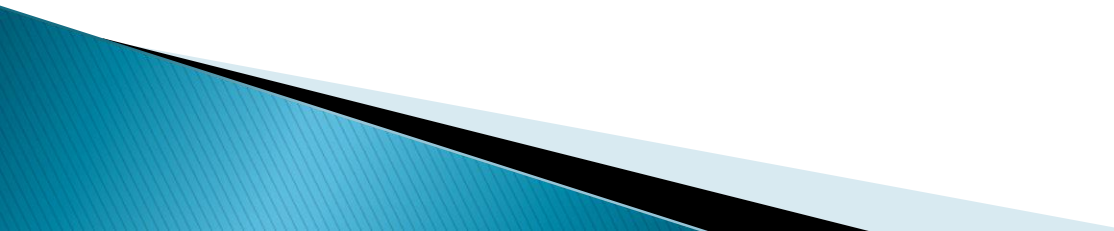


# Memanipulasi Karakter dan String

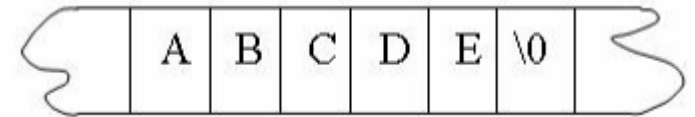
Pert. 9,10

# Pendahuluan

- ▶ String merupakan bentuk data yang biasa dipakai dalam bahasa pemrograman untuk keperluan menampung dan memanipulasi data teks, misalnya untuk menampung suatu kata atau bahkan kalimat.
  - ▶ Pada bahasa C, string bukanlah merupakan tipe data tersendiri, melainkan hanyalah kumpulan dari nilai-nilai karakter yang berurutan dalam bentuk array berdimensi satu array of char
- 

# Konstanta String

- ▶ Suatu konstanta string ditulis dengan diawali dan diakhiri tanda petik ganda, misalnya: "ABCDE"
- ▶ Nilai string ini disimpan dalam memori secara berurutan dengan komposisi sebagai berikut:



- ▶ Setiap karakter akan menempati memori sebesar 1 byte.
- ▶ Byte terakhir otomatis akan berisi karakter NULL (\0), dengan demikian maka akhir dari nilai suatu string akan dapat dideteksi.
- ▶ Sebagai sebuah array of char, karakter pertama dari nilai string mempunyai indeks ke-0, karakter kedua mempunyai indeks ke-1, dan seterusnya.

# Variabel String

- ▶ Variabel string adalah variabel yang dipakai untuk menyimpan nilai string.
- ▶ Misalnya : `char name[15];`  
merupakan instruksi untuk mendeklarasikan variabel string dengan panjang maksimal 15 karakter (termasuk karakter NULL).
- ▶ Deklarasi tersebut sebenarnya tidak lain merupakan deklarasi array bertipe char.

# Inisialisasi String

- ▶ Suatu variabel string dapat diinisialisasi seperti halnya array yang lain (dalam kurung kurawal dipisahkan koma).
- ▶ Namun tentu saja elemen terakhirnya haruslah berupa karakter NULL.
- ▶ Sebagai contoh : `char name[] = {'R','I','N','I','\0'};` yang menyatakan bahwa name adalah variabel string dengan nilai awal berupa string : “RINI”.
- ▶ Bentuk inisialisasi yang lebih singkat : `char name[] = “RINI”;` pada bentuk ini, karakter NULL tidak perlu ditulis. Secara IMPLISIT akan disisipkan oleh kompiler.

# Inisialisasi String

- Variabel string dapat diinisialisasi dengan cara seperti berikut :

`Char x[10]="Aplikasi";`

- x di memori setelah inisialisasi :

[0]				[4]				[8]	[9]
A	p	l	i	k	a	s	i	\0	?

- Karakter “\0” pada x [8] merupakan karakter null yang menandai berakhirnya suatu string

# Input Output Data String

- ▶ Untuk memasukkan atau menampilkan data String digunakan bisa beberapa fungsi standar yang ada di `stdio.h`.
- ▶ Untuk operasi input :
  - `scanf()`
  - `gets()`
  - `fgets()`
  - `cin`
- ▶ Untuk operasi output :
  - `puts()`
  - `printf()`
  - `cout`

# Input Data String (1)

- ▶ Input data string ke dalam suatu variabel biasa dilakukan dengan fungsi `gets()` atau `scanf()` atau `cin`.
- ▶ Bentuk umum pemakaiannya adalah sebagai berikut :  

```
#include <stdio.h>
    gets(nama_array);
atau #include <stdio.h>
    scanf("%s", nama_array);
atau #include <iostream>
    cin >> variabel
```



# Input Data String (2)

```
main()
```

```
{ char name[15];  
  printf("Input your name: ");  
  gets(name);  
  printf("\nHalo, %s. Good luck.\n", name);  
}
```

Ruang yang disediakan di deklarasi : char name[15]

name 

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Data yang masuk ke dalam variabel name : Chyquitha

name 

c	h	y	q	u	i	t	h	a	\0					
---	---	---	---	---	---	---	---	---	----	--	--	--	--	--

NULL 

Tidak terpakai 

# Input Data String (2)

- ▶ Fungsi `gets()` akan membaca seluruh karakter yang diketik melalui keyboard sampai tombol ENTER ditekan dengan tanpa mengecek batasan panjang array yang merupakan argumennya.
- ▶ Jika string yang dimasukkan melebihi ukuran array, maka sisa string (panjang string masukan dikurangi ukuran array plus karakter NULL) akan ditempatkan di lokasi sesudah bagian akhir dari array tersebut.

