

# TUGAS PRAKTIKUM

## MODUL VIII

### FUNGSI DAN PROSEDUR

Versi A.1

Prepared by: Gusti Arif H.P. & Juliver Napitupulu / 27 Februari 2016

#### Problem 1 : Mengurutkan n buah bilangan

#### *Definisi Masalah*

Buatlah sebuah program yang mengurutkan sejumlah n bilangan dan menggunakan fungsi atau prosedur. Pengguna akan memberikan jumlah bilangan (n) yang akan dimasukkan dan nilai n diasumsikan maksimal 1000000. Input pengguna berupa bilangan bulat. Fungsi atau prosedur yang dibuat akan memproses penyusunan urutan bilangan dari terbesar ke terkecil. Output dari program ini adalah hasil pengurutan bilangan.

#### Contoh Input dan Output

Input ke STDIN

```
4
3
2
1
4
```

Output ke STDOUT

```
Masukkan banyaknya bilangan (n) :
Masukkan bilangan ke - 1 :
Masukkan bilangan ke - 2 :
Masukkan bilangan ke - 3 :
Masukkan bilangan ke - 4 :
Hasil pengurutan bilangan :
4
3
2
1
```

Input ke STDIN

```
8
10
7
4
2
8
2
5
9
```

Output ke STDOUT

```

Masukkan banyaknya bilangan (n) :
Masukkan bilangan ke - 1 :
Masukkan bilangan ke - 2 :
Masukkan bilangan ke - 3 :
Masukkan bilangan ke - 4 :
Masukkan bilangan ke - 5 :
Masukkan bilangan ke - 6 :
Masukkan bilangan ke - 7 :
Masukkan bilangan ke - 8 :
Hasil pengurutan bilangan (terbesar ke terkecil) :
10
9
8
7
5
4
2
2

```

## Deliverable

Simpan program utama problem 1 dengan nama `problem1.c`. Jangan lupa memberikan identitas (*header file*) di awal file ini.

## Problem 2 : Kalkulator Berat Badan

### Definisi Masalah

Buatlah sebuah program yang dapat melakukan indentifikasi kondisi berat badan seseorang dengan menggunakan input berupa tinggi dan berat dari user. Rumus yang digunakan dalam melakukan kalkulasi adalah :

$$\text{Body Mass Index} = \text{Body Weight} \div (\text{Body Height} \times \text{Body Height})$$

Kalkulasi harus dibuat dalam bentuk fungsi yang nantinya akan dipanggil pada program utama. Cocokkan hasil BMI tersebut pada tabel dibawah ini :

BMI Score	Identification
< 18	Under Weight
18-25	Normal Weight

25-27	Over Weight
27 >	Obese

Diasumsikan seluruh input yang diberikan oleh user adalah benar dan nilai BMI yang ditampilkan memiliki ketelitian 2 angka di belakang koma.

### Contoh Input dan Output

Input ke STDIN

```
1.50
75
```

Output ke STDOUT

```
Kalkulator Berat Badan Ideal BMI
Masukkan tinggi badan anda ( Meter ):
Masukkan berat badan anda ( Kilogram):
BMI = 33.33, Obese
```

Input ke STDIN

```
1.75
50
```

Output ke STDOUT

```
Kalkulator Berat Badan Ideal BMI
Masukkan tinggi badan anda ( Meter ):
Masukkan berat badan anda ( Kilogram):
BMI = 16.33, Under Weight
```

Input ke STDIN

```
1.75
78
```

Output ke STDOUT

```
Kalkulator Berat Badan Ideal BMI
Masukkan tinggi badan anda ( Meter ):
Masukkan berat badan anda ( Kilogram):
BMI = 25.47, Over Weight
```

Input ke STDIN

```
1.75
75
```



Output ke STDOUT

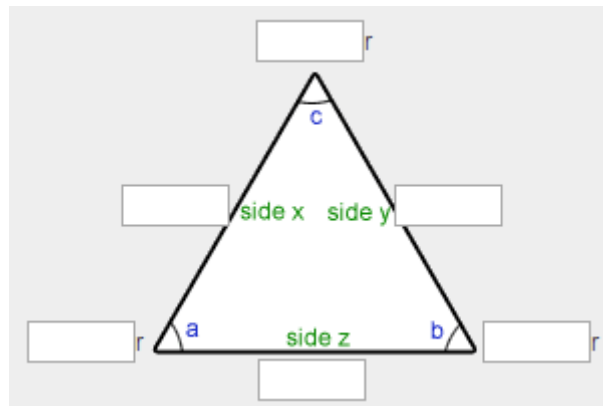
```
Kalkulator Berat Badan Ideal BMI
Masukkan tinggi badan anda ( Meter ):
Masukkan berat badan anda ( Kilogram):
BMI = 24.49, Normal Weight
```

## Deliverable

Simpan program utama problem 1 dengan nama problem2.c. Jangan lupa memberikan identitas (*header file*) di awal file ini.

