BAB IX
STRING & MATH

IF

Tujuan

1. Menjelaskan tentang konsep string
2. Menjelaskan operasi I/O
3. Menjelaskan cara mengkases elemen string
4. Menjelaskan berbagai fungsi mengenai string

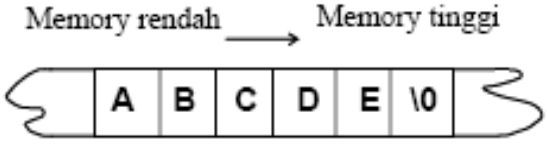
Dasar Teori

String merupakan bentuk data yang biasa dipakai dalam bahasa pemrograman keperluan menampung dan memanipulasi data teks. Misalnya untuk menampung (menyimpan) suatu kalimat. Pada bahasa pemrograman C, string bukanlah sebagai tipe data tersendiri, melainkan hanya jenis khusus dari array. String ditulis dengan diawali dan diakhiri tanda petik ganda dan nilai string selalu diakhiri dengan tanda NULL (\0).

Misalnya : “ABCDE”

1. Konstanta String

Konstanta “ABCDE” disimpan dalam memory secara berurutan, dengan komposisi sebagai berikut :



Setiap karakter akan menempati memory sebesar 1 byte. Byte terakhir secara otomatis akan berisi karakter NULL (\0).

2. Variabel String

Variabel string adalah variabel yang digunakan untuk menyimpan string.

Misal :

```
char nama[15];
```

merupakan instruksi untuk mendeklarasikan variabel string dengan panjang maksimal 15 karakter.

Inisialisasi String

Cara untuk menginisialisasi string, dapat dilakukan dengan salah satu cara dibawah ini :

```
char nama[]=”ini adalah string”;
char nama2[]={‘i’,’n’,’i’,’’,’s’,’t’,’r’,’i’,’n’,’g’,’\0’};
char nama3[5]=”ACER”;
char nama4[5]={‘A’,’C’,’E’,’R’};
```

Untuk mengisi suatu string caranya adalah :

```
stcpy (nama, “Ini string”);
nama5=”Ini juga string”;
```

Coba diperhatikan, untuk string yang dideklarasikan sebagai sebuah array karakter, pengisian nilainya adalah dengan menggunakan suatu perintah **`strcpy`** yang berguna untuk mengisikan suatu string ke string lain. Pengisiannya tidak boleh langsung. Tetapi jika string dideklarasikan sebagai sebuah pointer karakter, maka pengisiannya boleh diisikan secara langsung.

Input Data String dari Keyboard

Untuk memasukan data string kedalam suatu variabel dapat dilakukan dengan 2 jenis perintah/fungsi :

a. `gets (nama_array)`
b. `scanf(“%s”,&nama_array);`

Keterangan :

- nama_array adalah variabel bertipe `array_of_char`
- Pada isntruksi `scanf()`, didepan `nama_array` boleh diberikan operator &, boleh tidak.
- Prototype `gets()` ada pada file stdio.h
- `gets()` akan membaca seluruh karakter yang diketik sampai penekanan ENTER. Tidak ada pengecekan terhadap batasan dari array. Jika string yang dimasukan melebihi ukuran array, sisa string berikutnya akan ditempatkan sesudah bagian terakhir dari array. Hal ini akan menimbulkan kejadian yang tidak diinginkan, seperti berubahnya isi variabel.

Output Data String ke Layar

Untuk menampilkan isi variabel string ke layar, dapat dilakukan dengan 2 jenis perintah / fungsi :

```
a. puts(var_string);
b. printf("%s",var_string);
```

yang perlu diperhatikan adalah :

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

Fungsi-fungsi String

Merupakan kumpulan fungsi-fungsi pustaka string yang berada pada file **`string.h`**

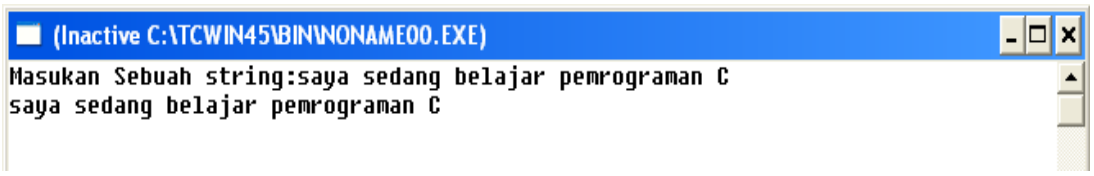
❖ **`gets`** dan **`puts`**

Fungsi **`gets`** digunakan untuk membaca data berupa string dari keyboard.

