**LAPORAN PRAKTIKUM**

**PEMROGRAMAN TERSTRUKTUR**

**MODUL 14 MENGENAL PROGRAM C DAN C++**

****

**Dikerjakan oleh:**

**Nama: Erick Erdiansyah**

**Kelas: TI A**

**NIM: 3202216014**

**Program Studi Teknik Informatika Jurusan Teknik Elektro**

**Politeknik Negeri Pontianak**

**Pontianak**

**2022**

**MODUL XIV**

**MENGENAL PROGRAM C DAN C++**

1. **TUJUAN**
   1. Mengenal bahasa pemrograman C dan C++
   2. Memahami cara mendeklarasikan Program C dan C++
   3. Memahami masukan dan keluaran dalam bahasa pemrograman C dan C++
   4. Mampu membuat program sederhana C dan C++
2. **DASAR TEORI**
   1. **Pendahuluan**
      1. Bahasa C

Bahasa C berakar dari Bahasa BCPL (*Basic Combined Programming Language*) yang dikembangkan oleh Martin Richards pada tahun 1966 dari Universitas Cambridge. Kemudian berdasarkan bahasa BCPL ini, Ken Thompson dari Bell Labs mengembangkan Bahasa B pada tahun 1969. Dan kemudian Dennis Ritchie mengembangkan Bahasa C antara tahun 1969 – 1972. Nama C dipilih sebagai huruf abjad lanjutan setelah B.

Bahasa C pertama kali digunakan di Komputer Digital Equipment Corporation PDP-11 yang menggunakan sistem operasi UNIX. Bahasa C adalah bahasa pemrograman yang dapat dikatakan berada antara bahasa beraras rendah (*low level language*) yang berorientasi kepada mesin dan bahasa beraras tinggi (*high level language*) yang berorientasi kepada manusia. Seperti diketahui, bahasa beraras tinggi mempunyai kompatibilitas yang tinggi antarplatform. Karena itu, amatlah mudah untuk membuat program pada berbagai jenis mesin. Berbeda halnya kalau menggunakan bahasa beraras rendah, sebab setiap perintahnya bergantung sekali pada jenis mesin.

* + 1. Bahasa C++

C++ diciptakan satu dekade setelah C oleh Bjarne Stroustrup di Laboratorium Bell, AT&T, pada tahun 1983. Bahasa ini bersifat kompatibel dengan bahasa pendahulunya, C. Pada mulanya C++ disebut “a better C”. Nama C++ sendiri diberikan oleh Rick Mascitti pada musim panas 1983. Adapun tanda ++ berasal dari nama operator penaikkan pada bahasa C.

C diambil sebagai landasan dari C++, mengingat keportabilitasan C yang memungkinkannya diterapkan pada berbagai mesin, dari PC hingga mainframe, serta pada pelbagai sistem operasi (DOS, UNIX, VMS, dan sebagainya).

* 1. **Struktur Program C dan C++**
     1. Bahasa C

File judul / Header File

#include < >

Fungsi Utama

main()

{

Tubuh Program

statement-statement;

}

Fungsi\_Fungsi\_Lain()

Fungsi-fungsi lain yang ditulis oleh pemrograman komputer

{

statement-statement;

}

* + 1. Bahasa C++

File judul / Header File

#include< >

Fungsi Utama

void main()

{

Tubuh Program

statement-statement;

}

* 1. **File Judul (*header file*) pada C dan C++**

File judul (*header file*) pada Pemrograman C dan C++ adalah berupa file yang berisi deklarasi dari fungsi-fungsi pustaka tertentu dalam bentuk *prototype*, sedangkan fungsi pustakanya sendiri terletak pada file pustaka (*library*) dengan ekstensi file .LIB. Penggunaan file header ini dapat dilakukan dengan cara menyertakannya menggunakan perintah preprosessor #include.

Salah satu contoh dari file judul yang menyediakan dukungan operasi secara umum dan fungsi dengan kemampuan input dan output adalah #include <stdio.h> untuk Pemrograman C dan #include <iostream> untuk Pemrograman C++ .

Daftar file judul yang lengkap untuk Pemrograman C dan C++ dapat dicari pada referensi-referensi yang ada, salah satunya dapat dilihat pada <https://en.cppreference.com/w/c/header>.

