

Algoritma dan Pemrograman

bagian 2

2009

Modul ini menjelaskan tentang bahasa C dan apa saja yang dibutuhkan bila kita akan menulis suatu program dengan bahasa C. Editor yang dipakai adalah Turbo C++ 4.5. Struktur program yang akan dipakai adalah struktur program bahasa C.

**Pengenalan C
dan struktur
program**

OPERASI

Ada beberapa operasi yang dapat dilakukan menggunakan bahasa C, yaitu

- a) operasi aritmatika
- b) operasi relasional (pertidaksamaan)
- c) operasi kondisional
- d) operasi logika

1. *Operasi aritmatika*

Operasi ini adalah operasi matematika yang sering kita lakukan. Operasi ini meliputi penjumlahan (+), pengurangan (-), perkalian (*), pembagian (/), dan modulus/sisa hasil bagi (%).

```
Contoh : a = b+c;
         p = q%r;
```

Di samping itu, di dalam bahasa C dikenal operasi aritmatika increment (++) dan decrement (--). Bila operator ini ditulis sebelum operand (pre increment atau pre decrement), maka operand tersebut akan ditambah (untuk increment) atau dikurang (untuk decrement) dengan 1, dan nilai operand tersebut akan berubah sebelum dikenai fungsi atau operasi lain (misal ditampilkan menggunakan fungsi printf). Dan bila operator ditulis setelah operand (post increment atau post decrement), maka operand tersebut akan digunakan terlebih dahulu (misalnya ditampilkan menggunakan fungsi printf) dan setelah itu nilai operand akan ditambah atau dikurang dengan 1.

```
Contoh : int a=5;
         printf("%d",++a); //nilai yang ditampilkan adalah 6 (5+1), a sekarang adalah 6
         int b=3;
         printf("%d",b++); //nilai yang ditampilkan adalah 3, a sekarang adalah 4
```

2. *Operasi relasional*

Operasi relasional adalah operasi pertidaksamaan, seperti yang pernah kita pelajari waktu SMA. Operasi ini membutuhkan 2 operand. Operasi ini bernilai *true* bila operasi tersebut benar. Demikian juga sebaliknya, operasi ini bernilai *false* bila operasi tersebut salah. Operator yang digunakan adalah > (lebih besar), < (lebih kecil), >= (lebih besar atau sama dengan), <= (lebih kecil atau sama dengan), dan != (tidak sama dengan).

```
Contoh : int c=10, d=4;
         if(c>d)
             printf("Algoritma");
         else
             printf("Pemrograman");
         //outputnya adalah Algoritma karena benar bahwa c>d
```

3. *Operasi kondisional*

Operasi kondisional dapat dikaitkan dengan operasi logika. Operator yang dapat digunakan adalah :

&&	op1 && op2	bernilai true bila op1 dan op2 bernilai true
	op1 op2	bernilai true bila op1 atau op2 bernilai true

!	!op1	bernilai true bila op1 bernilai false
&	op1 & op2	bitwise AND
	op1 op2	bitwise OR
^	op1 ^ op2	bernilai true bila salah satu operand bernilai true, tetapi tidak keduanya

Operasi bitwise adalah operasi bilangan biner. Jadi, untuk melakukan operasi ini, komputer akan mengubah bilangan desimal menjadi angka biner dan melakukan operasi AND atau OR, kemudian akan mengubah kembali menjadi bilangan desimal.

```
Contoh : int j = 1;           //bil. binernya = 01
          int k = 2;         //bil. binernya = 10
          printf("%d",j&k); //outputnya adalah 0 karena 01 & 10 = 00
```

4. Operasi logika

Operasi ini dapat dikatakan bagian dari operasi kondisional karena menggunakan operator &&, ||, dan !.

