

SEJARAH PENGEMBANGAN COBOL

Fatur

artur@gmail.com

Lisensi Dokumen:

Copyright © 2008 mr-amateur.co.cc

Seluruh dokumen di mr-amateur.co.cc dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari mr-amateur.co.cc.

Materi Pembahasan :

1. Sejarah perkembangan bhs C
2. Pengenalan Program C
3. Type data, Konstanta, Variabel, & Operator pada C
4. Latihan membuat program sederhana menggunakan C

Tujuan Praktikum

Praktikan mengetahui sejarah perkembangan bahasa C

Praktikan mengerti struktur program bahasa C

Praktikan mengerti konsep tipe data dalam bahasa C

Praktikan dapat membuat program sederhana dengan menggunakan bahasa C

PEMBAHASAN

1 Sejarah Perkembangan Bahasa C

Berasal dari bahasa BCPL (Basic Combined Programming Language) oleh MARTIN RICHARD, Cambridge tahun 1967, KEN THOMPSON membuat bahasa B untuk dipakai pada komputer DEC PDP-7 dibawah sistem operasi UNIX pada Bell laboratory, Murray Hill, New Jersey tahun 1970.

Bahasa B merupakan suatu bahasa pemrograman yang tidak memiliki jenis suatu data seperti halnya PL/M. Berdasarkan gambaran bahasa B, DENNIS RITCHIE menulis bahasa C. Nama C diambil berdasarkan urutan sesudah B dari bahasa BCPL. Tujuan bahasa C pada mulanya untuk

membentuk suatu sistem operasi yang akan digunakan pada mesin komputer DEC PDP-11 yang baru.

Pada tahun 1975, sistem operasi UNIX versi 6 dan bahasa C mulai diberikan kepada Universitas maupun Akademi. Dan pada tahun 1979, sistem operasi UNIX versi 7 dikeluarkan dengan bahasa C. Sistem operasi ini (versi 7) seluruhnya ditulis dalam bahasa C.

Pada 1978 Dennis Ritchie dan Brian Kernighan kemudian mempublikasikan buku *The C Programming Language* yang semakin memperluas pemakaiannya dan dijadikan standar oleh *ANSI* (*American National Standard Institute*) pada tahun 1989. C kemudian dikembangkan lagi oleh Bjarne Stroustrup menjadi C++ (1986). C dan/atau C++ banyak digunakan (sehingga menjadi 'standar') sebagai bahasa pemrograman untuk membuat sistem operasi.

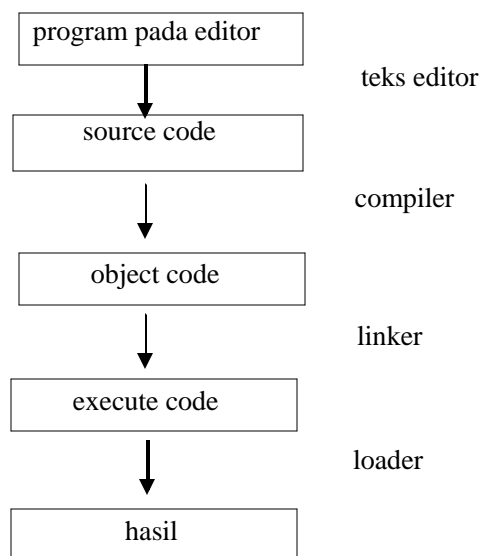
2 Pengenalan Program C

2.1 Proses Penerjemahan bahasa C

Untuk dapat dimengerti oleh komputer, bahasa C yang ditulis harus diterjemahkan terlebih dahulu ke dalam bentuk yang dikenal oleh bahasa mesin. Ada dua jenis translator atau penerjemah yang digunakan dalam bahasa C, yaitu interpreter dan compiler.

Interpreter merupakan suatu translator yang menerjemahkan bahasa C ke dalam bahasa yang dikenal mesin satu persatu, dan hasil terjemahan langsung dikerjakan. Sedangkan pada compiler, diterjemahkan secara keseluruhan dan hasil terjemahan tersebut disimpan dalam bentuk kode mesin (object code), dan kode eksekusi(execute code).

