**PERTEMUAN 6**

## INPUT, OUTPUT, ALGORITMA DAN PENGETAHUAN TERKAIT

**Tujuan Pembelajaran**

Setelah mempelajari materi pada pertemuan ini, mahasiswa mampu:

1. Membedakan operasi input dan output
2. Membuat program dengan operasi input dan output
3. Memahami penggunaan varibel dan konstanta dalam operasi input dan output
4. Memahami persoalan dan memahami penyelesaiannya.

## Uraian Materi

Menginput data melalui keyboard dapat diartikan sebagai kegiatan memasukkan data dengan mengetik melalui keyboard. Dalam prosesnya, permintaan pengetikan data dapat berlangsung saat ada program yang berjalan, sehingga diperlukan instruksi atau perintah input. Sedangkan untuk menampilkan dilayar bisa dilakukakan melalui perintah output.

Penerapan algoritma dengan ilmu terkait adalah merepresentasikan pengetahuan yang diketahui untuk menyusun algoritma kemudian melanjutkanya ke bahasa pemrograman.

### Output

cout

Dengan **cout** (dibaca C out) pemrogram dapat menempatkan suatu informasi ke standar output (umumnya berupa layar). Instruksi cout membutuhkan file iostream untuk di include **#include <iostream>**. include library iostream berlaku juga untuk fungsi cin.

### Mencetak konstanta tanpa format

Output:

Universitas Pamulang

#include<iostream>

**using namespace** std;

**int** main()

{

cout<<”Universitas Pamulang”;

}

Program di atas, ada satu buah instruksi mencetak sebuah konstanta string yaitu cout<<”Universitas Pamulang”; instruksi tersebut akan mencetak kalimat **Universitas Pamulang** di layar.

Beberapa contoh pencetakan nilai konstanta tanpa format pada

C++

|  |  |  |  |
| --- | --- | --- | --- |
| Instruksi cetak | Yang tercetak di layar | | |
| cout<<”Universitas Pamulang”; | Universitas Pamulang |  |  |
| cout<<”U”; | U |  |  |
| cout<<’U’; | U |  |  |
| cout<<123; | 123 |  |  |
| cout<<-32768 | -32768 |  |  |
| cout<<123.456 | 123.456 |  |  |
| cout<<123.4567 | 123.457 (lebih dari | 6 | digit |
|  | dibulatkan ke atas) |  |  |
| cout<<1234567.890 | 1.23457e+006 |  |  |
| cout<<123E2 | 12300 |  |  |
| cout<<123.45E3 | 123450 |  |  |

### Mencetak konstanta dengan format

Perhatikan contoh instruksi pada program c++ berikut:

|  |  |
| --- | --- |
| Code Program | Output |
| #include<iostream> | 123456789 |
| #include<iomanip>  **using namespace** std; | 1234 |
| i**nt** main() |  |
| { |  |
| **int** x; |  |
| x=1234; |  |
| cout<<123456789<<”\n”; |  |
| cout<<setw(9)<<x; |  |
| } |  |

Untuk dapat menggunakan instruksi setw() harus menambahkan #include <iomanip>. Instruksi setw(9) berfungsi untuk menentukan tata letak hasil cetakan elemen berikutnya yaitu variabel **x**, sehingga tercetak dengan lebar 9 lokasi dengan hasil cetakan rata kanan (right justified).

Instruksi setw(), hanya berlaku mengatur format cetakan satu elemen cetakan yang mengikutinya. Elemen” selanjutnya tidak dipengaruhi lagi oleh setw(). Karena setw hanya berlaku satu elemen cetakan berikutnya biasanya pemrogram lebih memilih menulis setw() digabung satu baris dengan elemen yang akan dicetak.

|  |  |
| --- | --- |
| Code Program | Output |
| #include<iostream> | 123456789 |
| #include<iomanip>  **using namespace** std; | 1234 |
| i**nt** main() | 5678 |
| {  **int** x,y,z; | 6789 |
| x=1234; |  |
| y=5678; |  |
| z=6789; |  |
| cout<<123456789<<endl; |  |
| cout<<setw(9)<<x; |  |
| cout<<endl; |  |
| cout<<setw(9)<<y<<endl; |  |
| cout<<z**;** |  |
| } |  |

Instruksi **endl** mempunyai fungsi yang sama dengan **“\n”** dimana kursor akan pindah ke posisi awal pada baris berikutnya. setw() dan endl biasa disebut manipulator.

