

**LAPORAN PRAKTIKUM  
STRUKTUR DATA**

**MODUL 2  
ADT LINE**

**DISUSUN OLEH :  
AJI KARTIKO HARTANTO-2350081062**



**PROGRAM STUDI INFORMATIKA  
FAKULTAS SAINS DAN INFORMATIKA  
UNIVERSITAS JENDERAL ACHMAD YANI  
TAHUN 2023**



# DAFTAR ISI

|                                     |    |
|-------------------------------------|----|
| DAFTAR GAMBAR .....                 | ii |
| BAB I. HASIL PRAKTIKUM.....         | 1  |
| I.1 Program garis.c.....            | 1  |
| I.1.A. Boolean.h .....              | 1  |
| I.1.B. Source Code garis.c .....    | 1  |
| I.1.C. Hasil .....                  | 6  |
| I.1.D. Analisa .....                | 6  |
| BAB II. TUGAS PRAKTIKUM .....       | 7  |
| II.1 Tugas ADT Line.....            | 7  |
| II.1.A. Source Code Boolean.h ..... | 7  |
| II.1.B. Source Code line.c.....     | 7  |
| II.1.C. Source Code line.h .....    | 11 |
| II.1.D. Source Code mline.c .....   | 13 |
| II.1.E. Source Code point.c .....   | 17 |
| II.1.F. Source Code point.h.....    | 22 |
| II.1.G. Hasil .....                 | 24 |
| II.1.H. Analisa.....                | 25 |
| BAB III. KESIMPULAN .....           | 26 |

## **DAFTAR GAMBAR**

|   |    |
|---|----|
| Gambar I.1 Output Program garis.c .....                               | 6  |
| Gambar II.1 Output Program Tugas ADT line (mline.c/main driver) ..... | 24 |

# BAB I. HASIL PRAKTIKUM

## I.1 Program garis.c

### I.1.A. Boolean.h

```
/*  
    Program      : boolean.h  
    Deskripsi    : header file boolean  
*/  
  
#ifndef boolean_H  
#define boolean_H  
#define true 1  
#define false 0  
#define boolean unsigned char  
#endif
```

### I.1.B. Source Code garis.c

```
/*  
    Program      : garis.c  
    Author       : 2350081062, Aji Kartiko Hartanto  
    Kelas       : C  
    Deskripsi    : main driver  
    Tanggal      : 20-03-2024  
*/  
  
#include <stdio.h>  
#include <conio.h>  
#include <math.h>  
  
//type  
typedef struct {  
    int abs;  
    int ord;  
} point;  
  
typedef struct {  
    point Pawal;
```

```

        point Pakhir;
    } garis;

//prototype
void CreateGaris(garis *G);
void SetGaris(garis *G, point P1, point P2);
void CetakGaris(garis G);

void MoveGaris(garis *G, int d);
float GradienGaris(garis G);
garis CopyGarisNeg(garis G);
point getPointGaris(garis G);
boolean Is2GarisTgkLurus(garis G1, garis G2);

//realisasi prototype
void CreateGaris(garis *G) {
    (*G).Pawal.abs = 0;
    (*G).Pawal.ord = 0;
    (*G).Pakhir.abs = 0;
    (*G).Pakhir.ord = 0;
}

void SetGaris(garis *G, point P1, point P2) {
    //algoritma
    (*G).Pawal = P1;
    (*G).Pakhir = P2;
}

void CetakGaris(garis G) {
    //kamus
    int x1, y1, x2, y2;

    //algoritma
    x1 = G.Pawal.abs;
    y1 = G.Pawal.ord;
    x2 = G.Pakhir.abs;
    y2 = G.Pakhir.ord;

    printf("G[P1(%d, %d), P2(%d, %d)]", x1, y1, x2, y2);
}

