

PRAKTIKUM 22

String 2

A. TUJUAN PEMBELAJARAN

1. Memanfaatkan berbagai *built-in function* untuk manipulasi string
2. Dapat menggunakan fungsi `strcat()`, `strcmp()` dan `strcmpi()` untuk memanipulasi string
3. Mahasiswa dapat menggunakan Algoritma String dalam menyelesaikan masalah

B. DASAR TEORI

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.

1. 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**. Contoh implementasinya bisa dilihat pada program **salinstr2.c** di bawah ini.

2. 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.

3. 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**.

4. 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);
```

Fungsi ini dipakai untuk membandingkan string **str1** dengan string **str2**. Keluaran dari fungsi ini bertipe *int* yang berupa nilai :

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

Pembandingan dilakukan untuk karakter pada posisi yang sama dari **str1** dan **str2**, dimulai dari karakter terkiri. Acuan pembandingan dari dua buah karakter didasarkan oleh nilai ASCII-nya. Misal, karakter 'A' lebih kecil daripada 'B' dan karakter 'B' lebih kecil daripada 'C'.

Fungsi `strchr()` untuk Mencari Nilai Karakter dalam String

Bentuk pemakaian

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

Fungsi ini dapat digunakan untuk mencari suatu nilai karakter yang berada dalam suatu nilai string. Dalam hal ini adalah mencari karakter **kar** dalam string **var_string**. Keluaran dari fungsi ini 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 fungsi ini akan memberikan hasil nilai pointer kosong (NULL).

C. TUGAS PENDAHULUAN

Untuk semua persoalan di bawah ini, desainlah algoritma dan flowchartnya.

D. PERCOBAAN

1. Ulangilah soal nomor 2, 3 & 4 pada modul prakt String 1 dengan menggunakan fungsi-fungsi standard (*built-in functions*)
2. a. Definisikanlah sebuah fungsi untuk membandingkan dua buah string yang menjadi parameternya secara CASE SENSITIVE (misalnya : `bandingstr1()`)
b. Pada fungsi `main()` terimalah input 2 buah string yang akan dibandingkan. Selanjutnya panggil fungsi `bandingstr1()` untuk membandingkan kedua string tsb dan tampilkan hasilnya di `main()`

3. a. Definiskanlah sebuah fungsi untuk membandingkan dua buah string yang menjadi parameternya secara NON CASE SENSITIVE (misalnya : `bandingstr2()`)
- b. Pada fungsi `main()` terimalah input 2 buah string yang akan dibandingkan.. Selanjutnya panggil fungsi `bandingstr2()` untuk membandingkan kedua string tsb dan tampilkan hasilnya di `main()`

Keterangan : lihat algoritma Fungsi Banding – NON CASE SENSITIVE

4. Lakukan percobaan untuk membandingkan 2 buah string dengan menggunakan fungsi `strcmp()` dan `strcmpi()`. Analisislah dan berikan kesimpulan tentang perbedaan dan contoh aplikasi untuk keduanya

NOTE :

- Kode ASCII untuk

A= 65	a = 97	→ selisihnya 32
Z = 90	z = 112	→ selisihnya 32
- `Kata[i] > 90` → berupa karakter dalam lowercase
- `Kata[i] <= 90` → berupa karakter dalam UPPERCASE
- Untuk nilai absolut gunakan fungsi standar `fabs()` → `math.h`
- Hasil operasi perbandingan 2 string secara NON CASE SENSITIVE:
 - -1 → `string1 < string2`
 - 0 → `string1 = string2`
 - 1 → `string1 > string2`

E. LAPORAN RESMI

1. Untuk setiap listing program dari percobaan-percobaan di atas, ambil *capture* outputnya.
2. Tuliskan kesimpulan dari percobaan yang telah anda lakukan.