PERTEMUAN 5
PREPROCESSOR DAN LIBRARY FUNCTION

A. Tujuan Pembelajaran

1. mahasiswa mampu memahami dan mengerti dari penggunaan dari setiap fungsi pada preprocessor dan library function. Serta mahasiswa mampu menerapkan penggunaannya di dalam pemrograman.

Uraian Materi

1. Definisi Preprocessor

Prepocessor include merupakan salah satu jenis pengarah preprocessor directive (#include) yang terdapat pada C++. Atau dengan kata lain dapat dikatakan bahwa preprocessor merupakan program yang sangat berperan penting dalam mengontrol jalannya alur program. Dimana preprocessor akan selalu dijalankan lebih dulu ketika suatu proses kompilasi sudah terjadi. Pada suatu compiler C++ proses pengerjaan preprocessor akan dilakukan dengan sendirinya sehingga dalam proses penggunaannya, seorang programmer tidak perlu menjalankan suatu proses preprocessor terlebih dahulu.

Dimana bentuk umum dari preprocessor adalah sebagai berikut:

include <nama_file>

Bentuk awal dari #include <nama_file> merupakan bahwa pencarian suatu file akan dilakukan pada direktori khusus yaitu yang merupakan direktori file include. Sedangkan bentuk yang kedua seperti #include “namafile” merupakan bahwa pencarian suatu file dilakukan pertama kali pada direktori aktif dimana tempat program sumber dan apabila tidak ditemukan maka pencarian akan dilanjutkan kepada direktori lainnya yang sesuai dengan perintah pada suatu sistem operasi.

Bentuk umum preprocessor tidak diakhiri dengan tanda semicolon, hal ini dikarenakan bentuk tersebut bukan merupaka suatu pernyataan, tetapi merupakan suatu preprocessor directive. Dimana baris tersebut akan memerintahkan kepada suatu compiler yang akan menyisipkan suatu file lain. Dalam hal ini adalah file yang berakhiran .h atau disebut sebagai file header.

#include <iostream.h>	Digunakan pada program yang akan melibatkan objek dengan cout
#include <conio.h>	Akan digunakan apabila sudah melibatkan clrscr(), yang merupakan suatu instruksi yang berfungsi membersihkan layar
#include <iomanip.h>	Akan digunakan apabila telah melibatkan instruksi setw() yang berfungsi mengatur lebar dari suatu tampilan data pada program.
#include <math.h>	Akan digunakan oleh suatu program dimana menggunakan perintah sqrt () yang berfungsi untuk sebuah operasi pada fungsi matematika kuadrat.

Adapun cara kerja suatu preprocessor adalah membaca perintah dari pengarah suatu preprocessor yang akan dilakukan dalam suatu program yang akan diawali dengan bentuk # atau pound.

Ada 3 jenis dari preprocessor sebagai berikut:

a. Include file

   Pada include file merupakan suatu preprocessor berfungsi untuk melibatkan pustaka yang berupa suatu file header yang terdapat pada sebuah program yang akan dikerjakan. Dimana dengan preprocessor ini maka kita dapat memasukkan suatu kode instruksi yang berada pada sebuah file header yang sedang dibuat. Dimana format penulisannya adalah sebagai berikut:

   #include <nama_file_header.h>

   #include “nama_file_header” dimana tanda kutip disini berfungsi untuk memasukkan suatu altternatif path dari suatu file dan akan disertakan pada suatu program.

b. Definisi makro

Pada definisi makro suatu preprocessor ini akan berfungsi sebagai penentu definisi dari suatu identifier tertentu yang akan ditulis dalam suatu program C++. Dimana preprocessor ini juga dapat digunakan untuk menentukan rumus dari suatu makro fungsi. Preprocessor ini akan dijalankan apabila dengan menyertakan suatu preprocessor directif #define, #undef dalam suatu kode program yang sering kita kerjakan. Adapun format penulisan directive #define adalah sebagai berikut:

#define nama_makro nilai_makro

c. Pengarah kondisional kompilasi

Pada proses ini, suatu preprocessor berfungsi mengarahkan akan kerja dari suatu program yang kita buat dengan beberapa pengarah dari preprocessor yang digunakan untuk memberikan dan mengatur akan sebuah solusi yang dapat dijalankan dalam program C++. Dimana preprocessor ini dijalankan dengan melibatkan preprocessor directive #if, #else, #elif, #ifdef, #ifndef ke dalam kode program yangs sedang kita buat.

