

PRAKTIKUM 10

STRING

A. Tujuan

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

B. DASAR TEORI

Konstanta dan Variabel String

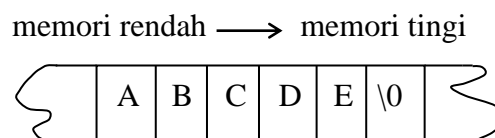
String merupakan bentuk data yang biasa dipakai dalam bahasa pemrograman untuk keperluan menampung dan memanipulasi data teks, misalnya untuk menampung (menyimpan) suatu kalimat. Pada bahasa C, string bukanlah merupakan tipe data tersendiri, melainkan hanyalah kumpulan dari nilai-nilai karakter yang 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 11.1 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 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. 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.

Input Output Data String

Memasukkan Data String

Pemasukan data string ke dalam suatu variabel biasa dilakukan dengan fungsi *gets()* atau *scanf()*.

```
#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.

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);
```

Fungsi-Fungsi Mengenai String

Berikut ini akan dibahas beberapa fungsi pustaka yang umumnya disediakan oleh kompiler C untuk mengoperasikan suatu nilai string. Fungsi-fungsi pustaka untuk operasi string, prototype-prototype nya berada di file judul **string.h**. Beberapa di antara fungsi pustaka untuk operasi string akan dibahas di bawah ini.

Fungsi *strcpy()* untuk Menyalin Nilai String

Bentuk pemakaian :

```
#include <string.h>
```

```
strcpy(tujuan, asal)
```

Fungsi ini dipakai untuk menyalin string **asal** ke variabel string **tujuan** termasuk karakter '\0'. Keluaran dari fungsi ini (*return value*) adalah string **tujuan**. Dalam hal ini, variabel **tujuan** haruslah mempunyai ukuran yang dapat digunakan untuk menampung seluruh karakter dari string **asal**.

Fungsi *strlen()* untuk Mengetahui Panjang Nilai String

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 *strcat()* untuk Menggabung Nilai String

Bentuk pemakaian :

```
#include <string.h>

strcat(tujuan, sumber);
```

Menggabungkan dua buah nilai string tidak dapat dilakukan dengan operator '+', karena operator ini bukan operator untuk operasi string. Penggabungan dua buah nilai string dapat dilakukan dengan fungsi pustaka *strcat()* dengan menambahkan string **sumber** ke bagian akhir dari string **tujuan**. Keluaran dari fungsi ini adalah string **tujuan**.

Fungsi *strcmp()* untuk Membandingkan Dua Nilai String

Membandingkan dua nilai string juga tidak dapat digunakan dengan operator hubungan, karena operator tersebut tidak untuk operasi string. Membandingkan dua buah nilai string dapat dilakukan dengan fungsi pustaka *strcmp()*.

Contoh bentuk pemakaian fungsi :

```
#include <string.h>

strcmp(str1, str2);
```

C. TUGAS PENDAHULUAN

1. Buat program untuk menghitung panjang string .
Tampilan:
Masukkan String:Andika
Jadi panjang stringnya adalah 6
2. Masukkan nama Anda dalam huruf besar, rubah ke dalam huruf kecil semua, balikkan urutan hurufnya, selanjutnya tampilkan hasilnya di layar.

D. PERCOBAAN

1. Ketikkan sebuah kalimat melalui keyboard kemudian didapatkan keluaran berupa laporan apakah kalimat tsb palindrom ataukah bukan. Misal :

Kalimat : KASUR RUSAK

Termasuk PALINDROM

Kalimat : MAKAN MALAM

Bukan PALINDROM

Catatan : disebut palindrom adalah bila urutan kalimat dibalik akan menghasilkan kalimat yang sama

2. Buatlah program untuk membaca sebuah string dan mengganti setiap kemunculan huruf 'a' pada string tersebut dengan huruf 'i' dan menampilkannya di layar .
3. Buatlah program yang membaca dua buah string dan membandingkan kedua string tersebut serta menampilkan pesan di layar 'kedua string sama' jika kedua string tersebut sama, dan 'kedua string tidak sama' jika sebaliknya.
4. Buatlah program untuk mengurutkan nama seseorang. Contoh:

Sebelum diurutkan

Nama ke-1: Suadi marwan

Nama ke-2: Abdi Raharjo

Nama ke-3:Shintiana

Nama ke-4:Budi Sulisty

Sesudah diurutkan

Nama ke-1: Abdi Raharjo

Nama ke-2: Budi Sulisty

Nama ke-3:Shintiana

Nama ke-4:Suadi Marwan

E. LAPORAN RESMI

1. Buatlah Flowchart dari percobaan yang telah dilakukan.

2. Buat program untuk menghasilkan keluaran sbb :

POLITEKNIK
POLITEKNI
POLITEKN
POLITEK
POLITE
POLIT
POLI
POL
PO
P

3. Modifikasilah program tersebut hingga menampilkan hasil sbb:

P
PO
POL
POLI
POLIT
POLITE
POLITEK
POLITEKN
POLITEKNI
POLITEKNIK