

FASE E

OPERATOR & PERCABANGAN

Informatika

DISAJIKAN OLEH

Johannes Alexander
Putra

DAFTAR ISI

Capaian Pembelajaran

01

Tujuan Pembelajaran

02

Operator

03

Percabangan

08

Daftar Pustaka

16

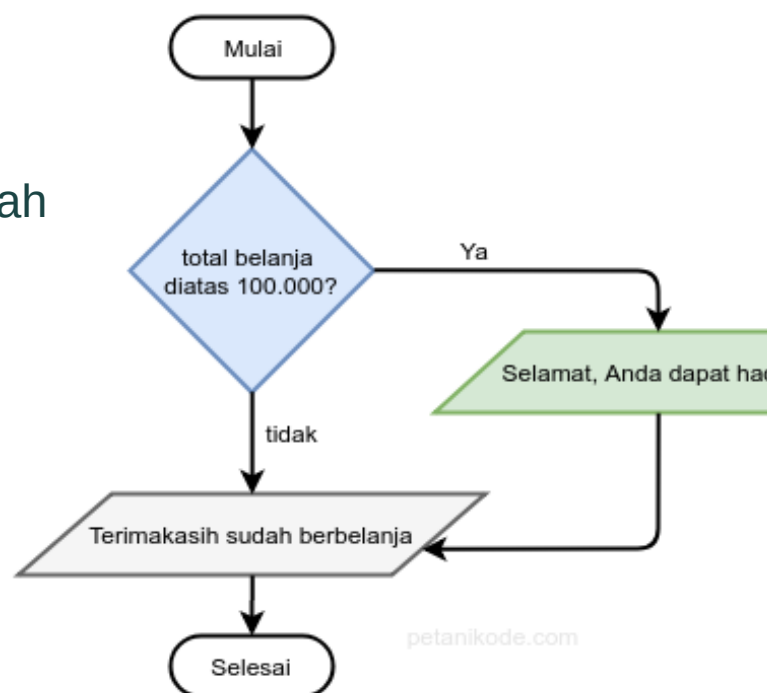
CAPAIAN PEMBELAJARAN

Pada akhir fase E peserta didik mampu melakukan pemrograman terstruktur, antara lain penerapan struktur data yang terdiri dari data statis (array baik dimensi, panjang, tipe data, pengurutan) dan data dinamis (list, stack), penggunaan tipe data, struktur kontrol perulangan dan percabangan pada proyek pengembangan perangkat lunak sederhana dan gim. Pemrograman berorientasi obyek

TUJUAN PEMBELAJARAN

Tujuan Pembelajaran setelah menyelesaikan pertemuan ini adalah siswa mampu:

- Memahami permasalahan yang berkaitan dengan operator
- Menentukan pemecahan masalah dengan operator
- Melakukan implementasi operator untuk suatu permasalahan
- Melakukan evaluasi terhadap penggunaan operator
- Memahami permasalahan yang berkaitan dengan percabangan
- Menentukan pemecahan masalah dengan percabangan
- Melakukan implementasi percabangan untuk suatu permasalahan
- Melakukan evaluasi terhadap penggunaan percabangan



OPERATOR

Operator menurut Sukamto (2018) adalah simbol atau tanda yang jika diletakan pada dua buah operan dapat menghasilkan sebuah hasil, contohnya di dalam operasi matematika di mana tanda '+' diletakan diantara kedua buah nilai akan menghasilkan angka lain hasil penjumlahan kedua bilangam sebelumnya. Terdapat beberapa jenis operator.

PEMBAGIAN OPERATOR

Berikut ini merupakan pembagian operator

Operator	Keterangan	Contoh
Operator Aritmetik	Operator yang biasa digunakan dalam matematika misalnya '+', '-', '/', '%', '*'	2 + 3
Operator Relasi	Operator yang digunakan untuk membandingkan dua buah nilai. Misalnya ==, <, <=, >, >=, !=	A<B
Operator Logika Boolean	Operator yang digunakan untuk mengaitkan dua buah ungkapan kondisi menjadi sebuah kondisi biasanya digunakan untuk melogikakan dua buah atau lebih kondisi contoh : && (AND) , (OR)	Kondisi_1 && Kondisi_2

OPERATOR ARITMATIKA

Operator aritmatika merupakan operator yang paling sering digunakan karena pengolahan data sering berhubungan dengan nilai numerik (Jubilee Enterprise, 2017). Tabel dibawah ini menunjukan operator-operator aritmatika dalam bahasa pemrograman C.