* 1. **Statemen**

Statemen pada Pemrograman C dan C++ adalah merupakan baris perintah yang akan dikompilasi oleh kompiler dan setiap baris perintah ini diakhiri dengan tanda titik koma (;).

Suatu statemen yang panjang dapat ditulis dalam beberapa baris penulisan. Akhir dari suatu baris yang menggunakan tanda ‘\’ menunjukkan bahwa baris berikutnya adalah baris sambungannya. Contohnya:

printf(“Politeknik Negeri Pontianak adalah merupakan \

salah satu perguruan tinggi vokasi yang terdapat \

di Kalimantan Barat”);

* 1. **Ekstensi File C dan C++**

Ekstensi file untuk Pemrograman C adalah .c, sedangkan untuk Pemrograman C++ adalah .cpp.

* 1. **Masukan dan keluaran**
     1. Bahasa C
        + - **scanf**
          - **printf**
     2. Bahasa C++
        + - **cin**
          - **cout**
  2. **Komentar pada C dan C++**

Pada pemrograman C dan C++, untuk memberikan keterangan pada bagian-bagian tertentu dari program dapat dilakukan dengan 2 cara:

* + 1. menggunakan /\* untuk mengawalinya dan diakhiri dengan \*/. Komentar dengan cara ini bisa digunakan untuk lebih dari satu baris.

Contohnya:

/\* Program Menghitung Luas Lingkaran

Dengan masukan berupa jari-jari

Yang dikalikan dengan Konstanta phi \*/

* + 1. menggunakan // yang diletakkan di awal komentarnya

Contohnya:

// Program Menghitung Luas Lingkaran

// Dengan masukan berupa jari-jari

// Yang dikalikan dengan Konstanta phi

* 1. **Kode Format (*Format Specifier*)**

Dalam Pemrograman C, jika kita ingin memasukkan (input) dan keluaran (ouput) harus diikuti dengan kode format yang menunjukkan format dari variabel yang digunakan.

Contoh penggunaan kode format pada masukan dapat dilihat pada potongan kode program di bawah ini:

printf("Masukkan Nilai Gaji Kotor Anda : ");

scanf("%f",&Gaji\_kotor);

Baris pertama dari potongan kode program tersebut ditujukan untuk menampilkan permintaan memasukkan nilai gaji kotor lewat fungsi printf(). Kemudian ketika nilai gaji kotor dimasukkan, akan dibaca sebagai masukkan oleh fungsi scanf(), kemudian “%f” akan menunjukkan format dari nilai yang dimasukkan, yang disesuaikan dengan tipe dari variabel (dalam hal ini variabel bertipe float).

Sedangkan contoh penggunaan kode format pada keluaran dapat dilihat pada potongan kode program di bawah ini:

printf("Nilai Gaji Kotor Anda adalah : %.2f",Gaji\_kotor);

Kode format “%.2f” pada baris program tersebut digunakan untuk memformat data dari variabel Gaji\_kotor yang akan ditampilkan sesuai dengan tipe dari variabelnya, yaitu bertipe float. Sedangkan “.2” pada baris program tersebut digunakan untuk mengatur jumlah digit angka di belakang koma yang akan ditampilkan. Pada contoh ini jumlah digit angka di belakang koma yang akan ditampilkan adalah 2 (dua).

Tabel berikut memperlihatkan kode format-kode format pada Pemrograman C.

|  |  |
| --- | --- |
| Kode Format | Kegunaan |
| %c  %d  %i  %h  %e  %f  %g  %s  %o  %x | Membaca sebuah karakter  Membaca sebuah nilai integer desimal  Membaca sebuah nilai integer desimal  Membaca sebuah nilai short integer desimal  Membaca sebuah data pecahan  Membaca sebuah data pecahan  Membaca sebuah data pecahan  Membaca sebuah data string  Membaca sebuah nilai octal  Membaca sebuah nilai heksadesimal |

Sumber: Konsep Dasar Pemrograman Bahasa C oleh Jogiyanto Hartono, MBA, Ph.D

* 1. **Gaya Penulisan Program C**

Dalam penulisan Program C biasanya diterapkan penulisan kode dimana beberapa statemen tertentu ditulis dengan menggeser beberapa spasi ke dalam dan diberi jarak beberapa spasi antar baris. Kompiler C akan mengabaikan spasi atau baris kosong.