Object code dari compiler baru dapat dilaksanakan setelah object code tersebut diubah menjadi execute code oleh linker. Secara lengkap prosesnya adalah sebagai berikut :



Gambar 2.1 Proses penerjemahan bahasa C

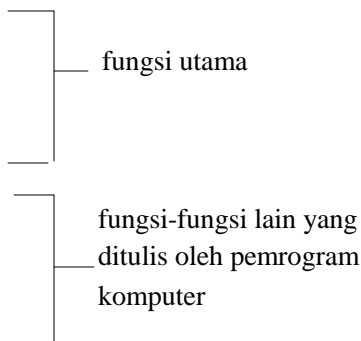
2.2 Struktur Program C

Untuk dapat memahami bagaimana suatu program ditulis, maka struktur dari program harus dimengerti terlebih dahulu. Tiap bahasa komputer mempunyai struktur program yang berbeda. Jika struktur dari program tidak diketahui, maka akan sulit bagi pemula untuk memulai menulis suatu program dengan bahasa yang bersangkutan.

Struktur dari program C terdiri dari koleksi satu / lebih fungsi-fungsi. Fungsi pertama yang harus ada di program C sudah ditentukan namanya, yaitu bernama **main()**. Suatu fungsi didalam program C dibuka dengan kurung kurawal buka ({) dan ditutup dengan kurung kurawal tutup (}). Diantara kurung kurawal dapat dituliskan statemen-statemen program C dan pada setiap statemen diakhiri dengan tanda titik koma (;). Berikut adalah struktur dari program C :

```
main()
{
    statement-statement;
}

Fungsi_Fungsi_Lain()
{
    statemen-statemen ;
}
```



Bahasa C dikatakan sebagai bahasa pemrograman terstruktur, karena strukturnya menggunakan fungsi-fungsi sebagai program-program bagian (subroutine). Fungsi-fungsi selain fungsi utama merupakan program-program bagian. Fungsi-fungsi ini dapat ditulis setelah fungsi utama atau diletakkan pada file pustaka dan akan dipakai di suatu program, maka nama judulnya (header file) harus dilibatkan di dalam program yang menggunakan *preprocessor directive* **#include**.

Header file merupakan file yang berisi dengan prototype (judul, nama, sintak) dari sekumpulan fungsi-fungsi pustaka tertentu. Jadi file ini hanya berisi dengan prototype dari fungsi-fungsi pustaka, sedangkan fungsi-fungsi pustakanya sendiri disimpan dalam file pustaka (library file dengan nama extension file-nya adalah .LIB) Misalnya prototype dari fungsi-fungsi pustaka printf() dan scanf() terdapat di file stdio.h, sehingga jika fungsi-fungsi ini digunakan di program, maka nama file judulnya harus dilibatkan dengan menggunakan preprocessor #include. File judul stdio.h berisi prototype fungsi-fungsi pustaka untuk operasi input dan output standar. Ada dua cara melibatkan file judul disuatu program C, yaitu :

```
#include<stdio.h>

atau

#include "stdio.h"
```

File judul selain berisi dengan prototype dari fungsi-fungsi pustaka, juga umumnya berisi dengan konstanta-konstanta terdefinisi dan makro-makro. Misalnya nama konstanta terdefinisi M_PI telah didefinisikan di file judul math.h oleh Turbo-C. Selanjutnya untuk menggunakan nilai phi, nama konstanta M_PI dapat digunakan yang telah berisi dengan nilai konstanta 3.14159265358979323846.

2.3 Program C Sederhana

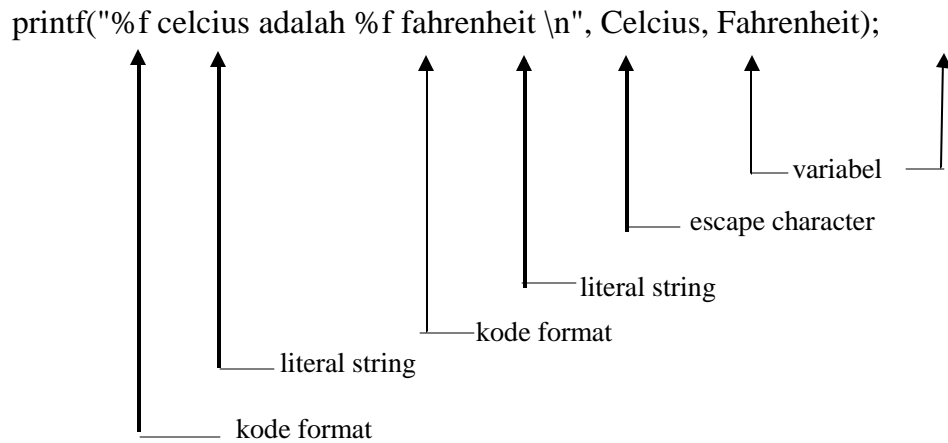
Berikut merupakan contoh program sederhana menggunakan bahasa C :

```
/* Program C yang sederhana */
#include <stdio.h>
main()
{
    float Celcius, Fahrenheit;
    printf("masukkan nilai celcius?");
    scanf("%f",&Celcius);
    Fahrenheit = Celcius * 1.8 + 3.2;
    printf("%f celcius adalah %f fahrenheit \n", Celcius, Fahrenheit);
}
```

