

**LAPORAN PRAKTIKUM  
STRUKTUR DATA**

**MODUL 2  
ADT LINE**

**DISUSUN OLEH :**

Ammar Bagas Fathurrahman Wantoro      2350081008



**PROGRAM STUDI INFORMATIKA  
FAKULTAS SAINS DAN INFORMATIKA  
UNIVERSITAS JENDERAL ACHMAD YANI  
TAHUN 2024**

# DAFTAR ISI

BAB I.	HASIL PRAKTIKUM.....	1
I.1	Line.h .....	1
I.1.A.	Source Code.....	1
I.1.B.	Hasil .....	3
I.1.C.	Analisis .....	3
BAB II.	TUGAS PRAKTIKUM .....	4
II.1	ADT LINE.....	4
II.1.A.	Point.h.....	4
II.1.A.1	Source code.....	4
II.1.A.2	Hasil.....	6
II.1.A.3	Analisis .....	6
II.1.B.	Point.c .....	7
II.1.B.1	Source code.....	7
II.1.B.2	Hasil.....	12
II.1.B.3	Analisis .....	12
II.1.C.	Line.c .....	12
II.1.C.1	Source code.....	12
II.1.C.2	Hasil.....	18
II.1.C.3	Analisis .....	18
II.1.D.	Boolean.h .....	18
II.1.D.1	Source code.....	18
II.1.D.2	Hasil.....	19
II.1.D.3	Analisis .....	19

II.1.E. mline.c .....	19
II.1.E.1 Source code.....	19
II.1.E.2 Hasil.....	25
II.1.E.3 Analisis .....	26
BAB III. KESIMPULAN .....	27

## DAFTAR GAMBAR

Gambar 1. 1 Hasil line.h.....	3
Gambar 2. 1 Hasil point.c.....	6
Gambar 2. 2 Hasil Point.c.....	12
Gambar 2. 3 Hasil Point.c.....	12
Gambar 2. 4 Hasil line.c .....	18
Gambar 2. 5 Hasil line.c .....	18
Gambar 2. 6 Hasil boolean.h .....	19
Gambar 2. 7 Hasil ADT LINE pada main driver.....	25
Gambar 2. 8 Hasil ADT LINE pada main driver.....	25

# BAB I. HASIL PRAKTIKUM

## I.1 Line.h

### I.1.A. Source Code

```
/*  
  
    Program: line.h  
  
    Author: 2350081008/Ammar Bagas Fathurrahman Wantoro  
  
    Kelas: A  
  
    Deskripsi: header file dari adt line  
  
    Tanggal: 03/20/2024  
  
*/  
  
#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

```

### I.1.B. Hasil

```

1  /*
2  Program: line.h
3  Author: 2150081008/Amar Bagas Fathurrahman Wantoro
4  Kelas: A
5  Deskripsi: header file dari adt line
6  Tanggal: 03/20/2024
7  */
8
9  #ifndef LINE_H
10 #define LINE_H
11
12 #include <stdio.h>
13 #include <stdlib.h>
14 #include "boolean.h"
15 #include "point.h"
16
17 #define FirstPoint(L) (L).PAwal
18 #define EndPoint(L) (L).PAkhir
19
20 typedef struct
21 {
22     POINT PAwal;
23     POINT PAkhir;
24 } LINE;
25
26 /*konstruktor*/
27 void CreateLine(LINE *L);
28
29 void CreateLine2(LINE *L, POINT First, POINT End);
30 /*selektor*/
31 POINT GetPAwal(LINE L);
32
33 POINT GetPAkhir(LINE L);
34 /*set nilai*/
35 void SetPAwal(LINE *L, POINT NewFirstP);
36
37 void setPAkhir(LINE *L, POINT NewEndP);
38 /*opsi dan utiliti*/
39 void BacaLine(LINE *L);
40
41 void CetakLine(LINE L);
42 /*operator relational*/
43 boolean IsEqLine(LINE L1, LINE L2);
44
45 boolean IsNEqLine(LINE L1, LINE L2);
46
47 boolean IsOnSBX(LINE L);
48
49 boolean IsOnSBY(LINE L);
50
51 boolean IsTegakLurus(LINE L1, LINE L2);
52
53 boolean IsSejajar(LINE L1, LINE L2);
54 /*operasi aritmatika*/
55 LINE GeserLine(LINE L, int deltaX, int deltaY);
56 /*operasi aritmatika*/
57 double Gradien(LINE L);
58
59 #endif