* 1. **Kata Kunci (Keyword) dan Identifier**

Kata Kunci (keyword) adalah berupa perintah dasar yang ada dalam Pemrograman C/C++. Kata Kunci ini tidak dapat digunakan sebagai identifier (seperti untuk variabel, konstanta, atau nama dari sebuah fungsi).

Sedangkan identifier adalah nama untuk menandakan variabel, konstanta, atau nama fungsi.

Contoh di bawah ini memperlihatkan penggunaan kata kunci dan identifier pada Pemrograman C/C++

int jumlah;

int menunjukkan sebuah keyword untuk tipe data integer, sedangkan jumlah adalah identifier dari sebuah variabel.

Aturan penulisan identifier pada Pemrograman C/C++ adalah sebagai berikut:

a. Identifier tidak boleh menggunakan keyword, misalnya int yang berupa kata kunci untuk tipe data integer tidak bisa digunakan untuk identifier dari sebuah variabel.

b. Identifier bisa terdiri dari huruf, angka, dan karakter underscore (garis bawah(\_) yang biasa digunakan sebagai pengganti spasi).

c. Karakter pertama dari identifier tidak boleh berupa angka, hanya bisa berupa huruf atau underscore. Walaupun dibolehkan menggunakan underscore sebagai awal dari identifier tetapi hal ini tidak disarankan.

d. Beberapa kompiler membatasi panjang identifier maksimal 31 karakter.

* 1. **Tipe Data**

Pada Pemrograman C terdapat 8 jenis tipe data yang dibagi dalam 4 kelompok, yaitu:

a. Tipe Data Dasar, terdiri dari **char** untuk menampung individual karakter, **integer** untuk menampung bilangan bulat, dan **float** untuk menampung bilangan pecahan.

b. Tipe Data Turunan, terdiri dari **array** berupa kumpulan data bertipe sama, **structure** berupa kumpulan data bertipe berbeda, serta **pointer** yang merupakan tipe data untuk mengakses alamat memory secara langsung.

c. Tipe Data Bentukan adalah tipe data yang dibuat dengan isi berupa data-data yang sudah ditentukan. Tipe data ini juga dikenal sebagai **Enumerated Data Type** (**enum**). Enum ini biasanya berupa kumpulan dari konstanta.

d. Tipe Data Void adalah tipe data khusus yang menyatakan tidak ada data. Tipe data ini digunakan untuk kondisi tertentu, seperti fungsi yang tidak mengembalikan nilai atau untuk mengisi fungsi dengan nilai kosong.

Pada Pemrograman C tipe data **string** dibentuk menggunakan array.

Tipe data Pemrograman C++ dibagi menjadi

a. Tipe Data Primitive, terdiri dari **integer** untuk menampung bilangan bulat, **float/double** untuk menampung bilangan pecahan, **boolean** yang berisi nilai benar atau salah, **char** untuk menampung individual karakter, dan **void** yang khusus menyatakan tidak ada data.

b. Tipe Data Non-Primitive, terdiri dari **string** yaitu tipe data untuk kumpulan karakter, misalnya “Polnep”, **array** berupa kumpulan tipe data sejenis, **structure** (**struct**) berupa kumpulan tipe data yang bisa lebih dari 1 jenis, **enum** berupa tipe data bentukan, serta **pointer** untuk mengakses alamat memory secara langsung.

* 1. **Konstanta dan Variabel**

Pada Pemrograman C dan C++, Konstanta dan Variabel digunakan untuk menyimpan nilai, baik itu berupa masukan yang berikutnya akan diproses maupun berupa keluaran yang merupakan hasil dari pemrosesan. Konstanta dan variabel memiliki perbedaan dimana selama proses berlangsung dari program, konstanta akan menyimpan nilai yang tetap sedangkan variabel akan menyimpan nilai yang berubah-ubah.

Agar konstanta dan variabel bisa menyimpan nilai dan dapat digunakan selama proses program, maka keduanya harus dideklarasikan berupa sebuah pengenal (*identifier*). Proses pendeklarasian ini ditujukan agar program dapat mengenal keduanya, kemudianmenggunakan nilai yang terdapat pada keduanya untuk proses yang akan dilakukan atau menyimpan nilai hasil dari pemrosesan pada keduanya.