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

```
main()
{
    char name1[40];
    char name2[40];
    char kar;

    cout<<"Input Name 1 : ";
    gets(name1);

    cout<<"Input Name 2 : ";
    cin>>name2;

    cout<<"Karakter : ";
    cin>>kar;
    cout<<endl;

    cout<<"Name 1 : "<<name1<<endl;
    cout<<"Name 2 : "<<name2<<endl;
    cout<<"Karakter : "<<kar<<endl;
    getch();
}
```

```
D:\BAHAN NGAJAR KULIAH\MATERI PENGAJARAN C
Input Name 1 : Universitas Bunda Mulia
Input Name 2 : Teknik
Karakter : H

Name 1 : Universitas Bunda Mulia
Name 2 : Teknik
Karakter : H
```

← Hasil 1

```
Input Name 1 : Universitas Bunda Mulia
Input Name 2 : Teknik Informatika
Karakter :
Name 1 : Universitas Bunda Mulia
Name 2 : Teknik
Karakter : I
```

↑ Hasil 2

# Output String

- ▶ Untuk menampilkan isi variabel string, fungsi yang digunakan adalah `puts()` atau `printf()` atau `cout`.
- ▶ Bentuk umum pemakaiannya adalah sebagai berikut : `#include puts(var_string);` atau `printf("%s",var_string);` Dalam hal ini `var_string` adalah sebuah variabel yang berupa sebuah array of char.
- ▶ Fungsi `puts()` akan menampilkan isi dari `var_string` dan secara otomatis menambahkan karakter `'\n'` di akhir string.
- ▶ Sedangkan fungsi `printf()` akan menampilkan isi variabel string tanpa memberikan tambahan `'\n'`. Sehingga, agar kedua pernyataan di atas memberikan keluaran yang sama, maka pada pernyataan `printf()` dirubah menjadi : `printf("%s\n", var_string);`

# Built-in Functions untuk manipulasi String

- ▶ Untuk manipulasi string, C telah menyediakan beberapa fungsi standar yang ada pada string.h
- ▶ Beberapa yang akan dibahas kali ini adalah :
  - Fungsi strcpy()
  - Fungsi strncpy
  - Fungsi strlen()
  - Fungsi strcat()
  - Fungsi strncat()
  - Fungsi strcmp()

# Fungsi strcpy()

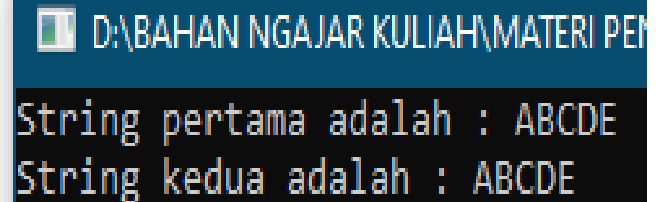
- ▶ Bentuk pemakaian :  

```
#include <string.h>
strcpy(tujuan, asal);
```
- ▶ Fungsi ini dipakai untuk mengcopy string asal ke variabel string tujuan termasuk karakter '\0'.
- ▶ Dalam hal ini, variabel tujuan haruslah mempunyai ukuran yang dapat digunakan untuk menampung seluruh karakter dari string asal

# Fungsi strcpy() -- contoh

```
#include <stdio.h>
#include <string.h>
#include <conio.h>
#define MAKS 80
main()
{
    char str1[MAKS];
    char str2[]="ABCDE";
    strcpy(str1, str2); //menyalin isi str2 ke str1
    printf("String pertama adalah : %s\n", str1);
    printf("String kedua adalah : %s\n", str2);

    getch();
}
```



D:\BAHAN NGAJAR KULIAH\MATERI PEM...  
String pertama adalah : ABCDE  
String kedua adalah : ABCDE

# Fungsi STRNCPY

- ▶ Kegunaan : menyalin nilai sumber ke variabel tujuan sebanyak n karakter □
- ▶ Syntax : □
  - Strncpy (var\_tujuan, var\_sumber / str\_sumber, n) □
  - Ket : var\_tujuan : variabel hasil (berupa variabel string)
  - Sumber bisa berupa variabel string maupun string biasa n : banyaknya karakter yang disalin □

Contoh : □

Strncpy (s1, s2, 3); → menyalin nilai di variabel s2 ke variabel s1 sebanyak 3 karakter □

