

TUGAS PENDAHULUAN

MODUL VII

POINTER DAN EXTERNAL FILE

Versi A.1

Prepared by: Alvin Lianto dan Agung Nuza / 10 Februari 2016

Problem 1 : Tulis file external

Definisi Masalah

Pada *problem* ke-1 ini, akan dilakukan penulisan suatu variabel yang sudah didefinisikan ke dalam file eksternal.

Dalam bahasa C terdapat *library* standar yang menyediakan interaksi dengan STDIN dan STDOUT. *Library* ini bernama `stdio.h` dan harus di-*include* dalam program bahasa C kita. Beberapa fungsi yang tersedia dalam *library* ini adalah sebagai berikut.

Function	Purpose
Formatted Input/Output Functions	
fprintf	Formatted File Write
fscanf	Formatted File Read
printf	Formatted Write
scanf	Formatted Read
sprintf	Formatted String Write
sscanf	Formatted String Read
vfprintf	Formatted File Write Using Variable Argument List
vprintf	Formatted Write Using Variable Argument List
vsprintf	Formatted String Write Using Variable Argument List
File Operation Functions	
fclose	Close File
fflush	Flush File Buffer
fopen	Open File
freopen	Reopen File
remove	Remove File
rename	Rename File
setbuf	Set Buffer (obsolete)
setvbuf	Set Buffer
tmpfile	Create Temporary File
tmpnam	Generate Temporary File Name

Pada bagian ini, kita akan mencoba membuat aplikasi sangat sederhana yaitu menulis *hello world* ke dalam file eksternal. Aplikasi ini cukup mencetak ke file “**output.txt**” sebuah kalimat “Hello World!”. Berikut ini adalah **contoh** kode aplikasi tersebut. Lengkapi kode tersebut dengan identitas (*header file*) yang sesuai (lihat petunjuk teknis praktikum). Kompilasi kode tersebut dengan GCC lalu jalankan dan lihat file keluarannya



```
#include <stdio.h>

int main (void)
{
    // Deklarasi variabel
    // pointer untuk akses ke
    file FILE *fileTarget;

    // buka file target dengan permission write
    (w) fileTarget = fopen("output.txt", "w");

    // tulis isi ke file
    fprintf(fileTarget, "Hello World!\n");

    return (0);
}
```

Tugas anda adalah : mengubah kode di atas untuk menulis

Hello praktikum !

Deliverable

Simpan tugas *problem* ke-1 dengan nama **problem1.c**. Jangan lupa memberikan identitas (*header file*) di awal file ini. Pastikan program dapat dikompilasi dan dijalankan dengan benar.

Problem 2 : Menggunakan Pointer

Pengantar

Pada *problem* ke-2 ini, kita akan Menggunakan pointer dalam bahasa pemrograman C. Pointer adalah jenis variabel yang digunakan untuk menyimpan alamat dari suatu variabel lainnya. Lengkapi kode tersebut dengan identitas (*header file*) yang sesuai (lihat petunjuk teknis praktikum).

Berikut adalah **contoh** program :

```
#include <stdio.h>

int main (void)
{
    // Deklarasi
    Variabel int *
    pointerToVar;
    int normalVariable, normalVariable2;

    // Algoritma
```



```

normalVariable = 10;
pointerToVar =
&normalVariable;

printf("Isi variabel biasa   : %d\n", normalVariable);
printf("Alamat variabel     : %p\n", &normalVariable);
printf("Isi variabel pointer : %p\n", pointerToVar);
printf("Isi alamat dirujuk pointer : %d\n\n", *pointerToVar);

// ubah isi variabel biasa
normalVariable = 20;
printf("Isi variabel biasa   : %d\n", normalVariable);
printf("Alamat variabel     : %p\n", &normalVariable);
printf("Isi variabel pointer : %p\n", pointerToVar);
printf("Isi alamat dirujuk pointer : %d\n\n", *pointerToVar);

// ubah isi variabel biasa melalui pointer
(* pointerToVar) = 15;
printf("Isi variabel biasa   : %d\n", normalVariable);
printf("Alamat variabel     : %p\n", &normalVariable);
printf("Isi variabel pointer : %p\n", pointerToVar);
printf("Isi alamat dirujuk pointer : %d\n\n", *pointerToVar);

// Pindahkan pointer ke alamat variabel baru
normalVariable2 = 100;
pointerToVar = &normalVariable2;
printf("Isi variabel biasa   : %d\n", normalVariable);
printf("Alamat variabel     : %p\n", &normalVariable);
printf("Isi variabel biasa 2 : %d\n", normalVariable2);
printf("Alamat variabel 2   : %p\n", &normalVariable2);
printf("Isi variabel pointer : %p\n", pointerToVar);
printf("Isi alamat dirujuk pointer : %d\n\n", *pointerToVar);

return 0;
}

