

Algoritma & Pemrograman #3

by antonius rachmat c, s.kom, m.cs

Bahasa C - Variabel

- Tentukan nama variabel yang benar :
 1. 9kepala
 2. _nilaimax
 3. data nilai
 4. _4445
 5. a_b

Review: Deklarasi Identifier

□ Variabel

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

□ Konstanta

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

Preprocessor Directive



Konstanta

□ Konstanta, bentuk 2:

➤ Contoh:

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

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

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

const float phi = 3.14;
const char nama[] = "antonius";

int main() {
    printf("%f\n", phi);
    printf("%s\n", nama);

    //phi = 3.16;
    //scanf("%s", nama);
    getch();
}
```

Error jika konstanta diubah “paksa”

Error pada #define:

C:\Users\Antonius Rachmat C\Docu...	In function 'int main()':
C:\Users\Antonius Rachmat C\Docu...	non-lvalue in assignment

Error pada const:

C:\Users\Antonius Rachmat C\Docu...	In function 'int main()':
C:\Users\Antonius Rachmat C\Docu...	assignment of read-only variable 'pi'

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 yg sudah didefinisikan

Karakter *Escape*

- ❑ Yaitu karakter yang diawali dengan \ (backslash)
- ❑ Walaupun berjumlah 2 karakter namun tetap dianggap 1 karakter
- ❑ Masing-masing memiliki makna tertentu

Karakter *Escape* (2)

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

Sifat Data Numerik *Integer*

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

□ Contoh:

➤ 2.38 \longrightarrow 2

➤ 4.928 \longrightarrow 4

Sifat Data Numerik *Integer* (2)

Jika nilai variabel yang **melebihi** jangkauannya akan dipotong sepanjang jumlah bit yang tersedia

□ Contoh:

Jika misal dideklarasikan variabel integer (16 bit = 2 byte) berarti hanya menyimpan sampai dengan 32,767.

Jika variabel diisi nilai 70,000 (1 0001 0001 0111 0000), padahal 70,000 menempati 17 bit maka bit paling kiri akan dipotong menjadi (0001 0001 0111 0000),

Sifat Data Numerik *Integer* (3)

□ Contoh pada **Turbo C**:

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

int main()
{   int x;



    x = 70000;

    printf("x = %d\n",x); //hasil 4464
}
```

Pada DevC++

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

int main() {
    int x=700000000000;
    printf("%d", x);
    getch();
}
```

 Compiler |  Resources |  Compile Log |  Debug |  Find Results |  Close

Line	File	Message
5	C:\Documents and Settings\@nton\...	In function `int main()': integer constant is too large for "long" type

Casting

- “Pemaksaan” suatu tipe data ke tipe data lain

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

int main(){
    int a=5;
    int b=2;
    float c = 5.0;
    float d = 2.0;

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

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

    printf("5/2 cast float 1 = %.2f\n", ((float) a/b));
    printf("5/2 cast float 2 = %.2f\n", ((float) a/ (float) b));

    printf("5.0/2.0 cast int 1 = %d\n", ((int) c/d));
    printf("5.0/2.0 cast int 2 = %d\n", ((int) c/ (int) d));
    getch();
}
```

```
5/2 dalam int = 2
5.0/2.0 dalam float = 2.50
5/2 dalam float = 2.50
5.0/2.0 dalam int = 0
5/2 cast float 1 = 2.50
5/2 cast float 2 = 2.50
5.0/2.0 cast int 1 = 0
5.0/2.0 cast int 2 = 2
```

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>

int main()
{
    char hrf;

    hrf = 'A';

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

    getch();
}
```


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));
```

A = 65

A = a

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 Aritmatika (binary operator)

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 (unary operator)

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

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

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

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

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

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

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

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

int main() {
    int x=2;
    int y=1;
    int h=x++ + ++y - ++x;
    printf("%d",h);
    getch();
}
```

Hasil : 1

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?

!0 || 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 '||', terserah pilih '||' yang mana duluan

Pemberian Komentar

- ❑ Program yang terdokumentasi dengan baik berarti alur dan logika program tersebut jelas, dapat dibaca dengan mudah pada lain waktu.
- ❑ Semua komentar 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.