```

```

}

void MoveGaris(garis *G, int d) {
    //kamus
    int x1, y1, x2, y2;

    //algoritma
    x1 = (*G).Pawal.abs + d;
    y1 = (*G).Pawal.ord + d;
    x2 = (*G).Pakhir.abs + d;
    y2 = (*G).Pakhir.ord + d;

    printf("garis pindah sebanyak %d jadi P1(%d, %d), P2(%d, %d)", d,
x1, y1, x2, y2);
}

float GradienGaris(garis G) {
    //kamus
    int x1, y1, x2, y2;
    float gradien;

    //algoritma
    x1 = G.Pawal.abs;
    y1 = G.Pawal.ord;
    x2 = G.Pakhir.abs;
    y2 = G.Pakhir.ord;

    gradien = (float) (y2 - y1) / (x2 - x1);

    return gradien;
}

garis CopyGarisNeg(garis G){
    //kamus
    garis garis1;

    //algoritma
    garis1.Pawal.abs = G.Pawal.abs * -1;
    garis1.Pawal.ord = G.Pawal.ord * -1;
    garis1.Pakhir.abs = G.Pakhir.abs * -1;
    garis1.Pakhir.ord = G.Pakhir.ord * -1;
}

```

```

        return garis1;
    }

point getPointGaris(garis G){
    return G.Pawal;
}

boolean Is2GarisTgkLurus(garis G1, garis G2){
    float TgkLurus;
    TgkLurus = GradienGaris(G1) * GradienGaris(G2);
    if (TgkLurus == -1)
    {
        return true;
    }
    else
    {
        return false;
    }
}

//main driver
int main() {
    //kamus
    point P1, P2, Poingaris;
    garis MyGaris, MyGaris1;
    int x1, y1, x2, y2;
    int titikbaru;
    float gradien;

    //algoritma
    CreateGaris(&MyGaris);
    printf("x1:"); scanf("%d,", &x1);
    printf("y1:"); scanf("%d,", &y1);
    printf("x2:"); scanf("%d,", &x2);
    printf("y2:"); scanf("%d,", &y2);
    P1.abs = x1;
    P1.ord = y1;

```



```

P2.abs = x2;
P2.ord = y2;

SetGaris(&MyGaris, P1, P2);
CetakGaris(MyGaris);
printf("\n");

printf("masukkan titik baru: "); scanf("%d", &titikbaru);
MoveGaris(&MyGaris, titikbaru);
printf("\n");

gradien = GradienGaris(MyGaris);
printf("cetak gradien: %2f \n", gradien);

MyGaris1 = CopyGarisNeg(MyGaris);
CetakGaris(MyGaris1);
printf("\n");

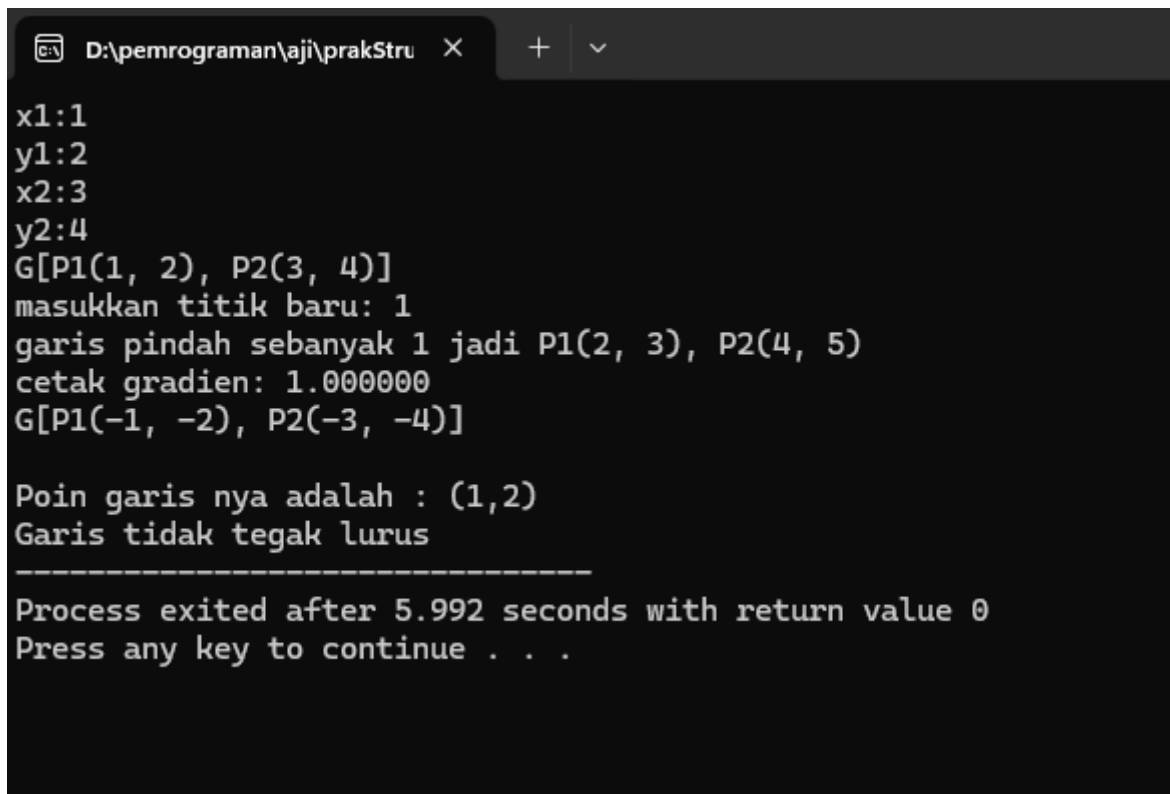
Poingaris = getPointGaris(MyGaris);
printf("\nPoin garis nya adalah : (%d,%d)", Poingaris.abs,
Poingaris.ord);

if (Is2GarisTgkLurus(MyGaris, MyGaris1))
{
    printf("\nGaris tegak lurus");
}
else
{
    printf("\nGaris tidak tegak lurus");
}

return 0;
}