Misal : s2 ← "Baru" maka s1 ← "Bar" □

Strncpy (s1, "Hello", 2); → menyalin string "He" ke variabel s1 □  
s1 ← "He"

# Bagaimana Menyalin substring yang di tengah ?

## ■ Misal :

- Char s2[10]

- Char s1[15] = "Jan. 30, 1996"

## ■ s2 bernilai berapa?

1. Strncpy (s2,s1,9) → s2 ?

- s2 bernilai "Jan. 30, "

2. Strncpy (s2, &s1[5], 2) → s2 ?

- s2 bernilai "30"



# Fungsi STRCAT

- Kegunaan : menggabungkan nilai pada 2 variabel string / menambahkan string pada sumber ke variabel tujuan
- Syntax :
  - Strcat (var\_tujuan, var\_sumber / str\_sumber)
  - Ket :  
var\_tujuan : variabel hasil (berupa variabel string)  
Sumber bisa berupa variabel string maupun string biasa
- Contoh :
  - Strcat (s1, s2); → nilai di variabel s2 ditambahkan ke variabel s1
    - Misal s1 ← "Belajar" , s2 ← "String"
    - maka s1 menjadi "BelajarString"
  - Strcat (s1, "Hello"); → Menambahkan string "Hello" ke variabel s1
    - Misal s1 ← "Belajar"
    - Maka s2 ← "BelajarHello"

# Fungsi STRNCAT

- Kegunaan : menambahkan string pada sumber ke variabel tujuan sebanyak n karakter
- Syntax :
  - Strncat (**var\_tujuan**, **var\_sumber** / **str\_sumber**, **n**)
  - Ket :
    - var\_tujuan : variabel hasil (berupa variabel string)
    - Sumber bisa berupa variabel string maupun string biasa
    - n : banyaknya karakter yang disalin
- Contoh :
  - Strncat (**s1**, **s2**, 3); → nilai di variabel s2 sebanyak 3 karakter, ditambahkan ke variabel s1
    - Misal s1 ← "Belajar" , s2 ← "String"
    - maka s1 menjadi "BelajarStr"
  - Strncat (**s1**, **"Hello"**, 2); → Menambahkan string "He" ke variabel s1
    - Misal s1 ← "Belajar"
    - Maka s2 ← "BelajarHe"

# Fungsi strlen()

- ▶ Bentuk pemakaian :  

```
#include <string.h>
    strlen(var_string);
```
- ▶ Fungsi ini digunakan untuk memperoleh banyaknya karakter di dalam string yang menjadi argumennya (var\_string).
- ▶ Keluaran dari fungsi ini adalah panjang dari var\_string(karakter NULL tidak ikut dihitung)

# Fungsi strlen()

```
#include <stdio.h>
#include <string.h>
#include <conio.h>
main() {
char salam[20];
printf("Ucapan :  ");
gets(salam);
printf("Panjang string %s = %d karakter\n", salam, strlen(salam));
getch();
}
```

 D:\BAHAN NGAJAR KULIAH\MATERI PENGAJARAN CHYQUITHA\PEMOGRAMAN DASAR\MAT

```
Ucapan :  selamat siang
Panjang string selamat siang = 13 karakter
_
```

# Fungsi strcmp()

- ▶ Membandingkan dua nilai string juga tidak dapat digunakan dengan operator hubungan, karena operator tersebut tidak untuk operasi string.
- ▶ Membandingkan dua buah nilai string secara case sensitive dapat dilakukan dengan fungsi strcmp(). • Contoh bentuk pemakaian fungsi :  
`#include strcmp(str1, str2);`
- ▶ Fungsi ini dipakai untuk membandingkan string str1 dengan string str2.
- ▶ Keluaran dari fungsi ini bertipe int yang berupa nilai :
  - -1, jika str1 kurang dari str2
  - 0, jika str1 sama dengan str2
  - 1, jika str1 lebih dari str2
- Pembandingan dilakukan untuk karakter pada posisi yang sama dari str1 dan str2, dimulai dari karakter terkiri yang didasarkan oleh nilai ASCIInya. Misal, karakter 'A' lebih kecil daripada 'B' dan karakter 'B' lebih kecil daripada 'C'.

# Fungsi strcmp()

```
#include <stdio.h>
#include <string.h>
#include <conio.h>
main(){
char str1[]="10";
char str2[]="01";
char str3[]="10";
printf("Hasil pembandingan %s dengan %s --> %d\n",
str1, str2, strcmp(str1, str2));
printf("Hasil pembandingan %s dengan %s --> %d\n",
str2, str1, strcmp(str2, str1));
printf("Hasil pembandingan %s dengan %s --> %d\n",
str1, str3, strcmp(str1, str3));
getch();
}
```

D:\BAHAN NGAJAR KULIAH\MATERI PENGAJARAN CH

```
Hasil pembandingan 10 dengan 01 --> 1
Hasil pembandingan 01 dengan 10 --> -1
Hasil pembandingan 10 dengan 10 --> 0
```

**TERIMA KASIH**