```

Gambar 1. 1 Hasil line.h

### I.1.C. Analisis

Line.h merupakan sebuah program yang dibuat untuk mendeklarasikan atau membuat type data bentukan yaitu ADT LINE atau ADT GARIS. Line.h merupakan header file dari ADT LINE yang kemudian akan di operasikan pada body file dan main driver. Kemudian pada file header ini berisi type data bentukan LINE yang terbentuk dari type data bentukan POINT yang akan digunakan juga nanti. Kemudian operasi konstruktor sampai dengan operasi aritmatika di deklarasikan apda file header.

## BAB II. TUGAS PRAKTIKUM

### II.1 ADT LINE

#### II.1.A. Point.h

##### II.1.A.1 Source code

```
/*  
  
    program          : point.h  
  
    author           : 2350081008/Ammar Bagas Fathurrahman  
Wantoro  
  
    kelas            : A  
  
    deskripsi        : header file dari prototype point  
  
    tanggal           : 3/16/2024  
  
*/  
  
#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 jarak0(POINT P);


void GeserKeSbX(POINT *P);

void GeserKeSbY(POINT *P);

#endif

```

## II.1.A.2 Hasil



```

1  C point.h > NEQ(POINT, POINT)
2  /*
3   * program : point.h
4   * author  : 2350081008/Amar Bagas Fathurrahman Mantoro
5   * kelas   : A
6   * deskripsi : header file dari prototype point
7   * tanggal : 3/16/2024
8   */
9  #ifndef _POINT_H
10 #define _POINT_H
11 #include "boolean.h"
12 #include <stdio.h>
13 #include <conio.h>
14 #define absis(p) (p).X
15 #define ordinat(p) (p).Y
16 #define PI 3.14159265
17
18 /*definisi type data POINT*/
19 typedef struct
20 {
21     int X;
22     int Y;
23 } POINT;
24
25 /*konstruktor POINT*/
26 void CreatePoint(POINT *P);
27 void CreatePoint2(POINT *P, int XBaru, int YBaru);
28
29 /*selektor POINT*/
30 int GetAbsis(POINT P);
31 int GetOrdinat(POINT P);
32
33 /*set nilai POINT*/
34 void SetAbsis(POINT *P, int XBaru);
35 void SetOrdinat(POINT *P, int YBaru);
36
37 /*kelompok interaksi dengan I/O device, Baca/tulis*/
38 void BacaPoint(POINT *P);
39 void CetakPoint(POINT *P);
40
41 /*kelompok operasi relasional terhadap POINT*/
42 boolean EQ(POINT P1, POINT P2);
43 boolean NEQ(POINT P1, POINT P2);
44 boolean LT(POINT P1, POINT P2);
45 boolean MT(POINT P1, POINT P2);
46
47 /*kelompok menentukan dimana P berada*/
48 boolean IsOrigin(POINT P);
49 boolean IsOnSbX(POINT P);
50 boolean IsOnSbY(POINT P);
51 int Kuadran(POINT P);
52
53 /*kelompok operasi lain terhadap type*/
54 POINT MirrorOf(POINT P, boolean SbX, boolean SbY);
55 float jarak0(POINT P);
56 void GeserKeSbX(POINT *P);
57 void GeserKeSbY(POINT *P);
58
59 #endif
60

```

Gambar 2. 1 Hasil point.c

## II.1.A.3 Analisis

Program line.h merupakan program header yang akan digunakan pada operasi ADT LINE karena type data bentukan point merupakan pembentuk dari type data bentukan ADT LINE. Line.h berisikan operasi atau primitif yang di deklarasikan terdapat konstruktor dengan procedure CreatePoint dan CreatePoint 2, Selector dengan GetAbsis dan GetOrdinat, Set nilai ADT dengan SetAbsis dan SetOrdinat, kelompok baca dan tulisa dengan BacaPoint dan CetakPoint, kelompok operasi relasional yang semua nya adalah function EQ NEQ LT dan MT, kelompok menentukan dimana P berada dengan function IsOrigin IsOnSbY IsOnSbX Kuadran, kemudian yang terakhir adalah kelompok operasi lain terhadap POINT dengan MirrorOf Jarak0 dan GeserKeSbx GeserKeSbY.