```

### I.1.C. Hasil



```
D:\pemrograman\aji\prakStru x + v
x1:1
y1:2
x2:3
y2:4
G[P1(1, 2), P2(3, 4)]
masukkan titik baru: 1
garis pindah sebanyak 1 jadi P1(2, 3), P2(4, 5)
cetak gradien: 1.000000
G[P1(-1, -2), P2(-3, -4)]

Poin garis nya adalah : (1,2)
Garis tidak tegak lurus
-----
Process exited after 5.992 seconds with return value 0
Press any key to continue . . .
```

Gambar I.1 Output Program garis.c

### I.1.D. Analisa

Program ini dibuat dalam Bahasa C menggunakan IDE dev c++, Program terdiri dari dua bagian utama: definisi tipe data dan fungsi-fungsi untuk operasi-operasi pada garis, serta bagian main() sebagai titik awal eksekusi program. Dalam definisi tipe data, digunakan dua struktur: titik yang mewakili titik dengan atribut abs dan ord, dan garis dengan dua titik sebagai ujungnya.

Ada beberapa program yang akan dijelaskan seperti pada program CopyGarisNeg, program ini adalah program yang memberikan nilai baru dan dari nilai awal menjadi negasi dari nilai awal. Selanjutnya ada getpawalgaris(ditulis dalam program nya getPointGaris), program ini adalah program yang dibuat untuk mengetahui nilai point awal dari garis.

Terakhir ada program Boolean Is2GarisTgkLurus, program ini menyatakan nilai dari 2 garis apakah nilai tersebut tegak lurus atau tidak tegak lurus.

## BAB II. TUGAS PRAKTIKUM

### II.1 Tugas ADT Line

#### II.1.A. Source Code Boolean.h

```
/*
    Program      : boolean.h
    Deskripsi    : Header dari file boolean
*/
#ifndef boolean_H
#define true 1
#define false 0
#define boolean unsigned char
#endif
```

#### II.1.B. Source Code line.c

```
/*
    Program      : line.c
    Author       : 2350081062, Aji Kartiko Hartanto
    Kelas       : C
    Deskripsi    : program line.c merupakan program yang menuliskan semua
operasi fungsi dari ADT
    Tanggal     : 21-03-2024
*/
#include <stdio.h>
#include "line.h"

void CreateLine(LINE *L)
```

```

{
    CreatePoint(&(*L).PAwal);
    CreatePoint(&(*L).PAkhir);
}

void CreateLine2(LINE *L, POINT First, POINT End)
{
    (*L).PAwal = First;
    (*L).PAkhir = End;
}

POINT GetPAwal(LINE L)
{
    return L.PAwal;
}

POINT GetPAkhir(LINE L)
{
    return L.PAkhir;
}

void SetPAwal(LINE *L, POINT NewFirstP)
{
    (*L).PAwal = NewFirstP;
}

void setPAkhir(LINE *L, POINT NewEndP)
{
    (*L).PAkhir = NewEndP;
}

void BacaLine(LINE *L)
{
    printf("Point Awal:\n ");
    BacaPoint(&(*L).PAwal);
    printf("\nPoint Akhir:\n ");
    BacaPoint(&(*L).PAkhir);
}

void CetakLine(LINE L)
{

