ALGORITMA & PEMROGRAMAN #3

TIPE DATA, OPERASI
&
OPERATOR

Sufajar Butsianto, M.Kom

Rev.00

Bahasa C - Variabel

- Tentukan nama variabel yang benar :
  - 9kepala
  - _nilaimax
  - data nilai
  - _4445
  - a_b

Review: Deklarasi Identifier

Variabel

- Bentuk umum: `<tipe_data> <nama_variabel>;`
- Contoh:
  - `int umur;`

Konstanta

Preprocessor Directive

- Bentuk umum: `#define <nama_konstanta> <nilai>`
- Contoh:
  - `#define pi 3.14`
  - `#define nama "antonius"`

Konstanta

Konstanta, bentuk 2:

- Contoh:
  - `const float phi = 3.14;`
  - `const char nama[] = "antonius";`

```
const <tipe_data> <nama_konstanta> = <nilai>;
```

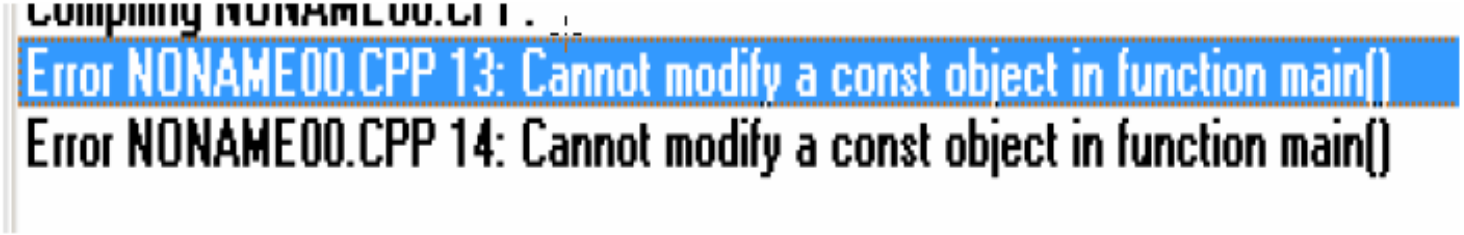
```c
#include <stdio.h>
const float phi = 3.14;
const char nama[] = "antonius";

void main(){
    printf("%f\n",phi);
    float luas = 7 * 7 * phi;
    printf("%f\n",luas);
    printf("%s\n",nama);
}
```

Error

Error karena #define:

Error karena const:



Preprocessor Directive

- Bagian yang berisi pengikutsertaan file atau **berkas-berkas fungsi** maupun pendefinisian **konstanta**

```
#include <stdio.h>
#define pi 3.14
```

Tidak diakhiri titik koma

Bahasa C – File Header

- Formatnya : [`namafile.h`]
- Ada 2 macam penulisan
  - Diapit tanda < dan >
    - Contoh : `#include <stdio.h>`, digunakan bila mengakses file header dari library standar
  - Diapit tanda " "
    - Contoh : `#include "tugas.h"`, digunakan bila mengakses file header tugas.h yang ada di direktori kerja

Karakter *Escape*

- Yaitu karakter yang diawali dengan \ (backslash)
- Masing-masing memiliki makna tertentu

Karakter *Escape* (2)

Karakter	Arti
\a	Bunyi bel (*speaker* komputer)
\b	Mundur satu spasi (*backspace*)
\f	Ganti halaman *(form feed)*
\n	Ganti baris baru *(new line)*
\r	Ke kolom pertama baris yang sama *(carriage return)*
\t	Tabulasi horisontal
\v	Tabulasi vertikal
\0	Nilai kosong *(null)*
\'	Karakter petik tunggal
\""	Karakter petik ganda
\\	Garis miring terbalik *(back slash)*

Sifat Data Numerik *Integer*

Nilai numerik pecahan yang disimpan dalam integer akan dibulatkan ke **bawah**. Jadi, nilai pecahan **dibuang**

- Contoh:
  - 2.38 → 2
  - 4.928 → 4

Casting

- Pemaksaan suatu tipe data ke tipe data lain

```c
#include <stdio.h>

void main(){
   int a = 5;
   int b = 2;

   printf("5/2 dlm integer = %d\n",a/b);
   //printf("%f",a/b);

   float c = 5.0;
   float d = 2.0;

   printf("5.0/2.0 dlm float = %.2f\n",c/d);
   printf("5.0/2.0 dlm int = %d\n",c/d);

   printf("5/2 casting ke float = %f\n",(float)a/b);
   printf("5/2 casting ke float = %f\n",(float)a/(float)b);
   printf("5.0/2.0 casting ke int = %d\n",(int)c/d);
   printf("5.0/2.0 casting ke int = %d\n",(int)c/(int)d);

}
```

Sifat Data Karakter

Karakter disimpan dalam memori berupa kode ASCII.