Fungsi **`puts`** digunakan untuk menampilkan suatu string ke layar (monitor)

```
#include <stdio.h>
main()
{
char string[80];
```

```
printf("Masukan Sebuah string:");gets(string);
puts(string);
return 0;
}
```



❖ **strlen**

Fungsi **strlen** digunakan untuk mengetahui panjang suatu string.

Contoh program :

```
#include <stdio.h>
#include <string.h>
int main(void)
{
      char string[80];
      int panjang;
      printf("Masukan String: ");gets(string);
      panjang=strlen(string);
      printf("Panjang String adalah %i karakter\n",panjang);
      return 0;
}
```



❖ **strcpy dan strncpy**

Fungsi **`strcpy`** adalah untuk menyalin isi suatu string ke string lain.

Fungsi **`strncpy`** berfungsi untuk menyalin isi suatu string ke string lain sebanyak n karakter.

Contoh program :

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
main()
{
      char string[31];
      char str1[31] = "Universitas Komputer Indonesia";
      char str2[31];
      strcpy(string, str1);
      printf("Isi String : %s\n", string);
      strncpy(str2,str1,15);
      str2[15]='\0';// menutup string
      printf("Isi Str2 : %s",str2);
      getch();
      return 0;
}
```



❖ **strcmp dan strncmp**

Fungsi **strcmp** digunakan untuk membandingkan 2 buah string secara case sensitive

Fungsi **strncmp** digunakan untuk membandingkan 2 buah string sebanyak n buah karakter secara case sensitive.

Fungsi **strcmpi** digunakan untuk membandingkan 2 buah string secara insensitive

Fungsi **strncmpi** digunakan untuk membandingkan 2 buah string sebanyak n buah karakter secara insensitive.

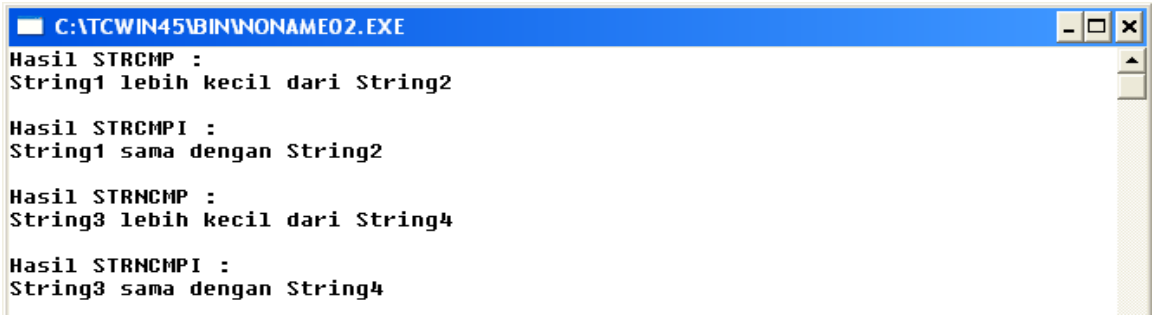
Semua fungsi tersebut akan menghasilkan sebuah nilai integer yang mempunyai ketentuan :

- Nilai return akan lebih dari 0 (>0) ketika string1 lebih besar dari string2
- Nilai terurn akan sama dengan 0 (==0) ketika string1 sama dengan string2

- Nilai return akan kurang dari 0 (<0) ketika string1 lebih kecil dari string2

Contoh Program

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
main()
{
char str1[5]="ABCD",str2[5]="abcd",str3[5]="BCDE",str4[5]="BCda";
int hasil;
hasil=strcmp(str1,str2);
printf("Hasil STRCMP : \n");
if(hasil==0) printf("String1 sama dengan String2\n"); else
if(hasil>0) printf("String1 lebih besar dari String2\n"); else
if(hasil<0) printf("String1 lebih kecil dari String2\n");
hasil=strcmpi(str1,str2);
printf("\nHasil STRCMPI : \n");
if(hasil==0) printf("String1 sama dengan String2\n"); else
if(hasil>0) printf("String1 lebih besar dari String2\n"); else
if(hasil<0) printf("String1 lebih kecil dari String2\n");
hasil=strncmp(str3,str4,3);
printf("\nHasil STRNCMP : \n");
if(hasil==0) printf("String3 sama dengan String4\n"); else
if(hasil>0) printf("String3 lebih besar dari String4\n"); else
if(hasil<0) printf("String3 lebih kecil dari String4\n");
hasil=strncmpi(str3,str4,3);
printf("\nHasil STRNCMPI : \n");
if(hasil==0) printf("String3 sama dengan String4\n"); else
if(hasil>0) printf("String3 lebih besar dari String4\n"); else
if(hasil<0) printf("String3 lebih kecil dari String4\n");
getch();
return 0;
}
```