Tanda	Nama	Contoh
*	Operator perkalian	2*3
/	Operator pembagian	2/3
%	Operator sisa pembagian (Operator Modulo)	5 % 3
+	Operator penjumlahan	2 + 3
-	Operator pengurangan	8 - 4

OPERATOR LOGIKA

Operator logika pada dasarnya membuat proses logika menjadi lebih kompleks. Dengan menggunakan operator ini pemrogram akan bisa menggabungkan beberapa kondisi. Contoh dari penggunaan operator ini: jika A dari Bandung DAN nilainya lebih tinggi dari 80, maka diberikan ucapan kelulusan menggunakan bahasa Sunda. Penggunaan kata 'dan' merupakan contoh dari operator logika (Jubilee Enterprise, 2017). Tabel di bawah ini menunjukan operator logika dalam bahasa C (Sukamto, 2018).

Keterangan Operator	Bahasa C															
<p>Operator dan</p> <p>Operator logika yang menyatakan logika dan</p> <p>Keterangan:</p> <p>B : Benar</p> <p>S : Salah</p> <table><tr><th>Kondisi 1</th><th>Kondisi 2</th><th>Hasil</th></tr><tr><td>B</td><td>B</td><td>B</td></tr><tr><td>B</td><td>S</td><td>S</td></tr><tr><td>S</td><td>B</td><td>S</td></tr><tr><td>S</td><td>S</td><td>S</td></tr></table>	Kondisi 1	Kondisi 2	Hasil	B	B	B	B	S	S	S	B	S	S	S	S	<p>Tanda: &&</p> <p>Contoh:</p> <p>(a>b) && (a !=b)</p>
Kondisi 1	Kondisi 2	Hasil														
B	B	B														
B	S	S														
S	B	S														
S	S	S														
<p>Operator atau</p> <p>Operator logika yang menyatakan logika atau</p> <table><tr><th>Kondisi 1</th><th>Kondisi 2</th><th>Hasil</th></tr><tr><td>B</td><td>B</td><td>B</td></tr><tr><td>B</td><td>S</td><td>B</td></tr><tr><td>S</td><td>B</td><td>B</td></tr><tr><td>S</td><td>S</td><td>S</td></tr></table>	Kondisi 1	Kondisi 2	Hasil	B	B	B	B	S	B	S	B	B	S	S	S	<p>Tanda: </p> <p>Contoh:</p> <p>(a>b) (a !=b)</p>
Kondisi 1	Kondisi 2	Hasil														
B	B	B														
B	S	B														
S	B	B														
S	S	S														
<p>Operator Bukan</p> <p>Operator yang menyatakan logika bukan (negasi)</p> <table><tr><th>Kondisi 1</th><th>Hasil</th></tr><tr><td>B</td><td>S</td></tr><tr><td>S</td><td>B</td></tr></table>	Kondisi 1	Hasil	B	S	S	B	<p>Tanda: !</p> <p>Contoh: !(a==b)</p>									
Kondisi 1	Hasil															
B	S															
S	B															

OPERATOR RELATIONAL

Operator relasi atau conditional operator adalah operator yang digunakan untuk membandingkan variable-variable yang umumnya bertipe data angka. Namun anda juga dapat menggunakannya untuk tipe data lainnya (Jubilee Enterprise, 2017). Table dibawah ini menunjukan operator relasi dalam bahasa C (Sukamto, 2018).

