

JOB SHEET 02

Pemrograman Dasar C

*"Everybody should learn to program a computer,
because it teaches you how to think."*

TUJUAN PEMBELAJARAN

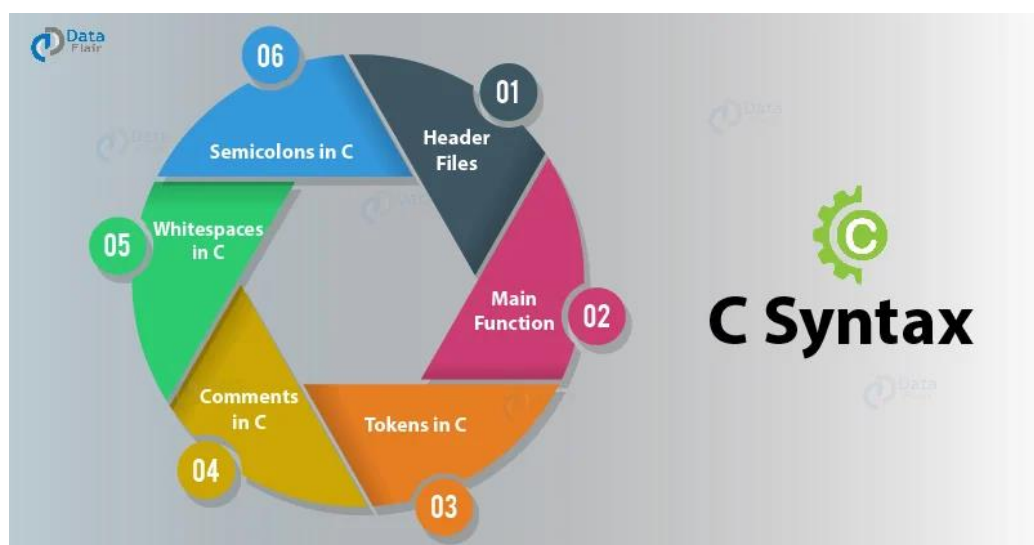
1. Mampu memahami struktur dasar penulisan bahasa pemrograman C.
2. Mampu mengimplementasikan *Basic Syntax C*, Proses *Input & Output* dalam pemrograman menggunakan IDE.

POKOK MATERI

1. Struktur Dasar C
2. Proses Input & Output

URAIAN MATERI

A. Sintaks



Dalam setiap bahasa pemrograman memiliki sintaks tersendiri yang dapat diartikan sebagai aturan penulisan kode pada bahasa pemrograman tersebut. Ini adalah struktur dasar yang wajib dipahami. Struktur program C dibagi menjadi dua bagian utama:

- 1) Bagian Include;
- 2) Blok Fungsi Main

```
#include <stdio.h>

int main(){
    // kode atau logika program kita di sini
    return 0;
}
```

```
#include <stdio.h>

int main(){
    printf("Hello World!");
    return 0;
}
```

```
1  #include <stdio.h>
2
3  int main(){
4      printf("Hello World");
5      return 0;
6  }
7
8
```

bagian include

blok fungsi main

statement

1) Header Files

Di baris paling awal, terdapat kode #include. Perintah #include digunakan untuk memasukkan sebuah file khusus yang memungkinkan untuk mengakses berbagai fitur tambahan dalam bahasa C. #include berfungsi untuk mengimpor fungsi-fungsi yang sudah didefinisikan pada header

file. Header file adalah file yang berisi definisi fungsi yang sudah dibuat. Tujuannya agar bisa digunakan pada program C yang lainnya. File ini berekstensi .h, contoh: stdio.h. Pada contoh di atas, kita mengimpor file stdio.h. File ini berisi deklarasi fungsi-fungsi dasar yang kita butuhkan untuk membuat program C, seperti printf().

