

# PEMROGRAMAN KOMPUTER

C Language Series

## MODUL 1

(Struktur Dasar Bahasa C dan Input-Output)

Tim Pengajar Algoritma dan Pemrograman

## Struktur Dasar Bahasa C dan Input-Output

---

Nama Mata Kuliah	: Praktikum Pemrograman Komputer
Bobot sks	: 1 SKS
Semester	: II (Dua)
Koordinator MK	: I Made Widiartha, S.Si.,M.Kom
Modul Praktikum	: Modul 1 – Struktur Dasar Bahasa C dan Input-Output

---

### 1. Deskripsi

Pada praktikum ini akan dipraktekan tentang struktur dasar Bahasa C dan fungsi input-output yang meliputi: format penulisan kode program; deklarasi tipe data; deklarasi konstanta; deklarasi variabel; operator; komentar; printf; dan scanf.

### 2. Tujuan Umum

Setelah mengikuti praktikum ini, mahasiswa mampu menggunakan berbagai tipe data yang ada pada bahasa pemrograman C dalam sebuah variabel sesuai dengan kebutuhan program yang ada. Mahasiswa juga diharapkan mampu memanfaatkan operator-operator yang dimiliki oleh bahasa pemrograman C yang akan digunakan ke dalam proses-proses logika maupun perhitungan sederhana beserta inputan dan keluarannya pada sebuah program.

### 3. Indikator Pencapaian

Adapun indikator pencapaian dari modul praktikum ini adalah mahasiswa dapat mengetahui struktur dasar bahasa pemrograman C dan mengimplementasikan ke dalam sebuah program sederhana.

## Struktur Dasar Bahasa C dan Input-Output

### 4. Teori

#### 4.1. Struktur Dasar Bahasa C

Program Bahasa C tidak mengenal aturan penulisan di kolom tertentu, jadi bisa dimulai dari kolom manapun. Namun demikian, untuk mempermudah pembacaan program dan untuk keperluan dokumentasi, sebaiknya penulisan bahasa C diatur sedemikian rupa sehingga mudah dan enak dibaca. Berikut contoh penulisan Program Bahasa C yang baik dan yang kurang baik:

```
#include "stdio.h"
void main()
{
    printf("Bahasa C\n");
}
```

```
#include "stdio.h"
void main() { printf("Bahasa C"); }
```

Kedua Program di atas bila dijalankan akan menghasilkan hasil yang sama berupa tulisan "Bahasa C" di layar, namun dari segi penulisannya program yang pertama tampaknya lebih mudah dibaca dan lebih rapih dibanding dengan program yang kedua.

##### 4.1.1. Tipe Data

Tipe data merupakan bagian program yang paling penting karena tipe data mempengaruhi setiap instruksi yang akan dilaksanakan oleh computer. Misalnya saja 5 dibagi 2 bisa saja menghasilkan hasil yang berbeda tergantung tipe datanya. Jika 5 dan 2 bertipe integer maka akan menghasilkan nilai 2, namun jika keduanya bertipe float maka akan menghasilkan nilai 2.5000000. Pemilihan tipe data yang tepat akan membuat proses operasi data menjadi lebih efisien dan efektif.

Dalam bahasa C terdapat lima tipe data dasar, yaitu :

No	Tipe Data	Ukuran	Range (Jangkauan)	Format	Keterangan
1	char	1 byte	- 128 s/d 127	%c	Karakter/string
2	int	2 byte	- 32768 s/d 32767	%i , %d	Integer/bilangan bulat
3	float	4 byte	- 3.4E-38 s/d 3.4E+38	%f	Float/bilangan pecahan
4	double	8 byte	- 1.7E-308 s/d 1.7+308	%lf	Pecahan presisi ganda
5	void	0 byte	-	-	Tidak bertipe

## Struktur Dasar Bahasa C dan Input-Output

---

```
{
    int x;
    float y;
    char z;
    double w;
    clrscr();           /* untuk membersihkan layar */
    x = 10;             /* variable x diisi dengan 10 */
    y = 9.45;           /* variable y diisi dengan 9.45 */
    z = 'C';            /* variable z diisi dengan karakter "C" */
    w = 3.45E+20;        /* variable w diisi dengan 3.45E+20 */
    printf("Nilai dari x adalah : %i\n", x); /* Menampilkan isi variable x */
    printf("Nilai dari y adalah : %f\n", y); /* Menampilkan isi variable y */
    printf("Nilai dari z adalah : %c\n", z); /* Menampilkan isi variable z */
    printf("Nilai dari w adalah : %lf\n", w); /* Menampilkan isi variable w */
    getch();
}
```

