



UJIAN AKHIR SEMESTER GASAL 2019 – 2020

Mata Kuliah	:	Algoritma dan Pemrograman (TF4204)
Hari / Tanggal	:	Kamis / 12 Desember 2019
Waktu	:	150 menit (10.30 – 13.00 WIB)
Sifat Ujian	:	Tertutup
Dosen	:	Tim Dosen

1. (30 poin: 10 poin CPMK 5.1, 20 poin CPMK 5.2)

Buatlah program untuk membandingkan *string* dengan 2 fungsi bantuan berikut ini:

(15 poin) Fungsi pertama adalah fungsi **Sanitize** yang menerima parameter sebuah *string* dan menghasilkan sebuah *string* yang sudah “dibersihkan” dari karakter selain alfabet, angka, dan spasi.

Contoh penggunaan fungsi:

```
string kal = "Sa.y*a m#akan n(a)si";  
kal = Sanitize (kal);  
cout << kal << endl; // outputnya adalah: Saya makan nasi
```

(15 poin) Fungsi kedua adalah fungsi **Compare** yang menerima parameter dua buah *string*. Fungsi akan menghasilkan nilai *true* jika kedua *string* sama dan *false* jika kedua *string* berbeda, tanpa memperhatikan karakter selain alfabet, angka, dan spasi. Untuk memudahkan pekerjaan anda, gunakan fungsi **Sanitize** di dalam fungsi **Compare** untuk menghilangkan karakter-karakter selain alfabet, angka, dan spasi pada kedua *string* sebelum membandingkan kedua string tersebut (*secara case-sensitive*).

Contoh penggunaan fungsi:

```
string string1 = "Sa.y*a m#akan n(a)si";  
string string2 = "S-ay-a _makan n_as_i";  
if(Compare(kal1, kal2))  
    cout << "String 1 sama dengan String 2" << endl;  
else  
    cout << "String 1 tidak sama dengan String 2" << endl;
```

Untuk memastikan fungsi sudah benar, buatlah program yang menerima input dua buah *string*, kemudian membandingkan kedua *string* tersebut dengan fungsi **Compare**.

Contoh:

Input: String 1: Sa.y*a m#akan n(a)si
String 2: S-ay-a _makan n_as_i
Output: String 1 sama dengan String 2

Input: String 1: Nas1 g0renG har^ga 5000
String 2: N@si go-r3ng ha_rga 5000
Output: String 1 tidak sama dengan String 2



Keterangan (tidak perlu dituliskan sebagai output):

Nilai string setelah "dibersihkan":

String 1: Saya makan nasi

String 2: Saya makan nasi

Sehingga String 1 = String 2

Nilai string setelah "dibersihkan":

String 1: Nas1 g0renG harga 5000

String 2: Nsi gor3ng harga 5000

Sehingga String 1 != String 2

2. (40 poin: 12 poin CPMK 5.1, 16 poin CPMK 5.2, 12 poin CPMK 5.3)

Buatlah sebuah *Array Management System*. Program ini berisi beberapa fitur yaitu mengatur array dan menyimpan array tersebut kedalam file. Ketika program pertama berjalan, akan muncul menu berikut ini:

1) Create Array (8 poin): menu ini akan meminta pengguna untuk melakukan input N, kemudian pengguna akan melakukan input int N kali. Setelah melakukan input N kali, program akan meminta input nama file, simpan array tersebut ke dalam file.

Contoh:

Input N: 3

Input Array Index 0: 2

Input Array Index 1: 8

Input Array Index 2: 3

Save Array into: Array.dat

2) Sort Array (8 poin): menu ini akan menerima input nama file input, nama file output dan mode sorting (Asc / Desc).

Contoh:

Input File: Array.dat *) sama dengan file diatas!

Output File: sorted_array3.bin

Mode (1 = ASC / 2 = DESC): 1

*) isi sorted_array3.bin: 2 3 8

3) Display Array (8 poin): menu ini akan menerima input nama file, kemudian keluarkan / tampilkan isi array dalam file tersebut ke dalam *console*.

Contoh:

Input: sorted_array3.bin

Output: 2 3 8

4) Exit: menu ini akan membuat program berhenti.



