | --- |
| **Tampilan Program** | **Hasil Run** |
| N    \ |  |
| **Penjelasan:** Pada program di atas, merupakan contoh penggunaan satu kondisi namun terdapat keluaran berupa pernyataan mejemuk. Pada program di atas, jika kita memasukkan total belanja 200000 maka akan didapatkan hasil output berupa kedua pernyataan yaitu “SELAMAT, ANDA MENDAPATKAN SATU HADIAH GRATIS” dan “Terimakasih Atas Kunjungan Anda”. Jika kita memasukan total belanja<200000 maka hanya akan didapatkan satu keluaran yaitu pernyataan “Terimakasih Atas Kunjungan Anda”. | |

**Contoh IF Satu Kondisi Dengan Dua Pernyataan Namun tidak muncul bersamaan dalam satu keluaran(if-else):**

|  |  |
| --- | --- |
| **Tampilan Program** | **Hasil Run** |
| N |  |
| **Program:** Mengecek Kadar Gula darah Sewaktu.  **Penjelasan:** Pada program di atas terdapat dua pernyataan di mana pernyataan 1 akan dijalankan jika kondisi terpenuhi dan pernyataan 2 akan muncul jika kondisi tidak terpenuhi. Artinya tetap ada keluaran yang muncul jika satu kondisi bernilai salah. Seperti pada hasil run di atas, jika kita menginputkan kadar gula darah (mg/dL) <200 maka akan didapatkan output berupa pernyataan “TIDAK NORMAL” dan jika kita menginputkan kadar gula darah 200 maka akan didapatkan output berupa pernyataan ”NORMAL”. | |

**Contoh Program Lebih dari Dua Kondisi (if-else bertingkat):**

|  |  |
| --- | --- |
| **Tampilan Program** | **Hasil Run** |
| N  else |  |
| **Penjelasan:** Pada program, jika terdapat lebih dari dua kondisi maka kita perlu menggunakan if-else bertingkat seperti pada program di atas. Pada program di atas, akan selalu terdapat dua pernyataan yang akan muncul sebagai hasil output untuk tiap kondisi kecuali untuk kondisi TotalBelanja<200000 maka hanya ada satu keluaran sebab satunya merupakan pernyataan kosong. Di mana, satu pernyataan yaitu pernyataan “Terimakasih Atas Kunjungan Anda” merupakan hasil keluaran akan selalu ditampilkan dalam kondisi apapun dan berlaku untuk semua kondisi. | |

1. Apa saja operator yang dapat digunakan dalam perintah **IF**? Berikan contoh programnya

**Jawab:** Operator relasi dan operator Logika.

* Operator Relasi.

Operator relasi merupakan operasi yang digunakan untuk membandingkan antara dua nilai.

Contoh operasi Relasi:

* Operasi Senilai direpresentasikan dengan ==
* Operasi tak senilai direpresentasikan dengan !=
* Operasi lebih dari direpresenntasikan dengan >
* Operasi lebih dari sama dengan direpresentasikan dengan >=
* Operasi kurang dari direpresentasikan dengan <
* Operasi kurang dari sama dengan direpresentasikan dengan <=
* Operator Logika

Merupakan operakor yang digunakan untuk menggabungkan dua atau lebih kondisi melalui suatu pertimbangan.

* Operator AND direpresentasikan dengan &&
* Operator OR direpresentasikan dengan | |
* Operator NOT direpresentasikan dengan !

**Contoh Operator Relasi(== dan !=):**

|  |  |
| --- | --- |
| **Tampilan Program** | **Hasil Run** |
|  |  |
| **Penjelasan:** Pada program di atas, jika kita memasukkan tinggi badan a dan tinggi bandan b, maka akan didapatkan keluaran berupa “Tinggi Badan A sama dengan Tinggi Bada B” jika kondisi pertama terpenuhi, jika kondisi kedua yang terpenuhi maka akan didapatkan hasil output berupa “Tinggi Badan A tidak sama dengan Tinggi Badan B”. Seperti yang tampak pada hasil run di atas, dimasukkan data berupa tinggi badan a 170 dan tinggi badan b 163 maka hasil outputnya adalah “Tinggi Badan A tidak sama dengan Tinggi Badan B”. | |