Untuk membuat konstanta dapat digunakan 2 cara, yaitu:

a. Dengan perintah #define yang ditulis sebelum main(), dengan format

#define nama\_konstanta nilai\_konstanta, contohnya

#include <stdio.h>

#define PHI 3.14

main()

{

b. Dengan perintah const yang ditempatkan di dalam tubuh program, dengan format

const tipe\_data nama\_konstanta = nilai\_konstanta;, contohnya

#include <stdio.h>

main()

{

const float phi = 3.14;

float luas;

int jari;

* 1. **Case Sensitivity**

Penulisan perintah, kata kunci, dan identifier pada Pemrograman C/C++ membedakan antara huruf besar dan huruf kecil, yang disebut juga case sensitive. Penulisan kata kunci pada Pemrograman C/C++ menggunakan huruf kecil, misalnya printf, sehingga jika ditulis Printf akan mengakibatkan error.

Untuk penulisan identifier, misalnya variabel Luas, akan berbeda dengan LUAS, dan juga akan berbeda dengan luas.

Hal in berbeda dengan penulisan perintah, kata kunci, dan identifier pada Pascal yang *case insensitive* (tidak case sensitive).

* 1. **Karakter *Escape (Escape Sequence*)**

Dalam pemrograman C dan C++ dikenal karakter *escape*, yaitu konstanta yang digunakan untuk membantu pengaturan tampilan, misalnya untuk perpindahan kursor ke baris baru, untuk membunyikan bel, dan sebagainya. Karakter *Escape* ini diawali dengan tanda “\”.

Berikut ini adalah tabel yang memperihatkan karakter *escape* pada Pemrograman C dan C++

|  |  |
| --- | --- |
| Karakter *Escape* | Kegunaan |
| \a  \b  \f  \n  \r  \t  \v  \0  \’  \”  \\  \?  \ooo  \xhh | Bunyi bel (*alarm or beep*)  Mundur satu spasi (*backspace*)  Berpindah halaman baru (*form feed*)  Berpindah baris baru (*line feed*)  Berpindah ke kolom pertama pada baris yang sama (*carriage return*)  Tabulasi horisontal  Tabulasi vertikal  Nilai kosong (*null*)  Tanda petik satu (*single quote*)  Tanda petik dua (*doube quote*)  Garis miring terbalik (*backslash*)  Tanda tanya (*question mark*)  Bilangan oktal (*octal number*)  Bilangan heksadesimal (*hexadecimal*) |

Sumber: <https://www.geeksforgeeks.org/escape-sequences-c/>

* 1. **Ungkapan (*Expression*)**

Ungkapan adalah formula atau rumus atau persamaan yang digunakan untuk melakukan sebuah operasi. Ungkapan merupakan kombinasi antara 2 atau lebih operan (*operand*) yang dihubungkan oleh operator. Operan pada ungkapan bisa berupa konstanta atau variabel.

Contoh ungkapan adalah sebagai berikut:

**Operan1**

**Operan2**

**Operan3**

**Operan4**

**Operator1**

**Operator2**

**Operator3**

**a = b + c \* d**

* 1. **Operator**

Operator adalah suatu tanda yang menghubungkan antara 2 operan, yang digunakan untuk membantu operasi matematika atau logika tertentu.

Jenis-jenis operator pada Pemrograman C dan C++ adalah:

1. Operator Aritmatika adalah operator yang membantu perhitungan matematika (aritmatika), terdiri dari

|  |  |
| --- | --- |
| Tanda | Keterangan |
| \* | Perkalian |
| / | Pembagian |
| % | Modulus (Sisa hasil bagi) |
| + | Penjumlahan |
| - | Pengurangan |

Operator “\*” dan “/” memiliki hirarki yang sama

1. Operator Unary adalah operator yang dikenakan pada satu operan saja, terdiri dari

|  |  |
| --- | --- |
| Tanda | Keterangan |
| - | Unary minus |
| ++ | Peningkatan dengan penambahan nilai 1 |
| -- | Penurunan dengan pengurangan nilai 1 |
| ! | Unary NOT |
| ~ | Unary komplimen satu (bitwise NOT) |
| & | Pointer (menghasilkan alamat memori operan) |
| \* | Pointer (menghasilkan nilai pengenal di alamatnya) |