### 4.1.2. Konstanta

Konstanta merupakan suatu nilai yang tidak dapat diubah selama proses program berlangsung. Konstanta nilainya selalu tetap. Konstanta harus didefinisikan terlebih dahulu di awal program. Konstanta dapat bernilai integer, pecahan, karakter dan string. Contoh konstanta : 50; 13; 3.14; 4.50005; 'A'; 'Bahasa C'. Selain itu, bahasa C juga menyediakan beberapa karakter khusus yang disebut karakter escape, antara lain :

- f \a : untuk bunyi bell (alert)
- f \b : mundur satu spasi (backspace)
- f \f : ganti halaman (form feed)
- f \n : ganti baris baru (new line)
- f \r : ke kolom pertama, baris yang sama (carriage return)
- f \v : tabulasi vertical
- f \0 : nilai kosong (null)
- f \' : karakter petik tunggal
- f \" : karakter petik ganda
- f \\ : karakter garis miring

### 4.1.3. Variabel

Variabel adalah suatu pengenalan (identifier) yang digunakan untuk mewakili suatu nilai tertentu di dalam proses program. Berbeda dengan konstanta yang nilainya selalu

- f Terdiri dari gabungan huruf dan angka dengan karakter pertama harus berupa huruf. Bahasa C bersifat case-sensitive artinya huruf besar dan kecil dianggap berbeda. Jadi antara **nim**, **NIM** dan **Nim** dianggap berbeda.
- f Tidak boleh mengandung spasi.
- f Tidak boleh mengandung symbol-simbol khusus, kecuali garis bawah (underscore). Yang termasuk symbol khusus yang tidak diperbolehkan antara lain : \$, %, #, !, &, \*, (, ), -, +, = dsb
- f Panjangnya bebas, tetapi hanya 32 karakter pertama yang terpakai.

%nilai mahasiswa, 80mahasiswa, rata-rata, ada spasi, penting!, dsb

```
int x; // Deklarasi x bertipe integer char
y, huruf, nim[10]; // Deklarasi variable bertipe char float
nilai; // Deklarasi variable bertipe float double
beta; // Deklarasi variable bertipe double int
array[5][4]; // Deklarasi array bertipe integer char
*p; // Deklarasi pointer p bertipe char
```

## Struktur Dasar Bahasa C dan Input-Output

---

### f Deklarasi Konstanta

Dalam bahasa C konstanta dideklarasikan menggunakan preprocessor `#define`.

Contohnya :

```
#define PHI 3.14
#define nim "0111500382"
#define nama "Sri Widhiyanti"
```

### f Deklarasi Fungsi

Fungsi merupakan bagian yang terpisah dari program dan dapat diaktifkan atau dipanggil di manapun di dalam program. Fungsi dalam bahasa C ada yang sudah disediakan sebagai fungsi pustaka seperti `printf()`, `scanf()`, `getch()` dan untuk menggunakannya tidak perlu dideklarasikan. Fungsi yang perlu dideklarasikan terlebih dahulu adalah fungsi yang dibuat oleh programmer.

Bentuk umum deklarasi sebuah fungsi adalah :

```
Tipe_fungsi nama_fungsi(parameter_fungsi);
```

Contohnya :

```
float luas_lingkaran(int jari);
void tampil();
int tambah(int x, int y);
```

## 4.1.5. Operator

### f Operator Penugasan

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

(`"=`").

Contoh :

```
nilai = 80;
A = x * y;
```

Artinya : variable "nilai" diisi dengan 80 dan variable "A" diisi dengan hasil perkalian

## Struktur Dasar Bahasa C dan Input-Output

---

antara x dan y.

### f Operator Aritmatika

Bahasa C menyediakan lima operator aritmatika, yaitu :

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

Catatan : operator % digunakan untuk mencari sisa pembagian antara dua bilangan.

