**MATERI/BAHAN PRAKTIKUM**

**PENDAHULUAN DAN PENGENALAN (IDENTIFIER)**

1. **TUJUAN**
2. Mahasiswa mengenal struktur bahasa pemrograman C dan C++
3. Mahasiswa mengenal perintah-perintah dasar C dan C++
4. Mahasiswa mengenal tipe data bahasa pemrograman C dan C++
5. **TEORI**

Bahasa C atau C++ adalah suatu bahasa pemrograman. Bahasa C termasuk sebagai bahasa pemrograman tingkat menengah, maksudnya bahasa C bisa dipelajari dengan lebih mudah karena mudah dimengerti tetapi mempunyai kemampuan yang tinggi.

**II.1. Struktur Program C**

Contoh:

/\* contoh program turbo C++ \*/ 🡪 komentar

 #include <iostream.h> 🡪 preprosesor/header file

 #include<stdio.h>

 main(){ 🡪 fungsi utama

 cout << “TEKKOM UNIKOM”; 🡪definisi fungsi utama

 printf(“TEKKOM UNIKOM”);

 return 0;

 }

Header file

Main( )

{

Deklarasi;

Statement;

}

**Fungsi main**

Merupakan fungsi yang menjadi inti dari program dan merupakan awal dan akhir eksekusi. Fungsi ini harus ada dalam setiap program. Tanda kurawal buka { merupakan tanda awal fungsi main dan tanda kurawal tutup } merupaka tanda berakhirnya fungsi *main*.

**Prepocessor directive/header file**

*Preprocessor directive* merupakan suatu pernyataan yang akan diikiutsertakan dalam program, dimana pernyataan tersebut akan di-*compile* sebelum proses compilasi yang sebenarnya dilakukan.

**Deklarasi**

Deklarasi diperlukan bila kita akan menggunakan pengenal (identifier) dalam program.

Identifier dapat berupa variable, konstanta dan fungsi.

**Konstanta**

Konstanta merupakan suatu nilai yang tidak dapat diubah selama proses program berlangsung. Konstanta nilainya selalu tetap.

**Variable**

Variabel adalah suatu pengenal (identifier) yang digunakan untuk mewakili suatu nilai tertentu di dalam proses program. Berbeda dengan konstanta yang nilainya selalu tetap, nilai dari suatu variable bisa diubah-ubah sesuai kebutuhan.

**Aturan umum penulisan bahasa C dan C++**

Bahasa C dan C++ :

1. membedakan penulisan huruf besar dan kecil.
2. Untuk memberi komentar pada suatu *statement* (keadaan), dapat menggunakan /\* di awal dan \*/ di akhir atau // dalam satu baris.
3. Awal dan akhir *subroutine* atau fungsi harus diapit kurung kurawal ({ })
4. Setiap *statement* harus diakhiri tanda titik koma, kecuali statement yang diawali oleh tanda kres “#”
5. Setiap variabel yang digunakan wajib dideklarasikan terlebih dahulu.
6. Untuk bahasa C setiap melakukan proses input dan outpu data harus selalu menyertakan format datanya, sedangkan di C++ tidak.

**Perintah-perintah dasar**

**C++ 🡪 cout << data;**

**C 🡪 printf(“format data”,data);**

Merupakan perintah yang biasa dipakai untuk menampilkan suatu keluaran pada layar

Contoh: cout << “Lab. Pemrograman Komputer”;

 cout << “Jurusan Teknik Elektro\n”;

**cin >>**

Merupakan perintah yang berguna untuk memasukan data, yang selanjutnya didefinisikan sebagai data variabel.

*example*: int age;

cin >> age;

**II.2. Tipe Data**

Tipe data merupakan bagian program yang paling penting karena tipe data mempengaruhi setiap instruksi yang akan dilaksanakan oleh komputer. Dalam bahasa C terdapat empat tipe data dasar, yaitu :

1. **Tipe data karakter 🡺** Untuk tipe data karakter kita gunakan perintah **char**.
2. **Tipe data bilangan bulat**

Ada beberapa tipe data standar yang digunakan untuk data bilangan bulat.

|  |  |  |  |
| --- | --- | --- | --- |
| **Tipe Data** | **Memori** | **Format** | **Jangkauan Nilai** |
| int | 2 byte | %d/%i | -32.768 s/d 32.767 |
| unsigned int | 2 byte | %u | 0 s/d 65.535 |
| char | 1 byte | %d/%i | -128 s/d 127 |
| unsigned char | 1 byte | %u | 0 s/d 255 |
| unsigned long | 4 byte | %lu | 0 s/d 4.294.967.295 |
| long | 4 byte | %ld/%li | -2.147.483.648 s/d 2.147.483.647 |