Beberapa contoh penggunaan manipulator endl

|  |  |
| --- | --- |
| Code Program | Output |
| cout<<”ABC”;  cout<<endl<<”DEF”; | ABC  DEF |
| cout<<”ABC”; cout<<endl;  cout<<”DEF”<<endl; | ABC DEF |
| cout<<”ABC”<<endl;  cout<<”DEF” | ABC  DEF |
| Cout<<”ABC”<<endl<<”DEF”; | ABC  DEF |

Manipulator yang digunakan dalam bahasa C++, selengkapnya di sub bab 1.2.2

### Manipulator

Umumrnya Manipulator digunakan untuk menata tampilan dari data. misalnya untuk menyisipkan karakter *newline* digunakan endl.

|  |  |
| --- | --- |
| Manipulator | Fungsi |
| endl  ends flush | End line, akan membuat kursor berpindah ke posisi awal pada baris berikutnnya sama dengan “\n” Meyisipkan karakter null  Mencetak isi buffer |

|  |  |
| --- | --- |
| Manipulator | Fungsi |
| Dec  hex oct | konversi ke bilangan decimal  konversi ke bilangan hexadecimal konversi ke bilangan octal |
| setbase(n) setw(n) setfill(c) | konversi ke bilangan basis n Menata lebar cetakan sebesar n Mengisi leading fields dengan  karakter c |
| setprecision(n) setiosflags(lf)  resetiosflags(lf) | Membuat lebar decimal point = n Menset atau mangatur format yang diatur dengan tanda format ::ios atau ios  Me-reset format yang diset atau di  atur oleh setiosflags() |

Setiap manipulator setxxxxx() atau resetxxxxx() digunakan harus menambahkan / menggunakan **#include<iomanip>**

### Contoh Penggunaan Manipulator

|  |  |
| --- | --- |
| Code program | Output |
| #include<iostream> #include<iomanip> **using namespace** std; i**nt** main()  {  **int** x; x=64;  cout<<123456789<<endl; cout<<setw(9)<<123<<endl; cout<<123<<endl; cout<<hex<<x<<endl; cout<<oct<<x<<endl; cout<<dec<<x<<endl;  cout<<setfill('.');cout<<setw(10)<<x<<endl;  cout<<x<<endl;  } | 123456789  123  123  40  100  64  ……..64 64 |

* 1. **Tanda Format**

mengatur format/bentuk cetakan diperlukan tanda format yang harus diset dengan menggunakan sebuah manipulator setiosflags() dan direset dengan menggunakan sebuah manipulator resetiosflags().

|  |  |
| --- | --- |
| Tanda Format | Fungsi |
| ios::left  ios::right | dicetak rata kiri  dicetak rata kanan |
| ios::scientific  ios::fixed | dicetak dalam bentuk scientific  dicetak dalam bentuk fixed point |

|  |  |
| --- | --- |
| Tanda Format | Fungsi |
| ios::dec ios::hex  ios::oct | dicetak dalam bentuk desimal dicetak dalam bentuk hexadesimal  dicetak dalam bentuk oktal |
| ios::uppercase  ios::showbase | dicetak dalam bentuk hexadesimal dicetak dengan huruf besar  Menambahkan 0x diawal hasil cetakan yang berbentuk hexademal atau 0 (nol) pada cetakan yang berbentuk oktal |
| ios::showpoint  ios::showpos | Menampilkan desimal point pada hasil cetakan yang mengandung pecahan  Menambahkan tanda + pada hasil cetakan yang bernilai positip |

Contoh Penggunaan tanda format

|  |  |
| --- | --- |
| Code program | Output |
| #include<iostream> |  |
| #include<iomanip> |  |
| **using namespace** std; |  |
| i**nt** main() |  |
| { |  |
| int x=255; |  |
| double y=123.44; |  |
| cout<<setiosflags(ios::showbase); |  |
| cout<<setiosflags(ios::left)<<setw(10)<<x<<endl; | 255 |
| cout<<resetiosflags(ios::left); |  |
| cout<<setiosflags(ios::right)<<setw(10)<<x<<endl; | 255 |
| cout<<setiosflags(ios::hex)<<x<<endl; | oxff |
| cout<<resetiosflags(ios::hex); |  |
| cout<<setiosflags(ios::oct)<<x<<endl; | 0377 |
| cout<<resetiosflags(ios::oct); |  |
| cout<<setiosflags(ios::dec)<<x<<endl; | 255 |
| cout<<setiosflags(ios::fixed)<<setprecision(5)<<y<<endl; | 123.44000 |
| cout<<resetiosflags(ios::fixed); |  |
| cout<<setiosflags(ios::scientific)<<setprecision(5)<<y<< | 1.23440e+00 2 |
| endl; |  |
| } |  |