Berikut adalah penjelasan dari beberapa instruksi – instruksi dari preprocessor yaitu sebagai berikut:

a. #define

Pada nstruksi #define berfungsi untuk melaksanakan proses substitusi makro dari suatu lembar teks yang satu ke lembar teks yang lain. Dimana proses ini akan melalui suatu file pada teks tersebut yang akan digunakan. Seperti instruksi #define nama karakter-sequence. Dimana pada instruksi ini tidak terdapat tanda titik koma. Sehingga apabila diawal karakter telah berhasil maka dia akan berhenti di akhir baris pada suatu program tersebut.

Misal:

“true” untuk angka 1

“false” untuk angka 0

Maka dapat dituliskan:

#define true 1

#define false 0

Dimana compiler akan mengganti 1 dan 0 setiap kali menemui true dan false.

b. #error

Pada perintah #error dalam suatu program disini berfungsi untuk memaksa suatu compiler dalam menghentikan suatu proses kompilasi pada suatu program. Dimana pada awalnya #error ini berfungsi di proses “debugging”. Perintah yang sering digunakan adalah #error message. Dimana apabila menemui #error akan muncul suatu notif yang berupa angka baris pada suatu program yang ingin dijalankan.

c. #if, #ifdef, #ifndef, #else, #elif, #endif

Perintah – perintah berupa #if, #ifdef, #ifndef, #else, #elif, #endif processor disini akan digunakan pada jenis versi perintah program yang dilakukan secara tertsruktur. Sehingga apabila ekspresi #if, #ifdef atau #ifndef bernilai benar maka kodenya tepat berada diantara salah satu dari instruksi tersebut dan #endif akan tersusun pada suatu program. Tetapi apabila yang terjadi adalah sebaliknya, maka akan terlewati atau tidak tersusun sama sekali. #endif disini berfungsi untuk menandai akhir dari suatu blok. Dan instruksi #else akan digunakan kepada salah satu yaitu instruksi – instruksi tersebut dengan cara yang sama seperti “else” pada perintah program C

d. #include

Instruksi #include berfungsi untuk memerintah suatu compiler membaca dan menyusun suatu file pada sumber yang lain.

e. Pragma

#pragma merupakan suatu perintah yang sudah diketahui perancangannya yang terdapat suatu jenis perintah yang harus

ditujukan pada suatu perintah dalam sebuah program. Suatu perintah akan terdapat sebuah pilihan untuk membantu dalam pembuatan dari trace suatu program. Dimana proses penggandaan (trace) ini akan ditentukan oleh instruksi dari suatu #pragma.

f. #undef

Pada instruksi #undef disini berfungsi untuk memindahkan suatu definisi yang telah ditentukan sebelumnya dari suatu nama makro yang mengikutinya. Dimana prinsi penggunaan suatu #undef adalah untuk memungkinkan dimana nama makro ditempatkan hanya berapa pada bagian code yang memerlukannya saja.

2. Definisi Library Function

Fungsi merupakan kumpulan dari suatu instruksi yang terdapat pada program yang akan dikelompokkan menjadi fungsi sendiri atau letaknya terpisah dari suatu program yang akan menggunakan fungsi tersebut. Sedangkan standar library function merupakan suatu fungsi standart yang sudah disediakan oleh progam Bahasa C. dimana untuk menggunakannya, maka kita harus mencantumkan suatu header file dari fungsi tersebut yaitu dengan menggunakan perintah #include dalam suatu program.

