



# **BAB 7**

# **STRING**

**Dosen : Sulistyowati,**

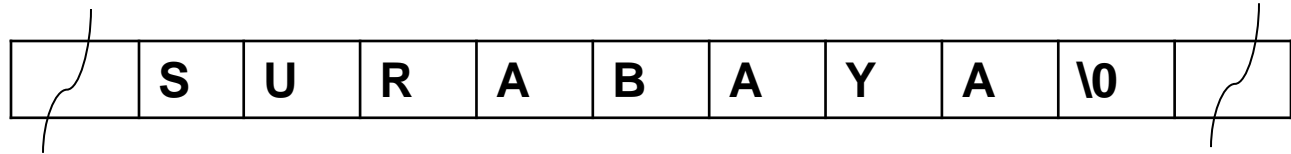


# Konstanta String

- String merupakan bentuk data yang biasa dipakai dalam bahasa pemrograman guna keperluan menampung dan memanipulasi data teks. Misalnya untuk menampung/menyimpan suatu kalimat.
- Dalam C++, string bukanlah sebuah tipe data tersendiri, melainkan hanya jenis khusus dari array.
- Suatu konstanta string ditulis dengan awalan dan akhiran tanda petik ganda(""). Misal : "SURABAYA".

# Konstanta String (*con't*)

- Konstanta string “SURABAYA” disimpan dalam memori secara berurutan, dengan susunan sbb :



- Setiap karakter akan menempati memori sebesar 1 byte dan byte terakhir secara otomatis akan berisi karakter NULL ( \0 ).
- Bila suatu string hanya berisi karakter NULL (tidak berisi sesuatu karakter-pun), string disebut sebagai string kosong dan ditulis sbb : ""

# Variabel String – *Deklarasi String*

- Variabel string adalah variabel yang digunakan untuk menyimpan string.
- Variabel string tak lain adalah array bertipe char yang mengandung elemen berupa karakter NULL, yang menyatakan akhir string.
- Bentuk deklarasi string :

**char N\_variabel [J\_karakter];**

Dimana :

- ~ Char : tipe data karakter
- ~ N\_variabel : nama variabel string
- ~ J\_karakter : jumlah/panjang maksimal karakter (sudah termasuk karakter NULL).

Misal :

```
char Nama[20];    char teks[35];
```

# Variabel String – *Inisialisasi String*

- Cara untuk menginisialisasi/memberi nilai awal string, dapat dilakukan dengan salah satu cara di bawah ini :
  - ✓ `char Nama[ ] = "SURABAYA";`
  - ✓ `char Nama[ ] = {'S','U','R','A','B','A','Y','A','\0'};`
  - ✓ `char Nama[9] = "SURABAYA";`
  - ✓ `char Nama[9] = {'S','U','R','A','B','A','Y','A'};`

# Input Data String – Dengan *cin*

- Untuk menginputkan data string dari piranti keyboard, dapat dilakukan dengan cara :

**cin >> N\_variabel;**

Misal :

```
...  
cout<<"Inputkan kata : ";  
cin>>teks;  
cout<<"Kata yang diinputkan = "<<teks;  
...
```



**PROGRAM 0**

Output :

Inputkan kata : **Teknik Informatika** ↩

Kata yang diinputkan = Teknik



# Input Data String – Dengan *cin.get( )*

- Karena operator >> pada *cin* hanya bisa membaca masukan hingga terdapat Spasi, Tab atau Enter, maka solusinya adalah menggunakan fungsi *cin.get()*

Bentuk :

*cin.get (N\_varibel, J\_karakter);*

Misal :

```
...  
cout<<"Inputkan kalimat : ";  
cin.get(teks,9);  
cout<<"Kalimat yang diinputkan = "<<teks;  
...
```



**PROGRAM 1**

Pada contoh diatas, argumen kedua yaitu **9**, menyatakan jumlah maksimal karakter yang dapat diterima oleh variabel **teks (termasuk karakter Null)**. 7

# Input Data String – Dengan *cin.get( )*

Pada contoh diatas, outputnya adalah :

Misal :

Inputkan kalimat : **Teknik Informatika**   
Kalimat yang diinputkan = Teknik I 

?