Misalnya :

- 9 % 2 = 1
- 9 % 3 = 0
- 9 % 5 = 4
- 9 % 6 = 3

### f Operator Hubungan (Perbandingan)

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

Operator	Arti	Contoh	
<	Kurang dari	$x < y$	Apakah x kurang dari y
<=	Kurang dari sama dengan	$x \leq y$	Apakah x kurang dari sama dengan y
>	Lebih dari	$x > y$	Apakah x lebih dari y
>=	Lebih dari sama dengan	$x \geq y$	Apakah x lebih dari sama dengan y
==	Sama dengan	$x == y$	Apakah x sama dengan y
!=	Tidak sama dengan	$x != y$	Apakah x tidak sama dengan y

### f 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)

## Struktur Dasar Bahasa C dan Input-Output

### *f* 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

### *f* Operator Unary

Operator Unary merupakan operator yang hanya membutuhkan satu operand saja.

Dalam bahasa C terdapat beberapa operator unary, yaitu:

Operator	Arti/Maksud	Letak	Contoh	Equivalen
-	Unary minus	Sebelum operator	A + -B * C	A + (-B) * C
++	Peningkatan dengan penambahan nilai 1	Sebelum dan sesudah	A++	A = A + 1
--	Penurunan dengan pengurangan nilai 1	Sebelum dan sesudah	A--	A = A - 1
sizeof	Ukuran dari operand dalam byte	Sebelum	sizeof(I)	-
!	Unary NOT	Sebelum	!A	-
~	Bitwise NOT	Sebelum	~A	-
&	Menghasilkan alamat memori operand	Sebelum	&A	-
*	Menghasilkan nilai dari pointer	Sebelum	*A	-

#### 4.1.6. Komentar

Komentar program hanya diperlukan untuk memudahkan pembacaan dan pemahaman suatu program (untuk keperluan dokumentasi program). Dengan kata lain, komentar program hanya merupakan keterangan atau penjelasan program. Untuk memberikan komentar atau penjelasan dalam bahasa C digunakan pembatas /\* dan \*/



## Struktur Dasar Bahasa C dan Input-Output

---

atau menggunakan tanda // untuk komentar yang hanya terdiri dari satu baris. Komentar program tidak akan ikut diproses dalam program (akan diabaikan).

*Contoh Program :*

```
#include "stdio.h"
#include "conio.h"
void main()
{
    clrscr();          /* Ini untuk membersihkan layar tampilan */
    printf("Contoh Penggunaan Komentar"); //komentar tidak ikut diproses
    getch();          /* Dan ini untuk menahan tampilan di layar */
}
```

## 4.2. Input dan Ouput

### 4.2.1. Masukan Data

Dalam bahasa C proses memasukkan suatu data bisa menggunakan beberapa fungsi pustaka yang telah tersedia. Beberapa fungsi pustaka yang bisa digunakan adalah :

#### f scanf()

- Fungsi pustaka scanf() digunakan untuk menginput data berupa data numerik, karakter dan string secara terformat.
- Hal-hal yang perlu diperhatikan dalam pemakaian fungsi scanf() :
  - ➊ Fungsi scanf() memakai penentu format
  - ➋ Fungsi scanf() memberi pergantian baris secara otomatis
  - ➌ Fungsi scanf() tidak memerlukan penentu lebar field
  - ➍ Variabelnya harus menggunakan operator alamat &

*Kode penentu format :*

- ➊ %c : Membaca sebuah karakter
- ➋ %s : Membaca sebuah string
- ➌ %i, %d : Membaca sebuah bilangan bulat (integer)
- ➍ %f, %e : Membaca sebuah bilangan pecahan (real)
- ➎ %o : membaca sebuah bilangan octal
- ➏ %x : Membaca sebuah bilangan heksadesimal
- ➐ %u : Membaca sebuah bilangan tak bertanda

## Struktur Dasar Bahasa C dan Input-Output

---

### Contoh Program :

```

/* Program memasukan inputan dengan beberapa tipe data */
#include <stdio.h>
#include <conio.h>
void main()
{
    int jumlah;
    char huruf, nim[10];
    float nilai;
    clrscr();

    printf("Masukkan sebuah bilangan bulat : ");
    scanf("%d", &jumlah); /* membaca sebuah bilangan bulat */
    printf("Masukkan sebuah karakter : ");
    scanf("%c", &huruf); /* membaca sebuah karakter */
    printf("Masukkan nim Anda : ");
    scanf("%s", &nim); /* membaca sebuah string */
    printf("Masukkan sebuah bilangan pecahan : ");
    scanf("%f", &nilai); /* membaca sebuah bilangan float */

    printf("\nNilai variable yang Anda masukkan adalah :\n");
    printf("jumlah = %d\n", jumlah);
    printf("huruf = %c\n", huruf);
    printf("nim = %s\n", nim);
    printf("nilai = %f\n", nilai);
    getch();
}