2) Main Function

Satu-satunya perintah yang harus ada di setiap kode program bahasa C adalah main(). Struktur main() sendiri pada dasarnya merupakan sebuah fungsi (function). Isi dari function ini diawali dan diakhiri dengan tanda kurung kurawal { dan } . Di dalam tanda kurung inilah “statement” dari kode program penyusun fungsi main() ditulis. Fungsi main() adalah fungsi utama dalam program. Fungsi ini akan dieksekusi pertamakali saat program dijalankan. Karena itu, kita harus menuliskan logika program di dalam fungsi ini.

```
int main(){  
    //...  
    return 0  
}
```

Maksud dari int di depan main adalah tipe data yang akan dikembalikan. Maka di dalam fungsi main(), wajib kita sertakan return 0. Artinya, fungsi main akan mengembalikan nilai 0 setelah selesai dieksekusi.

Cara lain kita bisa menggunakan void.

```
void main(){  
    //...  
}
```

void artinya tidak ada (kosong). Jika kita menggunakan void, maka kita tidak perlu menuliskan kata kunci return di akhir fungsi. Karena fungsi void tidak akan mengembalikan nilai apapun.

3) Statement

Statemen adalah perintah-perintah atau fungsi untuk melakukan sesuatu.

```
printf("Hello World!");
```

Artinya: kita menyuruh komputer untuk mencetak teks Hello World! ke console. Setiap statement di dalam program C wajib diakhiri dengan titik koma (;) atau semicolon.

4) Semicolons

Pada program, semikolon menandakan berakhirnya sebuah perintah atau statement. Artinya dalam program C semua statement harus diakhiri dengan semikolon kalau tidak program akan error.

5) Identifier

Identifier merupakan sebuah nama yang digunakan untuk mengidentifikasi variabel, fungsi, atau semua item yang didefinisikan oleh programmer. Adapun cara menuliskan identifier yaitu:

- (1) Diawali dengan huruf A s/d Z, a s/d z atau underscore (_).
- (2) Identifier dapat dibuat menggunakan digit angka 0 s/d 9 tapi tidak diperbolehkan untuk digunakan pada awal karakter pada identifier.
- (3) Tidak diperbolehkan menggunakan karakter lain selain yang telah disebutkan, artinya karakter seperti !, \$, %, @, - dan karakter lainnya tidak diperbolehkan dimanapun di dalam nama identifier.

- (4) Bahasa C merupakan bahasa case sensitive yang artinya alamat dan Alamat merupakan identifier yang berbeda.

```
// misalkan kita membuat variabel seperti ini
string nama = "Petani";
string Nama = "Kode";
```

Variabel `nama` dengan `Nama` akan dianggap variabel yang berbeda, karena ada yang pakai huruf besar (kapital) dan huruf kecil.

- (5) Nama identifier harus unik. Untuk menghindari kesalahan dalam penulisan program atau agar lebih mudah membaca kode program (tidak membingungkan) sebaiknya hindari memberi nama identifier yang mirip.

6) Comments

Comment atau komentar akan diabaikan oleh kompiler. Komentar biasanya digunakan untuk mendeskripsikan fungsi dari baris program atau mendokumentasi setiap bagian dari sebuah program. Komentar dapat terdiri dari satu baris komentar dengan menggunakan tanda `//` pada awal komentar dan untuk membuat komentar yang lebih dari satu baris dapat menggunakan tanda `/*` dan diakhiri dengan tanda `*/`.

```
// ini adalah komentar
```

```
#include <stdio.h>

int main(){
    // ini adalah komentar satu baris
    printf("Hello World!");

    /*
        ini adalah komentar
        yang lebih
        dari satu baris
    */
    return 0;
}
```

7) Whitespaces

Istilah whitespace dalam bahasa C digunakan untuk mendeskripsikan blank, tab, spasi, baris baru dan dapat juga berupa komentar. Whitespace digunakan untuk memisahkan bagian dari sebuah statemen yang memungkinkan kompiler mengidentifikasi elemen yang satu dan lainnya dalam sebuah statement.



```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    printf("Welcome to DataFlair");
    getch();
};
```

Annotations in the image:

- Including Header Files (points to `#include`)
- main() Function Must Be There (points to `main()`)
- Single Line Comment (points to `// helps to print the message "Welcome to DataFlair"`)
- Semicolon After Each Statement (points to `getch();`)
- Program Enclosed Within Curly Braces (points to the opening and closing braces of the `main` function)

B. Proses Input & Output

1. **Input** adalah sesuatu data yang kita masukan ke dalam program. Input biasanya diambil dari perangkat inputan seperti keyboard, mouse, kamera, microphone, dll.