Output dari listing program diatas adalah :
masukkan nilai celcius?10
10.000000 celcius adalah 50.000000 fahrenheit

Pembahasan program secara singkat :

1. Untuk memberi keterangan program, suatu komentar bebas dapat diletakkan dimanapun di program C. Komentar atau keterangan program diawali dengan bentuk `/*` dan diakhiri dengan bentuk `*/` yang terlihat sebagai berikut :
`/* Program C yang sederhana */`
2. Karena program ini menggunakan fungsi-fungsi pustaka `printf()` dan `scanf()` yang disediakan oleh C dan diletakkan di file pustaka, maka nama file yang berisi prototype dari fungsi-fungsi ini (file ini disebut header file) harus disebutkan dengan preprocessor directive `#include`. File judul (header file) untuk fungsi-fungsi `printf()` dan `scanf()` adalah dengan nama `stdio.h`. Nama extension file `.h` menunjukkan duatu header file.
3. Di dalam fungsi utama, digunakan lima buah statemen sebagai berikut :
`float Celcius, Fahrenheit;`
`printf("masukkan nilai celcius?");`
`scanf("%f",&Celcius);`
`Fahrenheit = Celcius * 1.8 + 3.2;`
`printf("%f celcius adalah %f fahrenheit \n", Celcius, Fahrenheit);`
Masing-masing statemen ditulis dengan diakhiri oleh titik koma. Statemen statemen (pernyataan-pernyataan) di program C dapat dibentuk dari kata-kata kunci (seperti misalnya `float`, `int`, `for` atau `if`), dibentuk dari fungsi-fungsi (misalnya `printf()`, `scanf()`) atau dibentuk dari suatu pengerjaan (seperti perhitungan, memindahkan suatu nilai ke suatu variabel)
4. Semua variabel yang digunakan di program C harus dideklarasikan terlebih dahulu. Deklarasi ini mempunyai maksud memberitahukan kepada C tipe dari variabel-variabel yang akan digunakan. Di contoh program ini digunakan dua buah variabel, yaitu `Celcius` dan `Fahrenheit`. Variabel-variabel ini dimaksudkan untuk dapat menyimpan nilai numerik pecahan (floating-point), sehingga harus dideklarasikan sebagai tipe `float`, sebagai berikut :
`float Celcius, Fahrenheit;`
5. Salah satu cara untuk menampilkan hasil di layar adalah dengan menggunakan statemen yang dibentuk dari fungsi standar `printf()`. Statemen terakhir di contoh program adalah statemen untuk menampilkan hasil di layar sebagai berikut :



è Kode format

Menunjukkan format dari variabel yang akan ditampilkan. Dalam contoh terdapat dua variabel yang akan ditampilkan nilainya, sehingga digunakan dua buah kode format, satu untuk variabel Celcius dan satu lagi untuk variabel Fahrenheit. Kode format `"%f"` menunjukkan tipe dari variabelnya adalah numerik pecahan (floating point). Jika tipe variabelnya adalah numerik integer (nilai bulat yang dihasilkan dengan kata kunci `int`).

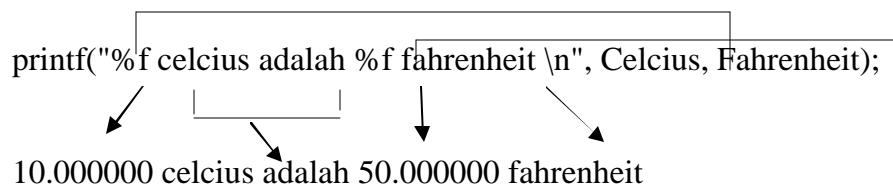
è Literal string

Adalah suatu konstanta string yang mempunyai bentuk yang tetap dan akan ditampilkan sesuai dengan apa yang ditulis.

