

**Topik:** Fungsi dan Prosedur pada ADT

**Tujuan Praktikum:**

Peserta praktikum memahami bahasa C secara lebih mendalam sebagai persiapan untuk membuat ADT (Abstract Data Type)

**Petunjuk Praktikum:**

1. Untuk setiap *file* yang anda buat, buatlah *header* sebagai berikut:

{ Nama/NIM	:	}
{ Tanggal	:	}
{ Topik	:	}
{ Deskripsi	:	}

2. Nama file hari ini adalah **P01-<NIM>-<DDMMYY>-<AA>-[main].<c|h>** dengan keterangan:

DD : tanggal pelaksanaan praktikum hari ini

MM : bulan pelaksanaan praktikum hari ini

YY : tahun pelaksanaan praktikum hari ini

AA : nomor soal (dalam angka)

main : khusus *file driver*

Contoh:

P00-18217001-190916-01.c

P00-18217001-190916-01.h

P00-18217001-190916-01-main.c

3. Seluruh *file* (.c dan/atau .h, tanpa *file* dengan ekstensi lain) dikompres menjadi satu file dengan penamaan **P01-<NIM>-<DDMMYY>.zip**
4. Ikuti petunjuk asisten dalam meng-*upload* pekerjaan Anda.
5. **Hanya file yang lulus *compile* yang akan dinilai.** File yang tidak lulus *compile* akan mendapatkan nilai 0.
6. Tugas ini bersifat **INDIVIDUAL**. Segala bentuk kecurangan apapun tidak akan ditoleransi dan akan diberikan nilai 0.

**SELAMAT BEKERJA**

### SOAL 1: *Abstract Data Type* (ADT) Point

Pada praktikum ini, buatlah 3 buah file yang berisi *file header*, *file body (implementasi file header)*, dan *program utama (driver)*.

Buatlah prosedur dan fungsi ADT Point sesuai dengan file header (Point.h) dibawah ini.

```
#ifndef point_H
#define point_H

/* Definisi ABSTRACT DATA TYPE POINT */
typedef struct{
    int x;
    int y;
} Point;

Point makePoint(int x, int y);
/* Membentuk sebuah Point dari komponen-komponennya */

int getAbsis(Point P);
/*Mengirimkan komponen Absis dari P*/

int getOrdinat (Point P);
/* Mengirimkan komponen Ordinat dari P*/

void bacaPoint(Point *P);
/*Makepoint (x,y,P) membentuk P dari x dan y yang dibaca */

void tulisPoint(Point P);
/* Nilai P ditulis ke layar dg format "(X,Y)" */

bool isEqual (Point P1, Point P2);
//Mengirimkan true jika P1=P2 : absis dan ordinatnya sama

bool isLessThan(Point P1, Point P2);
/* Mengirimkan true jika P1 < P2. Definisi lebih kecil: lebih
"kiribawah"
dalam bidang kartesian */

bool isGreaterThan(Point P1, Point P2);
/* Mengirimkan true jika P1 > P2. Definisi lebih besar: lebih
"kananatas"
dalam bidang kartesian */

int kuadran(Point P);
/* Menghasilkan kuadran dari P: 1,2,3, atau 4
Prekondisi : P bukan Titik Origin,
dan P tidak terletak di salah satu sumbu*/

Point tambah(Point P1, Point P2);
//Menghasilkan operasi penjumlahan vector P1+P2

Point kurang(Point P1, Point P2);
//Menghasilkan operasi pengurangan vector P1-P2

void geserPointKeSbX (Point *P);
```

```
//Geser point ke sumbu x dengan nilai x tetap  
  
void geserPointKeSbY (Point *P);  
//Geser point ke sumbu y dengan nilai y tetap  
  
#endif
```

Dengan mengimplementasikan semua fungsi diatas ke dalam program, buatlah sebuah program utama (*driver*) sederhana menggunakan *file header* yang telah Anda buat. Berikut ini adalah tampilan layar dari *driver* program tersebut

### Program Point

POINT 1:

x : **1**

y : **2**

POINT 1 adalah (1,2)

POINT 2:

x : **3**

y : **4**

POINT 2 adalah (3,4)

Kuadran POINT 1: Point 1 berada di kuadran 1

Kuadran POINT 2: Point 2 berada di kuadran 1

Perbandingan POINT 1 dan POINT 2 adalah sebagai berikut.

-Point 1 tidak sama dengan Point 2

-Point 1 lebih kecil dari Point 2

$\text{POINT 1} + \text{POINT 2} = (1,2) + (3,4) = (4,6)$

$\text{POINT 1} - \text{POINT 2} = (1,2) - (3,4) = (-2,-2)$

POINT 1 digeser sejajar sumbu X.

POINT 1 menjadi (1,0)

POINT 2 digeser sejajar sumbu Y.

POINT 2 menjadi (0,4)

POINT 3:

x : **-1**

y : **2**

POINT 3 adalah (-1,2)

Kuadran POINT 3: Point 3 berada di kuadran 2

POINT 4 adalah POINT hasil penjumlahan POINT 3 dan POINT 2.

POINT 4 adalah (-1,6)

Keterangan :

- Angka yang berwarna merah merupakan masukan dari user.
- Yang dikumpulkan harus 2 file yaitu ***driver*** (main.c) dan ***file implementasi header*** (point.c)

## SOAL 2: Abstract Data Type Garis (BONUS)

**Catatan:** Soal ini hanya akan diperiksa jika soal 1 sudah diselesaikan.

Buatlah file implementasi dari *file header* ADT garis dibawah ini. Buatlah *driver* sendiri untuk melakukan *testing* terhadap fungsi/prosedur yang ada. Kumpulkan hanya file **garis.c**.

```
typedef int bool;
#define true 1
#define false 0

#ifndef GARIS_H
#define GARIS_H
#include "Point.h"

//Definisi ABSTRACT DATA TYPE GARIS
typedef struct {
    Point P1; //titik awal
    Point P2; //titik akhir
}Garis;

Garis makeGARIS (Point P1, Point P2);
//Membentuk sebuah GARIS dari komponen-komponennya

void bacaGARIS (Garis *G);
//I.S: sembarang
//F.S mengulangi membaca dua buah nilai P1 dan P2 sehingga membentuk
//garis yang valid
//makeGARIS (P1,P2) membentuk G dari dua titik yang dibaca

void tulisGARIS (Garis G);
//Nilai G ditulis ke layar dengan format ((x1,y1),(x2,y2))

Point getPAw (Garis G);
//Mengirimkan komponen titik awal dari garis

Point getPAk (Garis G);
//Mengirimkan komponen titik akhir dari garis

void geser (Garis *G, int deltaX, int deltaY):
//I.S: G terdefinisi
//F.S: absis G digeser sejauh deltaX dan ordinatnya sejauh deltaY

#endif
```