2. **Proses** adalah langkah-langkah yang harus dilakukan program untuk menghasilkan output.
3. **Output** adalah informasi yang dihasilkan setelah dilakukan proses. Output biasanya ditampilkan ke layar komputer.

Pada bahasa pemrograman C, terdapat beberapa fungsi dasar untuk menampilkan output dan mengambil input.

Ada dua macam fungsi yang bisa digunakan untuk menampilkan output:

1. Fungsi `printf()`

Fungsi `printf()` merupakan fungsi untuk menampilkan output ke layar komputer. Fungsi ini terdapat pada library `stdio.h`. Oleh sebab itu, ketika kita diharuskan untuk menuliskan `#include <stdio.h>` di bagian atas program agar bisa menggunakan fungsi ini.

Berikut ini struktur dasar fungsi `printf()`:



Perhatikan:

"format" adalah sebuah teks (string) untuk ditampilkan. Lalu tanda ... akan berisi sebuah variabel atau nilai untuk ditampilkan berdasarkan format yang diberikan pada teks "format".

```
#include <stdio.h>

int main(){
    printf("Hello, ini adalah teks output\n");
    printf("Nama saya %s\n", "Dian");
    printf("Usia saya %d\n", 20);
    return 0;
}
```

Coba buat dan jalankan program diatas menggunakan DevC++!

Pada fungsi printf() kita menggunakan simbol %s, %d, dan \n untuk format teks. Simbol-simbol ini disebut format specifier.

Mari kita bahas arti dari simbol tersebut:

%s adalah simbol untuk menampilkan nilai string;


%d adalah simbol untuk menampilkan nilai angka atau bilangan desimal;

\n adalah simbol untuk membuat baris baru.

<i>Simbol</i>	<i>Arti atau Fungsi</i>
%c	untuk menampilkan karakter
%s	untuk menampilkan teks (string)
%d, %i	untuk menampilkan bilangan desimal
%f	untuk menampilkan bilangan pecahan
%o	untuk menampilkan bilangan oktal
%x	untuk menampilkan bilangan heksadesimal
\t	untuk membuat tabs

Simbol tersebut disebut dengan *Format Specifier*, yang merupakan format yang digunakan untuk input dan output dalam C. Sementara *Escape Sequence* adalah penulisan karakter yang tidak bisa diwakili secara langsung dalam penulisan karakter atau string literal.

<code>\a</code>	<i>Alarm or Beep</i>
<code>\b</code>	<i>Backspace</i>
<code>\f</code>	<i>Form Feed</i>
<code>\n</code>	<i>New Line</i>
<code>\r</code>	<i>Carriage Return</i>
<code>\t</code>	<i>Tab (Horizontal)</i>
<code>\v</code>	<i>Vertical Tab</i>
<code>\\</code>	<i>Backslash</i>
<code>\'</code>	<i>Single Quote</i>
<code>\"</code>	<i>Double Quote</i>
<code>\?</code>	<i>Question Mark</i>
<code>\ooo</code>	<i>octal number</i>
<code>\xhh</code>	<i>hexadecimal number</i>
<code>\0</code>	<i>Null</i>



2. Fungsi puts()

Fungsi `puts()` adalah fungsi untuk menampilkan output. Fungsi ini juga terdapat pada library `stdio.h`. Perbedaanya dengan `printf()`, fungsi `puts()` tidak memerlukan format. Fungsi `puts()` akan selalu membuat baris baru. Jadi tidak perlu menggunakan simbol `\n` seperti pada `printf()`.



```
#include <stdio.h>

int main(){
    puts("Hello World");
    puts("Nama Saya Dian");

    return 0;
}
```