1. **Tipe data bilangan pecahan**

Tipe data untuk bilangan pecahan terdiri dari beberapa jenis yaitu :

|  |  |  |  |
| --- | --- | --- | --- |
| **Tipe Data** | **Memori** | **Format** | **Jangkauan Nilai** |
| float | 4 byte | %f | 3.4\*(10^-38) - 3.4\*(10^+38 |
| double | 8 byte | %f | 1.7\*(10^-308) - 1.7\*(10^+308) |
| long double  | 10 byte | %lf | 3.4\*(10^-4932) - 1.1\*(10^+4932) |

1. **Tipe data string**

Dalam pemrograman C, untuk variabel yang menampung data string tidak ada perintah khusus, karena dalam bahasa C, string adalah sebuah array karakter atau sebuah pointer ke sebuah variabel char.

**II.3. Operator**

* **Operator Penugasan**

Operator Penugasan (*Assignment operator*) dalam bahasa C berupa tanda sama dengan

(“=”).

* **Operator Aritmatika**

Bahasa C menyediakan lima operator aritmatika, yaitu :

* + \*: untuk perkalian
	+ / : untuk pembagian
	+ % : untuk sisa pembagian (modulus)
	+ + : untuk pertambahan
	+ - : untuk pengurangan
* **Operator Hubungan (Perbandingan)**

Operator Hubungan digunakan untuk membandingkan hubungan antara dua buah operand (sebuah nilai atau variable. Operator hubungan dalam bahasa C :

**< , <= , > ,>= , ==, !=**

* **Operator Logika**

Jika operator hubungan membandingkan hubungan antara dua buah operand, maka operator logika digunakan untuk membandingkan logika hasil dari operator-operator hubungan.

Operator logika ada tiga macam, yaitu :

* && : Logika AND (DAN)
* || : Logika OR (ATAU)
* ! : Logika NOT (INGKARAN)
* **Operator Bitwise**

Operator bitwise digunakan untuk memanipulasi bit-bit dari nilai data yang ada di memori.

Operator bitwise dalam bahasa C :

* ( : Pergeseran bit ke kiri
* >> : Pergeseran bit ke kanan
* & : Bitwise AND
* ^ : Bitwise XOR (exclusive OR)
* | : Bitwise OR
* ~ : Bitwise NOT
	+ **Operator Unary**

Operator Unary merupakan operator yang hanya membutuhkan satu operand saja. Dalam bahasa C terdapat beberapa operator unary, yaitu :



**II.4 Kode – kode format untuk Fungsi Printf() pada Bahasa C**

|  |  |
| --- | --- |
| **Kode format** | **Fungsi** |
| %c%s%d%i%u%x%o%f%e%g%p | Menampilkan sebuah karakterMenampilkan nilai stringMenampilkan nilai decimal integerMenampilkan nilai decimal integerMenampilkan nilai decimal integer tak bertandaMenampilkan nilai heksadesimal integerMenampilkan nilai octal integerMenampilkan nilai pecahanMenampilkan nilai pecahan dalam bentuk saintifikSebagai pengganti %f atau %e tergantung yang terpendekMenampilkan suatu alamat memeori untuk pointer |

* + Nilai string

|  |  |
| --- | --- |
| **Format** | **Arti** |
| “%s”“%Ns”“%-Ns”“%N.Ms”“%N.-Ms” | Menampilkan semua nilai stringMenampilkan semua karakter rata kanan dengan lebar N posisi; N adalah konstanta numeric bulatMenampilkan semua karakter rata kiri dengan lebar N posisiMenampilkan rata kanan hanya M buah karakter pertama saja dengan lebar N posisi; M dan N adalah konstanta numericMenampilkan rata kiri hanya M buah karakter pertama saja dengan lebar N posisi; M dan N adalah konstanta numeric |

* + Nilai integer

|  |  |
| --- | --- |
| **Format** | **Arti** |
| “%i” atau “%d”“%Ni” atau “%Nd”“%-Ni” atau “%-Nd”%MNi” atau “MNd”  | Menampilkan selebar digit yang adaMenampilkan selebar N digit rata kanan; n berupa numerikMenampilkan selebar N digit rata kiriMenampilkan selebar N digit rata kanan dengan blank diisi oleh M; M bias beerupa karakter atau numenrik |