- ASCII
  - Berdasarkan English Alphabet
  - Dipublikasikan tahun 1967
  - Di-*update* tahun 1986
  - Terdiri dari 95 karakter yang *printable* (33-126) dan 32 (0-31) *non-printable/control character*

Sifat Data Karakter (1)

- Menggunakan tanda petik satu (`)
- Contoh:

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

void main()
{    char hrf;

     hrf = 'A';

     printf("Nilai desimal karakter %c adalah %d",
     hrf, hrf);
}
```

Sifat Data Karakter (2)

Pada tipe data karakter dapat dilakukan operasi matematika

- Contoh:

```
char hrf;
hrf = 'A';
printf("Nilai desimal karakter %c adalah
%d\n",hrf, hrf);
printf("Huruf kecilnya = %c",(hrf+32));
```

Sifat Data String

- Bahasa C tidak memiliki tipe data ***String***
- ***String*** diperlakukan sebagai *array of character* (kumpulan karakter)
- Menggunakan tanda petik dua (")

- Deklarasi:

```
char nama[20]="anton";
printf("%s",nama);
```

Operator

Kategori	Operator
Arithmetic	+ - * / %
Logical (boolean and bitwise)	& | ^ ! ~ && || true false
String concatenation	+
Increment, decrement	++ - -
Shift	<< >>
Relational	== != < > <= >=
Assignment	= += -= *= /= %= &= |= ^= <<= >>=

Operator (2)

Kategori	Operator
Member access	.
Indexing	[ ]
Cast	( )
Conditional	?:
Delegate concatenation and removal	+ -
Type information	As is sizeof typeof
Overflow exception control	Checked unchecked
Indirection and Address	* -> [ ] &

Operator Aritmatika

Oprtr	Contoh	Keterangan
+	op1 + op2	Menjumlahkan dua operand
-	op1 - op2	Mengurangkan dua operand
*	op1 * op2	Mengalikan dua operand
/	op1 / op2	Membagi dua operand
%	op1 % op2	Menghasilkan sisa hasil bagi dari pembagian operand

- Operator Modulus tidak dapat dioperasikan ke tipe data float atau double

Operator Aritmatika (2)

Oprtr	Contoh	Keterangan
++	op++	**Op dinaikkan nilainya 1 setelah dilakukan operasi pada op**
++	++op	**Op dinaikkan nilainya 1 sebelum dilakukan operasi pada op**
--	op--	Op diturunkan nilainya 1 setelah dilakukan operasi pada op
--	--op	Op diturunkan nilainya 1 sebelum dilakukan operasi pada op
-	-op	Menegaskan nilai op menjadi positif jika negatif atau sebaliknya

```c
void main(){
    int a = 5;
    int h = a++;
    printf("h = %d\n",h);
    printf("a = %d",a);
}
```

```
h = 5
a = 6
```

```c
#include <stdio.h>

void main(){
    int a = 5;
    int h = ++a;
    printf("h = %d\n",h);
    printf("a = %d",a);
}
```