è Escpae character

Merupakan suatu konstanta karakter yang ditulis dengan diawali oleh garis miring.

Hasil dari statement yang dibentuk dari fungsi `printf()` ini adalah sebagai berikut:



6. Salah satu cara untuk memasukkan data dari keyboard adalah dengan menggunakan fungsi pustaka `scanf()` sebagai berikut :

```
scanf("%f",&Celcius);
```

Di fungsi ini, yang ditulis iantara tanda petik dua adalah kode format dan yang ditulis diluar tanda petik dua adalah variabel yang akan digunakan untuk menerima nilai yang diketikkan dari keyboard. Untuk fungsi `scanf()`, nama variabelnya harus ditulis dengan diawali variabel pointer `&`, sehingga pada contoh untuk variabel Celcius ditulis menjadi `&Celcius`.

3 Tipe Data , Konstanta, Variabel, & Operator

3.1 Tipe Data

Bahasa C menyediakan lima macam tipe data dasar, yaitu tipe data integer (nilai numerik bulat yang dideklarasikan dengan `int`), floating-point (nilai numerik pecahan ketepatan tunggal yang

dideklarasikan dengan float), double-precision (nilai numerik pecahan ketepatan ganda yang dideklarasikan dengan double), karakter (dideklarasikan dengan char), dan kosong (dideklarasikan dengan void). Int, float, double dan char dapat dikombinasikan dengan pengubah (modifier) signed, unsigned, long, dan short. Hasil dari kombinasi tersebut diantaranya dapat dilihat pada tabel berikut :

Tipe data	length	range
unsigned char	8 bit	0 s/d 255
char	8 bit	-128 s/d 127
enum	16 bit	-32,768 s/d 32,767
unsigned int	16 bit	0 s/d 65,535
short int	16 bit	-32,768 s/d 32,767
int	16 bit	-32,768 s/d 32,767
unsigned long	16 bit	0 s/d 4,294,967,295
long	32 bit	-2,147,483,648 s/d 2,147,483,647
float	32 bit	$3.4 * (10^{*-38})$ s/d $3.4 * (10^{*+38})$
double	32 bit	$1.7 * (10^{*-308})$ s/d $1.7 * (10^{*+308})$
long double	32 bit	$3.4 * (10^{*-4932})$ to $1.1 * (10^{*+4932})$

3.1.1 Tipe Data Numerik Integer

Karakteristik dari nilai numerik integer adalah sebagai berikut :

- Nilai numerik pecahan yang disimpan di tempat nilai integer akan dibulatkan ke bawah. Misalkan nilai 34.56 disimpan dan ditampilkan sebagai nilai 34 untuk variabel numerik integer.
- Nilai numerik integer negatif disimpan di memori dengan cara komplemen dua. Misal nilai numerik 7 akan disimpan di memori untuk tipe numerik integer 16 bit dalam bentuk nilai binary 111 dengan perhitungan :

$$\begin{array}{rcl}
 1 \times 2^2 & = & 1 \times 4 = 4 \\
 1 \times 2^1 & = & 1 \times 2 = 2 \\
 1 \times 2^0 & = & 1 \times 1 = 1 \\
 & & \text{-----} + \\
 & & 7
 \end{array}$$

- Nilai variabel yang melebihi jangkauannya akan dipotong sepanjang jumlah bit yang tersedia.

3.1.2 Tipe Data Karakter

Variabel tipe karakter dideklarasikan dengan tipe char atau signed char atau unsigned char dan dimaksudkan untuk menampung nilai sebuah huruf (karakter). Kode format "%c" digunakan untuk menampilkan nilai tipe char. Karakter yang disimpan di memori dengan deklarasi tipe char menempati posisi 1 byte (8 bit) yang diwakili kode ASCII. Misal karakter 'A' dalam memori akan diwakilkan dengan nilai biner 01000001. Nilai binari itu dalam desimal bernilai 65.

3.1.3 Tipe Data String

Bahasa C tidak menyediakan deklarasi variabel untuk tipe string. Nilai string adalah kumpulan dari nilai-nilai karakter yang berurutan dalam bentuk larik dimensi satu. Larik dimensi satu dideklarasikan dengan menyebutkan jumlah dari elemennya yang ditulis diantara operator '[' (bahasa C menganggap '[' sebagai operator. Kode format untuk menampilkan nilai string adalah "%s".