❖ **strcat dan strncat**

Fungsi **strcat** adalah untuk menggabungkan 2 buah string

Fungsi **strncat** berfungsi untuk menggabungkan 2 buah string sebanyak n karakter

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
main()
{
char str1[80];
char str2[15]="Universitas ";
char str3[20]="Komputer Indonesia";
clrscr();
strcpy(str1,str2);
strcat(str1,str3);
printf("Hasil penggabungan dengan STRCAT : %s\n",str1);
strcpy(str1,str2);
strncat(str1,str3,8);
printf("Hasil penggabungan dengan STRNCAT : %s\n",str1);
getch();
return 0;
}
```



❖ **strlwr dan strupr**

Fungsi **strlwr** berguna untuk mengubah isi string menjadi huruf kecil.

Fungsi **strupr** berguna untuk mengubah isi string menjadi kapital

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
main()
{
   char str1[80]="Universitas Komputer Indonesia";
   clrscr();
   printf("Normal : %s\n",str1);
   strupr(str1);
   printf("UpperCase : %s\n",str1);
   strlwr(str1);
   printf("LowerCase : %s\n",str1);
getch();
return 0;
}
```



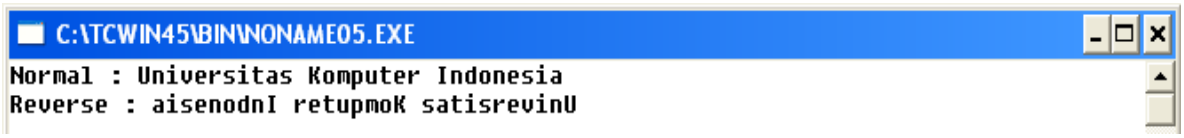
❖ **strrev**

Fungsi **`strrev`** berguna untuk membalikan urutan string (reverse)

Contoh program :

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
main()
{
      char str1[80]="Universitas Komputer Indonesia";
      clrscr();
      printf("Normal : %s\n",str1);
      strrev(str1);
      printf("Reverse : %s\n",str1);
getch();
return 0;
```

```
}
```



❖ **strset dan strnset**

Fungsi **strset** berguna untuk mengganti isi suatu string dengan suatu karakter tertentu

Fungsi **strnset** berguna untuk mengganti isi suatu strin fdengan suatu karakter tertentu sebanyak n buah data.

Contoh program :

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
main()
{
      char str1[31]="Universitas Komputer Indonesia";
      clrscr();
      strnset(str1,'A',12);
      printf("Setelah strnset 12 : %s\n",str1);
      strset(str1,'x');
      printf("Setelah strset : %s\n",str1);
      getch();
      return 0;
}
```

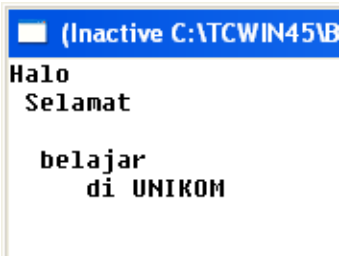


❖ **gotoxy**

```
#include <stdio.h>

main()
   {
     gotoxy(1,1);printf("Halo\n");
     gotoxy(2,2);printf("Selamat");
     gotoxy(3,4);printf("belajar");
     gotoxy(6,5);printf("di UNIKOM");

   }
```



Math

Dibawah ini akan diberikan beberapa fungsi dasar penggunakan fungsi matematika yang sering digunakan. Fungsi math ini menggunakan pustaka dari “math.h”

- **sqrt**

  sqrt digunakan untuk menghitung akar dari suatu bilangan

- **sin, cos, tan**

  digunakan dalam perhitungan trigonometri

- **pow**

  digunakan untuk menghitung pangkat

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

main()
      {
            float h_sin,h_cos,h_tan;
            int sudut=30;
            int x,y;
            float akar,pangkat;
            x=3;
            y=4;

            h_sin=sin(sudut*M_PI/180);
            h_cos=cos(sudut*M_PI/180);
            h_tan=tan(sudut*M_PI/180);

            akar=sqrt(y);
            pangkat=pow(x,y);

            printf("Sin %i adalah = %.3f",sudut,h_sin);
            printf("\nCos %i adalah = %.3f",sudut,h_cos);
            printf("\nTan %i adalah = %.3f",sudut,h_tan);
            printf("\n");
            printf("\nAkar dari %i adalah = %g",y,akar);
            printf("\n%i pangkat %i adalah = %g",x,y,pangkat);

            getch();
            return 0;
      }
```