Tampak bahwa hanya 8 karakter saja yang disimpan pada variabel teks (karena argumen kedua dari fungsi **cin.get()** diisi dengan 9 dimana satu karakter lagi berisi karakter NULL).

- Parameter kedua dari fungsi `cin.get()` harus diisi dengan nilai yang tidak melebihi ukuran array.  
Jika string yang dimasukkan melebihi ukuran array, maka sisa string berikutnya akan ditempatkan sesudah bagian terakhir dari array.



# Input Data String – Dengan *cin.get( )*

- Solusinya ????

Gunakan bentuk berikut :

**cin.get (N\_varibel, sizeof (N\_variabel));**

Misal :

```
...  
cout<<"Inputkan kalimat : ";  
cin.get(teks,sizeof(teks));  
cout<<"Kalimat yang diinputkan = "<<teks;  
...
```



**PROGRAM 2**

Output :

Inputkan kalimat : **Teknik Informatika**   
Kalimat yang diinputkan = Teknik Informatika

# Input Data String – Dengan *cin.getline()*

- Perhatikan contoh ini :

```
...
cout<<"Input Nama :
cin.get(nama,sizeof(nama));
cout<<"Inputkan Pesan : ";
cin.get(teks,sizeof(teks));
cout<<"NAMA      = "<<nama<<endl;
cout<<"PESAN     = "<<teks<<endl;
...
```



**PROGRAM 3**

- Output :  
Input Nama : **Aulia** ←  
Input Pesan : NAMA = Aulia  
PESAN =

# Input Data String – Dengan *cin.getline( )*

- Pada contoh diatas, `cin.get()` pertama digunakan untuk membaca nama dan yang kedua digunakan untuk membaca pesan.

Tapi program tidak memberikan kesempatan user untuk mengisi pesan.

Hal ini terjadi karena `cin.get()` yang pertama tidak mengabaikan Spasi, Tab atau newline, sehingga fungsi `cin.get()` yang kedua tidak berfungsi sebagaimana mestinya.

- Solusinya ..... dengan menggunakan fungsi :  
**`cin.getline (N_varibel, sizeof (N_variabel));`**

# Input Data String – Dengan *cin.getline()*

```
...  
cout<<"Input Nama :  
cin.getline(nama,sizeof(nama));  
cout<<"Inputkan Pesan : ";  
cin.getline(teks,sizeof(teks));  
cout<<"NAMA      = "<<nama<<endl;  
cout<<"PESAN     = "<<teks<<endl;  
...
```



**PROGRAM 4**

Output :

```
Input Nama : Aulia ↵  
Input Pesan : Anak yang cantik dan lucu ↵  
  
NAMA      = Aulia  
PESAN     = Anak yang cantik dan lucu
```

# Fungsi-fungsi String

Prototype fungsi-fungsi pada pengolahan data string ada pada file : **string.h**

## ■ Fungsi *cin.gcount()*

Untuk menghitung jumlah karakter

Contoh :

```
...  
cout<<"Inputkan kalimat : ";  
cin.getline(teks,sizeof(teks));  
cout<<"JUMLAH KARAKTER ="<<cin.gcount()-1;  
...
```



**PROGRAM 5**

Output :

Inputkan Kalimat : Surabaya Kota Pahlawan ↵  
JUMLAH KARAKTER = 22

# Fungsi-fungsi String (con't)

## ■ Fungsi *cin.getline()*

Untuk membaca sejumlah baris hingga suatu karakter yang telah ditentukan ditemukan.