```

```

    printf("Garis dari (%d,%d) ke (%d,%d)\n", L.PAwal.X, L.PAwal.Y,
L.PAkhir.X, L.PAkhir.Y);
}

boolean IsEQLine(LINE L1, LINE L2)
{
    if      (EQ(GetPAwal(L1),      GetPAwal(L2))      &&      EQ(GetPAkhir(L1),
GetPAkhir(L2)))
    {
        return true;
    }else
    {
        return false;
    }
}

boolean IsNEQLine(LINE L1, LINE L2)
{
    if      (NEQ(GetPAwal(L1),      GetPAwal(L2))      &&      NEQ(GetPAkhir(L1),
GetPAkhir(L2)))
    {
        return true;
        /* code */
    }else
    {
        return false;
    }
}

boolean IsLOnSbX(LINE L)
{
    if (IsOnSbX(GetPAwal(L)) && IsOnSbX(GetPAkhir(L)))
    {
        return true;
    }else
    {
        return false;
    }
}

```

```

    }
}

boolean IsLOnSbY(LINE L)
{
    if (IsOnSbY(GetPAwal(L)) && IsOnSbY(GetPAkhir(L)))
    {
        return true;
    }
    else
    {
        return false;
    }
}

boolean IsTegakLurus(LINE L1, LINE L2)
{
    int M1, M2, result;

    M1 = (GetPAkhir(L2).Y - GetPAwal(L1).Y) / (GetPAkhir(L2).X -
GetPAwal(L1).X);
    M2 = (GetPAkhir(L2).Y - GetPAwal(L1).Y) / (GetPAkhir(L2).X -
GetPAwal(L1).X);

    result = M1 * M2;

    if (result == -1)
    {
        return true;
    }
    else
    {
        return false;
    }
}

boolean IsSejajar(LINE L1, LINE L2)
{
    if (Gradien(L1) == Gradien(L2))
    {
        return true;
    }
}

```

```

    }else
    {
        return false;
    }
}

LINE GeserLine(LINE L, int deltaX, int deltaY)
{
    LINE newL;

    newL.PAwal.X = L.PAwal.X + deltaX;
    newL.PAwal.Y = L.PAwal.Y + deltaY;
    newL.PAakhir.X = L.PAakhir.X + deltaX;
    newL.PAakhir.Y = L.PAakhir.Y + deltaY;

    return newL;
}

double Gradien(LINE L)
{
    return (double)(L.PAakhir.Y - L.PAwal.Y) / (double)(L.PAakhir.X -
L.PAwal.X);
}

```

### II.1.C. Source Code line.h

```

/*
    Program      : line.h
    Author       : 2350081062, Aji Kartiko Hartanto
    Kelas        : C
    Deskripsi    : header file dari adt line
    Tanggal      : 21-03-2024
*/

#ifndef LINE_H
#define LINE_H

#include <stdio.h>
#include <stdlib.h>
#include "boolean.h"
#include "point.h"

```

```

#define FirsPoint(L) (L).PAwal
#define EndPoint(L) (L).PAkhir

typedef struct
{
    POINT PAwal;
    POINT PAkhir;
} LINE;

/*konstruktor*/
void CreateLine(LINE *L);

void CreateLine2(LINE *L, POINT First, POINT End);
/*selektor*/
POINT GetPAwal(LINE L);

POINT GetPAkhir(LINE L);
/*set nilai*/
void SetPAwal(LINE *L, POINT NewFirstP);

void setPAkhir(LINE *L, POINT NewEndP);
/*baca dan tulis*/
void BacaLine(LINE *L);

void CetakLine(LINE L);
/*operator relasional*/
boolean IsEQLine(LINE L1, LINE L2);

boolean IsNEQLine(LINE L1, LINE L2);

boolean IsLOnSbX(LINE L);

boolean IsLOnSbY(LINE L);

boolean IsTegakLurus(LINE L1, LINE L2);

boolean IsSejajar(LINE L1, LINE L2);
/*operasi aritmatika*/
LINE GeserLine(LINE L, int deltaX, int deltaY);
/*operasi aritmatika*/