Ada beberapa statement yang dapat ditulis dalam bentuk lain tetapi memiliki arti sama.

```
Contoh : a += b    //statement ini sama dengan a = a+b
          c *= d    //statement ini sama dengan c = c*d
          (demikian juga halnya dengan pengurangan dan pembagian)
```

PEMBERIAN KOMENTAR

Pemberian komentar adalah salah satu hal penting di dalam kita membuat program. Pemberian komentar merupakan salah satu cara untuk *mendokumentasi* program yang kita buat sehingga pada lain kesempatan kita dapat mengingat kembali program yang seperti apa yang telah kita buat dan bagaimana alur program tersebut. Pemberian komentar juga dapat digunakan untuk *mengkomunikasikan* program yang kita buat kepada orang lain. Semua komentar pada bahasa C tidak akan dibaca oleh compiler bahasa C. Untuk komentar yang hanya terdiri dari 1 baris, maka kita hanya perlu untuk menambahkan // pada awal komentar. Untuk komentar yang terdiri dari beberapa baris, kita akan menuliskan /* pada awal komentar dan */ pada akhir komentar.

INPUT DAN OUTPUT

Untuk memberikan suatu input ataupun menampilkan suatu output, ada kode format yang digunakan agar apa yang kita inputkan atau outputkan adalah sesuatu yang terformat. Beberapa kode format yang akan sering kita gunakan di dalam praktikum ini adalah sebagai berikut.

kode format	kegunaan
%c	input atau output berupa sebuah karakter
%s	input atau output berupa string
%d, %i	input atau output berupa integer
%ld, %li	input atau output berupa long integer
%f	input atau output berupa bilangan pecahan (float atau double)

%x	output berupa bilangan hexadesimal
%p	output berupa alamat memory

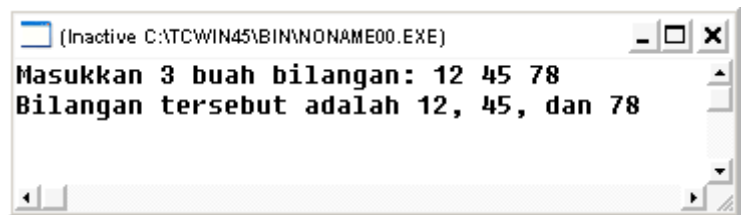
Input

Dalam mata kuliah ini kita tidak akan mempelajari semua fungsi input yang ada di dalam bahasa C. Fungsi-fungsi yang akan kita gunakan adalah :

- a. Di dalam header stdio.h
 - i. gets() – untuk menginputkan string (dapat menerima karakter spasi)
 - ii. scanf() – untuk menginputkan berbagai bentuk data (tidak hanya string)
- b. Di dalam header conio.h
 - i. getche () – input karakter tanpa enter dan karakter terlihat
 - ii. getchar () – input karakter dengan enter dan karakter terlihat
 - iii. getch () – input karakter tanpa enter dan karakter tidak terlihat

Kita dapat menginputkan beberapa nilai sekaligus menggunakan scanf(). Caranya adalah dengan memberikan spasi atau enter bila akan menginputkan nilai berikutnya. Lihat contoh di bawah ini.

```
#include<stdio.h>
void main()
{
    int a,b,c;
    printf("Masukkan 3 buah bilangan: ");
    scanf("%i%i%i",&a,&b,&c);
    printf("Bilangan tersebut adalah %i, %i, dan %i",a,b,c);
}
```



Latihan :

Buatlah sebuah program seperti yang sering kita lihat di mesin ATM. Saat kita memasukkan password, yang terjadi adalah angka yang kita masukkan tidak terlihat, tetapi yang terlihat hanyalah karakter bintang.

contoh :

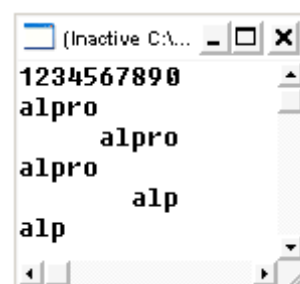
```
Masukkan password Anda : **** //input sekaligus output
Tampilkan password : 1234 //hanya output
```

Output

Di dalam bahasa C, output yang kita tampilkan dapat kita atur sehingga menjadi output lebih terformat. Output terformat dibentuk dari angka yang diletakkan setelah tanda %. Contoh-contoh berikut akan lebih memperjelas.