## II.1.B. Point.c

### II.1.B.1 Source code

```
/*  
  
    program            : point.h  
  
    author             : 2350081008/Ammar Bagas Fathurrahman Wantoro  
  
    kelas              : A  
  
    deskripsi          : header file dari prototype point  
  
    tanggal             : 3/16/2024  
  
*/  
  
#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));

}

void GeserKeSbX(POINT *P){
    if (GetOrdinat((*P)) != 0) {
        SetOrdinat(&(*P), 0);
    }
}

void GeserKeSbY(POINT *P){
    if (GetAbsis((*P)) != 0) {
        SetAbsis(&(*P), 0);
    }
}
}

```

## II.1.B.2 Hasil

```
42. bool isMirorOf(PPOINT P)
43. {
44.     return false;
45. }
46. int Kuadran(PPOINT P)
47. {
48.     if (GetAbsis(P) > 0 && GetOrdinat(P) > 0)
49.     {
50.         return 1;
51.     }
52.     else if (GetAbsis(P) < 0 && GetOrdinat(P) > 0)
53.     {
54.         return 2;
55.     }
56.     else if (GetAbsis(P) < 0 && GetOrdinat(P) < 0)
57.     {
58.         return 3;
59.     }
60.     else if (GetAbsis(P) > 0 && GetOrdinat(P) < 0)
61.     {
62.         return 4;
63.     }
64.     else
65.     {
66.         return 0;
67.     }
68. }
69. //Melakukan operasi lain terhadap type*/
70. PPOINT MirorOf(PPOINT P, bool isMiror, bool isMiror)
71. {
72.     if (isMiror == true)
73.     {
74.         P.X = GetAbsis(P) * -1;
75.     }
76.     else if (isMiror == true)
77.     {
78.         P.Y = GetOrdinat(P) * -1;
79.     }
80.     return P;
81. }
82. float Jarak(PPOINT P)
83. {
84.     float Jarak;
85.     Jarak = sqrt(pow((GetAbsis(P) - 0), 2) + pow((GetOrdinat(P) - 0), 2));
86. }
87. void GeserKediri(PPOINT *P)
88. {
89.     if (GetOrdinat(*P) != 0)
90.     {
91.         SetOrdinat(&*P, 0);
92.     }
93. }
94. void GeserKediri(PPOINT *P)
95. {
96.     if (GetAbsis(*P) != 0)
97.     {
98.         SetAbsis(&*P, 0);
99.     }
100. }
```

Gambar 2. 3 Hasil Point.c

```
/*
 * Program : point.h
 * Author : 2350081008/Ammar Bagas Fathurrahman Wantoro
 * Kelas : A
 * Deskripsi : header file dari prototype point
 * Tanggal : 3/16/2024
 */
#include "point.h"
#include <stdio.h>
#include <conio.h>
#include <math.h>
//konstruktur PPOINT*/
void CreatePoint(PPOINT *P)
{
    (*P).X = 0;
    (*P).Y = 0;
}
void CreatePoint2(PPOINT *P, int XBaru, int YBaru)
{
    (*P).X = XBaru;
    (*P).Y = YBaru;
}
//dekonstruktur PPOINT*/
int GetAbsis(PPOINT P)
{
    return P.X;
}
int GetOrdinat(PPOINT P)
{
    return P.Y;
}
//set nilai PPOINT*/
void SetAbsis(PPOINT *P, int XBaru)
{
    (*P).X = XBaru;
}
void SetOrdinat(PPOINT *P, int YBaru)
{
    (*P).Y = YBaru;
}
//Melakukan interaksi dengan I/O Device, Baca/Tulis*/
void BacaPoint(PPOINT *P)
{
    int X, Y;
    printf("Masukkan nilai Absis X :");
    scanf("%d", &X);
    printf("Masukkan nilai Ordinat Y :");
    scanf("%d", &Y);
    CreatePoint(&*P, X, Y);
}
void CetakPoint(PPOINT *P)
{
    printf("Nilai Point adalah (%d,%d)", (*P).X, (*P).Y);
}
```

Gambar 2. 2 Hasil Point.c

## II.1.B.3 Analisis

Program point.c merupakan program body atau body file dari ADT POINT yang berisikan operasi – operasi yang sudah di deklarasikan pada file header. Isi dari program ini adalah operasi dari masing – masing primitive pada ADT POINT dengan begitu primitive yang sudah di deklarasikan bisa berjalan dengan baik. Program ini akan saling berhubungan dengan ADT LINE karena operasi yang ada di dalam nya di butuhkan untuk membuat sebuah garis.

