



MODUL PRAKTIKUM DASAR-DASAR PEMROGRAMAN



```
index = 0;
while (index < 10)
{
    printf("The index is %d \n", index);
    index++;
}

/* for (index = 0; index <10; index++)
{
    printf("The index is %d" index);
}
```

The screenshot shows the CodeBlocks IDE interface. The main editor displays a C program using a `while` loop to print the index from 0 to 9. The output window on the right shows the execution results, displaying the index values. The status bar at the bottom indicates that the program was compiled and executed successfully.

Disusun Oleh:
Prio Handoko, S.Kom., M.T.I.

FAKULTAS DESAIN & TEKNOLOGI
PROGAM STUDI INFORMATIKA
UNIVERSITAS PEMBANGUNAN JAYA
TANGERANG SELATAN 2019

MODUL PRAKTIKUM I

STRUKTUR BAHASA C DAN I/O

Modul ini membahas mengenai struktur bahasa C dan sekaligus membahas I/O (input output) dalam pemrograman bahasa C.

Baris Komentar

Baris komentar adalah baris-baris yang menjelaskan maksud dari perubah yang digunakan atau maksud dari program itu sendiri. Hal ini dimaksudkan untuk memudahkan pelacakan atas perubah yang digunakan apabila program yang digunakan cukup besar atau memudahkan orang lain memahami program yang kita buat. Dalam program, baris komentar diletakkan diantara tanda `/*` dan `*/` dan baris ini tidak dikerjakan oleh komputer, hanya dianggap sebagai baris kosong.

Struktur Bahasa C

Bentuk program C mirip dengan kebanyakan program bahasa tingkat tinggi lainnya. Bentuk programnya adalah :

Judul Program

Header File

Deklarasi

Deskripsi

Judul Program

Judul program sifatnya sebagai dokumentasi saja, tidak signifikan terhadap proses program. Ditulis dalam bentuk baris komentar.

Syntax:

```
/*<judul program>*/
```

Contoh:

```
/* Program Menghitung Rata-Rata */
```

Header File

C menyediakan sejumlah file judul (*header file*) yaitu file yang umumnya berisi prototipe fungsi, definisi makro, variabel dan definisi tipe. File ini mempunyai ciri yaitu namanya diakhiri dengan extension `.h` (dot h).

Syntax:

```
#define <<nama header>.h>
```

Tabel 1. Daftar header file

Header File	Fungsi
<code><assert.h></code>	Melakukan diagnosa terhadap kesalahan program yang kita buat, yaitu dengan mengirimkannya ke dalam suatu file yang sering disebut dengan <i>standard error file</i> .
<code><ctype.h></code>	Melakukan pengecekan dan konversi terhadap sebuah karakter, <i>bukan</i> string. Fungsi-fungsi yang terdapat di dalam file header ini diawali dengan awalan <i>is...</i> dan <i>to....</i>
<code><errno.h></code>	Mengatasi kesalahan-kesalahan (<i>error</i>) yang muncul di dalam program.
<code><float.h></code>	Menyimpan nilai-nilai konstan yang didefinisikan untuk melakukan perhitungan-perhitungan pada bilangan riil.
<code><limits.h></code>	Menyimpan makro yang merupakan nilai-nilai konstan yang siap pakai.
<code><math.h></code>	Mendefinisikan fungsi-fungsi yang digunakan untuk melakukan perhitungan- perhitungan matematika.
<code><signal.h></code>	Mengetahui signal atau tanda-tanda yang perlu dilaporkan dalam eksekusi sebuah program.
<code><stdarg.h></code>	membentuk fungsi yang terdiri dari parameter (argumen) dinamis, artinya banyaknya parameter dapat berubah-ubah dalam setiap fungsi tersebut dipanggil.
<code><stddef.h></code>	menyimpan beberapa variabel dan makro standar. Beberapa diantaranya juga dapat Anda temui di dalam file header yang lain.
<code><stdio.h></code>	Header ini adalah header yang hampir digunakan dalam semua program C, yaitu untuk melakukan proses input-output.
<code><stdlib.h></code>	menyimpan fungsi-fungsi yang akan digunakan untuk melakukan pengkonversian bilangan, alokasi memori dan pekerjaan-pekerjaan pemrograman lainnya yang sejenis.
<code><string.h></code>	menyimpan fungsi-fungsi yang digunakan untuk melakukan manipulasian string.
<code><time.h></code>	menyimpan fungsi-fungsi yang digunakan untuk melakukan pembacaan maupun pengkonversian terhadap waktu dan tanggal.

Contoh:

```
#include <stdio.h>
```

Keterangan: menyatakan bahwa agar membaca file bernama **stdio.h** saat pelaksanaan kompilasi.

Deklarasi

Deklarasi adalah bagian untuk mendefinisikan semua nama yang dipakai dalam program. Nama tersebut dapat berupa nama tetapan (konstanta), nama variabel, nama tipe, nama prosedur, nama fungsi.