Contoh :

```
...  
cout<<"Inputkan kalimat, akhiri dengan .  
      dan Enter : ";  
cin.getline(teks,sizeof(teks),' .');  
cout<<"KALIMAT YANG DIINPUTKAN = "<<teks;  
...
```



**PROGRAM 6**

Output :

Inputkan kalimat, akhiri dengan . dan Enter : **ali. baba** ↵  
KALIMAT YANG DIINPUTKAN = ali

# Fungsi-fungsi String (con't)

## ■ Fungsi *strcpy()*

Untuk menyalin isi suatu string ke string lain.

- Menyalin string berbeda dengan menyalin bilangan.

Misalnya **teks1** dan **teks2** adalah variabel string, maka untuk menyalin isi data string dari variabel **teks1** ke variabel **teks2** tidak bisa dengan cara :

**teks2 = teks1;**

- Menyalin string tanpa fungsi :

```
char teks1 []="PEMROGRAMAN";  
char teks2 [25];  
  
for (int i=0; teks1[i]; i++)  
    teks2[i]=teks1[i];  
teks2[i]='\0';
```

# Fungsi-fungsi String (con't)

Bentuk fungsi untuk menyalin string :

**strcpy (string\_target, string\_asal);**

Contoh :

```
char teks1 []="PEMROGRAMAN";  
char teks2 [25] ;  
  
strcpy (teks2 , teks1) ;
```



**PROGRAM 7**

Hasil :

```
teks1[ ] = "PEMROGRAMAN"  
teks2[ ] = "PEMROGRAMAN"
```



# Fungsi-fungsi String (con't)

## ■ Fungsi *strlen()*

Untuk menghitung panjang string.

Bentuk :

**strlen (N\_variabel)**

Contoh :

```
char teks1 []="PEMROGRAMAN" ;  
char teks2 [25]="";
```

```
Panjang1=strlen (teks1) ;  
Panjang2=strlen (teks2) ;
```



**PROGRAM 8**

Hasil :

```
Panjang1 =11  
Panjang2= 0
```

# Fungsi-fungsi String (con't)

## ■ Fungsi *strcat()*

Untuk menggabungkan 2 buah string (yaitu menggabungkan string 2 ke dalam string 1).

Bentuk :

**strcat (string\_1,string\_2);**

Contoh :

```
char teks1 []="PEMROGRAMAN";  
char teks2 []="TERSTRUKTUR";  
char teks3 []="satu";
```

```
strcat (teks1,teks2) ;  
strcat (teks3," dua tiga") ;
```



**PROGRAM 9**

Hasil : teks1="PEMROGRAMAN TERSTRUKTUR"  
teks3="satu dua tiga"

# Fungsi-fungsi String (con't)

## ■ Fungsi *strncat()*

Untuk menggabungkan 2 buah string sebanyak n karakter.

Bentuk :

**strncat (string\_1,string\_2,J\_karakter);**

J\_karakter = jumlah karakter maksimal dari string\_2 yang akan ditambahkan ke string\_1.

Contoh :

```
char teks1 []="PEMROGRAMAN";  
char teks2 []="TERSTRUKTUR";  
  
strncat (teks1 , teks2 , 3) ;
```



**PROGRAM 10**

Hasil :

teks1="PEMROGRAMAN TER"

# Fungsi-fungsi String (con't)

## ■ Fungsi *strcmp()*

Untuk membandingkan 2 buah string.

Bentuk :

**strcmp (string\_1,string\_2)**

Fungsi pembandingan ini akan memberikan nilai bertipe **int**, berupa :

- ❖ Nilai negatif, apabila  $\text{string\_1} < \text{string\_2}$
- ❖ Nilai nol, apabila  $\text{string\_1} = \text{string\_2}$
- ❖ Nilai positif, apabila  $\text{string\_1} > \text{string\_2}$

Pembandingan string pada **strcmp()** berdasarkan urutan karakter dalam ASCII. Misal 'A' lebih kecil daripada 'a'.