**Contoh Operator Logika (OR):**

|  |  |
| --- | --- |
| **Tampilan Program** | **Hasil Run** |
|  | Tampilan Hasil Run jika Stok Barang Habis:  Tampilan Hasil Run jika Stok Barang tersedia: |
| **Penjelasan:** Pada program di atas, jika diinput stok barang lebih dari 1 atau tidak sama dengan 0 maka program akan memproses dengan mengeluarkan aksi berupa hasil output “Barang Masih Tersedia” dan jika diinput stok barang =0 atau kurang dari 1 maka program juga akan mengeluarkan aksi berupa hasil output “Barang Habis”. | |

1. Apakah kondisi **A==B** boleh disederhanakan menjadi **A=B** dalam struktur pemilihan?

**Jawab:** Tidak, dikarenakan jika **A==B** diubah menjadi **A=B** maka akan didapatkah hasil run yang berbeda jika dimasukkan angka A dan B yang berbeda(angka A tidak sama dengan B). Selain itu, penggunaan **==** dengan **=** berbeda, **=** digunakan untuk memberi nilai sedangkan **==** digunakan untuk membandingkan. Jadi, **A==B tidak dapat disederhanakan menjadi A=B.** Berikut adalah contoh tampilan program.

**Contoh program A==B:**

|  |  |
| --- | --- |
| **Tampilan Program** | **Hasil Run** |
| Tampilan Hasil Run jika A tidak sama dengan B: | Tampilan Hasil Run jika A sama dengan B: |
| **Penjelasan:** Pada program di atas, jika digunakan **A==B** untuk kondisi pada **IF** maka jika diinput angka yang sama untuk A dan B, didapatkan hasil output berupa “A sama dengan B” sedangkan jika diinput angka A dan B yang berbeda maka tidak didapatkan hasil keluaran(output) atau keluaran berupa pernyataan kosong. | |

**Yang terjadi jika A==B diubah menjadi A=B:**

|  |  |
| --- | --- |
| **Tampilan Program** | **Hasil Run** |
|  | Tampilan Hasil Run jika A sama dengan B:  Tampilan Hasil Run jika A tidak sama dengan B: |
| **Penjelasan:** Pada program di atas saat **A==B** diubah menjadi **A=B** memiliki perbedaan output yakni pada hasil keluaran yang kedua saat dimasukkan angka A dan B yang berbeda maka seharusnya tidak ada keluaran yang muncul sebab jika A tidak sama dengan B maka itu tidak memenuhi kondisi yang diminta. Namun saat **A==B** diubah menjadi **A=B** output “A sama dengan B” tetap dikeluarkan. Hal ini tentu sangat tidak sesuai dengan kondisi yang diminta yakni jika A tidak sama dengan B seharusnya tidak ada output yang keluar. Sehingga dapat disimpulkan bahwa **A==B** tidak dapat disederhanakan menjadi **A=B**. | |

1. Apakah operator **==** berlaku untuk tipe data String?

Jawab: Ya, sebab operator **==** dapat berlaku untuk semua tipe data.

**Contoh Program menggunakan == pada tipe data String:**

|  |  |
| --- | --- |
| **Tampilan Program** | **Hasil Run** |
|  | Tampilan Hasil Run jika Hobi Aku dan dia sama:      Tampilan Hasil Run jika Hobi Aku dan dia berbeda: |
| **Penjelasan:** Program di atas adalah contoh penggunaan == untuk tipe data **String**. Pada program di atas, jika kita menginput hobi dalam karakter dan jumlah kata yang sama maka program akan memproses dengan menampilkan hasil output “Kami Punya Hobi yang Sama” dan jika diinput hobi yang berbeda untuk aku dan dia maka tidak akan ada hasil output atau keluaran yang ditampilkan meskipun memasukkan hobi dengan jumlah kata yang sama, namun jika karakternya berbeda maka hobi Aku tidak sama dengan hobi dia. **Sehingga dari contoh program di atas, terbukti bahwa == berlaku untuk tipe data String.** | |

1. Bagaimana cara memberikan syarat **“dan”** (kondisi dimana ada dua syarat yang harus dipenuhi) pada struktur pemilihan?

Jawab:Dengan Menambahkan Simbol && di antara dua kondisi(syarat) yang berbeda seperti pada contoh.