```
//ini komentar satu baris  
/* ini komentar banyak baris  
ini baris kerikutnya  
ini juga */
```

Contoh komentar

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

//fungsi bernama xxx bersifat rekursif

int xxx(int A[],int n){
    //jika n adalah 0
    if(n==0) return A[0];
    else return xxx(A, (n-1) %4) +A[n];
}

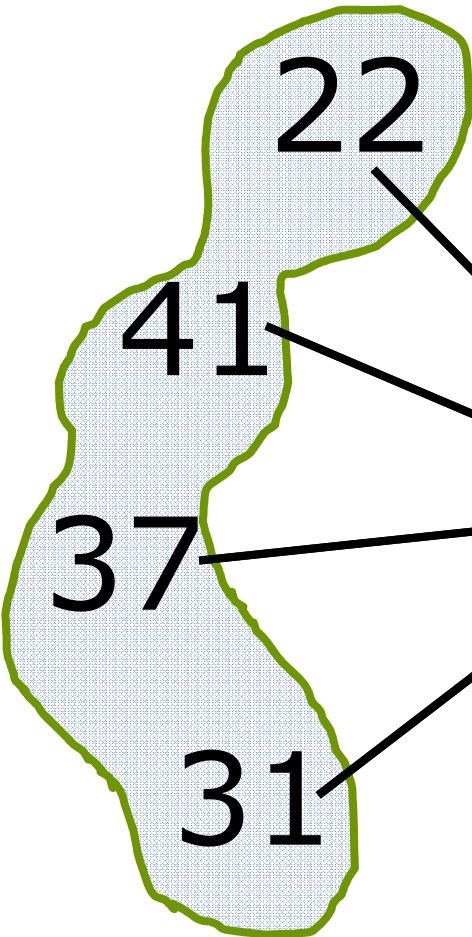
/*
di dalam fungsi main dibuat array 1 dimensi
bernama A berjumlah 5 elemen yg bertipe integer
tujuannya untuk mencoba fungsi xxx diatas
*/

int main() {
    int A[5] = {1,2,3,4,5};
    printf("%d ",xxx(A,1));
    getch();
}
```

Multi-Channel Marketing



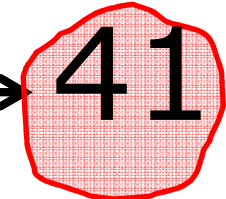
Input



Proses



Output



Mencari bilangan terbesar



1. Tentukan bilangan pertama adalah bilangan terbesar
2. Bandingkan bilangan terbesar tersebut dengan bilangan-bilangan berikutnya
3. Jika ada bilangan yang lebih besar, maka bilangan tersebut ditentukan sebagai yang terbesar
4. Jika seluruh bilangan sudah dibandingkan, bilangan terbesar yang terakhir ditemukan adalah bilangan yang terbesar dari keseluruhan bilangan yang ada

Output di Bahasa C

- Header "stdio.h"
 - **printf**(<string>,[<variabel>])
 - **puts**(<string>)
 - **putchar**(<char>)
- Output dapat diberikan kepada:
 - Layar, device, error, printer
- Output ada 2 jenis:
 - Terformat
 - Tidak terformat

Output Tidak Terformat

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

```
int main()
{
    char N,D[15] = "antonius rc";
    N = 'X';
    putchar(N);
    puts(D);
    getch();
}
```

Hasilnya adalah : Xantonius rc

Output Tidak Terformat

- (+) Bentuknya sederhana
- (-) 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
<code>%c</code>	Menampilkan sebuah karakter
<code>%s</code>	Menampilkan nilai string
<code>%d</code>	Menampilkan nilai desimal integer
<code>%i</code>	Menampilkan nilai desimal integer
<code>%u</code>	Menampilkan nilai desimal integer tak bertanda
<code>%x</code>	Menampilkan nilai heksa desimal integer
<code>%o</code>	Menampilkan nilai oktal integer
<code>%f</code>	Menampilkan nilai pecahan
<code>%e</code>	Menampilkan nilai dalam notasi saintifik
<code>%g</code>	Sebagai pengganti <code>%f</code> atau <code>%e</code> tergantung yg terpendek
<code>%p</code>	Menampilkan suatu alamat memori untuk pointer

```
printf("Selamat datang!");
```

```
int a = 10;
```

```
printf("Nilai a : %d", a);
```



"Format specifier"
untuk bilangan bulat

```
int a = 10; float b = 0.75 * a;
```

```
printf("Nilai b : %f", b);
```