String :

```
#include<stdio.h>
void main()
{
    char a[10] = "alpro";
    printf("1234567890\n");
    printf("%s\n",a); //rata kiri
```



```

printf("%10s\n",a); //lebar 10 karakter, rata kanan
printf("%-10s\n",a); //lebar 10 karakter, rata kiri
printf("%10.3s\n",a); //lebar 10 karakter, rata kanan dan diambil 3 karakter saja
printf("%-10.3s\n",a); // lebar 10 karakter, rata kiri dan diambil 3 karakter saja
}

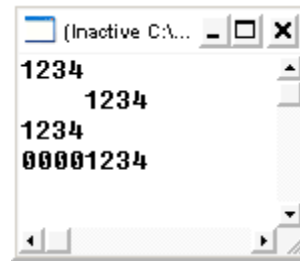
```

Integer :

```

#include<stdio.h>
void main()
{
    int b=1234;
    printf("%i\n",b); //rata kiri
    printf("%8i\n",b); //lebar 8 karakter, rata kanan
    printf("%-8i\n",b); //lebar 8 karakter, rata kiri
    printf("%08i\n",b); //lebar 8 karakter, di sisi kiri pada bagian yang kosong diisi dengan 0
}

```

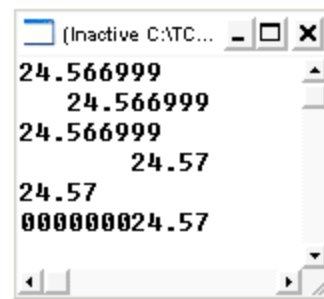


Float (bilangan pecahan) :

```

#include<stdio.h>
void main()
{
    float c=24.567;
    printf("%f\n",c); //rata kiri
    printf("%12f\n",c); //lebar 12 karakter, rata kanan
    printf("%-12f\n",c); //lebar 12 karakter, rata kiri
    printf("%12.2f\n",c); //lebar 12 karakter, rata kanan, dibulatkan 2 angka di belakang koma
    printf("%-12.2f\n",c); //lebar 12 karakter, rata kiri, dibulatkan 2 angka di belakang koma
    printf("%012.2f\n",c); /*lebar 12 karakter, rata kanan, dibulatkan 2 angka di belakang koma, sisi kiri yang kosong diisi dengan 0*/
}

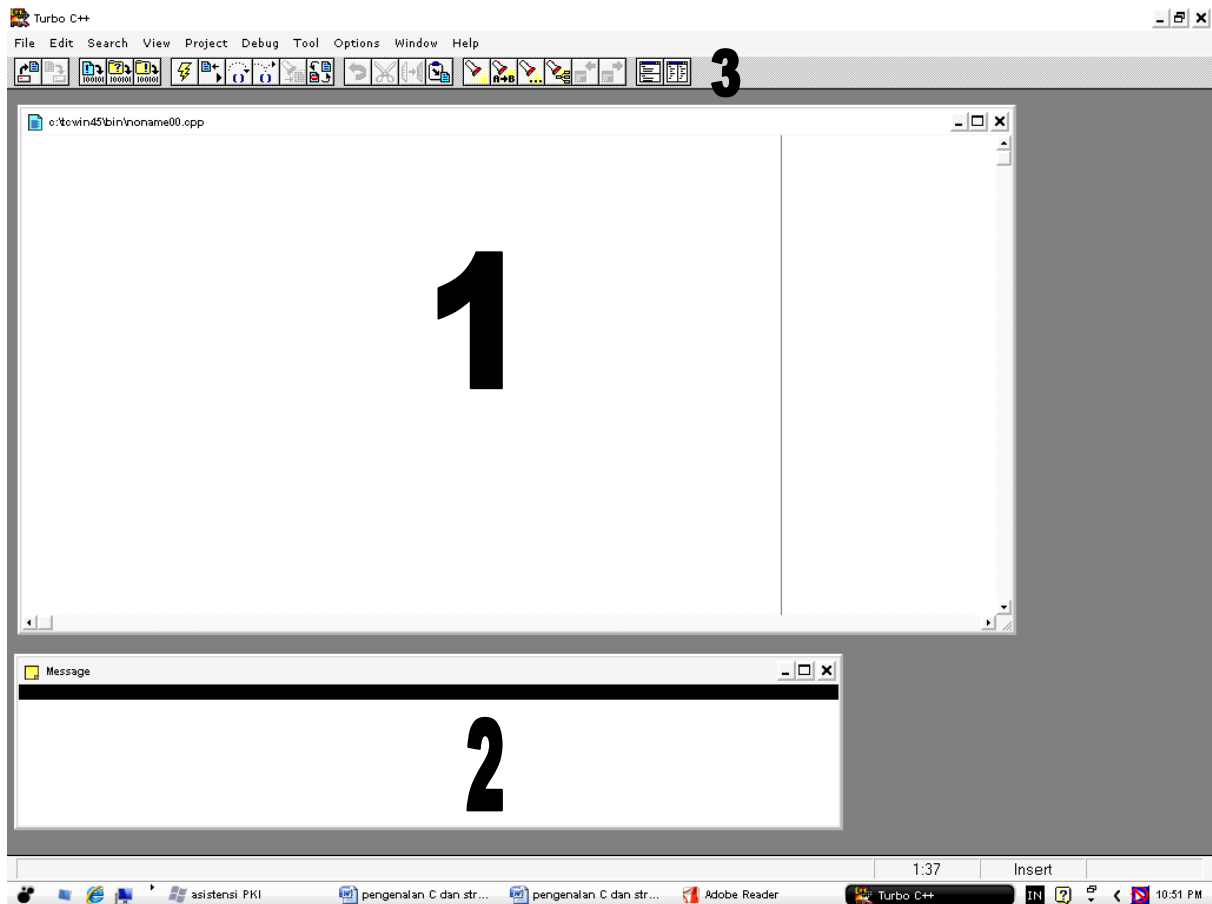
```