```
h = 6
a = 6
```

Operator Relasional

Oprtr	Contoh	Keterangan
>	op1 > op2	Menghasilkan true jika op1 lebih besar dari op2
<	op1 < op2	Menghasilkan true jika op1 lebih kecil dari op2
>=	op1 >= op2	Menghasilkan true jika op1 lebih besar atau sama dengan op2
<=	op1 <= op2	Menghasilkan true jika op1 lebih kecil atau sama dengan op2
!=	op1 != op2	Menghasilkan true jika op1 tidak sama dengan op2

Operator Kondisional

Oprtr	Contoh	Keterangan
&&	op1 && op2	Menghasilkan true jika op1 dan op2 true
||	op1 || op2	Menghasilkan true jika op1 atau op2 true
!	!op1	Menghasilkan true jika op1 bernilai false
&	op1 & op2	Bitwise AND
|	op1 | op2	Bitwise OR
^	op1 != op2	Menghasilkan true jika salah satu true, tetapi tidak keduanya

Contoh

- Misalnya, A bernilai 5, B bernilai 7, dan C bernilai 'a', maka ungkapan di bawah ini mempunyai hasil akhir benar atau salah?

A < B || B == 7 && C > 'z'

Contoh: Hasil

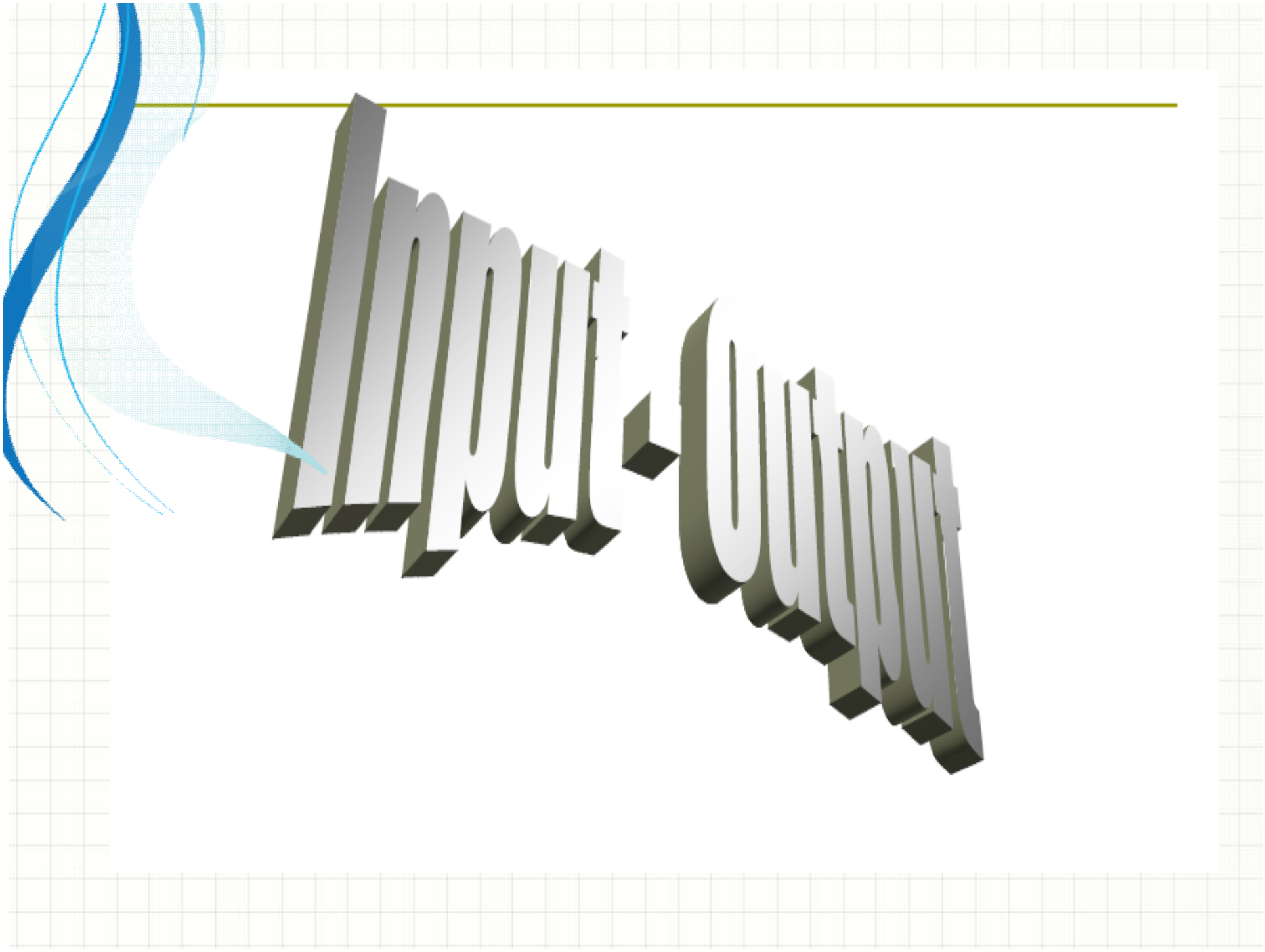
- Hasil akhir: **benar**
- Langkah-langkah:
  1. Jenjang operator relasional lebih tinggi dibandingkan dengan operator logika, jadi operator relasional dikerjakan lebih dahulu
  2. Operator logika '&&' mempunyai jenjang lebih tinggi dari operator '||', sehingga operator '&&' dikerjakan lebih dahulu
  3. Bagian yang paling akhir dikerjakan adalah operator '||'

Beberapa Ungkapan

Ungkapan	Arti
X / Y	X dibagi Y
X = 10	X diisi nilainya dengan 10
Y = Y + 1	Y diisi dengan nilai Y sebelumnya ditambah 1
Y = X	Y diisi dengan nilai X
X += Y	Sama dengan X = X + Y
X /= Y	Sama dengan X = X / Y

Pemberian Komentar

- Pemberian komentar sangat penting dalam menulis program agar program tersebut terdokumentasi dengan baik.
- Program yang terdokumentasi dengan baik berarti alur dan logika program tersebut jelas, dapat dibaca dengan mudah pada lain waktu.
- Semua komenter dalam bahasa C tidak akan dibaca atau akan diabaikan oleh compiler bahasa C.
- Komentar dalam banyak baris diawali dengan tanda /* , kemudian setelah semua komentar ditulis, diakhiri dengan tanda */ sebagai penutupnya.
- Sedangkan untuk komentar dalam satu baris saja, ditulis dengan tanda // di awal kalimat komentar.


Input-Output

Output di Bahasa C

- Header stdio.h
- **printf**(<string>,[<variabel>]),
  puts(<string>) atau **putchar**(<char>)

Contoh di C:

```
#include <stdio.h>

void main(){

        char nama[50] = "anton";

        printf("Hallo saya bernama %s, saya sedang belajar C!",nama);

}
```

Output Tidak Terformat

- **putchar(char)** dan **puts(char[]).**
- puts diakhiri dgn enter

Contohnya:

```
#include <stdio.h>
void main()
{     char N,D[15] = "antonius rc";
      N = 'X';
      putchar(N);
      puts(D);
}
```

Hasilnya adalah : **Xantonius rc**

Output Tidak Terformat

- **(+)** Bentuknya sederhana

- **(-)** Tidak tidak dapat digunakan untuk menampilkan bentuk yang rumit
- **(-)** Hanya dapat menggunakan sebuah argumen saja.

Output Terformat

Perintah untuk menampilkan hasil terformat adalah **printf()**

Kode Format	Kegunaan
%c	Menampilkan sebuah karakter
%s	Menampilkan nilai string
%d	Menampilkan nilai desimal integer
%i	Menampilkan nilai desimal integer
%u	Menampilkan nilai desimal integer tak bertanda
%x	Menampilkan nilai heksa desimal integer
%o	Menampilkan nilai oktal integer
%f	Menampilkan nilai pecahan
%e	Menampilkan nilai dalam notasi saintifik
%g	Sebagai pengganti %f atau %e tergantung yg terpendek
%p	Menampilkan suatu alamat memori untuk pointer

Menampilkan Karakter

- Menampilkan karakter di C secara terformat, kita dapat menggunakan **"%c".**
- Untuk menampilkan sebuah karakter dengan lebar 3 posisi (tiga karakter di depan, karakternya blank), maka gunakan **"%3c"**
- Untuk membuat rata kiri (blank ada di sebelah kanan karakternya) dapat digunakan simbol (*flag*) minus, misalnya "**%-3c**".

```
#include <stdio.h>

void main(){
      char c = 'a';
      printf("%3c\n",c);
      printf("%-3c\n",c);
}
```

Hasilnya:

```
  a
a
```

Menampilkan String Terformat

FORMAT	KETERANGAN
"%s"	Menampilkan semua nilai karakter pada nilai string
"%Ns"	Menampilkan semua karakter rata kanan dengan lebar N posisi; N adalah konstanta numerik bulat.
"%-Ns"	Menampilkan semua karakter rata kiri dengan lebar N posisi; N adalah konstanta numerik bulat.
"%N.Ms"	Menampilkan rata kanan hanya M buah karakter pertama saja dengan lebar N posisi; M dan N adalah konstanta numerik bulat.
"%-N.Ms"	Menampilkan rata kiri hanya M buah karakter pertama saja dengan lebar N posisi; M dan N adalah konstanta numerik bulat.

Contoh Output Terformat