## Problem 3 : Menghitung nilai setiap sudut dan luas dari sebuah segitiga sebarang

### Definisi Masalah



Gambar 2.1 Tampilan sebuah segitiga

Buatlah sebuah program yang menghitung nilai setiap sudut dan luas dari sebuah segitiga sebarang. Pengguna akan memberikan nilai sisi x, y, dan z (dalam cm) dan buatlah sebuah prosedur untuk menampung nilai sisi ini. Input pengguna berupa bilangan bulat. Fungsi dipergunakan dalam program ini untuk memroses perhitungan nilai sudut dan luas. Output dari program ini adalah nilai sudut a, b, dan c (dalam derajat) serta luas segitiga (dalam cm<sup>2</sup>) dengan ketelitian 3 angka di belakang koma.

### Contoh Input dan Output

Input ke STDIN

```
20
30
40
```

Output ke STDOUT

```
Masukkan sisi x :
Masukkan sisi y :
Masukkan sisi z :
```

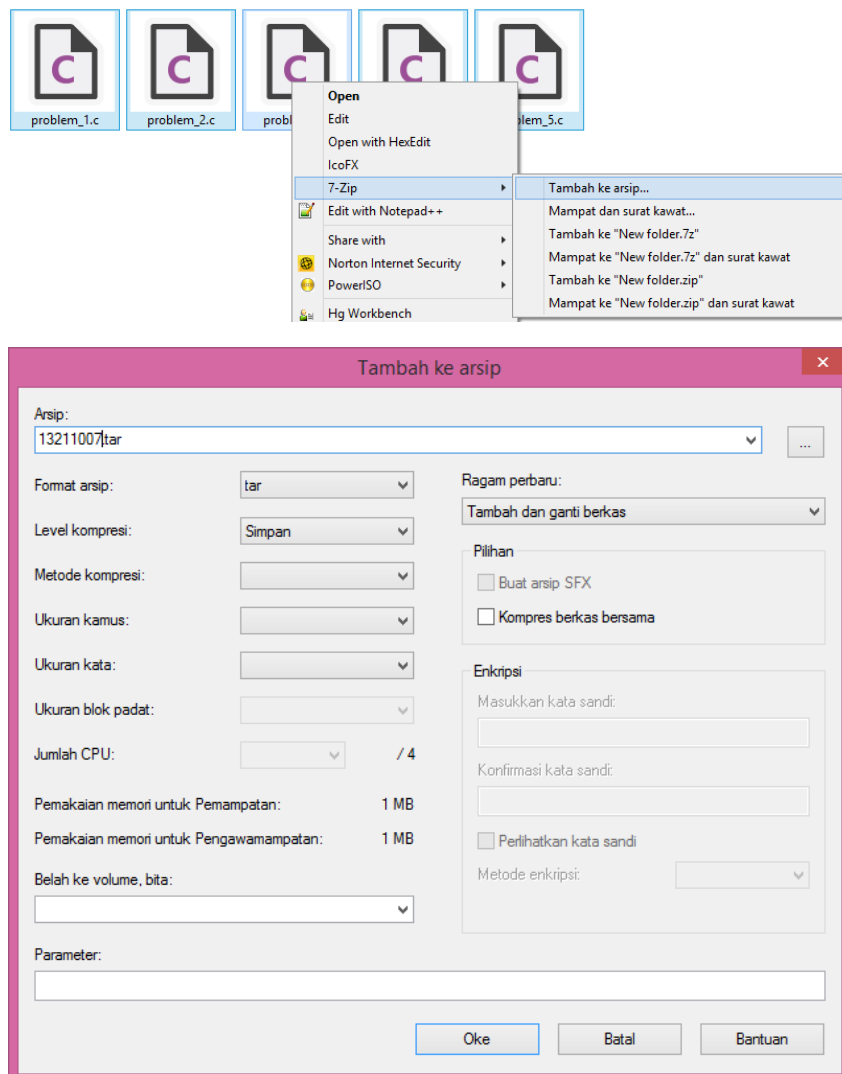
Nilai sudut a : 46.567 derajat  
Nilai sudut b : 29.955 derajat  
Nilai sudut c : 104.478 derajat  
Luas segitiga : 290.474 cm<sup>2</sup>

## Deliverable

Simpan program utama problem 2 dengan nama problem3.c. Jangan lupa memberikan identitas (*header file*) di awal file ini.

### Petunjuk Penyerahan Tugas Praktikum Modul VIII

Simpan file problem1.c, problem2.c, dan problem3.c dalam satu folder. Gunakan program 7-zip untuk mengkompresi menjadi arsip TAR (.tar). Penamaan file TAR bebas (disarankan menggunakan NIM). File TAR ini yang akan di-submit ke server MIKU saat pengumpulan tugas pendahuluan saat memasuki laboratorium. Hanya file kode saja yang dimasukkan ke dalam arsip TAR. File *executable* tidak perlu dimasukkan.



Selesai