"Format specifier"
untuk bilangan
pecahan

Output Terformat

- ❑ 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>
#include <conio.h>

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

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>
#include <conio.h>

int main()
{
    char D[15] = "Antonius Rachmat C";
    printf("12345678901234567890\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 */
    getch();
}
```

Hasil

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

Integer Terformat

FORMAT	ARTI
<code>%%d</code> , <code>%%i</code>	signed int
<code>%%u</code>	unsigned int
<code>%%ld</code> , <code>%%li</code>	long int
<code>%%hi</code>	short int
<code>%%hu</code>	unsigned short int
<code>%%lu</code>	unsigned long int

Contoh Integer Terformat

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

int main()
{
    int i=1234;
    printf("%i\n",i);
    printf("%5i\n",i);
    printf("%7d\n",i);
    printf("%07d\n",i);
    printf("%-7d\n",i);
    getch();
}
```

Hasilnya:

```
1234
 1234
   1234
0001234
1234
```

Menampilkan Bilangan Pecahan

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

Contoh Pecahan

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

int main()
{
    float x=123.4567;
    printf("%3f %15f %020f\n",x,x,x);
    printf("%3e %15e %020e\n",x,x,x);
    getch();
}
```

```
123.456703      123.456703 00000000000123.456703
1.23457e+02     1.23457e+02 00000000001.23457e+02
```

Contoh Pecahan

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

int 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);
    getch();
}
```

```
      12345.678711
          12345.68
000000012345.68
12345.68
                123456
```

Bilangan Hexadesimal & Octal

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

int main()
{
    int x = 1234;
    printf("%x\n",x); //hasil = 4d2
    getch();
}
```

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

int main()
{
    int o=1234;
    printf("%o\n"); //Hasil = 2322
    getch();
}
```

Membersihkan Layar & Meletakkan Kursor

□ Menggunakan **system("cls")**

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

int main()
{
    system("cls");
    printf("Layar sudah bersih...");
    getch();
}
```

Rangkuman Format

Table 4.1 Basic Data Types

Type	Constant Examples	printf chars
char	'a', '\n'	%c
_Bool	0, 1	%i, %u
short int	—	%hi, %hx, %ho
unsigned short int	—	%hu, %hx, %ho
int	12, -97, 0xFFE0, 0177	%i, %x, %o
unsigned int	12u, 100U, 0xFFu	%u, %x, %o
long int	12L, -2001, 0xffffL	%li, %lx, %lo
unsigned long int	12UL, 100ul, 0xffeeUL	%lu, %lx, %lo
long long int	0xe5e5e5e5LL, 50011	%lli, %llx, %llo
unsigned long long int	12ull, 0xffeeULL	%llu, %llx, %llo
float	12.34f, 3.1e-5f, 0x1.5p10, 0x1P-1	%f, %e, %g, %a
double	12.34, 3.1e-5, 0x.1p3	%f, %e, %g, %a
long double	12.341, 3.1e-51	%Lf, %Le, %Lg

Input Data

- Dapat menerima masukan dari pengguna
- Header **stdio.h**:
 - **gets()**
 - **scanf()**
- Header **conio.h**:
 - **getche()**
 - **getchar()**
 - **getch()**
- Input:
 - Terformat dan Tidak Terformat

Input Data Karakter Tidak Terformat

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

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

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

Hasilnya :

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

Input Data String tidak terformat

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

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

int 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

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

int 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 Karakter Terformat

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

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

Hasilnya :

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

Input Data String Terformat

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

int 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`` artinya Enter). Sehingga dengan demikian semua karakter termasuk spasi dan tabulasi akan dibaca sampai ditemui penekanan tombol Enter.
- Atau dengan **gets(<string>)**

Contoh

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

int 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:**
 - **$P1 = \frac{3}{4}$**
 - **$P2 = \frac{2}{3}$**
 - **Hasil = $(3 \times 2) / (4 \times 3) \Rightarrow \frac{6}{12}$**
- Program konversi dolar ke rupiah
 - Gunakan konstanta! 1 dolar = Rp. 8900
- Menghitung upah gaji per jam seorang pegawai, jika per jam @ 5000!

Challenging

- **Buatlah program untuk menghitung jumlah 2 buah pecahan dan sederhanakan**
 - **Input: $3/2 + 2/3$**
 - **Proses: $9/6 + 4/6$**
 - **Output: $13/6 \Rightarrow 2 \ 1/6$**

 - **Input: $2/6 + 1/8$**
 - **Proses: $6/48 + 16/48 = 22/48 \Rightarrow 11/24$**

NEXT

- Percabangan IF Sederhana