Buatlah 2 fungsi dibawah ini untuk membantu menu sort array:

- (6 poin) Fungsi swap:

```
int a = 3, b = 5;

swap(a, b);
cout << a << endl; // 5
cout << b << endl; // 3
```

- (10 poin) Fungsi sort_array:

```
int arr[5] = {3, 5, 1, 2, 4};
int co = 5;
int mode = 1; // mode ascending

sort_array(arr, co, mode);
// isi arr = {1, 2, 3, 4, 5};
```

*) File penyimpanan boleh dalam bentuk file text / file biner

3. (30 poin: 14 poin CPMK 5.1, 16 poin CPMK 5.3)

Buatlah program sederhana untuk menggunakan *console window* sebagai simulasi penggambaran garis-garis horizontal dan vertikal pada sebuah *canvas*. Asumsikan *canvas* berukuran 32 x 16 unit dengan koordinat (0, 0) sebagai posisi ujung kiri atas dan koordinat (31, 15) sebagai posisi ujung kanan bawah. Gunakan variabel *array 2D* untuk menyimpan *canvas* dan gunakan karakter '*' untuk menampilkan garis-garis yang digambar di *canvas* ke layar. Program dapat menyimpan *canvas* ke dalam sebuah *file* (format *file* dan metode penyimpanan bebas) dan dapat membuka kembali *file canvas* tersebut.

Buatlah program tersebut dengan menu berikut ini:

1. **View Canvas** (2 poin): Menampilkan isi *canvas* sekarang
2. **Draw Horizontal Line** (4 poin): *Input* dari *user* adalah *X*, *Y*, dan *L*. *X* dan *Y* adalah koordinat di mana garis horizontal dimulai, dan *L* adalah panjang garis tersebut.
3. **Draw Vertical Line** (4 poin): *Input* dari *user* adalah *X*, *Y*, dan *L*. *X* dan *Y* adalah koordinat di mana garis vertikal dimulai, dan *L* adalah tinggi garis tersebut.
4. **Erase Rectangle Area** (4 poin): Menghapus area segi-empat pada *canvas*. *Input* dari *user* adalah *X*, *Y*, *W*, dan *H*. Area yang dihapus dimulai dari koordinat (*X*, *Y*) dan memiliki lebar sebesar *W* dan tinggi sebesar *H*.
5. **Save To File** (8 poin): Menyimpan *canvas* ke sebuah *file*. Nama *file* yang digunakan untuk menyimpan *canvas* merupakan *input user*.



6. Load From File (8 poin): Membuka kembali *canvas* yang sudah disimpan ke sebuah *file*. Nama file yang dibuka merupakan *input user*.

Contoh:

Pilihan Menu: 2

X: 5

Y: 7

L: 20

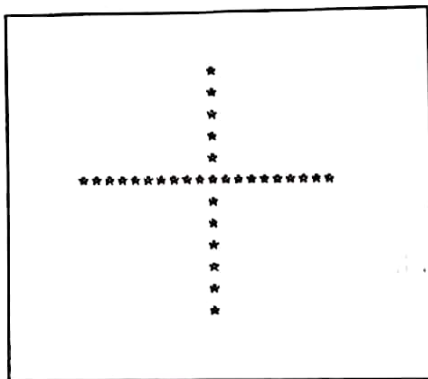
Pilihan Menu: 3

X: 15

Y: 2

L: 12

Pilihan Menu: 1



Pilihan Menu: 4

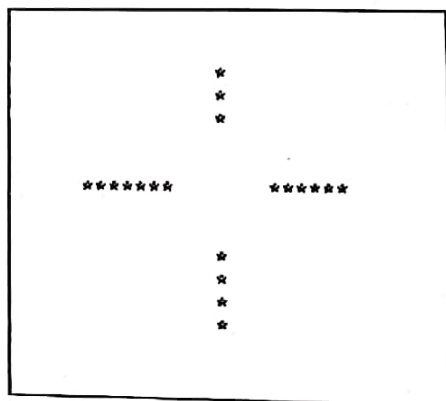
X: 12

W: 7

Y: 5

H: 5

Pilihan Menu: 1



~~ SELAMAT MENGERJAKAN DAN TUHAN MEMBERKATI ~~