* + Nilai pecahan

|  |  |
| --- | --- |
| **Format** | **Arti** |
| “%f”“%Nf”“%-Nf”“%N.Pf”“%MN.Pf | Menampilkan selebar digit yang adaMenampilkan selebar N digit rata kanan; N berupa numericMenampilkan selebar N digit rata kiriMenampilkan selebar N digit dan P digit dibelakang koma rata kanan; N dan P berupa numericMenampilkan selebar N digit rata kanan, P digit di belakang koma dan M blank dengan rata kanan; M,N,P berupa numeric  |

**VI. Daftar pustaka**

1. [www.ilmukomputer.com](http://www.ilmukomputer.com)
2. <http://id.wikipedia.org>

## VII.Latihan

* 1. Program latih1-1.cpp

#include<iostream.h>

#include<conio.h>

main()

{

clrscr();

cout<<"hello"<<endl;

cout<<"hello"<<'\n';

cout<<"hello"<<endl;

}

* 1. Program latih1-2.cpp

#include <stdio.h>

#include <conio.h>

main()

{

 char k1,k2;

 k1=’A’;

 k2=k1;

 printf(“Nilai variable K1 adalah %c\n”,k1);

 printf(“Nilai variable K2 dalam bentuk angka = %d\n”,k2);

 getch();

 return 0;

}

* 1. Program latih1-3.cpp

/\*Program contoh bitwise \*/

#include<iostream.h>

#include<conio.h>

main()

{

int a,b,c;

a = 5; b = 3;

cout<<"program operator bitwise"<<endl;

c = a<<2;

cout<<" 1. operasi pergeseran kekiri ; a<<2 " <<endl;

cout<<" a = " <<a<<endl;

cout<<" c = " <<c<<endl;

cout<<endl;

c = a>>2;

cout<<" 2. operasi pergeseran kekanan ; a>>2 " <<endl;

cout<<" a = " <<a<<endl;

cout<<" c = " <<c<<endl;

cout<<endl;

c = a & b;

cout<<" 3. operasi AND ; a&b "<<endl;

cout<<" a = " <<a<<endl;

cout<<" b = " <<b<<endl;

cout<<" c = " <<c<<endl;

cout<<endl;

c = a | b;

cout<<" 4. operasi OR ; a|b " <<endl;

cout<<" a = " <<a<<endl;

cout<<" b = " <<b<<endl;

cout<<" c = " <<c<<endl;

cout<<endl;

c = a ^ b;

cout<<" 5. operasi XOR ; a^b " <<endl;

cout<<" a = " <<a<<endl;

cout<<" b = " <<b<<endl;

cout<<" c = " <<c<<endl;

cout<<endl;

c = ~a ;

cout<<" 6. operasi NOT ; ~a " <<endl;

cout<<" a = " <<a<<endl;

cout<<" c = " <<c<<endl;

cout<<endl;

}

* 1. Program latih1-4.cpp

/\*Program contoh relational \*/

#include<stdio.h>

#include<conio.h>

main()

{

int a,b,c;

a = 5; b=3;

printf(“program operator realtional");

c = a<b;

cout<<" 1. opeator < " <<endl;

cout<<" a = " <<a<<endl;

cout<<" b = " <<b<<endl;

cout<<" c = " <<c<<endl;

cout<<endl;

}

* 1. Program latih1-5.cpp

/\*program menampilkan luas persegi panjang\*/

#include<iostream.h>

#include<conio.h>

main()

{

int luas, lebar, panjang;

lebar = 3;

panjang = 5;

luas = lebar \* panjang;

cout<<"luas persegi panajng = "<<luas<<endl;

}

**Tugas Praktikum**

* + 1. Modifikasi soal latihan1-4.cpp dengan memakai relational “ **< , <= , > ,>= , ==, != ”** dengan perintah dalam bahasa C++.
		2. Modifikasi soal latihan1-5.cpp dengan perintah atau sintak dalam bahasa C.
		3. Buatlah Program untuk menghitung luas permukaan bola dan volume bola, dimana nilai jari-jarinya sudah disediakan diprogram.

Petunjuk :

V = 4/3 \* pi \* r \* r \* r

L = 4\*pi\*r\*r

Dengan :

V : Volume bola

L : Luas permukaan bola

r : Jari – jari

pi : 3.14

* + 1. Buatlah program untuk menampilkan jumlah jam, menit dan detik dari 7272 detik.

Petunjuk :

7272 detik = 2 jam, 1 menit, 12 detik