3.2 Konstanta

Konstanta adalah suatu nilai yang tidak berubah selama proses dari program. Misalnya suatu statemen ungkapan sebagai berikut :

Fahrenheit=Celcius*1.8+32;

Celcius dan Fahrenheit adalah variabel yang nilainya dapat berubah selama proses program. Nilai Celcius dapat berubah tergantung dari nilai yang dimasukkan sebagai input data dan nilai Fahrenheit akan berubah tergantung nilai dari Celcius. Nilai 1.8 dan 32 sebaliknya tidak akan pernah berubah di dalam proses program, karena nilai-nilai ini adalah nilai-nilai konstanta.

3.2.1 Konstanta Numerik Integer

Konstanta numerik integer merupakan nilai numerik bilangan bulat. Konstanta-konstanta integer dapat ditulis dalam bentuk desimal, heksadesimal, maupun oktal. Berikut contoh dalam bentuk nilai desimal :

123	(integer bertanda)
-123	(integer bertanda)
37000U	(integer tidak bertanda)
-75000L	(integer panjang bertanda)
4185988210UL	(integer panjang tidak bertanda)

Bila user ingin suatu konstan diinterpretasikan sebagai oktal, tambahkan digit 0 didepan.

Contoh :

015 ekivalen dengan 13 dalam decimal

Jika yang diinginkan bilangan hexadesimal, tambahkan 0x di depan.

Contoh :

0x20 ekivalen dengan 32

3.2.2 Konstanta Numerik Pecahan

Merupakan nilai numerik yang dapat mempunyai nilai pecahan dibelakang titik desimal.

Konstanta numerik pecahan juga dapat ditulis dengan notasi saintifik (dengan notasi e atau E). Berikut contoh-contoh konstanta numerik pecahan :

123.
-123.00
12.34e29

3.2.3 Konstanta Karakter dan Konstanta String

Konstanta karakter merupakan nilai sebuah karakter yang ditulis diantara tanda petik tunggal. Konstanta string merupakan nilai sebuah atau lebih karakter yang ditulis dalam tanda petik ganda. Pengertian karakter adalah semua karakter yang sah seperti yang didefinisikan oleh kode ASCII, yaitu dapat meliputi huruf(a samapi z), digit (0 samapi 9), karakter-karakter khusus(misal +=\$). Berikut contoh dari konstanta-konstanta karakter dan string :

'a'	konstanta karakter huruf a bernilai ASCII 97
'7'	konstanta karakter huruf a bernilai ASCII 55
"a"	konstanta string huruf a

3.2.4 Konstanta Karakter Escape

Konstanta ini banyak digunakan distatemen-statement untuk menampilkan hasil, misal membuat kursor kemabli ke kolom pertama(carriage return).

Karakter	Keterangan
\n	Pindah kursor ke baris baru
\t	Horisontal tab, pindah kursor ke posisi tab
\a	Membunyikan bel sistem atau beep
\\	Mencetak atau menampilkan tanda \ (backslash)
\"	Mencetak atau menampilkan "

3.3 Operator

Operator atau tanda operasi adalah suatu tanda atau simbol yang digunakan untuk suatu operasi tertentu. Bahasa C menyediakan banyak sekali tanda-tanda operasi, yaitu operator pengerjaan, operator aritmatika, operator tipe, operator hubungan, operator logika, dan lain-lain.

3.3.1 Operator Aritmatika

Operator	Deskripsi	Contoh	Hasil
*	Perkalian	2*3	6
/	Pembagian	9/3	3
%	Pembagian Modulus (pembagian sisa)	2+3	5
-	Pengurangan	3-2	1
+	Pertambahan	10%3	1

Prioritas operator aritmatika

Operator	Operasi	Keterangan
()	Parenthesis	Operasi didalam tanda () akan dikerjakan terlebih dahulu. Apabila bersarang, yang dikerjakan pertama kali adalah tanda () yang paling dalam. Apabila terdapat tanda ()

		satu level (tidak bersarang) dikerjakan dari kiri sampai kanan.
*, /, %	Kali, bagi, modulo	Dievaluasi kedua setelah parenthesis. Apabila terdapat beberapa, dieksekusi dari kiri dan kanan.
-, +	Tambah, kurang	Dievaluasi terakhir. Apabila terdapat beberapa, dieksekusi dari kiri dan kanan.