### Input

Perintah input pada program C++ berbeda dengan bahasa pada program C yaitu tidak menggunakan format. Pada program C++, tipe data yang dinput tergantung tipe data penerimanya. Bila tipe data atau varibel penerimanya bertipe int, maka data yang diinput, apapun bentuknya dan besarnya akan diterima oleh variabel tersebut sebagai tipe int.

cin >> variable

**cin** (dibaca C in) disebut objek yang digunakan untuk menerima data dari standar input (keyboard)

Berikut ini ilustrasi menginput data melalui keyboard:

**8**

**A**

**2**

**B**

**10**

**T**

**8**

**2**

keyboard

instruksi input

**10**

instruksi cetak

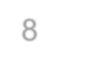
screen

memory

Data yang diketik melalui keyboard harus diterima oleh suatu area dalam memori baik baik dalam sebuah atau sekumpulan variabel.

Misal yang di input nilai integer 8 dan 2, dalam memory disiapkan 3 buah variabel (**A**, **B** dan **T**). **A** untuk menampung nilai **8** dan **B** untuk menampung nilai **2**. Kemudian **T** untuk menampung hasil penjumlahan **A** dan **B**. kemudian mencetak nilai **T** yaitu **10**.

|  |
| --- |
| **Program C++** |
| #include<iostream> **using namespace** std; **int** main()  {  int A, B, T;  cin>>A; //cin = instruksi input cin>>B;  T=A+B;  cout<<”T = ”<<T;//cout = intstruksi output  } |
| **Output :**  **T = 10** |



|  |  |  |
| --- | --- | --- |
| **Flowchart** | **Proses** | |
|  |  | **Nilai A, B dan T belum ditetapkan,** |
| Mulai |  | **isinya ada tapi tidak diketahui.**  **Sewaktu melaksanakan instruksi** |
|  |  | **input ke A, kursor kedip-kedip di** |
|  |  | **layar** |
| cin | **1**  \_ | **Pemakai mengetik 8 kemudian menekan tombol <enter>. Angka 8**  **akan muncul di layar dan masuk** |
| cin |  | **ke variabel A sebagai nilai**  **numerik bertipe data integer, dan** |
|  |  | **kursor turun ke baris berikutnya.** |
|  | **2** | **Pemakai mengetik 2 <enter>, 2** |
| T=A+B | 8   \_ | **akan muncul dilayar dan masuk ke variabel B sebagai nilai numeric,**  **dan kursor turun ke baris** |
|  |  | **berikutnya karena menekan** |
| cout T |  | **tombol <enter>.** |
|  | **3** |  |
|  | 8  | **Proses T=A+B tidak terlihat di** |
| Selesai | 2   10 | **layar. yang terlihat nilai 10 hasil**  **instruksi mencetak T** |
|  |  | **tanda**  **hanya untuk menyatakan** |
|  |  | **ditekan tombol enter. Tanda**  |
|  |  | **tidak munculdi layar.** |

Untuk mempermudah pengguna lain dalam menjalankan program, sebaiknya kita menggunakan program berikut:



#include<iostream> **using namespace** std; **int** main()

{

int A, B, T;

cout<<”Program Menjumlahkan dua buah bilangan”<<endl;

cout<<”Masukkan angka pertama : “; cin>>A;

cout<<”Masukkan angka kedua : “; cin>>B;

T=A+B;

cout<<”Total dari penjumlahan ”<<A<<”+”<<B<<” =

”<<T;

}

**Program C++**

### Menginput karakter melalui keyboard

* + 1. Menggunakan cin >> var;

|  |  |
| --- | --- |
| Contoh | Output |
| #include<iostream> | Masukkan sebuah karakter : U  |
| **using namespace** std;  **int** main() | **U** |
| { |  |
| char A; cout<<”Masukkan sebuah | Bila karakter yang diinput adalah U. |
| karakter : ”; |  |
| cin A; |  |
| cout<<A; |  |
| } |  |

Setelah menginput karakter, harus diakhiri dengan menekan tombol enter ()

|  |  |
| --- | --- |
| **Bila diinput** | **Output** |
| A | A |
| 75 | 7 |
| BCA | B |
| ‘UNPAM’ | ‘ |
| “UNPAM’ | “ |

hanya karakter pertama saja yang akan tercetak di layar.