Keterangan Operator	Contoh dalam Bahasa C
Operator sama dengan (=)	Tanda: == Contoh: a == b
Operator tidak sama dengan	Tanda: != Contoh: a != b
Operator lebih dari (>)	Tanda: > Contoh a > b
Operator kurang dari (<)	Tanda: < Contoh a<b
Operator lebih dari sama dengan (>=)	Tanda : >= Contoh a>=b
Operator Kurang dari sama dengan (<=)	Tanda : <= Contoh a<=b

PERCABANGAN

Suatu program yang hanya berisi runtutan instruksi biasanya terdapat dalam persoalan sederhana. Persoalan yang lebih kompleks melibatkan analisis berbagai kasus kemungkinan yang terdapat di dalamnya. Untuk setiap kasus terdapat persyaratan yang harus dipenuhi dan aksi apa yang harus dilakukan jika persyaratan dipenuhi. Dengan adanya analisis kasus maka instruksi tidak dikerjakan secara sekuensial lagi akan tetapi melakukan aksi jika persyaratan dipenuhi (Munir & Lidya, 2016).

Percabangan satu Kasus

Percabangan satu kasus dinotasikan menggunakan konstruksi IF-THEN (Jika-Maka) berbentuk:

```
if Kondisi then
    aksi
end if
```

Pada instruksi di atas aksi dilaksanakan jika kondisi bernilai benar (true). Bila kondisi bernilai salah (false), maka tidak ada aksi yang dilakukan. Kata end if ditambahkan untuk mempertegas awal dan akhir dari struktur if-then (Munir & Lidya, 2016).

Percabangan Dua Kasus

Konstruksi IF-THEN sebelumnya hanya menyediakan satu alternatif aksi jika suatu kondisi terpenuhi. Kadang kita perlu memilih untuk melakukan aksi alternatif jika aksi pertama tidak terpenuhi,. Maka dari itu diperlukan notasi algoritma baru. Notasi tersebut adalah IF-THEN-ELSE

```
If kondisi then  
  Aksi1  
else  
  Aksi2  
endif
```

Pernyataan di atas berarti jika aksi1 dikerjakan jika kondisi benar dan aksi2 dikerjakan jika kondisi salah (Munir & Lidya, 2016).

Percabangan Tiga Kasus atau Lebih

Masalah yang memiliki tiga kasus atau lebih dapat dianalisis menggunakan konstruksi IF-THEN-ELSE bertingkat-tingkat (Munir & Lidya, 2016).

```
if kondisi1 then  
  aksi1  
else  
  if kondisi2 then  
    aksi2  
    else  
    aksi3  
  end else  
  end if  
end if
```

Switch/Depend On

Percabangan depend on biasanya digunakan untuk dua kondisi atau lebih bergantung pada nilai sebuah variable, syarat kondisi pada percabangan depend on biasanya hanya sebuah nilai. Jika nilai yang diperiksa memenuhi syarat maka akan dikerjakan.

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int a = 5;
7      switch(a){
8          case 1:
9              cout <<"Angka nya adalah 1";
10             break;
11          case 3:
12              cout <<"Angkanya adalah 3";
13              break;
14          case 5:
15              cout <<"Angkanya adalah 5";
16              break;
17          default:
18              cout <<"bukan 1, 3, maupun 5";
19              break;
20      }
21  }
22
```

Angkanya adalah 5

Keterangan tambahan

Switch pada bahasa pemrograman C++ tidak bisa digunakan untuk string dan float

Harus selalu menggunakan break pada setiap akhir case jika tidak maka akan tetap melaksanakan aksi selanjutnya

Aksi default: adalah aksi yang pasti dikerjakan jika aksi sebelumnya tidak terpenuhi dan/atau aksi sebelumnya tidak menggunakan break

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int a = '5';
7      switch(a){
8          case 1:
9              cout <<"Angka nya adalah 1";
10
11             case 3:
12                 cout <<"Angkanya adalah 3";
13
14             case 5:
15                 cout <<"Angkanya adalah 5";
16
17             default:
18                 cout <<"bukan 1, 3, maupun 5";
19
20         }
21     }
22
```



bukan 1, 3, maupun 5

CONTOH SOAL

1. Harry Potter adalah seekor Cuybara yang sedang belajar sihir. Guru sihir Harry telah mengajarkan bahwa kalimat yang bersifat magical adalah kalimat yang huruf pertama dan huruf terakhirnya sama atau jarak perbedaannya hanya satu. Sebagai contoh, berikut ini adalah contoh kalimat-kalimat yang bersifat magical:

- abra kadabra: huruf pertama (a) dan huruf terakhirnya (a) sama
- lab acadak: huruf pertama (l) dan huruf terakhirnya (k) berjarak 1
- bac: huruf pertama (b) dan huruf terakhirnya (c) berjarak 1

Buatlah sebuah program yang menerima masukan berupa sebuah kalimat dan akan menentukan apakah kalimat tersebut bersifat magical atau tidak!

Spesifikasi Masukan

Masukan program ini adalah sebuah kalimat. Sebuah kalimat hanya dapat mengandung huruf non kapital.

Sebuah kalimat dapat mengandung spasi.

Spesifikasi Keluaran

Keluarkan "magical" bila kalimat tersebut bersifat magical dan "tidak magical" bila sebaliknya.

Contoh Masukan dan Keluaran

No	Masukan	Keluaran yang diharapkan
1	abra kadabra	magical
2	lab acadak	magical
3	dab	tidak magical

Step 1. Memahami Masalah

Memahami masalah: Menentukan apa yang diketahui dan ditanyakan kemudian memberikan keterangan apa yang sudah diketahui cukup untuk menjawab pertanyaan.

Soal tersebut adalah soal untuk mengambil input dari keyboard berupa String dan nantinya dibandingkan karakter pertama dan terakhirnya apakah sama atau apakah selisihnya 1. Jika sama atau selisihnya 1 maka akan mengeluarkan output "magical" dan jika tidak memenuhi maka akan mengeluarkan output tidak magical.

Step 2- Mengidentifikasi Masalah

Merencanakan penyelesaian masalah : Mengidentifikasi masalah dan mencari jalan yang tepat untuk menyelesaikannya

Masalah pada soal ini pada umumnya adalah masalah mengenai perbandingan antara karakter pertama dalam String dan karakter terakhir pada suatu String. Disini untuk mempermudah bisa memakai `charAt` pada Java dan `str.at()` pada C++. Kemudian membandingkan dengan operator `==` atau jika selisihnya 1 tinggal kurangi char terakhir- dengan char pertama hasil yang menyatakan bahwa selisihnya adalah satu adalah jika menghasilkan nilai -1 atau 1

Step 3- Melaksanakan Pemecahan Masalah



```
#include <iostream>

using namespace std;

int main()
{
    string s;
    getline(cin, s);
    int panjang = s.length();
    char pertama = s.at(0);
    char terakhir = s.at(panjang-1);
    if(pertama == terakhir){
        cout << "Magical";

    }else if((pertama - terakhir) == 1)
    {
        cout << "Magical";

    }else if((pertama-terakhir) == -1){
        cout << "Magical";
    }else{
        cout << "Tidak Magical";
    }
}
```


Step 4-Memeriksa Kembali

Memeriksa kembali hasil, langkah ini dilakukan dengan memeriksa kebenaran jawaban

Pada bagian ini kita harus memeriksa kembali

1. Pada program tersebut jika diinputkan String tidak akan mengalami error
2. method `charAt` pada java maupun method `str.at` sudah tepat
3. Masukan pada `c++` jangan menggunakan `cin` karena `cin` tak dapat memuat whitespace sehingga gunakan `getline`
4. Menghasilkan keluaran yang tepat

DAFTAR PUSTAKA

- Polya, G. (2015). *How to Solve It: A New Aspect of Mathematical Method*. New Jersey: Princeton University Press.
- Sukamto, R. A. (2018). *Logika Algoritma dan Pemograman Dasar*. Bandung: Modula.
- Wahyono, dkk. (2021). *Buku Panduan Guru Informatika SMA Kelas 10*. Jakarta: Pusat Kurikulum dan Perbukuan