```



```
double Gradien(LINE L);  
  
#endif
```

#### II.1.D. Source Code mline.c

```
/*  
    Program      : mline.c  
    Author       : 2350081062, Aji Kartiko Hartanto  
    Kelas        : C  
    Deskripsi    : Mline.c merupakan program yang menjalankan Semua Prosedur  
dan function  
    Tanggal      : 22-03-2024  
*/  
#include <stdio.h>  
#include "line.c"  
#include "point.c"  
  
int main()  
{  
    LINE L1, L2, L3, L4;  
    int deltaX, deltaY;  
  
    BacaLine(&L1);  
    printf("Line1 = ");  
    CetakLine(L1);  
  
    CreateLine(&L2);  
    BacaLine(&L2);  
  
    printf("Line2 = ");  
    CetakLine(L2);  
  
    printf("\nGaris Line 1 akan digeser sejauh: \n");  
    printf("delta X: ");
```

```

scanf("%d", &deltaX);

printf("delta Y: ");
scanf("%d", &deltaY);

L3 = GeserLine(L1, deltaX, deltaY);
printf("Line 3 = ");
CetakLine(L3);

L4 = L1;
printf("Line 4 = ");
CetakLine(L4);

SetAbsis(&L2.PAwal, 0);
SetOrdinat(&L2.PAwal, 0);

printf("\nPengecekan operator relasional\n");
if (IsEQLine(L1, L2))
{
    printf("Nilai Line 1 sama dengan Line 2\n");
}

if (IsNEQLine(L1, L2))
{
    printf("Nilai Line 1 tidak sama dengan Line 2\n");
}

if (IsEQLine(L3, L4))
{
    printf("Nilai Line 3 sama dengan Line 4\n");
}

if (IsNEQLine(L3, L4))
{
    printf("Nilai Line 3 tidak sama dengan Line 4\n");
}

printf("\nPengecekan garis apakah tegak lurus\n");
if (IsTegakLurus(L1, L2))
{
    printf("Garis Line 1 Tegak lurus dengan Line 2\n");
}

```

```

}
else
{
    printf("Garis Line 1 Tidak Tegak lurus dengan Line 2\n");
}

if (IsTegakLurus(L2, L3))
{
    printf("Garis Line 2 Tegak lurus dengan Line 3\n");
}
else
{
    printf("Garis Line 2 Tidak Tegak lurus dengan Line 3\n");
}

if (IsTegakLurus(L3, L4))
{
    printf("Garis Line 3 Tegak lurus dengan Line 4\n");
}
else
{
    printf("Garis Line 3 Tidak Tegak lurus dengan Line 4\n");
}

if (IsTegakLurus(L4, L1))
{
    printf("Garis Line 4 Tegak lurus dengan Line 1\n");
}
else
{
    printf("Garis Line 4 Tidak Tegak lurus dengan Line 1\n");
}

printf("\nPengecekan garis apakah sejajar\n");
if (IsSejajar(L1, L2))
{
    printf("Garis L1 sejajar dengan L2\n");
}
else
{
    printf("\nGaris Line 1 tidak sejajar dengan Line 2\n");
}

```

```

}

if (IsSejajar(L1, L3))
{
    printf("Garis Line 1 sejajar dengan Line 3\n");
}
else
{
    printf("Garis Line 1 tidak sejajar dengan Line 3\n");
}

if (IsSejajar(L1, L4))
{
    printf("Garis Line 1 sejajar dengan Line 4\n");
}
else
{
    printf("Garis Line 1 tidak sejajar dengan Line 4\n");
}

printf("\nPengecekan operator posisi\n");
if (IsLOnSbX(L1))
{
    printf("Garis Line 1 berada pada sumbu X\n");
}

if (IsLOnSbX(L2))
{
    printf("Garis Line 2 berada pada sumbu X\n");
}

if (IsLOnSbX(L3))
{
    printf("Garis Line 3 berada pada sumbu X\n");
}

if (IsLOnSbX(L4))
{
    printf("Garis Line 4 berada pada sumbu X\n");
}

```

```

    if (IsLOnSbY(L1))
    {
        printf("Garis Line 1 berada pada sumbu Y\n");
    }

    if (IsLOnSbY(L2))
    {
        printf("Garis Line 2 berada pada sumbu Y\n");
    }

    if (IsLOnSbY(L3))
    {
        printf("Garis Line 3 berada pada sumbu Y\n");
    }

    if (IsLOnSbY(L4))
    {
        printf("Garis Line 4 berada pada sumbu Y\n");
    }

    return 0;
}