Selain fungsi `puts()`, terdapat juga fungsi `putchar()` yang khusus untuk menampilkan satu karakter saja.

```
#include <stdio.h>

int main(){
    putchar('A');

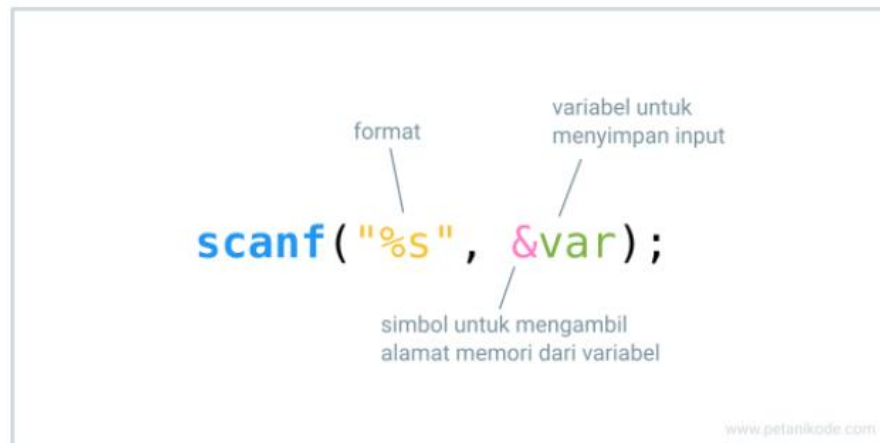
    return 0;
}
```

C. Fungsi Input pada C

Pada bahasa C, terdapat beberapa fungsi untuk mengambil input. Ini adalah fungsi untuk mengambil input dari keyboard:

1. Fungsi scanf()

Fungsi scanf() adalah fungsi untuk mengambil input dari keyboard. Fungsi ini memiliki format seperti fungsi printf().



Format yang diberikan tergantung dari jenis tipe data apa yang ingin kita ambil. Misalnya mau mengambil angka, maka kita bisa pakai `%d` atau `%i`.

```
#include <stdio.h>

int main () {
    // membuat variabel
    char name[20], web_address[30];

    printf("Nama: ");
    scanf("%s", &name);

    printf("Alamat web: ");
    scanf("%s", &web_address);

    printf("\n-----\n");
    printf("Nama yang diinputkan: %s\n", name);
    printf("Alamat Web yang diinputkan: %s\n", web_address);

    return 0;
}
```

Tapi saat kita menginputkan teks yang mengandung spasi, hasilnya akan dipecah menjadi dua. Untuk mengatasi masalah ini, kita bisa ubah format yang digunakan pada scanf() menjadi seperti ini:

```
printf("Nama: ");  
scanf("%[^\n]s", name);
```

Maka fungsi scanf() akan menerima spasi.

Dalam menggunakan scanf(), kita dianjurkan menggunakan simbol & sebelum nama variabel.

```
#include <stdio.h>  
  
void main(){  
  
    int a, b, c;  
  
    printf("Inputkan nilai a: ");  
    scanf("%i", &a);  
  
    printf("Inputkan nilai b: ");  
    scanf("%i", &b);  
  
    c = a + b;  
  
    printf("Hasil a + b: %i", c);  
  
}
```

Simbol & berfungsi untuk mengambil alamat memori dari sebuah variabel. Fungsi scanf() membutuhkan tempat untuk menyimpan nilai yang akan diinputkan. Karena itu kita memberikan simbol & di depan nama variabel untuk menentukan alamat memori yang akan digunakan oleh scanf().

2. Fungsi gets()

Fungsi gets() adalah fungsi untuk mengambil input dalam satu baris. Fungsi gets() tidak memerlukan format seperti scanf().



