

Dalam Tugas ini anda harus menyelesaikan dengan standar 3 file (driverttk.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 (driverttk.c). Dibawah ini seluruh header fungsi dan spesifikasi dari masalah yang harus anda selesaikan.

Keterangan :

I.S. = Initial State (keadaan awal)

F.S. = Final State (keadaan akhir)

*/*Variabel Global untuk absis dan ordinat*/*

```
float X; // Ordinat
float Y; // Absis
```

*/*Fungsi Konstruktor*/*

void MakePoint(float a,float b);

/ MakePoint(a,b) membentuk sebuah titik dari a dan b dengan a sebagai absis dan b sebagai ordinat
/

*/*Selektor*/*

float GetAbsis();

*/*Fungsi untuk mengambil nilai absis*/
/*mengembalikan nilai absis dari variable global*/*

float GetOrdinat();

*/*Fungsi untuk mengambil nilai ordinat*/
/*mengembalikan nilai ordinat dari variable global*/*

*/*Fungsi Kolektor*/*

void SetOrdinat(float *Ordinat,float newx);

*/*Fungsi untuk menyetel nilai Ordinat suatu titik*/
/*di variable Global*/*

void SetAbsis(float *Absis,float newy);

*/*Fungsi untuk menyetel nilai Absis suatu titik*/
/*di variable Global*/*

*/*Fungsi Baca Tulis*/*

void 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
/

```
void Tulis();
/*
{ 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*/
float AddP(float x1, float y1, float x2, float y2);
/*
{ Menghasilkan titik bernilai <x1,y1> +- <x2,y2>2 }
{ Buatlah spesifikasi penambahan dua buah titik }
*/

float MinP(float x1, float y1, float x2, float y2);
/*
{ Menghasilkan titik bernilai <x1,y1> - <x2,y2>2 }
{ Buatlah spesifikasi pengurangan dua buah titik }
*/

float MulCros(float x1, float y1, float x2, float y2);
/*
{ Operasi perkalian <x1,y1>*<x2,y2> : melakukan operasi cross
product}
{ *** Kelompok operasi relasional terhadap titik *** }
*/

/*pembanding*/
boolean EqP(float x1, float y1, float x2, float y2);
/*Mengirimkan true jika <x1,y1> sama dengan <x2,y2>
*/
boolean NEqP(float x1, float y1, float x2, float y2);
/*Mengirimkan true jika <x1,y1>tidak sama dengan <x2,y2>
*/
boolean LTP(float x1, float y1, float x2, float y2);
/*
Mengirimkan true jika <x1,y1> < <x2,y2>. Definisi lebih besar:
lebih "kiri-bawah dalam bidang kartesian.
*/
boolean GTP(float x1, float y1, float x2, float y2);
/*
Mengirimkan true jika <x1,y1> > <x2,y2>. Definisi lebih besar:
lebih "kanan-atas" dalam bidang kartesian.
*/
```

```

/*Posisi*/
boolean IsOrigin(float x, float y);
/*
  IsOrigin(x,y) benar jika <x,y> adalah titik origin yaitu titik
  <0,0>
*/
boolean IsOnSbX(float x, float y);
/*Menghasilkan true jika <x,y> terletak pada sumbu X*/

boolean IsOnSbY(float x, float y);
/*Menghasilkan true jika <x,y> terletak pada sumbu Y*/

int Kuadran(float x, float y);
/*
  { Menghasilkan kuadran dari <x,y>: 1, 2, 3, atau 4 }
  { Prekondisi : <x,y> bukan Titik Origin, }
  { dan <x,y> tidak terletak di salah satu sumbu }
*/
float Koor2Point(float x1, float y1, float x2, float y2);
/*Koord titik diantara 2 titik lainnya
  di ket P1(x1,y1) dan P3(x2,y2), P2 terletak di antaranya.
  Rumus :
  P2(xp2,yp2)
  tentukan m dan n dengan
  Y
  ^
  |           + P3
  |           n/|
  |   P2 +   |
  |       m/| | |
  |       + | |
  |   P1 | | |
  +---+---+----->X
      p1 p2p3
  m:n = p1p2:p2p3
  m:n = p2x-p1x:p3x-p2x
  P1(1,3) P3(-2,-5) ==> m:n=3-(-5):1-(-2)==>8:3
  m:n=8:3
  xp2=(m.x2+n.x1)/m+n
  yp2=(m.y2+n.y1)/m+n
*/

/*Operasi Tipe*/
float NextX(float x1, float y1 );
/*
  Mengirim salinan <x,y> dengan absis ditambah satu
*/

```

```

float NextY(float x, float y );
/*
Mengirim salinan <x,y> dengan ordinat ditambah satu
*/
void PlusDelta(float x, float y,float deltax,float deltay);
/*
Menghasilkan <x,y> yang absisnya adalah Absis(x) + deltaX dan
ordinatnya adalah Ordinat(y) + deltaY
*/
float MirrorOf(float x, float y, boolean sbx, boolean sby);
/*
Menghasilkan <x,y> 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
*/
float JarakPst(float x, float y);
/*
Menghitung jarak <x,y> ke (0,0)
*/
float Panjang(float x1, float y1, float x2, float y2);
/*
{ Menghitung panjang garis yang dibentuk <x1,y1> dan <x2,y2> }
*/
void Geser(float *x, float *y,float deltax,float deltay);
/*
{ I.S. x dan y sudah terdefinisi }
{ F.S. x dan y digeser sebesar deltaX & ordinatnya sebesar
deltaY }
*/
void GeserSbX(float *x, float *y);
/*
{ 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. }
*/

void GeserSbY(float *x , float *y );
/*
{ 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. }
*/
void MirrorP(float *x1, float *y1, boolean sbx, boolean sby);
/*
{ 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 }
*/
void Putar(float *x, float *y,float sudut);
/*
{ I.S. x dan y sudah terdefinisi }
{ F.S. x dan y diputar sebesar sudut derajat }
*/
```