1. Operator Penugasan (*Assignment Operator*) adalah operator yang digunakan untuk memberikan nilai dari ungkapan ke variabel, seperti berikut ini:

|  |  |  |
| --- | --- | --- |
| Tanda | Contoh | Arti |
| = | A = B + C | Memberikan nilai ungkapan B + C ke variabel A |

Operator Penugasan ini juga bisa digabungkan dengan operator lain, seperti Operator Aritmatika atau Operator *Bitwise*, seperti contoh berikut ini:

|  |  |  |  |
| --- | --- | --- | --- |
| Tanda | Keterangan | Contoh | Arti |
| += | Gabungan Operator Penugasan dan Operator Arimatika | A += 1 | A = A + 1 |
| -= | Gabungan Operator Penugasan dan Operator Arimatika | A -= 1 | A = A – 1 |
| \*= | Gabungan Operator Penugasan dan Operator Arimatika | A \*= 1 | A = A \* 1 |
| /= | Gabungan Operator Penugasan dan Operator Arimatika | A /= 1 | A = A / 1 |
| %= | Gabungan Operator Penugasan dan Operator Arimatika | A %= 1 | A = A % 1 |
| <<= | Gabungan Operator Penugasan dan Operator *Bitwise* | A <<= 1 | A = A << 1 |
| >>= | Gabungan Operator Penugasan dan Operator *Bitwise* | A >>= 1 | A = A >> 1 |
| &= | Gabungan Operator Penugasan dan Operator *Bitwise* | A &= 1 | A = A & 1 |
| != | Gabungan Operator Penugasan dan Operator *Bitwise* | A != 1 | A = A ! 1 |
| ^= | Gabungan Operator Penugasan dan Operator *Bitwise* | A ^=1 | A = A ^ 1 |

1. Operator Hubungan adalah operator yang digunakan untuk menunjukkan hubungan dari dua operan, terdiri dari

|  |  |
| --- | --- |
| Tanda | Keterangan |
| < | Lebih kecil |
| <= | Lebih kecil atau sama dengan |
| > | Lebih besar |
| >= | Lebih besar atau sama dengan |
| == | Sama dengan |
| != | Tidak sama dengan |

1. Operator Logika adalah operator yang digunakan untuk membandingkan logika hasil dari operator-operator hubungan, terdiri dari

|  |  |
| --- | --- |
| Tanda | Keterangan |
| && | Logika DAN (AND) |
| !! | Logika ATAU (OR) |
| ! | Operator Unary Komplimen |

Hubungan logika dan hasilnya terlihat seperti di bawah ini:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X | Y | X && Y | X !! Y | !X |
| 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 | 1 |

1. Operator Koma adalah operator yang digunakan untuk menggabungkan beberapa ungkapan dengan proses yang berurutan.

Sebagai contoh, nilai A adalah 5 dan B adalah 7, maka hasil ungkapan-ungkapan di bawah ini adalah

|  |  |  |  |
| --- | --- | --- | --- |
| Ungkapan | Hasil Akhir | | |
| A | B | X |
| X = (B=5, B\*2) | 5 | 5 | 10 |
| X = (B=B+5, B\*2) | 5 | 12 | 24 |
| X = (B=B+5, B=B\*2) | 5 | 24 | 24 |
| X = (A=B, B\*2) | 7 | 7 | 14 |
| X = (A=B+5, B\*2) | 12 | 7 | 14 |
| X = (A=B+5, A\*2) | 12 | 7 | 24 |
| X = (A=3, A=A+5, B=A\*2) | 8 | 16 | 16 |
| X = (A=3, A+5, B=A\*2) | 3 | 6 | 6 |
| X = (A=A, B, 3) | 5 | 7 | 3 |

1. Operator *Bitwise* adalah operator yang digunakan untuk manipulasi bit

|  |  |
| --- | --- |
| Tanda | Keterangan |
| << | Pergeseran ke kiri satu bit |
| >> | Pergeseran ke kanan satu bit |
| & | Bitwise AND |
| ^ | Bitwise XOR (Exclusive OR) |
| ! | Bitwise OR |

Hubungan bit dan hasilnya untuk operasi bitwise logika

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| bit | | AND  b1 & b2 | OR  b1 ! b2 | XOR  b1 ^ b2 |
| b1 | b2 |
| 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 |

1. **LATIHAN SOAL**
2. **Nomor Program : 1401**
3. Tulis program berikut!