```

Kompilasi kode tersebut dengan GCC lalu jalankan dan lihat hasilnya. Program di atas adalah referensi mengenai bagaimana pointer bekerja.

Definisi Masalah

Buatlah sebuah program untuk **menukar** dua buah variabel yang dimasukkan, dengan menggunakan **prosedur** saja (bukan fungsi). **Prosedur** yang dibuat **hanya memiliki 2 buah variabel proses** yaitu alamat dari bilangan A dan alamat dari bilangan B.



Input ke STDIN

```
10
2
```

Output ke STDOUT

```
Masukkan bilangan A :
Masukkan bilangan B :
2 10
```

Input ke STDIN

```
100
20
```

Output ke STDOUT

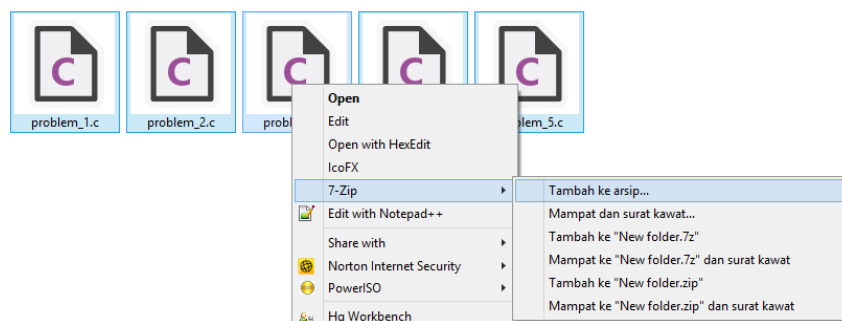
```
Masukkan bilangan A :
Masukkan bilangan B :
20 100
```

Deliverable

Simpan tugas *problem* ke-2 dengan nama `problem2.c`. Jangan lupa memberikan identitas (*header file*) di awal file ini. Pastikan program dapat dikompilasi dan dijalankan dengan benar.

Petunjuk Penyerahan Tugas Pendahuluan Modul VII

Simpan file dengan nama `problem1.c`, `problem2`. Gunakan program 7-zip untuk mengkompresi seluruh file `.c` menjadi arsip TAR (`.tar`). Penamaan file TAR bebas (disarankan menggunakan NIM). File TAR ini yang akan di-submit ke server MIKU.. Hanya file kode saja yang dimasukkan ke dalam arsip TAR. File *executable* tidak perlu dimasukkan.



Tambah ke arsip

Arsip: 13211007[tar] ...

Format arsip: tar

Level kompresi: Simpan

Metode kompresi:

Ukuran kamus:

Ukuran kata:

Ukuran blok padat:

Jumlah CPU: / 4

Pemakaian memori untuk Pemampatan: 1 MB

Pemakaian memori untuk Pengawamampatan: 1 MB

Belah ke volume, bita:

Parameter:

Ragam perbaru: Tambah dan ganti berkas

Pilihan

☐ Buat arsip SFX

☐ Kompres berkas bersama

Enkripsi

Masukkan kata sandi:

Konfirmasi kata sandi:

☐ Perlihatkan kata sandi

Metode enkripsi:

Oke Batal Bantuan

Selesai

