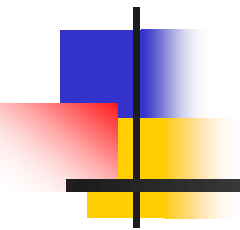


STRING





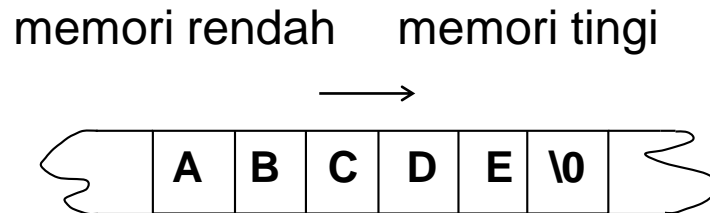
SASARAN

- Menjelaskan tentang konsep string
- Menjelaskan operasi I/O pada string.
- Menjelaskan cara mengakses elemen string
- Menjelaskan berbagai fungsi mengenai string

KONSEP STRING

- String merupakan bentuk data untuk memanipulasi data teks, misalnya untuk menampung suatu kalimat dengan menggunakan array dimensi 1.
- String ditulis dengan diawali dan diakhiri tanda petik ganda

Misal : "ABCDE"



Komposisi penyimpanan string dalam memori

- Nilai string “selalu” diakhiri dengan tanda NULL (\0)



KONSEP STRING - Lanjutan

- Variabel string

```
char name[15];
```

var string dengan panjang maksimal 15 karakter
(termasuk karakter NULL)

- Inisialisasi String

```
char name[] = {'R', 'I', 'N', 'I', '\0'};
```

```
char name[] = "RINI";
```

Jika tanpa karakter null ('\0'), maka akan keluar karakter aneh.

Dengan pembatasan array name :

```
char name[5] = {'R','I','N', 'I'}; //Jika diisi 4 → error
```

```
char name[5] = "RINI"; //Jika diisi 4 → error
```

INPUT DATA STRING

Dengan fungsi *gets()* atau *scanf()*

```
gets(nama_array);
```

```
scanf("%s", &nama_array);
```

```
fgets(nama_array, sizeof nama_array, stdin);
```

Keterangan :

a. **nama_array** adalah variabel bertipe *array of char*

b. Kalau memakai *scanf()*, data string masukan tidak boleh mengandung spasi.

c. Untuk *scanf()*, di depan *nama_array* boleh diberi & boleh tidak

Contoh :

```
char name[15];
```

```
printf("Masukkan nama Anda : ");
```

```
gets(name);
```

```
printf("\nHalo, %s. Selamat belajar string.\n", name);
```



OUTPUT DATA STRING

- Fungsi yang digunakan adalah *puts()* atau *printf()*.

```
puts (var_string) ;
```

```
printf ("%s", var_string) ;
```

Keterangan :

- **var_string** adalah berupa *array of char*
- *puts()* secara otomatis menambahkan karakter '\n' di akhir string



CONTOH STRING

```
#include <stdio.h>
#define MAKS 256
main()
{
    int i, jumkar = 0;
    char teks[MAKS];
    puts("Masukkan suatu kalimat (maks 255 karakter).");
    //masukan dr keyboard
    gets(teks);
    for(i=0; teks[i]; i++)
        jumkar++;
    printf("\nJumlah karakter = %d\n", jumkar);
}
```

PENGECEKAN AKHIR STRING



- Pemeriksaan terhadap **teks[i]** selama tidak berupa karakter NULL (dimulai dari indeks ke-0) dilakukan dengan instruksi

```
for (i=0; teks[i]; i++)  
    jumkar++;
```

Kondisi **teks[i]** pada *for* mempunyai makna yang secara implisit berupa

```
teks[i] != '\0';
```




CONTOH STRING

```
#include <stdio.h>
#define MAKS 30
main()
{
    int i;
    char keterangan[] = "Saya menyukai bahasa C";
    char kalimat[MAKS];
    i=0;
    while (keterangan[i] != '\0')
    {
        kalimat[i] = keterangan[i];
        i++;
    }
    kalimat[i] = '\0'; /* beri karakter NULL */
    printf("Isi kalimat : %s\n", kalimat);
}
```

Untuk yang lebih sederhana :

```
i=0;
while (kalimat[i] = keterangan[i])
    i++;
```



FUNGSI-FUNGSI STRING

Fungsi-fungsi pustaka untuk operasi string, prototype-prototypenya berada di file judul string.h

A. Fungsi *strcpy()* untuk Menyalin Nilai String

Bentuk pemakaian :

```
#include <string.h>
strcpy(tujuan, asal);
```

NB : menyalin string **asal** ke string **tujuan** termasuk karakter '\0'.