```
#include <stdio.h>

main()
{
      char D[15] = "Antonius Rachmat C";
      printf("123456789012345678901\n");
      printf("%s\n",D); /* semua karakter, rata kiri */
      printf("%20s\n",D); /* lebar 20, rata kanan */
      printf("%-20s\n",D); /* lebar 20, rata kiri */
      printf("%20.5s\n",D); /* 5 karakter lbr 20, rata kanan */
      printf("%-20.5s\n",D); /* 5 karakter lbr 20, rata kiri */
}
```

Hasil

```
12345678901234567890
Antonius Rachmat C
  Antonius Rachmat C
Antonius Rachmat C
               Anton
Anton
```

Integer Terformat

FORMAT	ARTI
"%d", "%i"	signed int
"%u"	unsigned int
"%ld","%li"	long int
"%hi"	short int
"%hu"	unsgined short int
"%lu"	unsigned long int

Contoh Integer Terformat

```
#include <stdio.h>
main()
{
            int i=1234;
            printf("%i\n",i);
            printf("%5i\n",i);
            printf("%7d\n",i);
            printf("%07d\n",i);
            printf("%-7d\n",i);
}
```

Hasilnya:

```
1234
 1234
   1234
0001234
1234
```

Menampilkan Bilangan Pecahan

FORMAT	ARTI
"%f"	float dgn nilai pecahan
"%e"	float dgn notasi saintifik
"%g"	terpendek dari "%f" atau "%e"
"%lf", "%le" atau "lg"	double
"%Lf", "%Le" atau "Lg"	long double

Contoh Pecahan

```
#include <stdio.h>
void main()
{
   float x=123.4567;
   printf("%3f %15f %020f\n",x,x,x);
   printf("%3e %15e %020e\n",x,x,x);
}
```

Hasilnya:

```
123.456703      123.456703 0000000000123.456703
1.23457e+02     1.23457e+02 0000000001.23457e+02
```

Contoh2:

```
#include <stdio.h>
void main()
{
    float F=12345.6789;
    printf("%15f\n",F);
    printf("%15.2f\n",F);
    printf("%015.2f\n",F);
    printf("%-15.2f\n",F);
    printf("%15.0f\n",F);
}
```

Hasilnya:

```
    12345.678711
        12345.68
000000012345.68
12345.68
             123456
```

Contoh Pecahan

Contoh3:

```
#include <stdio.h>
void main(){
                printf("123456789\n\n");
                printf("%4c\n",'a');
                printf("%-4c\n",'a');
                printf("%4d\n",50);
                printf("%-4d\n",50);
                printf("%6f\n",7.56);
                printf("%6.2f\n",7.56);
                printf("%6.3f\n",7.56);
                printf("%-6.3f\n",7.56);
}
```

Hasil:

Membersihkan Layar & Meletakkan Kursor

Menggunakan header **conio.h** dengan fungsi yang bernama **clrscr()**

Contoh :

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

void main()
{     clrscr();
      printf("Layar sudah bersih...");
}
```

Mengggunakan header **conio.h** dengan fungsi yang bernama **gotoxy()**

Contoh :

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

void main()
{    clrscr();
     gotoxy(20,1);printf("Layar sudah bersih...");
     gotoxy(20,3);printf("Ini baris ke 3, kolom 20");
     gotoxy(20,5);printf("Ini baris ke 5, kolom 20");
     gotoxy(25,6);printf("Ini kolom 25, baris ke 6");
     gotoxy(20,7);printf("Ini kolom 20, Baris ke 7");
}
```

Input Data

- Header **stdio.h:**
  - **gets()**
  - **scanf()**
- Header **conio.h:**
  - **getche()**
  - **getchar()**
  - **getch()**

Input Data Karakter Tidak Terformat

- getche(): Tanpa Enter, karakter terlihat
- getchar(): Dengan Enter, Karakter terlihat
- getch(): Tanpa Enter, karakter tdk terlihat

Contoh 1:

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

void main()
{ char hrf;
  printf("Masukkan sebuah karakter : "); hrf = getche();
  printf("\nNilai yang dimasukkan : %c\n",hrf);
}
```

Hasilnya :

```
Masukkan sebuah karakter : N
Nilai yang dimasukkan : N
```

Contoh Input

Contoh 2:

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

void main()
{       char hrf;
        printf("Masukkan sebuah karakter : ");
        hrf = getchar();
        printf("\nNilai yang dimasukkan : %c\n",hrf);
}
```

Hasilnya :