```

### 4.2.2. Menampilkan Data

#### *Menampilkan data ke layer monitor*

- 9 Menggunakan fungsi printf(), puts(), dan putchar().
- 9 Fungsi printf() digunakan untuk menampilkan semua jenis data (numeric dan karakter)
- 9 Fungsi puts() digunakan untuk menampilkan data string dan secara otomatis akan diakhiri dengan perpindahan baris.
- 9 Fungsi putchar() digunakan untuk menampilkan sebuah karakter.

#### *Mengatur tampilan bilangan pecahan (float).*

Bentuk umum :

```
printf("%m.nf", argument);
```

- 9 m : menyatakan panjang range
- 9 n : menyatakan jumlah digit di belakang koma.
- 9 argument : nilai atau variable yang akan ditampilkan.

Contoh :

```
printf("%5.2f", nilai);
```

artinya variable **nilai** akan ditampilkan sebanyak 5 digit dengan 2 digit di belakang koma.

## Struktur Dasar Bahasa C dan Input-Output

---

```

/* Program untuk menampilkan data berupa bilangan pecahan */
#include "stdio.h"
#include "conio.h"
void main()
{   float nilai;
    clrscr();
    puts("Masukkan nilai Anda : "); scanf("%f", &nilai);
    printf("Anda memperoleh nilai %5.2f", nilai);
    printf("Apakah Anda telah puas mendapat nilai %6.4f ?", nilai);
    getch();
}

```

## 5. Referensi

1. Munir, R. 1999. *Algoritma dan Pemrograman Dalam Bahasa Pascal dan C*. Bandung: Informatika.
2. Kadir, A dan Heriyanto. 2005. *Algoritma Pemrograman Menggunakan C++*. Yogyakarta: Penerbit Andi.
3. Tosin, R. 1997. *Flowchart untuk Siswa dan Mahasiswa*. Jakarta: DINASTINDO.

## 6. Pelaksanaan Praktikum

### 6.1. Tools

Tools yang dapat digunakan pada untuk menunjang praktikum kali ini antara lain:

1. Code :: Block IDE sebagai IDE bahasa pemrograman C

### 6.2. Langkah-langkah Praktikum

Berikut akan dipaparkan langkah-langkah pembuatan program sederhana dengan menggunakan bahasa pemrograman C.

1. Pemanggilan library-library yang dibutuhkan oleh program

```

#include <stdio.h>
#include <stdlib.h>

```

## Struktur Dasar Bahasa C dan Input-Output

---

Library di atas merupakan library yang dibutuhkan untuk standar proses input-output pada bahasa pemrograman C. Dalam praktikum kali ini, kita membutuhkan kedua library tersebut untuk memanggil fungsi-fungsi masukan dan keluaran data.

### 2. Pembuatan fungsi utama

```
int main()
{
    return 0;
}
```

Dalam bahasa pemrograman C, pada setiap program yang dibangun haruslah memiliki fungsi utama (*main function*) dalam hal ini yang kita buat adalah `int main() { }`. Fungsi "`main()`" ini berfungsi sebagai fungsi pertama yang akan dieksekusi oleh program, sehingga tanpa adanya fungsi "`main()`" ini, maka tidak ada acuan untuk memulai eksekusi sebuah program.

### 3. Pendefinisian Konstanta

```
#define nama "adi darmawan"
```

Konstanta merupakan suatu nilai yang tidak dapat diubah selama proses program berlangsung. Konstanta nilainya selalu tetap. Konstanta harus didefinisikan terlebih dahulu di awal program sebelum pendefinisian fungsi "`main()`".