```

### II.1.E. Source Code point.c

```

/*
    program          : point.c
    author           : 2350081062, Aji Kartiko Hartanto
    kelas            : C
    deskripsi        : header file dari prototype point
    tanggal           : 21-03-2024

*/

#include "point.h"
#include <stdio.h>
#include <conio.h>
#include <math.h>

/*konstruktor POINT*/
void CreatePoint(POINT *P)

```

```

{
    (*P).X = 0;
    (*P).Y = 0;
}

void CreatePoint2 (POINT *P, int XBaru, int YBaru)
{
    (*P).X = XBaru;
    (*P).Y = YBaru;
}

/*selektor POINT*/
int GetAbsis (POINT P)
{
    return P.X;
}

int GetOrdinat (POINT P)
{
    return P.Y;
}

/*set nilai POINT*/
void SetAbsis (POINT *P, int XBaru)
{
    (*P).X = XBaru;
}

void SetOrdinat (POINT *P, int YBaru)
{
    (*P).Y = YBaru;
}

/*kelompok interaksi dengan I/O device, Baca/Tulis*/
void BacaPoint (POINT *P)
{
    int X, Y;

    printf("\nMasukan nilai Absis X :");
    scanf("%d", &X);
    printf("Masukan nilai Ordinat Y :");
    scanf("%d", &Y);
    CreatePoint2 (&(*P), X, Y);
}

```

```

void CetakPoint (POINT *P)
{
    printf("\nNilai Point adalah (%d,%d)", (*P).X, (*P).Y);
}

/*kelompok operasi relasional terhadap POINT*/
boolean EQ (POINT P1, POINT P2)
{
    if (GetAbsis(P1) == GetAbsis(P2) && GetOrdinat(P1) == GetOrdinat(P2))
    {
        return true;
    }
    else
    {
        return false;
    }
}
boolean NEQ (POINT P1, POINT P2)
{
    if (GetAbsis(P1) == GetAbsis(P2) && GetOrdinat(P1) == GetOrdinat(P2))
    {
        return false;
    }
    else
    {
        return true;
    }
}
boolean LT (POINT P1, POINT P2)
{
    if (GetAbsis(P1) < GetAbsis(P2) && GetOrdinat(P1) < GetOrdinat(P2))
    {
        return true;
    }
    else
    {
        return false;
    }
}
boolean MT (POINT P1, POINT P2)
{

```

```

        if (GetAbsis(P1) > GetAbsis(P2) && GetOrdinat(P1) > GetOrdinat(P2))
        {
            return true;
        }
        else
        {
            return false;
        }
    }

/*kelompok menentukan dimana P berada*/
boolean IsOrigin(POINT P)
{
    if (GetAbsis(P) == 0 && GetOrdinat(P) == 0)
    {
        return true;
    }
    else
    {
        return false;
    }
}

boolean IsOnSbX(POINT P)
{
    if (GetOrdinat(P) == 0)
    {
        return true;
    }
    else
    {
        return false;
    }
}

boolean IsOnSbY(POINT P)
{
    if (GetAbsis(P) == 0)
    {
        return true;
    }
    else
    {

```



```

        return false;
    }
}
int Kuadran(POINT P)
{
    if (GetAbsis(P) > 0 && GetOrdinat(P) > 0)
    {
        return 1;
    }
    else if (GetAbsis(P) < 0 && GetOrdinat(P) > 0)
    {
        return 2;
    }
    else if (GetAbsis(P) < 0 && GetOrdinat(P) < 0)
    {
        return 3;
    }
    else if (GetAbsis(P) > 0 && GetOrdinat(P) < 0)
    {
        return 4;
    }
    else
    {
        return 0;
    }
}

/*kelompok operasi lain terhadap type*/
POINT MirrorOf(POINT P, boolean SbX, boolean SbY)
{
    if (SbX == true)
    {
        P.X = GetAbsis(P) * -1;
    }
    else if (SbY == true)
    {
        P.Y = GetOrdinat(P) * -1;
    }

    return P;
}