```
Masukkan sebuah karakter : N↵
Nilai yang dimasukkan : N
```

Contoh 3:

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

void main()
{       char hrf;
        printf("Masukkan sebuah karakter : ");
        hrf = getch();
        printf("\nNilai yang dimasukkan : %c\n",hrf);
}
```

Hasilnya :

```
Masukkan sebuah karakter :
Nilai yang dimasukkan : N
```

Input Data String tidak terformat

- Untuk memasukkan nilai string dapat dipakai fungsi **gets()**

Contoh:

```
#include <stdio.h>

void main()
{     char kata[10];
      printf("Masukkan suatu nilai string : ");
      gets(kata);
      printf("Nilai string yang dimasukkan : %s\n",kata);
}
```

Hasilnya :

```
Masukkan suatu nilai string : anton
Nilai string yang dimasukkan : anton
```

Input Data Terformat

- **Menggunakan scanf(kodeformat,variabel)**

Kode Format	Kegunaan
%c	Membaca sebuah karakter
%s	Membaca sebuah data string
%d	Membaca sebuah nilai desimal integer
%i	Membaca sebuah nilai desimal integer
%x	Membaca sebuah nilai heksa desimal
%o	integer
%f	Membaca sebuah nilai oktal integer
%e	Membaca sebuah data pecahan
%g	Membaca sebuah data pecahan
	Membaca sebuah data pecahan

Input Data Karakter Terformat

Contoh:

```
#include <stdio.h>

void main()
{    char c1,c2,c3;
     printf("Masukkan 3 nilai karakter : ");
     scanf("%c%c%c",&c1,&c2,&c3);
     printf("\nAnda memasukkan : %c %c %c\n",c1,c2,c3);
}
```

Hasilnya :

```
Masukkan 3 nilai karakter : abcde
Anda memasukkan : a b c
```

Input Data String Terformat

Contoh:

```
#include <stdio.h>

void main()
{    char kata[10];
     printf("Masukkan suatu nilai string : ");
     scanf("%s",kata);
     printf("Nilai string yang dimasukkan : %s\n",kata);}
```

Hasilnya :

```
Masukkan suatu nilai string : Anton
Nilai string yang dimasukkan : Anton
```

Perhatian

Scanf(<format>,<variabel>):

- Jika string yang dimasukkan memiliki whitespace karakter, maka input string hanya akan dibaca sampai dengan karakter sebelum whitespace saja!

Solusi:

- kode format **"%s"** dapat diganti dengan "**%[^\n]**"
  - Berarti bahwa karakter nilai string akan dibaca terus sampai ditemui penekanan tombol Enter (bentuk '^' menunjukkan maksud 'tidak' dan karakter '\n' artinye Enter). Sehingga dengan demikian semua karakter temasuk spasi dan tabulasi akan dibaca sampai ditemui penekanan tombol Enter.
- Atau dengan **gets(<string>)**

Contoh

```
#include <stdio.h>

void main()
{    char kata[20];
     printf("Masukkan suatu nilai string : ");
     scanf("%[^\n]",kata);
     printf("Nilai string yang dimasukkan : %s\n",kata);
}
```

Hasilnya :

```
Masukkan suatu nilai string : Antonius RC
Nilai string yang dimasukkan : Antonius RC
```

Memasukkan Nilai Numerik

- Menggunakan %d untuk integer
- Menggunakan %i untuk integer
- Menggunakan %ld atau %li untuk long integer
- Menggunakan %f untuk double dan float
- Menggunakan %le atau %lf dan %lg untuk long double

Soal-soal

- Buatlah program menghitung luas persegi panjang!
- Buatlah program menghitung luas lingkaran!
- Buatlah program penghitung rumus sebagai berikut:
  - $E = mc^2$
- Buatlah program konversi suhu, dari Celcius, Reamur, dan Farenheit.
  - F = 9/5 * C + 32
  - R = 4/5 * C
- Buatlah program konversi **detik** ke hari, jam, menit, detik!
  - Rumus : 1 hari = 86400 detik; 1 jam = 3600 detik dan 1 menit = 60 detik.

Soal - soal

- Hitung jarak tempuh, dengan kec v, dan waktu t (detik)!
  - S = v * t
- Perkalian 2 pecahan:
  - P1 = ¾
  - P2 = 2/3
  - Hasil = (3 x 2) / (3 x 4)
- Program konversi dolar ke rupiah
  - Gunakan konstanta!
- Menghitung upah gaji per jam seorang pegawai, jika per jam @ 5000!

TERIMA KASIH