//Program C Pertamaku

#include <stdio.h>

main()

{

printf("Selamat Datang di");

printf("Pemrograman C");

}

1. Apa output dari program 1401 di atas?

Jawab : ‘selamat datang dipemrograman C’

1. Apa yang harus ditambahkan agar ada spasi antara “Selamat Datang di” dan “Pemrograman C”?

Jawab : spasi bisa ditambahkan setelah kata ‘di’ atau sebelum kata ‘pemrograman’

1. Rubah #include <stdio.h> menjadi komentar dengan menambahkan tanda //, kemudian lakukan kompilasi. Apa yang terjadi? Jelaskan!

Jawab : program error karena stdio adalah library input/output dari C

1. Tambahkan karakter escape (*escape sequence*) “\n” setelah string “Selamat Datang di”. Apa outputnya? Apa fungsi dari “\n”.

Jawab : “\n” berfungsi untuk membuat newline atau tab

1. **Nomor program: 1402**

a. Tulis program berikut ini!

//Input Output Pemrograman C

#include <stdio.h>

main()

{

float a;

printf("Masukkan sebuah nilai: ");scanf("%f",&a);

printf("Nilai yang dimasukkan adalah : %25.2f",a);

}

b. Apa hasil dari program 1402 di atas?

Jawab : program diatas meminta user input dan menampilkan hasilnya

c. Apa fungsi keyword (kata kunci) float a pada program di atas?

Jawab : float a adalah sebuah variable

d. Apa fungsi dari keyword (kata kunci) scanf dan printf pada program di atas?

Jawab : scanf menyimpan user input kedalam variable dan printf untuk menampilkan program ke terminal

e. Apa fungsi dari “%f” pada program di atas?

Jawab : ‘%f’ merupakan format spesifier untuk float atau bilangan pecahan

f. Apa fungsi dari & pada program di atas?

Jawab : ‘&’ berupakan sebuah pointer yang merujuk pada lokasi variable a di memory

g. Rubah %f menjadi %.2f dan amati ouput yang dihasilkan. Jelaskan!

Jawab : akan ada 2 angka dibelakang koma

g. Rubah %.2f menjadi %.8f dan amati output yang dihasilkan. Jelaskan perubahan apa yang terjadi!

Jawab : angka dibelakang koma menjadi 8

h. Tambah angka 25 pada kode format menjadi %25.2f. Apa hasil running program?

Jawab : akan ada jarak sepanjang 25 spasi diantara string dan angka

1. **Nomor program: 1403**

a. Tulis program berikut ini!

//Program Menghitung luas\_lingkaran

#include <stdio.h>

#include <stdlib.h>

main()

{

float luas, phi = 3.14;

int r;

printf("Program Menghitung Luas Lingkaran\n");

printf("---------------------------------\n");

printf("\n");

printf("Diketahui : \n");

printf("Jari-jari = ");scanf("%d",&r);

luas = phi\*r\*r;

system("cls");

printf("Jadi Hasil Perhitungan Luas Lingkaran\n");

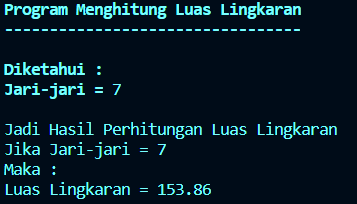
printf("Jika Jari-jari = %d\n",r);

printf("Maka : \n");

printf("Luas Lingkaran = %.2f\n",luas);

}

b. Apa hasil dari program 1403 di atas?



1. **Nomor Program : 1404**
   1. Tulis program berikut!

//Selamat Datang di Program C++ Pertamaku

#include <iostream>

main()

{

std::cout << "Selamat Datang di" << std::endl;

std::cout << "Pemrograman C";

}

* 1. Apa hasil dari program 1404 di atas?



Catatan:

Perintah std::cout terdiri dari 2 bagian yaitu **namespace std** dan perintah **cout**. Namespace std berisi fitur-fitur standar library pada Pemrograman C, seperti cin, cout, endl, vector, string, pair, map queue, deque dan lain sebagainya.

* 1. Tambahkan << std::endl setelah “Selamat Datang di”. Apa hasil yang ditampilkan? Apa fungsi dari endl?