# Fungsi-fungsi String (con't)

## ■ Contoh :

```
...  
cout<<"Inputkan kalimat 1 : ";  
cin.getline(teks1,sizeof(teks1));  
cout<<"Inputkan kalimat 2 : ";  
cin.getline(teks2,sizeof(teks2));  
Hasil=strcmp(teks1,teks2);  
cout<<"HASIL PEMBANDINGAN = "<<Hasil;  
...
```



**PROGRAM 11**

Output :

```
Inputkan kalimat 1 = SAYA  ↵  
Inputkan kalimat 2 = aku   ↵  
HASIL PEMBANDINGAN = -14 → Karena 'S' < 'a'
```

# Fungsi-fungsi String (con't)

## ■ Fungsi *stricmp()* - lanjutan *strcmp()*

Untuk membandingkan 2 buah string tanpa membedakan huruf kecil ataupun huruf kapital.

Bentuk : **stricmp (string\_1,string\_2)**

Contoh :

```
char teks1 []="SAYA";  
char teks2 []="saya";  
char teks3 []="SAYAP";  
char teks4 []="sama";
```

```
A=stricmp (teks1 , teks2) ;  
B=stricmp (teks2 , teks3) ;  
C=stricmp (teks1 , teks4) ;
```



**PROGRAM 12**

Output :

A = 0

B = -80 → Karena '/' < 'P'

C = 12 → Karena 'Y' > 'm'

# Fungsi-fungsi String (con't)

## ■ Fungsi *strncmp()* - lanjutan *strcmp()*

Untuk membandingkan 2 buah string dan jumlah karakter yang dibandingkan maksimal sebanyak n.

Bentuk :

**strncmp (string\_1,string\_2,J\_karakter)**

Contoh :

```
char teks1 []="SAYA";  
char teks2 []="SAYa";  
  
A=strncmp (teks1 , teks2 , 2) ;  
B=strncmp (teks1 , teks3 , 4) ;
```



**PROGRAM 13**

Output :

A = 0

B = -32

→ Karena 'A' < 'a'

# Fungsi-fungsi String (con't)

## ■ Fungsi *strlwr()*

Untuk merubah string menjadi huruf kecil.

Bentuk :

**strlwr (string);**

Contoh :

```
char teks []="PEMROGRAMAN";  
strlwr (teks);
```



**PROGRAM 14**

Output :

teks = "pemrograman"



# Fungsi-fungsi String (con't)

## ■ Fungsi *strupr()*

Untuk merubah string menjadi huruf kapital.

Bentuk :

**strupr (string);**

Contoh :

```
char teks []="pemrograman";  
strupr (teks) ;
```



**PROGRAM 15**

Output :

```
teks ="PEMROGRAMAN"
```

# Fungsi-fungsi String (con't)

## ■ Fungsi *strstr()*

Untuk mencari suatu substring dalam sebuah string.

Bentuk :

**strstr (string\_asal,substring);**

Contoh :

```
...  
cout<<"Input kalimat : ";  
cin.getline(teks,sizeof(teks));  
if (strstr(teks,"aman"))  
    cout<<"Kata ditemukan dalam kalimat";  
...
```



**PROGRAM 16**

Output :

Input kalimat : **pemrograman** ↵  
Kata ditemukan dalam kalimat

# Fungsi-fungsi String (con't)

## ■ Fungsi *strrev()*

Untuk mencari suatu substring dalam sebuah string.

Bentuk :

**strrev (N\_variabel);**

Contoh :

```
...  
cout<<"Input kalimat : ";  
cin.getline(teks,sizeof(teks));  
strrev(teks);  
cout<<"Kalimat terbalik = "<<teks;  
...
```



**PROGRAM 17**

Output:

Input kalimat : **pemrograman** ↵  
Kalimat terbalik = namargormep



# TUGAS

Membuat flowchart dan program untuk :

- Menghitung panjang kalimat (tanpa fungsi)
- Merubah kalimat yang diinputkan menjadi huruf kapital dan sebaliknya (tanpa fungsi)
- Membalik kalimat (tanpa fungsi)
- Men-copy kalimat (tanpa fungsi)