

TUGAS PENDAHULUAN

MODUL IV

PROCEDURE, FUNCTION, POINTER & FILE EKSTERNAL

Problem 1 : Menuliskan Barisan Bilangan Fibonacci

Definisi Masalah

Dalam matematika, bilangan Fibonacci adalah barisan yang dapat didefinisikan secara rekursif. Barisan ini didapatkan dengan menjumlahkan kedua bilangan berurutan sebelumnya. Dengan aturan ini, barisan bilangan Fibonacci pertama adalah sebagai berikut

0,1,1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, ...

Buatlah sebuah program yang akan meminta *input* sebuah bilangan integer yang menunjukkan jumlah bilangan Fibonacci yang akan ditampilkan dan akan menampilkan barisan bilangan tersebut pada *console*. Untuk menghasilkan barisan tersebut, buatlah **sebuah prosedur** yang dapat melakukan kalkulasi hingga menghasilkan barisan seperti di atas.

Kompilasi kode tersebut dengan GCC lalu jalankan dan lihat hasilnya. Untuk menjalankan *executable file*, gunakan *command prompt* pada Windows lalu berpindah ke direktori tempat *executable file* berada. Kemudian, tulis nama *executable file* tersebut lalu tekan Enter.

Contoh Input dan Output

Contoh 1

Input ke STDIN

7

Output ke STDOUT

```
Masukkan nilai n yang diinginkan =  
Barisan bilangan Fibonacci adalah =  
0  1  1  2  3  5  8
```

Contoh 2

Input ke STDIN

-5

Output ke STDOUT

```
Masukkan nilai n yang diinginkan =  Barisan bilangan  
Fibonacci adalah =          Nilai n harus lebih besar sama  
dengan 1!!!
```

Simpan tugas dengan nama `problem1.c`. Jangan lupa memberikan identitas (*header file*) di awal file ini. Pastikan program dapat dikompilasi dan dijalankan dengan benar. **Karena hanya terdapat satu problem, silakan unggah satu file kode C saja (`problem.c`) tanpa perlu melakukan kompresi ke TAR.**

Problem 2 : Menggunakan Pointer

Pengantar

Pada *problem* ke-2 ini, kita akan Menggunakan pointer dalam bahasa pemrograman C. 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("Isi alamat variabel : %p\n", &normalVariable);
    printf("Isi variabel pointer : %d\n", *pointerToVar);
    printf("Isi alamat pointer : %p\n\n", pointerToVar);

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

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

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

Pointer dapat mereferensikan alamat langsung dari suatu variabel, sehingga memungkinkan bagi suatu fungsi / prosedur untuk mengubah nilai variabel secara langsung.

Buatlah sebuah program untuk menukar dua buah variabel yang dimasukkan, dengan menggunakan **prosedur** saja (bukan fungsi).

Template untuk output :

```
printf("Masukkan bilangan A :\n");  
printf("Masukkan bilangan B :\n");  
  
printf("%d %d\n", bilangan_a, bilangan_b);
```

Contoh Input dan Output

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
```

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 VI

Simpan kedua file 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 melalui server KULI **ITERA**. Hanya file kode saja yang dimasukkan ke dalam arsip TAR. File *executable* tidak perlu dimasu