Deskripsi

Bagian inti dari suatu program yang berisi uraian langkah-langkah penyelesaian masalah. Program C pada hakekatnya tersusun atas sejumlah blok fungsi. Sebuah program minimal mengandung sebuah fungsi. Setiap fungsi terdiri dari satu atau beberapa pernyataan, yang secara keseluruhan dimaksudkan untuk melaksanakan tugas khusus.

Bagian pernyataan fungsi (disebut tubuh fungsi) diawali dengan tanda “{” (kurung kurawal buka) dan diakhiri dengan tanda “}” (kurung kurawal tutup)

Variabel

Variabel dalam program digunakan untuk menyimpan suatu nilai tertentu dimana nilai tersebut dapat berubah-ubah. Setiap variabel mempunyai tipe dan hanya data yang bertipe sama dengan tipe variabel yang dapat disimpan di dalam variabel tersebut. Setiap variabel mempunyai nama. Pemisahan antar variabel dilakukan dengan memberikan tanda koma.

Syntax:

```
<nama_tipe_data> <nama_variabel>;
```

Contoh :

```
int jumlah;
float harga_per_unit, total_biaya;
```

Dari contoh diatas, variabel jumlah hanya boleh menerima data yang bertipe integer (bulat), tidak boleh menerima data bertipe lainnya. Variabel **harga_per_unit** dan **total_biaya** hanya bisa diisi dengan bilangan *float* (pecahan).

Konstanta

Berbeda dengan variabel yang isinya bisa berubah selama eksekusi program berlangsung, nilai suatu konstanta tidak bisa berubah.

Syntax:

Cara 1.

```
<nama_tipe_data> <nama_variabel> = <nilai_variabel>;
```

Cara 2.

```
#define <nama_variabel> <nilai_variabel>
```

Contoh :

```
const int m = 8;
#define pajak 0.05
```

Fungsi main() atau int main() atau int main(void)

Fungsi main() harus ada pada program, karena fungsi inilah yang menjadi titik awal dan titik akhir eksekusi program. Tanda “{” di awal fungsi menyatakan awal tubuh fungsi sekaligus awal eksekusi program, sedangkan tanda “}” di akhir fungsi merupakan akhir tubuh fungsi dan sekaligus akhir eksekusi program.

Operasi Input dan Output

Berkenaan dengan mempelajari bahasa pemrograman C, tentunya tidak terlepas dari sebuah proses penting yang sering digunakan hampir di setiap program C yang dibuat, yaitu sebuah proses yang dinamakan masukan (input) dan keluaran (output). Proses ini penting karena dalam sebuah program terkadang pemrogram membutuhkan proses memasukkan data untuk mendapatkan sebuah keluaran tertentu. Input adalah sebuah proses menerima inputan dari pengguna dan output merupakan proses menampilkan data/nilai dari sebuah proses. Nilai yang dimasukkan dan ditampilkan oleh program nantinya akan disimpan dalam sebuah wadah penyimpanan nilai yang disebut dengan *variabel*. Tabel 2. di bawah ini memperlihatkan beberapa tipe data yang akan sering digunakan dalam bahasa C.

Tabel 2. Tabel tipe data

Jenis	Ukuran	Range	Format	Keterangan
Char	1 byte	-128 s/d 127	%c	Karakter/Huruf
Int	2 byte	-32768 s/d 32767	%d	Integer/Bilangan Bulat
Float	4 byte	-3.4E-38 s/d 3.4E+38	%f	Float/Bilangan Pecahan
Double	8 byte	-1.7E-308 s/d 1.7E+308	%lf	Pecahan Presisi Ganda

Fungsi Perintah Input

scanf()

Fungsi ini digunakan untuk menampilkan data yang dimasukkan dari keyboard. Guna membedakan antara setiap tipe data yang digunakan untuk melakukan proses input, maka dalam penulisannya perlu dituliskan simbol string kontrol yang tepat untuk mewakili tipe data yang digunakan.

Syntax:

```
scanf("<simbol_string_kontrol>", &<nama_variabel>);
```

Tabel 3. berikut menunjukkan beragam simbol string kontrol yang dapat dituliskan untuk mewakili tipe data yang digunakan.

Tabel 3. Simbol string kontrol

Simbol	Tipe Data
%d	Integer
%c	Char
%s	String
%f	Float
%lf	Double/Long Float
%s	String
%[^\n]	Until breakline. Menerima inputan hingga enter. (seperti gets)

Contoh:

```
/*Contoh penggunaan fungsi scanf()*/
#include <stdio.h>
void main()
{
    int angka;
    scanf("%d", &angka);
    fflush(stdin); //fflush digunakan untuk membersihkan buffer.
}
```

Keterangan: Jika menggunakan **scanf()**, pada penulisan perintahnya harus disertai dengan simbol “&” (dan) di depan variabel penampung. Hal ini akan dibahas nanti saat Alasannya akan dijelaskan saat mempelajari tentang *pointer*.

gets()

Fungsi yang digunakan untuk menerima inputan berupa kata/kalimat yang akan dibaca sampai negasi enter (\n).