* + 1. Menggunakan **var=getch() dan var=getche()**

Jika menggunakan cin, setelah karakter di input maka harus menekan tombol enter (), sedangkan fungsi getch() dan gethce() berguna untuk membaca karakter tanpa harus menekan enter. Selain

itu, fungsi ini juga dipakai untuk membaca karakter seperti spasi, tabulasi ataupun enter. Untuk dapat menggunakan fungsi getch() dan getche() harus menambahkan library **<conio.h>**

Perbedaaan dari kedua fungsi ini adalah:

* + - 1. **getch() :** tidak menampilkan karakter dari tombol yang ditekan.
      2. **getche() :** menampilkan karakter dari tombol yang ditekan.

|  |  |
| --- | --- |
| Contoh getch() | Output |
| #include<iostream> | Masukkan sebuah karakter : **U**  Bila karakter yang diinput adalah U.  Karakter U yang tampil di layar bukanlah yang diketik, tetapi hasil instruksi cetak **cout<<A**. apa yang diketik di keyboard tidak terlihat di layar. |
| #include<conio.h> |
| **using namespace** std; |
| **int** main() |
| { |
| char A; |
| cout<<”Masukkan sebuah |
| karakter : ”; |
| A=getch(); |
| cout<<A; |
| } |
| Contoh getche() | Output |
| #include<iostream> | Masukkan sebuah karakter :  **UU**  Bila karakter yang diinput adalah U.  Ada dua huruf U yang tampil di layar. pertama adalah hasil ketikan A=getche(), kedua adalah instruksi cetak  **cout<<A**. |
| #include<conio.h> |
| **using namespace** std; |
| **int** main() |
| { |
| char A; |
| cout<<”Masukkan sebuah |
| karakter : ”; |
| A=getche(); |
| cout<<A; |
| } |

getch() dapat dimanfaatkan untuk menunggu sembarang tombol di keyboard ditekan. Pada kondisi seperti ini tidak diperlukan variabel.

|  |  |
| --- | --- |
| Contoh getch() | Output |
| #include<iostream> | Silakan tekan tombol |
| #include<conio.h> | apapun untuk mengakhiri program ini. |
| **using namespace** std; |  |
| **int** main() |  |
| { |  |
| cout<<”Silakan tekan |  |
| tombol apapun untuk |  |
| mengakiri program |  |
| ini”<<endl; |  |
| getch(); |  |
| } |  |

### Menginput String

String pada bahasa C++ pengertiannya sama dengan pengertian pada umumnya, yaitu deretan atau kombinasi dari sejumlah karakter.

* + 1. Menggunakan cin>>var

#include<iostream>

**using namespace** std;

**int** main()

{

char U[5]; //array U cin>>U;

cout<<U;

}

 Yang tersimpan di dalam array U

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 |
| U | N | P | A | M |  |

**U** bila diketik UNPAM

Hanya bagian ini yang bisa di isi

|  |  |
| --- | --- |
| Bila di input | Output |
| UNPAM | UNPAM |
| UN PAM | UN |
| 85 | 85 |
| “UNP” | “UNP” |
| ‘UNP’ | ‘UNP’ |

Bila dalam string yang diinput mengandung spasi seperti: **UN PAM**, maka hanya **UN** yang tersimpan di array U.

* + 1. Menggunakan **cin.getline(var,sizeof(var));**

Penggunaan cin.getline(var.sizeof(var)) dapat menyimpan karakter spasi di array.

Output:

UN PAM

Bila yang diketikan UN PAM

#include<iostream>

**using namespace** std;

**int** main()

{

char U[7]; //array U cin.getline(U,7); cout<<U;

}

Agar tidak salah mengetik, maka sebaiknya instruksi: **cin.getline(U,7)**;

diganti dengan **cin.getline(U,(sizeof(U))**;

### Menginput nilai numerik melalui keyboard

Output:

30 

30

Bila yang diinput adalah angka 30

#include<iostream>

**using namespace** std;

**int** main()

{

int U; cin>>U; cout<<U;

}

Untuk tipe data lain long int, float atau double. Digunakan instruksi yang sama perbedaanya hanya pada tipe data dan jangkauan nilai yang tersimpan dalam variabel tersebut.