➔ Bila kita perhatikan, angka yang tertulis di layar untuk output pertama, kedua, dan ketiga tidak sesuai dengan yang kita deklarasikan. Ini disebabkan tipe bilangan float secara default memiliki 6 angka di belakang koma. Namun, kita dapat membulatkannya seperti yang tertulis pada contoh di atas.

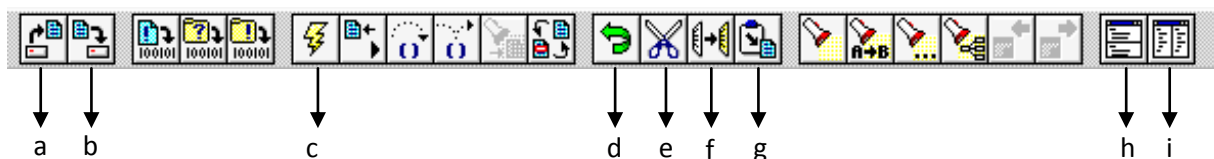
Editor Turbo C++ 4.5

Di dalam praktikum ini kita akan menggunakan editor Turbo C++ 4.5. Editor ini akan membuat source code dengan ekstensi .cpp. Tampilan window-nya adalah sebagai berikut.



Ketiga bagian (yang diberi nomor) tersebut adalah bagian yang akan sering kita gunakan.

1. **Layar tulis**
Layar ini adalah layar yang akan kita gunakan untuk menulis source code untuk program yang kita buat.
2. **Message**
Layar ini digunakan untuk menampilkan peringatan-peringatan, misalnya menampilkan error yang kita lakukan.
3. **Toolbar**
Seperti window lain, window ini juga memiliki toolbar yang berisi icon-icon yang menjalankan perintah-perintah tertentu. Beberapa icon yang akan kita gunakan adalah :



- a. Open a File
Untuk membuka source code (.cpp) yang sudah kita buat.
- b. Save File As
Untuk menyimpan source code (.cpp) yang kita buat ke dalam harddisk.
- c. Run (Ctrl + F9)
Untuk menjalankan (mengeksekusi) source code yang kita buat. Hal ini akan menyebabkan terciptanya sebuah file baru dengan ekstensi .exe.
- d. Undo (Ctrl + z)
Untuk membatalkan perintah. Sedangkan untuk mengembalikan perintah yang kita batalkan, kita dapat melakukan Redo (Shift + Ctrl + z).
- e. Cut
Untuk mendelete bagian yang kita blok dan mem-paste-nya ke tempat lain.
- f. Copy
Untuk menggandakan bagian yang kita blok ke tempat yang lain.
- g. Paste
Untuk memunculkan bagian yang kita cut atau copy.
- h. Icon ini digunakan untuk membagi layar tulis menjadi beberapa bagian secara horizontal.
- i. Icon ini digunakan untuk membagi layar tulis menjadi beberapa bagian secara vertikal.