

Modul PDP-08

TITIK

Dalam Tugas ini anda harus menyelesaikan dengan standar 3 file (drvttitik.c, titik.h, titik.c dan boolean.h). Masalah yang harus anda selesaikan adalah mengenai pengolahan suatu titik dalam koordinat kartesius (seperti dalam matematika smp/sma). Semua fungsi dan prosedur (titik.h dan titik.c) yang anda buat harus di test dalam driver (drvttitik.c). Dibawah ini seluruh **header fungsi titik.h** dan spesifikasi dari masalah yang harus anda selesaikan.

Keterangan :

I.S. = Initial State (keadaan awal)

F.S. = Final State (keadaan akhir)

/*

File driverttk.c

***/**

#include<stdio.h>

#include<stdlib.h>

#include<ctype.h>

#include<limits.h>

#include "boolean.h"

/*Variabel Global untuk absis dan ordinat*/

float X; // Ordinat

float Y; // Absis

/*Konstruktor*/

void MakePoint(float, float);

/*Selektor*/

float GetAbsis();

float GetOrdinat();

/*Kolektor*/

void SetOrdinat(float *Ordinat, float newx);

void SetAbsis(float *Absis, float newy);

/*Baca Tulis*/

void BacaPoint();

void Tulis(float, float);

/*Operasi aritmatika*/

void AddP(float x1, float y1, float x2, float y2);

void MinP(float x1, float y1, float x2, float y2);

void MulDot(float x1, float y1, float x2, float y2);

/*Posisi*/

boolean IsOrigin();

boolean IsOnSbX();

boolean IsOnSbY();

int Kuadran();

/*Operasi Tipe*/

void NextX();

void NextY();

```

void PlusDelta(float deltax,float deltay);
void MirrorOf(boolean sbx);
float JarakPst();
void Geser(float deltax,float deltay);
void GeserSbX();
void GeserSbY();
void MirrorP(boolean sbx);
void Putar(float sudut);

int main()
{
    MakePoint(2.0,2.0);
    Tulis(X,Y);
    BacaPoint();
    Tulis(X,Y);
    printf("  GetAbsis          : %d\n", (int)GetAbsis());
    printf("  GetOrdinat         : %d\n", (int)GetOrdinat());
    SetAbsis(&X,4.0); SetOrdinat(&Y,4.0);
    printf("  P4 stlh diset 4,4   : ");Tulis(X,Y);

    MakePoint(-2,2);
    printf("  P5 stlh diset -2,2 : ");Tulis(X,Y);

    AddP(2.0,2.0,-2.0,2.0);
    printf("  Tulis P5 =P1+P2     : ");Tulis(X,Y);
    MinP(2.0,2.0,-2.0,2.0);
    printf("  Tulis P5 =P1-P2     : ");Tulis(X,Y);
    MulDot(2.0,2.0,-2.0,2.0);
    printf("  Tulis P5 =P1*P2     : ");Tulis(X,Y);

    printf("  Origin              ? %d\n",IsOrigin());
    printf("  IsOnSbX              ? %d\n",IsOnSbX());
    printf("  IsOnSbY              ? %d\n",IsOnSbY());
    printf("  X,Y di Kuadran Ke    : %d\n",Kuadran());
    NextX();
    printf("  Next X 1 titik      : ");Tulis(X,Y);
    NextY();
    printf("  Next Y 1 titik      : ");Tulis(X,Y);
    PlusDelta(2,2);
    printf("  PlusDelta (2,2) : ");Tulis(X,Y);
    MirrorOf(true);
    printf("  Miror ke SbX        : ");Tulis(X,Y);
    Geser(2,3);
    printf("  Geser ke (2,3)      : ");Tulis(X,Y);
    GeserSbX();
    printf("  Geser ke SbX        : ");Tulis(X,Y);
    GeserSbY();
    printf("  Geser ke SbY        : ");Tulis(X,Y);
}