Misal:

```
#include <stdio.h>
void main()
{
  int C, D, R;
  C = 7;
  D = 8;
  R = C + D;
  Printf (“%i”,  T);
}
```

Berdasarkan contoh diatas, dimana instruksi printf () tersebut bukanlah merupakan suatu instruksi yang secara langsung melakukan proses pekerjaan untuk mencetak, tetapi hanya memanggil sebuah fungsi tersebut yang bernama printf(). Dimana fungsi printf (“%i”, T) adalah untuk memerintahkan computer

supaya menjalankan semua perintah yang ada dalam sebuah fungsi yang dinamakan dengan printf(). Sehingga untuk memanggil suatu fungsi adalah hanya dengan menggunakan nama pada fungsi tersebut.

Contoh penggunaan library function yaitu:

#include <stdio.h>

Adapaun beberapa fungsi yang ada pada stdio.h adalah sebagai berikut:

a. Printf ()
b. Scanf()
c. Getchar()
d. Gets()
e. Puts()

Sehingga dalam merancang suatu function ada beberapa hal yang harus diperhatikan, seperti::

a. Input adalah data yang akan menjadi masukka suatu sistem
b. Proses, yaitu bagaimana proses algorima dari program yang akan digunakan dalam suatu fungsi tersebut.
c. Output, merupakan suatu informasi yang akan dikembalikan oleh fungsi kepada si pemanggil instruksi tersebut.

Penulisan fungsi dibagi menjadi 2 macam, yaitu:

1. Function Prototype

   Fungsi prototype merupakan suatu pendeklarasian dari fungsi yang disebut sebagai kepala atau judul fungsi atau dikenal sebagai pengenal fungsi dari suatu program.

2. Function Definition

   Function definition merupakan suatu penulisan fungsi ang dilakukan secara lengkap dalam suatu program.

Contoh function dalam C++ yaitu:

```
#include <iostream>
using namespace std;
int addition (int x, int y) //called or calling
```

```
function berisi formal parameter
{    int s;
     s=x+y;
     return s;
}
int main ()
{    int z;
     z = addition (4,1); // caller function berisi
actual parameter
     cout << "The result is " << z;
     return 0;
}
```

Jawab:

int addition (int x, int y)

z = addition(4, 1)

Dimana nilai dari 4 dan 1 merupakan nilai yang dikirim oleh suatu fungsi pemanggil atau yang disebut sebagai caller function ke suatu fungsi called function melalui suatu parameter yang ada. Dimana argument pada caller function disini merupakan actual parameter dan suatu argument pada caliing function merupakan formal parameter

Terdapat 3 macam passing pada suatu function dalam C++ yaitu:

a. Passing by value

   Contoh:

```
//passing by value
#include <iostream>
using namespace std;
void foo(int val)
{   val = 7;
}
int main()
{   int value = 6;
cout << "value = " << value << '\n';
foo(value);
    cout << "value = " << value << '\n';
    system("pause");
    return 0;
```

```
}
```

b. Passing by reference

Contoh:

```
//passing by reference
#include <iostream>
using namespace std;
void foo(int &val)
{   val = 7;
}
int main()
{   int value = 6;
cout << "value = " << value << '\n';
foo(value);
cout << "value = " << value << '\n';
system("pause");
return 0;
}
```

c. Passing by address

Contoh:

```
//passing by address
#include <iostream>
using namespace std;
void foo(int *ptr)
{   *ptr = 7;
}
int main()
{   int value = 6;
cout << "value = " << value << '\n';
foo(&value);
cout << "value = " << value << '\n';
system("pause");
return 0;
}
```

