Praktikum 9 (1/3)

STRING

A. TUJUAN

1. Memahamkan konsep tipe data string
2. Memahamkan string sebagai array dari karakter
3. Dapat memasukkan dan menampilkan isi variabel string

B. DASAR TEORI

Konstanta dan Variabel String

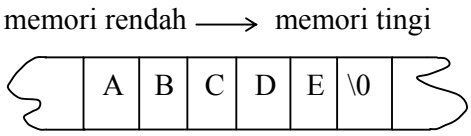
String merupakan bentuk data yang biasa dipakai dalam bahasa pemrograman untuk keperluan menampung dan memanipulasi data teks, misalnya untuk menampung (menyimpan) suatu kalimat. Pada bahasa C, string bukanlah merupakan tipe data tersendiri, melainkan hanyalah kumpulan dari nilai-nilai karakter yang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

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:



Gambar 9.1.1 Komposisi penyimpanan string dalam memori

Setiap karakter akan menempati memori sebesar 1 byte. Byte terakhir otomatis akan berisi karakter NULL (\0). Dengan mengetahui bahwa suatu string diakhiri nilai

NULL, maka akhir dari nilai suatu string akan dapat dideteksi. Sebagai sebuah array karakter, 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 utuk 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. 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. Perlu diperhatikan, bila name dideklarasikan sebagai string, penugasan (assignment) suatu string ke variabel string seperti

```
name = “RINI”;
```

adalah tidak diperkenankan. Pengisian string ke variabel string akan dibahas pada sub bab berikutnya.

Memasukkan Data String

Pemasukan data string ke dalam suatu variabel biasa dilakukan dengan fungsi gets() atau scanf(). Bentuk umum pemakaiannya adalah sebagai berikut :

```
#include <stdio.h>
gets(nama_array);
```

atau

```
#include <stdio.h>
scanf("%s", nama_array);
```

Perhatikan : → nama_array adalah variabel bertipe array of char yang akan digunakan untuk menyimpan string masukan.

→ Di depan nama_array tidak perlu ada operator & (operator alamat), karena nama_array tanpa kurung siku sudah menyatakan alamat yang ditempati oleh elemen pertama dari array tsb.

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

Perlu diketahui, fungsi gets() akan membaca seluruh karakter yang diketik melalui keyboard sampai tombol ENTER ditekan. Dalam hal ini tidak ada pengecekan terhadap 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. Tentu saja kejadian seperti ini bisa menimbulkan hal yang tidak diinginkan, misalnya berubahnya isi variabel yang dideklarasikan sesudah array tersebut karena tertumpuki oleh string yang dimasukkan (overwrite), atau perilaku program yang sama sekali berbeda dengan kemauan user yang dalam hal ini pelacakan kesalahannya (debugging) sangat sulit dilakukan, atau bahkan terjadi penghentian program secara tidak normal.

Untuk mengatasi hal itu, disarankan untuk menggunakan fungsi fgets() untuk menggantikan fungsi gets() dalam memasukkan data string.

Bentuk umum pemakaian fgets() adalah :

```
#include <stdio.h>
fgets(nama_array, sizeof nama_array, stdin);
```

Menampilkan Isi Variabel String

Untuk menampilkan isi variabel string, fungsi yang digunakan adalah puts() atau printf(). Bentuk umum pemakaiannya adalah sebagai berikut :

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

C. PERCOBAAN.

1. Buatlah program yang dapat menampikan identitas anda, NIP, Nama, Alamat,Telepon, Tanggal lahir, Hobby.

2. Buatlah program sebagai berikut:

   Input:

   Nama Anda =

   Output:

   "Wahai <Nama Anda> , ketahuilah bahwa Anda sedang belajar String"

3. Buatlah program sebagai berikut:

   Input:

   Masukkan berupa huruf dari keybord (type data char) yang di tampung di dalam array dari karakter, sampai ditekan enter atau sebanyak 100 karakter,.

   Output:

   Array dari karakter yang diinputkan tadi ditampilkan sebagai String.

4. Dengan mengaju cara pada Program no 3 buatlah program untuk membuat kata sandi, dengan cara menggeser 1 huruf yaitu A→ B, B→C,…… Z→A, dan a→b, b→c,… z→a

   Input:

   Masukkan kata :

   Output:

   Kata yang disandikan:

   Misal:

   Masukkan kata : PENS ITS

   Kata yang disandikan: QFOT JUT

5. Dengan mengacu cara pada Program no 3 buatlah, program yang digunakan untuk menghitung banyaknya karakter dari kalimat yang dimasukkan melalui keyboard (termasuk karakter spasi). Untuk mengakhiri pemasukan kalimat, tombol ENTER ('`\n`') harus ditekan

   Input : Ketikkan sembarang kalimat

   Output : jumlah karakter = m

   jumlah spasi =n

D. LAPORAN RESMI

1. Tulis listing program dari semua percobaan yang dilakukan.

2. Kemudian tuliskan outputnya. Terangkan kenapa demikian.

3. Buatlah program untuk melengkapi program no 5 dengan menghitung jumlah masing-masing karakter.