Jawab : ‘<< std::end’ berfungsi seperti tab atau ‘\n’ pada c

* 1. Rubah kode program menjadi seperti berikut ini:

//Selamat Data di Program C++ Pertamaku

#include <iostream>

using namespace std;

main()

{

cout << "Selamat Datang di" << endl;

cout << "Pemrograman C";

}

Apa hasil yang ditampilkan?



Catatan:

**using namespace std** berfungsi untuk penggunaan fitur-fitur standar library Pemrograman C, seperti cin, cout, endl, vector, string, pair, map queue, deque dan lain sebagainya. Dengan penggunaan **using namespace std** ini, kita tidak perlu lagi menuliskan std::cin, std::cout dan sebagainya. Namun penggunaan **using namespace std** ini bukan merupakan praktek yang baik, karena timbulnya error karena adanya tabrakan perintah standar. Oleh karena itu sebaiknya tetap menggunakan std untuk penulisan perintah standar.

1. **Nomor Program 1405**
   1. Tulis program berikut!

//Input Output Pemrograman C++

#include <iostream>

int main()

{

float a;

std::cout << "Masukkan sebuah nilai: ";std::cin >> a;

std::cout << "Nilai yang dimasukkan adalah : " << a ;

}

* 1. Lakukan kompilasi dan kemudian jalankan program.



* 1. Beri masukan dengan bilangan bulat, misalnya 10. Apa hasil running program?



* 1. Beri masukan dengan bilangan pecahan, misalnya 12.59892198. Apa hasil running program?



* 1. Beri masukan dengan bilangan pecahan lain, misalnya 12.5981234. Apa hasil running program?



* 1. Beri masukan dengan bilangan pecahan lain, misalnya 12.59895198. Apa hasil running program?
  2. 
  3. Rubah kode program menjadi seperti berikut ini:

//Input Output Pemrograman C++

#include <iostream>

#include <iomanip>

int main()

{

float a;

std::cout << "Masukkan sebuah nilai: ";std::cin >> a;

std::cout << "Nilai yang dimasukkan adalah : " \

<< std::setprecision(8) << a ;

}

* 1. Beri masukan dengan bilangan bulat, misalnya 10. Apa hasil running program?



* 1. Beri masukan dengan bilangan pecahan, misalnya 12.59892198. Apa hasil running program?



* 1. Ganti angka 8 pada std::setprecision() menjadi angka 2. Apa hasil running program?



* 1. Rubah kode program menjadi seperti berikut ini:

//Input Output Pemrograman C++

#include <iostream>

#include <iomanip>

int main()

{

float a;

std::cout << "Masukkan sebuah nilai: ";std::cin >> a;

std::cout << "Nilai yang dimasukkan adalah : " << std::fixed \

<< std::setprecision(2) << a ;

* 1. Beri masukkan dengan bilangan bulat, misalnya 10. Apa hasil running program?



* 1. Apa fungsi std::fixed?

Jawab : memastikan ada angka dibelakang koma

* 1. Rubah kode program menjadi seperti berikut ini:

//Input Output Pemrograman C++

#include <iostream>

#include <iomanip>

int main()

{

float a;

std::cout << "Masukkan sebuah nilai: ";std::cin >> a;

std::cout << "Nilai yang dimasukkan adalah : " << std::fixed \

<< std::setw(25) << std::setprecision(2) << a ;

}

* 1. Beri masukkan dengan bilangan bulat, misalnya 10. Apa hasil running program?



1. **Nomor Program : 1406**
   1. program berikut dan amati outputnya!

//Program Menghitung luas\_lingkaran

#include <iostream>

#include <iomanip>

#include <windows.h>

main()

{

float luas, phi = 3.14;

int r;

std::cout << "Program Menghitung Luas Lingkaran" << std::endl;

std::cout << "---------------------------------" << std::endl;

std::cout << std::endl;

std::cout << "Diketahui : " << std::endl;

std::cout << "Jari-jari = ";std::cin >> r;

luas = phi\*r\*r;

system("CLS");

std::cout << "Jadi Hasil Perhitungan Luas Lingkaran" << std::endl;

std::cout << "Jika Jari-jari = " << r << std::endl;

std::cout << "Adalah : " << std::endl;

std::cout << "Luas Lingkaran = " << std::fixed\

<<std::setprecision(2)<< luas;

}

* 1. Apa hasil dari program 1406 di atas?