Output adalah string **tujuan**.



Contoh Fungsi strcpy()

```
#include <stdio.h>
#include <string.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);
}
```

FUNGSI STRING – Cont. 1

B. Fungsi *strlen()* untuk Mengetahui Panjang Nilai String

Bentuk pemakaian :

```
#include <string.h>
strlen(var_string);
```

NB : Untuk memperoleh banyaknya karakter di dalam string (var_string).

Output adalah panjang dari var_string.

Karakter NULL tidak ikut dihitung.

```
//Contoh Program dengan Fungsi strlen()
#include <stdio.h>
#include <string.h>
main()
{
    char salam[] = "Halo";
    printf("Panjang string = %d karakter\n", strlen(salam));
}
```

FUNGSI STRING – Cont. 2

C. Fungsi *strcat()* untuk Menggabung Nilai String

Bentuk pemakaian :

```
#include <string.h>
strcat(tujuan, sumber);
```

NB : Menambahkan string **sumber** ke bag akhir dari string **tujuan**.
Output adalah string **tujuan**.

//Contoh Program dengan strcat()

```
#include <stdio.h>
#include <string.h>
#define PJJ 15
main()
{
    char str1[PJJ], str2[PJJ];
    strcpy(str1, "sala");    /* str1 diisi "sala" */
    strcpy(str2, "tiga");    /* str2 diisi "tiga" */
    strcat(str1, str2);      /* tambahkan str2 ke akhir str1 */
    printf("str1 → %s      str2 → %s\n", str1, str2);
}
```

FUNGSI STRING – Cont. 3

D. Fungsi *strcmp()* untuk Membandingkan Dua Nilai String

Bentuk pemakaian :

```
#include <string.h>
strcmp(str1, str2);
```

NB : Untuk membandingkan string str1 dengan string str2. Acuan perbandingan dimulai dari karakter terkecil dan didasarkan pada nilai ASCII-nya, mis : 'A' < 'B'

Output adalah bertipe *int* yang berupa nilai :

- > -1, jika str1 kurang dari str2
- > 0, jika str1 sama dengan str2
- > 1, jika str1 lebih dari str2

Contoh Fungsi strcmp()

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

```
Hasil perbandingan HALO dengan Halo --> -1
Hasil perbandingan Halo dengan HALO --> 1
Hasil perbandingan HALO dengan HALO --> 0
```

FUNGSI STRING – Cont. 4

E. Fungsi *strchr()* untuk Mencari Nilai Karakter dalam String

Bentuk pemakaian :

```
#include <string.h>
strchr(var_string, kar);
```

NB : Untuk mencari nilai karakter yang berada dalam nilai string. Output adalah alamat posisi dari karakter pertama pada nilai string, yang sama dengan karakter yang dicari. Jika karakter yang dicari tidak ada dalam nilai string, maka akan diberikan hasil nilai pointer kosong (*null*).

Contoh Fungsi strchr()

```
#include <stdio.h>
#include <string.h>
main()
{
    char str[]="ABcde"; // inialisasi string
    char *hasil1,*hasil2;
    /* var bertipe pointer to char, agar bisa ditampilkan
       isi dari alamat yang ditunjuk oleh hasil1 & hasil2 */
    hasil1 = strchr(str, 'B');
    hasil2 = strchr(str, 'X');
    printf("Dari string ABcde\n");
    printf("Mencari karakter B = %s\n", hasil1);
    printf("Mencari karakter X = %s\n", hasil2);
}
```

```
Dari string ABcde
Mencari karakter B = Bcde
Mencari karakter X = <null>
```



LATIHAN

1. Ketikkan sebuah kalimat melalui keyboard dengan menggunakan `gets()` (atau `fgets()`) kemudian didapatkan keluaran berupa laporan tentang jumlah huruf kecil dan huruf kapital dalam kalimat tsb.
2. Masukkan nama Anda, rubah ke dalam huruf besar semua, balikkan urutan hurufnya, selanjutnya tampilkan hasilnya di layar.
3. Ketikkan sebuah kalimat, hitung dan tampilkan jumlah spasinya.
4. Ketikkan sebuah kalimat, kemudian tampilkan kalimat tsb satu kata perbaris. Asumsikan ada satu spasi yang memisahkan setiap kata dan kalimat diakhiri dengan sebuah tanda titik.



5. Buat program untuk menghasilkan keluaran sbb :

POLITEKNIK

POLITEKNI

POLITEKN

POLITEK

POLITE

POLIT

POLI

POL

PO

P

Petunjuk :

gunakan fungsi *strlen()* untuk menghitung panjang string yang prototypenya ada di file <string.h>

gunakan *function call* (pemanggilan fungsi)

index array dimulai dari indeks ke-0