## Struktur Dasar Bahasa C dan Input-Output

---

### 4. Pendefinisian dan Inisialisasi Variabel

```
int x;
float y;
char z;
double w;

x = 10;           // variable x diisi dengan 10
y = 9.45;        // variable y diisi dengan 9.45
z = 'A';         // variable z diisi dengan karakter 'A'
w = 3.0123456789; // variable w diisi dengan 3.0123456789
```

### 5. Operator

```
int a = 10;
int b = 4;
int hasil;
float hasil2;

hasil = a + b;
hasil = a - b;
hasil = a * b;
hasil2 = a / b;
hasil = a % b;
```

### 6. Komentar

```
int x;
x = 10; // variable x diisi dengan nilai 10

/*

Nama Program : Program Pertamaku
Creator      : Adi
Deskripsi    : Ini program pertamaku, semoga setelah ini
               aku bisa membuat program yang lebih baik lagi

*/
```

Komentar program hanya diperlukan untuk memudahkan pembacaan dan pemahaman suatu program (untuk keperluan dokumentasi program). Dengan kata lain, komentar program hanya merupakan keterangan atau

## Struktur Dasar Bahasa C dan Input-Output

---

penjelasan kode program. Untuk memberikan komentar atau penjelasan dalam bahasa C dapat menggunakan 2 cara :

- f Tanda // digunakan untuk komentar yang terdiri dari 1 baris saja.
- f Tanda /\* dan \*/ digunakan untuk komentar/ penjelasan program yang lebih dari satu baris.

Komentar program tidak akan ikut diproses dalam program (akan diabaikan).

### 7. Input – Output

```
int bilangan1, bilangan2, hasil;

printf("Masukan Bilangan 1 : ");
scanf("%d", &bilangan1);
printf("Masukan Bilangan 2 : ");
scanf("%d", &bilangan2);

hasil = bilangan1 * bilangan2;

printf("\n\n");
printf("Hasil Perkalian Bilangan 1 dan Bilangan 2 adalah %d", hasil);
printf("\n\n");
```

- f Fungsi "printf()" digunakan untuk menampilkan semua jenis data (numeric dan karakter) ke layar monitor.
- f Fungsi pustaka "scanf()" digunakan untuk menginput data dari user berupa data numerik, karakter dan string secara terformat.

## 7. Laporan

Buatlah sebuah program yang dapat menyimpan data diri mahasiswa berupa nama, NIM, kelas, alamat, nomor hp, dan tanggal lahir. Setelah data dimasukkan oleh pengguna maka nantinya akan menghasilkan output berupa nama, NIM, kelas, alamat, nomor hp, dan tanggal lahir yang diinputkan oleh pengguna tadi serta umur dari pengguna. Untuk mencari umur gunakan tanggal hari ini sebagai acuannya. Jelaskan juga *code* yang telah kamu buat!

## Struktur Dasar Bahasa C dan Input-Output

---

```

Nama: I Komang Tryana Mertayasa
NIM: 1908561044
Kelas: E
Alamat: Jln. Raya Uluwatu
No HP: 082xxxxxx
Tanggal Lahir: Denpasar 14 Maret 2001

*****Terimakasih Telah Mendaftar!*****

Nama: I Komang Tryana Mertayasa
NIM: 1908561044
Kelas: E
Alamat: Jln. Raya Uluwatu
No HP: 082xxxxxx
Tanggal Lahir: Denpasar 14 Maret 2001
Umur: 19
PS C:\Users\User\Desktop\Modul1>

```

Laporan diketik rapih pada kertas A4 dan dikumpulkan pada saat demo program.

Format Penulisan Laporan :

- Font Tulisan : Times New Roman 12 pt
- Font Source Code : Courier New 10 pt
- Line Spacing : Single
- Margin : 4, 3, 3, 3 (left, top, right, bottom)
- Page Numbering : a). Cover = tanpa halaman  
b). Kata Pengantar s/d sebelum BAB I = Center Bottom, angka romawi kecil (i,ii,iii)  
c). BAB = Center Bottom, angka arab (1, 2, 3)  
d). Bagian BAB = Top Right, angka arab (1, 2, 3)

## Struktur Dasar Bahasa C dan Input-Output

---

- Isi Laporan :
  - COVER
  - KATA PENGANTAR
  - DAFTAR ISI
  - BAB I LANDASAN TEORI
  - BAB II PERMASALAHAN
  - BAB III PEMBAHASAN
  - (Tugas Pendahuluan dan
  - Tugas Praktikum)
  - DAFTAR PUSTAKA