```

```

float jarakO(POINT P)
{
    float Jarak;

    Jarak = sqrt(pow((GetAbsis(P) - 0), 2) + pow((GetOrdinat(P) - 0),
2));

    return Jarak;
}
void GeserKeSbX(POINT *P)
{
    if (GetOrdinat((*P)) != 0)
    {
        SetOrdinat(&(*P), 0);
    }
}
void GeserKeSbY(POINT *P)
{
    if (GetAbsis((*P)) != 0)
    {
        SetAbsis(&(*P), 0);
    }
}

```

### II.1.F. Source Code point.h

```

/*
    program          : point.h
    author           : 2350081062, Aji Kartiko Hartanto
    kelas            : C
    deskripsi        : header file dari prototype point
    tanggal           : 21-03-2024

*/
#ifndef _POINT_H
#define _POINT_H
#include "boolean.h"
#include <stdio.h>
#include <conio.h>
#define absis(p) (p).X
#define ordinat(p) (p).Y

```

```

#define PI 3.14159265

/*definisi type data POINT*/
typedef struct
{
    int X;
    int Y;
} POINT;

/*konstruktor POINT*/
void CreatePoint(POINT *P);
void CreatePoint2(POINT *P, int XBaru, int YBaru);

/*selektor POINT*/
int GetAbsis(POINT P);
int GetOrdinat(POINT P);

/*set nilai POINT*/
void SetAbsis(POINT *P, int XBaru);
void SetOrdinat(POINT *P, int YBaru);

/*kelompok interaksi dengan I/O device, Baca/Tulis*/
void BacaPoint(POINT *P);
void CetakPoint(POINT *P);

/*kelompok operasi relasional terhadap POINT*/
boolean EQ(POINT P1, POINT P2);
boolean NEQ(POINT P1, POINT P2);
boolean LT(POINT P1, POINT P2);
boolean MT(POINT P1, POINT P2);

/*kelompok menentukan dimana P berada*/
boolean IsOrigin(POINT P);
boolean IsOnSbX(POINT P);
boolean IsOnSbY(POINT P);
int Kuadran(POINT P);

/*kelompok operasi lain terhadap type*/
POINT MirrorOf(POINT P, boolean SbX, boolean SbY);
float jarakO(POINT P);
void GeserKeSbX(POINT *P);

```

```
void GeserKeSbY (POINT *P);  
  
#endif
```

## II.1.G. Hasil

Gambar II.1 Output Program Tugas ADT line (mline.c/main driver)

### **II.1.H. Analisa**

Program yang dibuat adalah program yang mengoperasikan objek-objek line dan dapat melakukan berbagai operasi pada garis, seperti membaca input pengguna, menggeser garis, mengecek hubungan, mengecek kedudukan, dan mencetak hasilnya. Program terstruktur dengan baik untuk melakukan operasi khusus pada garis, yang memudahkan pemeliharaan dan pengembangan lebih lanjut lagi.

Objek garis dan nilai pergeseran disimpan dalam variabel. Fungsi seperti BacaLine(), CreateLine(), dan GeserLine, antara lain, memungkinkan program untuk memiliki modularitas dengan membagi tugas-tugas tertentu ke dalam fungsi-fungsi yang berbeda. Selain itu, program menggunakan struktur kontrol seperti pernyataan kondisional (if) dan pengulangan (for atau while) untuk mengatur alur eksekusi program sesuai dengan logika operasi-operasi yang dilakukan pada garis. Program menjadi lebih mudah dimengerti, diperbaiki, dan dikembangkan untuk memenuhi kebutuhan yang berbeda berkat struktur yang terorganisir dengan baik ini.

### **BAB III. KESIMPULAN**

Kesimpulan yang diambil dari praktikum modul 2 ini adalah Kedua program ini menunjukkan bagaimana tipe data, fungsi, dan prosedur dapat digunakan untuk merepresentasikan dan melakukan operasi pada objek geometri sederhana seperti garis dalam bahasa pemrograman C. Pendekatan moduler yang digunakan dalam kedua program memungkinkan untuk memisahkan logika operasi pada garis dari logika program utama, yang meningkatkan keterbacaan dan kemudahan pemeliharaan. Oleh karena itu, program-program ini memberikan fondasi yang kuat untuk pengembangan lebih lanjut dari aplikasi geometri dan perangkat lunak yang lebih kompleks.