Step 1. $y = 2 * 5 * 5 + 3 * 5 + 7;$ (Leftmost multiplication)
 $2 * 5$ is 10

Step 2. $y = 10 * 5 + 3 * 5 + 7;$ (Leftmost multiplication)
 $10 * 5$ is 50

Step 3. $y = 50 + 3 * 5 + 7;$ (Multiplication before addition)
 $3 * 5$ is 15

Step 4. $y = 50 + 15 + 7;$ (Leftmost addition)
 $50 + 15$ is 65

Step 5. $y = 65 + 7;$ (Last addition)
 $65 + 7$ is 72

Step 6. $y = 72;$ (Last operation—place 72 in y)

3.3.2 Operator Pengerjaan

digunakan untuk memindahkan nilai dari suatu ungkapan ke suatu pengenalan.

Operator	Contoh	Keterangan
=	$A=B+C$	Mengerjakan $B+C$ ke A
+=	$A+=1$	$A=A+1$
++	$A++B$	$A=A+B$
=	$A=B$	$A=A*B$
/=	$A/=B$	$A=A/B$
%=	$A\%=B$	$A=A\%B$

Contoh :

Misalnya variabel I dan J adalah variabel-variabel type integer dengan nilai awal keduanya adalah 10. Maka

$I+=3$; artinya : $I=I+3$;

$I=10+3$;

$I=13$;

$I/=J-8$; artinya : $I=I/(J-8)$;

$I=10/(10-8)$;

$I=5$;

3.3.3 Operator Hubungan

Digunakan untuk menunjukkan hubungan antara dua buah operand

Operator	Keterangan
<	Lebih kecil dari
<=	Lebih kecil atau sama dengan
>	Lebih besar dari
>=	Lebih besar atau sama dengan
==	Sama dengan
!=	Tidak sama dengan

Operator hubungan banyak digunakan untuk penyeleksian kondisi dengan statemen if, do-while atau while.

3.3.4 Operator Logika

Digunakan untuk membandingkan logika hasil dari operator-operator hubungan.

Hubungan Logika

X	Y	$X \& \& Y$	$X \parallel Y$	$!X$
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

3.3.5 Operator Unary

Operator	Keterangan
-	Unary minus
++	Peningkatan dengan penambahan nilai 1
--	Penurunan dengan pengurangan nilai 1
(type)	Cast
sizeof	Ukuran dari operand dalam byte
!	Unary NOT
&	Menghasilkan alamat memory operand
*	Menghasilkan nilai pengenalan alamatnya
~	Komplemen satu

Catatan :

ü Operator Unary minus

Digunakan untuk memberi nilai minus suatu nilai numerik (bukan untuk pengurangan).

Misalnya ungkapan $A+-B*C$ akan diartikan sebagai $A+(-B)*C$. Penulisan $A--B*C$ berarti

$A-(-B)*C$

ü Operator ++ dan --

Operator ini banyak dijumpai di statemen perulangan yang berbentuk sebagai berikut :

`for(i=1;i<=10;i=i+1)`

dapat ditulis dengan operator ++ sebagai berikut :

`for(i=1;i<=10;i++)`

Operator ++ dan -- yang digunakan di suatu ungkapan akan berarti lain jika ditulis sebelum atau sesudah operandnya.

Contoh :

`X=5;`

`A=X++;`

Hasil

`X=6 dan A=5`

`X=5;`

`A=++X;`

Hasil

`X=6 dan A=6`

4 Latihan sederhana menggunakan C

a. Contoh 1

```
1  /* Fig. 2.4: fig02_04.c
2     Printing multiple lines with a single printf */
3  #include <stdio.h>
4
5  /* function main begins program execution */
6  int main()
7  {
8     printf( "welcome\nto\nC!\n" );
9
10    return 0; /* indicate that program ended successfully */
11
12 } /* end function main */
```

Output :

```
Welcome
To
C!
```

b. Contoh 2

Tentukan outputnya

```
#include <stdio.h>
int A;
int B;
int C;
void main()
{
    A=25;
    B=35;
    C=A+B;
    printf("Hasil Jumlahnya adalah C=%d",C);
    getch();
};
```

1. Latihan 1

Buatlah program dengan tampilan output seperti berikut :

```
Masukkan nilai A : 15
Masukkan Nilai B : 13
Hasil jumlahnya = 28
Hasil Pengurangannya = 2
```

2. Buatlah program dengan tampilan output sebagai berikut :

Ketikkan nama anda : Budi

Masukkan umur anda : 25

Hai Budi

Anda berumur : 25 tahun

Selamat Belajar Pemrograman dengan bahasa C