



# Pertemuan 9

## Pemrograman Terstruktur

String

## Dasar- dasar String

- ❖ String sebenarnya merupakan penggunaan tipe data char secara berkelompok
- ❖ Tipe data string merupakan array dari tipe data char

## Deklarasi Variabel String

❖ Karena string merupakan array dari char, maka pendeklarasiannya sama dengan mendeklarasikan array dari char, yaitu :

- **Char** nama\_var [ jml\_karakter ]

Contoh :

- **char** alamat [40] → deklarasi variabel alamat dengan tipe data string. Nilai Variabel alamat terdiri dari beberapa karakter maksimal 40 karakter (0 s/d 30)

## Inisialisasi Variabel 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

## Array dari String

- ❖ Karena string merupakan array dari char, maka **array string** adalah **array dua dimensi dari array char**
- ❖ **Contoh** : mendeklarasikan array yang menyimpan 30 nama, masing-masing nama terdiri dari 25 karakter

```
#define jml_orang 30
```

```
#define pjg_nama 25
```

```
....
```

```
Char nama [jml_orang] [pjg_nama]
```

## Inisialisasi Array dari String

- ❖ Inisialisasi array dari string dapat dilihat pada **contoh** berikut :

```
Char bulan[12] [10] = {"Januari",  
"Februari", "Maret", "April", "Mei",  
"Juni", "Juli", "Agustus", "September",  
"Oktober", "Nopember", "Desember"}
```

# Input/Output (I/O) dgn Printf & Scanf (1)

- ❖ Printf dan scanf dapat menangani argumen string dengan menggunakan format string %s
- ❖ Contoh:  

```
Printf("Topik : %s\n", string_var);
```
- ❖ Secara default penulisan string menggunakan alignment rata kanan

## I/O dgn Printf & Scanf (2)

- ❖ Untuk menuliskan string secara rata kiri (left justified) maka tuliskan tanda minus (-) didepan s
- ❖ Contoh : `printf (“%-20s\n”,presiden);`
- ❖ Tabel Hasil (perbedaan tanda minus dan tanpa tanda minus

Tanda minus

<code>printf(“%20s\n”,presiden);</code>	<code>printf(“%-20s\n”,presiden);</code>
George Washington	George Washington
John Adams	John Adams
Thomas Jefferson	Thomas Jefferson
James Madison	James Madison



## I/O dgn Printf & Scanf (3)

- Scanf digunakan untuk pemasukan string (input of string)
- Contoh program yang menggunakan scanf dan printf untuk I/O String

```
#include <stdio.h>
#define STRING_LEN 10

int main (void)
{
    char dept[STRING_LEN];
    int course_num;
    char days[STRING_LEN];
    int time;

    printf("Masukkan kode departemen, nomer kursus, hari dan ");
    printf("Waktu seperti berikut : \n> COSC 2060 MWF 1410\n> ");
    scanf("%s%d%s%d", dept, &course_num, days, &time);
    printf("%s %d ada pada %s jam %d\n", dept, course_num, days, time);
    return (0);
}
```

# Fungsi-fungsi pada Librari String

- ❖ `String.h` merupakan **librari** yang menyimpan fungsi-fungsi yang digunakan untuk menangani **string** ataupun **substring**
- ❖ Fungsi-fungsi dalam librari `string.h` antara lain : **`strcpy`**, **`strncpy`**, **`strcat`**, **`strncat`**, **`strcmp`**, **`strncmp`**, **`strlen`**

# Fungsi STRCPY (String Assignment)

- ❖ Kegunaan : menyalin nilai sumber ke variabel tujuan
- ❖ Syntax :
  - Strcpy (var\_tujuan, var\_sumber / str\_sumber)
  - Ket : var\_tujuan : variabel hasil (berupa variabel string)  
Sumber bisa berupa variabel string maupun string biasa
- ❖ Contoh :
  - Strcpy (s1, s2); → menyalin nilai di variabel s2 ke variabel s1
    - Misal : s2 ← “Baru” maka s1 ← “Baru”
  - Strcpy (s1, “Hello”); → menyalin string “Hello” ke variabel s1
    - s1 ← “Hello”

# Fungsi STRNCPY (String Assignment 2)

- ❖ 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 (Concatenation)

- ❖ 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 (Concatenation n karakter)

- ❖ 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”

## Perbedaan char dan string

- ❖ Char 'Q' berbeda dengan string "Q"
- ❖ Lihat perbedaanya pada penggambaran memori berikut :