```

```

    printf(" Jarak dr (0,0) : %.2f\n",JarakPst());

    printf(" Titik sbkm rot 30 CCW : ");Tulis(X,Y);
    Putar(30.0);
    printf(" Titik stlh rot 30 CCW : ");Tulis(X,Y);
    Putar(-30.0);
    printf("Pl stlh rot -30 CCW : ");Tulis(X,Y);

    printf("Hello world!\n");
    return 0;
}
/*Fungsi Konstruktor*/
Procedure MakePoint(Input float,float);
/* MakePoint(a,b) membentuk sebuah titik dari a dan b dengan a
sebagai absis dan b sebagai ordinat
*/
/*Selektor*/
Function GetAbsis()→Real;
/*Fungsi untuk mengambil nilai absis*/
/*mengembalikan nilai absis dari variable global*/
Function GetOrdinat()→Real;
/*Fungsi untuk mengambil nilai ordinat*/
/*mengembalikan nilai ordinat dari variable global*/
/*Fungsi Kolektor*/
Procedure SetOrdinat(Input/Output Ordinat:Real, Input newx:Real);
/*Fungsi untuk menyetel nilai Ordinat suatu titik*/
/*di variable Global*/
Procedure SetAbsis(Input/Output Absis:Real,Input newy:Real);
/*Fungsi untuk menyetel nilai Absis suatu titik*/
/*di variable Global*/
/*Fungsi Baca Tulis*/
Procedure BacaPoint();
/*Fungsi untuk membaca nilai suatu titik dari keyboard*/
/*baik nilai absis atau ordinat, perlu variable local
dalam prosedur untuk di assign ke dalam variable global
Procedure Tulis(Input float,float);
/*
{ Nilai variable global x dan y ditulis ke layar dengan format
"(X,Y)" }
{ I.S. <x,y> terdefinisi }
{ F.S. <x,y> tertulis di layer dengan format "(X,Y)" }
*/
/*Operasi aritmatika*/
Function AddP(Input x1,y1,x2,y2:Real)→Real;
/*
{ Menghasilkan titik bernilai <x1,y1> +- <x2,y2>2 }
{ Buatlah spesifikasi penambahan dua buah titik }
*/

```

```

Function MinP(Input x1,y1,x2,y2:Real)→Real;
/*
{ Menghasilkan titik bernilai  $\langle x1,y1 \rangle - \langle x2,y2 \rangle^2$  }
{ Buatlah spesifikasi pengurangan dua buah titik }
*/

Function MulDot(Input x1,y1,x2,y2:Real)→Real;
/*
{ Operasi perkalian  $\langle x1,y1 \rangle * \langle x2,y2 \rangle$  : melakukan operasi dot product}
{ *** Kelompok operasi relasional terhadap titik *** }
*/
/*Posisi*/

Function IsOrigin(Input x,y:Real)→Boolean;
/*
IsOrigin(x,y) benar jika  $\langle x,y \rangle$  adalah titik origin yaitu titik  $\langle 0,0 \rangle$ 
*/

Function IsOnSbX()→Boolean;
/*Menghasilkan true jika  $\langle x,y \rangle$  terletak pada sumbu X*/

Function IsOnSbY()→Boolean;
/*Menghasilkan true jika  $\langle x,y \rangle$  terletak pada sumbu Y*/

Function Kuadran()→Integer;
/*
{ Menghasilkan kuadran dari  $\langle x,y \rangle$ : 1, 2, 3, atau 4 }
{ Prekondisi :  $\langle x,y \rangle$  bukan Titik Origin, }
{ dan  $\langle x,y \rangle$  tidak terletak di salah satu sumbu }
*/

/*Operasi Tipe*/

Function NextX()→Real;
/*
Mengirim salinan  $\langle x,y \rangle$  dengan absis ditambah satu
*/

Function NextY()→Real;
/*
Mengirim salinan  $\langle x,y \rangle$  dengan ordinat ditambah satu
*/

Procedure PlusDelta(Input deltax,deltay:Real);
/*
Menghasilkan  $\langle x,y \rangle$  yang absisnya adalah Absis(x) + deltax dan ordinatnya adalah Ordinat(y) + deltay
*/

Function MirrorOf(Input sbx:Boolean)→Real;
/*
Menghasilkan  $\langle x,y \rangle$  yg dicerminkan dg nilai SbX dan SbY
Jika SbX bernilai true, maka dicerminkan terhadap sumbu X
Jika SbY bernilai true, maka dicerminkan terhadap sumbu Y
*/

```

```

Function JarakPst()→Real;
/*
Menghitung jarak <x,y> ke (0,0)
*/
Procedure Geser(Input deltax,deltay:Real) ;
/*
{ I.S. x dan y sudah terdefinisi }
{ F.S. x dan y digeser sebesar deltaX & ordinatnya sebesar
deltaY }
*/
Procedure GeserSbX() ;
/*
{ I.S. x dan y sudah terdefinisi }
{ F.S. x dan y di Sumbu X dengan ordinat yang sama dgn semula. }
{ Proses : x dan y digeser ke sumbu X. }
*/
Procedure GeserSbY() ;
/*
{ I.S. x dan y sudah terdefinisi }
{ F.S. x dan y di Sumbu Y dengan ordinat yang sama dgn semula. }
{ Proses : x dan y digeser ke sumbu Y. }
*/
Procedure MirrorP(Input sbx:Boolean) ;
/*
{ I.S. x dan y sudah terdefinisi }
{ F.S. x dan y dicerminkan tergantung nilai SbX atau SbY }
{ Jika SbX true maka dicerminkan terhadap sumbu X }
{ Jika SbY true maka dicerminkan terhadap sumbu Y }
*/
Procedure Putar(Input sudut:Real) ;
/*
{ I.S. x dan y sudah terdefinisi }
{ F.S. x dan y diputar sebesar sudut derajat }
*/

```