Syntax:

gets(<nama_variabel>);

Contoh:

```
/*Contoh penggunaan fungsi gets()*/
#include <stdio.h>
void main()
{
    char nama[20];
    gets(nama);
    fflush(stdin); //fflush digunakan untuk membersihkan buffer.
}
```

getchar()

Fungsi yang digunakan untuk menerima inputan berupa karakter.

Syntax:

getchar();

Contoh:

```
/*Contoh penggunaan fungsi getchar()*/
#include <stdio.h>
void main()
{
    char grade;
    grade = getchar();
    fflush(stdin); //fflush digunakan untuk membersihkan buffer.
}
```

*Fungsi Perintah Ouput***printf()**

Merupakan fungsi yang digunakan untuk menampilkan output ke layar. Dengan menggunakan fungsi ini, tampilan dapat diatur (diformat) dengan mudah.

Syntax :

Opsi 1.

printf("string");

Contoh:

```
/*Program Opsi 1*/
#include <stdio.h>
int main()
{
    printf("Belajar Pemrograman Komputer");
    getchar();
}
```

Opsi 2.

printf("string string_kontrol", argumen);

Contoh:

```
/*Program Opsi 2*/
#include <stdio.h>
int main()
{
    int nilai;
    printf("Nilai saya : %d", nilai); getchar();
}
```

String kontrol dapat berupa keterangan beserta penentu format (seperti %d, %f). Argumen adalah data yang akan ditampilkan, dapat berupa variabel, konstanta, maupun ungkapan.

puts()

Merupakan fungsi untuk menampilkan string (kata dan kalimat) disertai dengan breakline (\n).

Syntax:

```
puts("string/kalimat");
```

Contoh:

```
/* Program penggunaan fungsi puts() */
#include <stdio.h>
int main()
{
    puts("Halo, selamat datang di Prodi Informatika");
    puts("Selamat belajar giat 😊 ");
    getchar();
}
```

putchar()

Merupakan fungsi untuk menampilkan karakter.

Syntax:

```
putchar("nama_variabel");
```

Contoh:

```
/*Program penggunaan fungsi putchar()*/
#include <stdio.h>
void main()
{
    char grade;

    //proses input grade
    printf("Grade = ");
    scanf("%c", &grade);
    fflush(stdin);

    //proses output
    printf("Grade anda  : ");
    putchar(grade);
    getchar();
}
```


LATIHAN.

Buatlah program-program berikut.

1. Menampilkan:

- A. tulisan **“Halo, siapa namamu?”**, lalu meminta pengguna memasukkan namanya;
- B. tulisan **“Berapa usiamu?”**, lalu meminta pengguna memasukkan usianya ;
- C. tulisan **“Dimanakah tempat tinggalmu?”**, lalu meminta pengguna memasukkan alamat tempat tinggalnya;
- D. tulisan **“Dimanakah tempat kuliahmu?”**, lalu meminta pengguna memasukkan nama kampus tempat kuliah;
- E. tulisan **“Program studimu apa?”**, lalu meminta pengguna memasukkan nama program studinya;

dan akhirnya menuliskan pesan **“Hallo <nama>, senang berteman denganmu. Usiamu sekarang sudah <usia> tahun ya? Makin keren aja kamu, apalagi sekarang kamu kuliah di <nama kampus> di program studi <nama prodi>. Rumah kamu di <alamat rumah> kan? Kapan-kapan kita pergi sama-sama ya ke kampus? Aku juga mahasiswa <nama kampus>”**.

Keterangan:

<nama>, **<usia>**, **<nama kampus>**, **<nama prodi>**, dan **<alamat rumah>** adalah data yang dibaca dari hasil input sebelumnya.

2. Menukarkan dua buah nilai dari dua buah variabel. Misalnya, sebelum pertukaran nilai a=5, nilai b=3, maka setelah pertukaran nilai a=3, nilai b=5.
3. Menghitung luas dan keliling persegi panjang. Data masukan dibaca dari piranti masukan dan luas dan keliling bangun persegi panjang ditampilkan sebagai keluaran.
4. Mengkonversikan total detik menjadi jam menit detik. Petunjuk: 1 menit = 60 detik dan 1 jam = 3600 detik.
5. Menampilkan hasil konversi jarak x dalam km ke dalam meter dan sentimeter dimana setiap kali hasil konversi akan dimunculkan didahului dengan penekanan tombol Enter. Ingat bahwa 1 m = 100 cm dan 1 km = 1000 m = 100.000 cm.