### Algoritma Merepresentasikan apa yang diketahui

Pada pertemuan ini akan diperlihatkan, bahwa algoritma adalah “sekedar” menuliskan atau merepresentasikan apa yang kita ketahui. “Tidak ada yang diketahui, tidak ada yang ditulis, tidak ada yang ditulis, tidak ada

algoritma”. Memahami persoalan dan memahami penyelesaiannya lebih sulit dan memakan waktu lebih lama dibandingkan menulis algoritmanya.

Contoh :

Susun Algoritma untuk mencetak luas sebuah lingkaran bila diketahui jari-jari = 7 cm.

7 cm

Agar dapat menyusun algoritma untuk menjawab soal di atas, ada tiga hal yang perlu dikuasai:

# 1. Paham? 2. Design! (solving)

3.

# Application (coding)

* 1. Mengerti dan memahami maksud dari persoalan yang akan diselesaikan. Jika tidak mengerti maksud soal tersebut, maka tidak ada yang dapat dikerjakan. Untuk contoh soal di atas, Anda harus mengerti apa yang dimaksud dengan luas lingkaran. Kemudian mengetahui cara atau rumus untuk menghitung luas Lingkaran jika jari-jari sudah diketahui.

Luas Lingkaran = phi \* r2 Luas Lingkaran = phi \* r \* r

phi = 3.14 = 22/7 r = jari-jari

Pengetahuan mengenai luas lingkaran, bukan didapat dari mata kuliah algoritma, tapi merupakan pengetahuan umum yang didapatkan dari jenjang pendidikan sebelumnya.

Tidak paham rumus atau cara menghitung luas lingkaran maka tidak dapat membuat algoritma.

* 1. Setelah paham menghitung luas lingkaran dari soal yang dicontohkan: Maka mulailah menuangkan jalan pikiran tersebut dalam bentuk langkah- langkah yang terurai secara: **rinci**, **lengkap** dan **tersusun secara logis.**

r = 7

phi = 3.14

L = phi \* r \* r Cetak L

**r** untuk jari-jari L untuk luas

* 1. Membuat Program komputer

Setelah jalan pikiran tertuang dalam bentuk **algoritma**, maka algoritma tersebut dapat diberikan ke komputer sebagai langkah-langkah penyelesaian pekerjaan. Untuk itu algoritma tersebut harus dituangkan ke dalam suatu **bahasa pemrograman**, misal bahasa C++. Algoritma yang ditulis dalam suatu bahasa pemrograman, disebut **program**.

**Output :**

**Luas Lingkaran = 153.86**

**#include<iostream> using namespace std; int main()**

**{**

**float phi = 3.14, r, L; r=7;**

**L = phi\*r\*r;**

**cout<<"Luas Lingkaran = "<<L;**

**}**

## Soal Latihan / Tugas

1. Susunlah program untuk mencetak empat buah bilangan bulat kemudian mencetak total keempat buah bilangan tersebut!
2. Jika diketahui nilai X=7, Y=5, Z=3. Berapa isi X, Y, dan Z jika dikenai instruksi sebagai berikut:

a. Z = Y b. Z=Z+X c. Z=Z-X\*Y d. Z=X%Y

1. Jika diketahui A=5, B=2, berapa isi A dan B dan T jika dikenai instruksi sebagai berikut:

T=A A=B B=T

## Referensi

A.S, R. (2018). *LOGIKA ALGORITMA dan PEMROGRAMAN DASAR.*

BANDUNG: MODULA.

Charibaldi, N. (2004). *Modul Kuliah Algoritma Pemrograman II Edisi Kedua.*

Yogyakarta.

Davis, S. R. (2014). *C++ For Dummies* (7th ed.). John Wiley & Sons, Inc. Deitel, P., & Deitel, H. (2014). *C++ How To Program* (9th ed.). United State of

America: Pearson.

Munir, R. (2005). *Algoritma dan Pemrograman dalam Bahasa Pascal danC.*

Bandung: Penerbit Informatika.

Sjukani, M. (2014). *Algoritma dan Struktur Data 1 dengan C, C++ dan Java* (Edisi 9 ed.). Jakarta: MItra Wacana Media.