Q
---

Q	\0	?	?	?	?	?	?	?
---	----	---	---	---	---	---	---	---

Character 'Q'

String "Q" (pada string Q terdapat karakter null diakhir string)

- Operasi string tidak dapat dilakukan untuk tipe data char



## Fungsi STRLEN

- ❖ Kegunaan: menghitung jumlah karakter dari suatu string
- ❖ Mengembalikan nilai integer (return value = int)
- ❖ Syntax :
  - Strlen(s1) → menghitung jumlah karakter string s1
  - Contoh :
    - Char a[8] = “hitung” , b[8] = “panjang”
    - s = strlen(a) + strlen(b)
    - s = 13

## Fungsi STRCMP(Comparison)

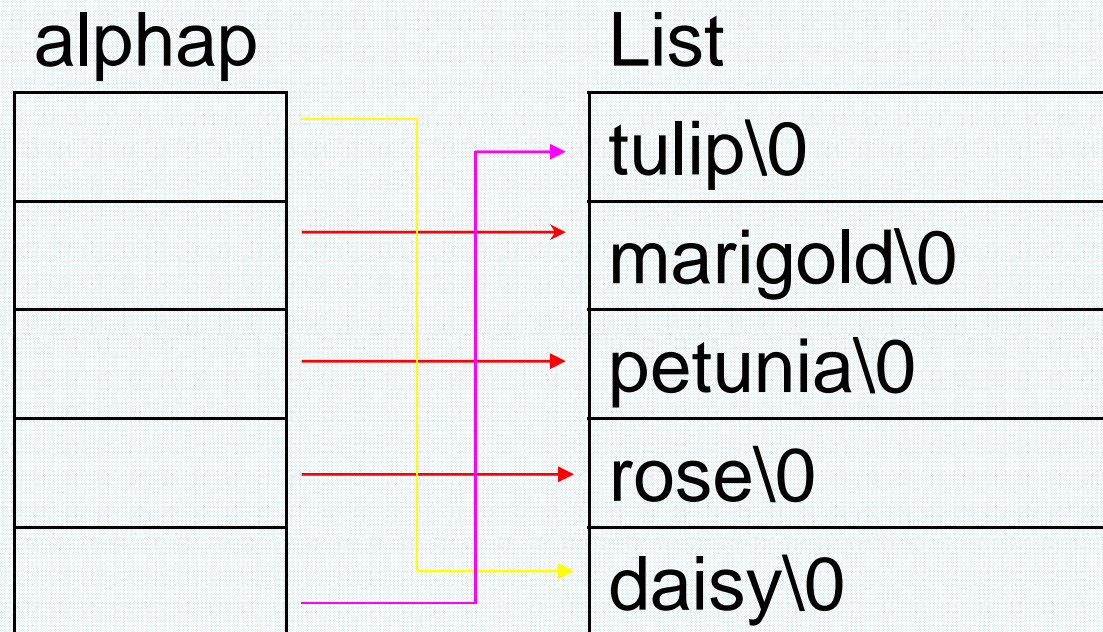
- ❖ Kegunaan : membandingkan 2 buah string
- ❖ Nilai kembalian : integer
- ❖ Syntax : strcmp(s1,s2)

Hubungan	Nilai Kembalian	Contoh :
$s1 < s2$	Negative integer	s1 : "marigold" s2 : "tulip"
$s1 = s2$	Zero (0)	s1=s2="end"
$s1 > s2$	Positive integer	s1 : "shrimp" s2 : "crab"

## Pointer Array (1)

- ❖ Char list[5][9] =  
{"tulip", "marigold", "petunia", "rose", "daisy"}
- ❖ Char \*alphap[5] → deklarasi pointer array
- ❖ Listing nilai dari alphap :
  - alphap[0] alamat dari "daisy"
  - alphap[1] alamat dari "marigold"
  - alphap[2] alamat dari "petunia"
  - alphap[3] alamat dari "rose"
  - alphap[4] alamat dari "tulip"

## Pointer Array (2)



Untuk mencetak nilai yang ditunjuk oleh alphap dapat menggunakan perulangan berikut :

```
For (i=0; i<5; ++i)  
    Printf("%s\n", alphap[i]);
```



# Operasi Karakter

## Referensi

- ❖ Bab 9, “Strings”, *Problem Solving and Program Design in C*, Jeri R. Hanly dan Elliot B. Koffman, Addison Wesley, 2002