2.1 Penggunaan Library Function

Adapun beberapa penggunaan library function adalah sebagai berikut:

a. Library function yang berhubungan dengan penanganan tanggal dan waktu

Dalam penangana tanggal dan waktu pada Bahasa C mempunyai suatu pustaka sendiri, yaitu time.h. dimana fungsi tersebut memiliki beberapa kumpulan fungsi sebagai berikut:

Fungsi	Deskripsi
Difftime	Untuk menghitung perbedaan waktu yang terjadi dari dua nilai time_t
Time	Untuk mengembalikan suatu nilai waktu yang ada saat ini
Clock	Untuk mengembalikan suatu nilai perhitungan clock dari sebuah prosesor
ctime	Untuk melakukan konversi nilai time_t ke representasi tekstual
Strftime	Digunakan untuk melakukan konversi suatu objek struct tm pada representasi tekstual
Wcsftime	Digunakan untuk melakukan konversi objek struct tm ke suatu representasi custom wide string
gmtime	Digunakan untuk melakukan suatu konversi nilai time_t ke suatu ekspresi kalender menggunakan koordinat universal GMT

b. Library function yang berhubungan dengan fungsi matematika

Library function matematika digunakan untuk operasi matematika. Library function yang digunakan dalam matematika adalah math.h. sehingga dengan penggunaan library ini, seorang programmer harus menuliskan suatu instruksi yaitu #include <math.h>. adapun beberapa fungsi yang digunakan dalam fungsi matematika ini adalah sebagai berikut:

Fungsi	Deskripsi
Fabs()	Digunakan untuk memberikan nilai kembalian yang berupa suatu nilai yang absolut

Ceil()	Digunakan untuk memperoleh suatu bilangan bulat yang paling kecil dimana nilai yang muncul tidak boleh kurang dari nilai argumennya
Floor()	Digunakan untuk memperoleh suatu bilangan bulat yang muncul tidal lebih besar dari nilai argumennya
Round()	Digunakan untuk memperoleh suatu nilai bilangan bulat yang terdekat dari nilai argumennya
Trunc()	Digunakan untuk memperoleh suatu nilai bilangan bulat dari argumennya
Pow()	Digunakan untuk menghitung perpangkatan
Pow10()	Digunakan untuk menghitung perpangkatan 10
Sqrt()	Digunakan untuk menghitung akar
Hypot()	Digunakan untuk menghitung sisi miring dari segitiga siku - siku
Log()	Digunakan untuk menghitung nilai algoritma

Contoh:

a. Fungsi akar kuadrat sqrt()

```
#include <stdio.h>
#include <math.h>
void main()
{
Int X, Y;
X = 16
Y = sqrt(X);
Printf (“%i”, Y);
}
```

Penjelasan:

X = 16

Y = 4 (akar dari 16)

Y = sqrt(X). Pada perintah ini bukan untuk menghitung akar kuadrat X tetapi hanyalah sebuah perintah yang memanggil fungsi sqrt() supaya semua fungsi yang ada di dalam sqrt() dapat dilaksanakan kemudian. Dan hasill perhiungan dari sqrt disimpan oleh Y.

b. Fungsi pangkat (pow)

```
#include <stdio.h>
#include <math.h>
void main()
{int X, Y, Z;
X = 7;
Y = 2;
Z = pow (X, Y);
Printf("%i", Z);
}
```

Penjelasan:

X = 7

Y = 2 (merupakan pangkat)

Maka Z = 7^2 = 49

Soal Latihan/Tugas

1. Buatlah program sederhana dengan menggunakan bahasa C dengan menggunakan salah satu library function pada matematika?
2. Jelaskanlah perbedaan dari function prototype dengan fungsi definition dan sebutkan contohnya?
3. Apakah function hanya bisa dituliskan diabawah main(). Jelaskan?
4. Apakah bisa dilakukan pemanggilan function oleh function lain? Lalu bagaimana pengaturan letaknya? Jelaskan?
5. Apakah bisa menuliskan function tanpa menuliskan prototype? Jelaskan bagaimana caranya?

Referensi

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

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.