**Contoh Program Mengunakan && (AND):**

|  |  |
| --- | --- |
| **Tampilan Program** | **Hasil Run** |
|  |  |
| **Penjelasan:** Cara memberikan syarat **“dan”** adalah dengan mengetikkan **if(kondisi)** seperti biasa, namun jika ada 2 syarat atau kondisi maka dapat ditambahkan simbol **&&** di antara dua syarat/kondisi tersebut sehingga penulisannya menjadi **if(kondisi1&&kondisi2).** Seperti pada program di atas, jika ada syarat HASILAKHIR>=85 dan HASILAKHIR<92 maka penulisannya menjadi **else if(HASILAKHIR>=85&& HASILAKHIR<92)** penambahan kata else karena program di atas merupakan if else bertingkat**.** Jika kita menginput HASILAKHIR maka program akan memproses dengan mengeluarkan aksi berupa hasil keluaran yaitu “ANDA MENDAPAT NILAI A+” dan jika HASILAKHIR 85 dan HASILAKHIR<92 maka program akan memproses dengan mengeluarkan hasil output berupa pernyataan “ANDA MENDAPAT NILAI A” dan seterusnya. | |

7. Apakah pernyataan **SWITCH** bisa menggantikan pernyataan **IF**?

**Jawab:** Ya, **SWITCH** dapat menggantikan **IF** jika dan hanya jika kondisi pada **if** tersebut menggunakan operator logika **==, selain kondisi itu SWITCH tidak dapat menggantikan IF**. Pada **SWITCH**, kondisi hanya dinyatakan dengan bilangan bulat atau karakter/string sedangkan pada **IF** kondisi dapat dinyatakan dengan menggunakan operasi seperti <,>,<=,>=dll. Jika kita menggunakan **SWITCH** maka kita harus menggunakan **BREAK** yang berfungsi untuk mengakhiri statetment. Selain itu, Dengan perintah ini, program percabangan akan semakin mudah dibuat dan dipelajari.

Kapan kita dapat menggunakan **IF** dan **SWITCH**?

Kita dapat menggunakan **IF** jika seleksi memiliki lebih dari 1 kondisi dan nilai. Sedangkan kita menggunakan **SWITCH** jika hanya ada satu kondisi dengan nilai bervariasi.

**Contoh Program yang dapat menggunakan if-else dan SWITCH case:**

**Dengan if-else:**

|  |  |
| --- | --- |
| **Tampilan Program** | **Hasil Run** |
|  |  |
| **Program:** Menentukan Kelompok Usia sesuai rentang usia  **Penjelasan:** Pada program di atas, jika kita menggunakan if-else untuk membuat program di atas, maka jika kita menginput nomor kategori usia antara 1-3 akan didapatkan hasil output sesuai dengan kondisi yang diminta. Seperti pada hasil run di atas, diinput nomor kategori usia yaitu ‘2’ maka didapatkan hasil keluaran berupa “Kelompok Usia Anda: Remaja”. | |

**Dengan SWITCH Case:**

|  |  |
| --- | --- |
| **Tampilan Program** | **Hasil Run** |
|  |  |
| **Program:** Menentukan Kelompok Usia sesuai rentang usia  **Penjelasan:** Program di atas juga dapat dibuat dengan menggunakan **SWITCH CASE** yaitu dengan membuat **CASE**1-3 kemudian ketik statetment yang akan dikeluarkan pada hasil output jika kondisi terpenuhi. Selanjutnya ketik **BREAK** di setiap bagian akhir dari **CASE** untuk mengakhiri satetment. Pada program di atas, jika kita menginput nomor kategori usia antara 1-3 akan didapatkan hasil output sesuai dengan kondisi yang diminta. Seperti pada hasil run di atas, diinput nomor kategori usia yaitu ‘2’ maka didapatkan hasil keluaran berupa “Kelompok Usia Anda: Remaja”. | |

8. Apakah **BREAK** harus diletakkan pada bagian akhir setiap **CASE**, pada struktur pemilihan dengan **SWITCH**?

**Jawab:** Ya, harus, sebab kata kunci **BREAK** digunakan untuk mengakhiri setiap **CASE** sehingga kita harus meletakannya pada bagian akhir setiap **CASE**. Hal ini dimaksudkan agar nilai ekspresi **SWITCH** sama dengan salah satu nilai **CASE** (jika kondisi pemilihan terpenuhi) sehingga interpreter akan berhenti memeriksa nilai-nilai **CASE** selanjutnya.

**Contoh Program menggunakan SWITCH CASE dan peletakan BREAK:**

|  |  |
| --- | --- |
| **Tampilan Program** | **Hasil Run** |
|  |  |
| **Penjelasan:** Pada program di atas, jika kita menginput Nomor urutan bulan sesuai dengan kalennder maka program akan memproses dengan mengeluarkan aksi berupa hasil output nama bulan. Pada hasil run, diinput bullan ke -9 sehingga muncul hasil keluaran berupa “Sekarang Bulan September”. | |