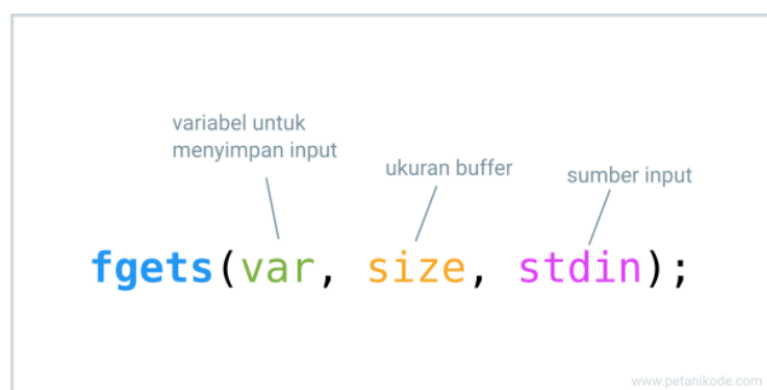
Saat melakukan kompilasi, kita akan mendapatkan peringatan:

```
warning: the `gets' function is dangerous and should not be used.
```

Ini bukan error, melainkan hanya peringatan saja agar tidak menggunakan fungsi `gets()`. Mengapa? Karena dapat menyebabkan masalah buffer overflow pada program. Lalu bagaimana solusinya? Kita bisa menggunakan `fgets()`.

3. Fungsi fgets()

Fungsi `fgets()` lebih aman dibandingkan `gets()`, karena kita bisa menentukan ukuran buffer dan sumber inputan.



Ukuran buffer adalah batas ukuran panjang string yang diinputkan. Lalu `stdin` adalah sumber inputan dari keyboard. Kita juga bisa mengambil input dari file dengan `fgets()` dengan memasukan pointer FILE pada `stdin`.

```
#include <stdio.h>

int main(){
    char name[50], email[50];

    printf("Nama: ");
    fgets(name, sizeof(name), stdin);

    printf("Email: ");
    fgets(email, sizeof(email), stdin);

    printf("\n-----\n");
    printf("Nama anda: %s", name);
    printf("Alamat email: %s", email);

    return 0;
}
```

Pada contoh program di atas, kita menggunakan fungsi `sizeof()` untuk mengambil ukuran buffer dari variabel. Ukuran ini akan menjadi batas maksimum dari inputan.

LATIHAN

1. Buatlah sebuah program dengan algoritma berikut;
 - Program menampilkan tulisan “Hello, siapa nama lengkapmu?”
 - Pengguna menginputkan nama lengkap <nama>
 - Menampilkan pesan “Selamat Datang <nama> dalam Pemrograman C!”
2. Buatlah sebuah program yang meminta inputan nama, NIM, program studi, fakultas, Nilai Praktikum, UTS, UAS. Hitung Nilai Akhir!
Nilai Akhir = 30% Nilai Praktikum + 30% UTS + 40% UAS

Keluaran output yang diharapkan:

Nama :
NIM :
Prodi :
Fakultas :
Nilai Praktikum :
Nilai UTS :
Nilai UAS :
Nilai Akhir :

3. Buatlah program untuk menghitung luas dari segitiga berikut, jika diketahui panjang alas = 8 cm, dan tinggi = 5 cm.
4. Buatlah program untuk menghitung luas persegi panjang dengan panjang = 10 cm, dan lebar = 5 cm.

REFERENCES

1. The C Programming Language. 2nd Edition
2. [https://en.wikipedia.org/wiki/C_\(programming_language\)](https://en.wikipedia.org/wiki/C_(programming_language))
3. https://en.wikipedia.org/wiki/Imperative_programming
4. <https://www.petanikode.com/tutorial/c/>
5. <https://www.learn-c.org/>
6. <https://www.tutorialspoint.com/cprogramming/index.htm>
7. <https://www.programiz.com/>
8. <https://www.dicoding.com/>
9. <https://data-flair.training/blogs/c-tutorials-home/>