## II.1.C. Line.c

### II.1.C.1 Source code

```
/*
 *
 * Program: line.c
 *
 * Author: 2350081008/Ammar Bagas Fathurrahman Wantoro
 *
 * Kelas: A
 *
 * Deskripsi: program line.c merupakan program yang menuliskan
 * semua operasi fungsi dari ADT
 *
 * Tanggal: 03/20/2024
 *
 */
```



```

#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);
}

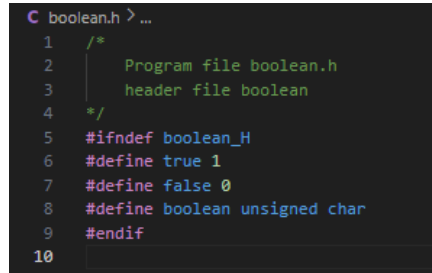
```



```
#define boolean unsigned char

#endif
```

### II.1.D.2 Hasil



```
C boolean.h > ...
1  /*
2     Program file boolean.h
3     header file boolean
4  */
5  #ifndef boolean_H
6  #define true 1
7  #define false 0
8  #define boolean unsigned char
9  #endif
10
```

*Gambar 2. 6 Hasil boolean.h*

### II.1.D.3 Analisis

Program Boolean.h merupakan program header yang mendefinisikan type data Boolean karena pada Bahasa C tidak terdapat type data Boolean. Di dalam source code dapat terlihat bahwa true di definisikan menjadi 1 dan false menjadi 0.

## II.1.E. mline.c

### II.1.E.1 Source code

```
/*
    Program: main.c
    Author: 2350081008/Ammar Bagas Fathurrahman Wantoro
    Kelas: A
    Deskripsi: Mline.c merupakan program yang menjalankan
               Semua Prosedur dan function
    Tanggal: 03/20/2024
*/

#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);

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.2 Hasil

```

PS D:\kulia\semester 2\struktur data\praktikum\p2\ADT LINE> cd "d:\kulia\semester 2\struktur data\praktikum\p2\ADT LINE\" ; if ($?) { gcc mline.c -o mline } ; if ($?) { .\mline }
Point Awal:
Masukan nilai Absis X :1
Masukan nilai Ordinat Y :6
Point Akhir:
Masukan nilai Absis X :-3
Masukan nilai Ordinat Y :2
Line1 = Garis dari (1,6) ke (-3,2)
Point Awal:
Masukan nilai Absis X :1
Masukan nilai Ordinat Y :7
Point Akhir:
Masukan nilai Absis X :3
Masukan nilai Ordinat Y :7
Line2 = Garis dari (1,7) ke (3,7)
Garis Line 1 akan digeser sejauh:
delta X: 10
delta Y: 3
Line 3 = Garis dari (11,9) ke (7,5)
Line 4 = Garis dari (1,6) ke (-3,2)
Pengecekan operator relasional
Nilai Line 1 tidak sama dengan Line 2
Nilai Line 3 tidak sama dengan Line 4
Pengecekan garis apakah tegak lurus
Garis Line 1 Tidak Tegak lurus dengan Line 2
Garis Line 2 Tidak Tegak lurus dengan Line 3
Garis Line 3 Tidak Tegak lurus dengan Line 4
Garis Line 4 Tidak Tegak lurus dengan Line 1

```

Gambar 2. 7 Hasil ADT LINE pada main driver

```

Pengecekan operator relasional
Nilai Line 1 tidak sama dengan Line 2
Nilai Line 3 tidak sama dengan Line 4

Pengecekan garis apakah tegak lurus
Garis Line 1 Tidak Tegak lurus dengan Line 2
Garis Line 2 Tidak Tegak lurus dengan Line 3
Garis Line 3 Tidak Tegak lurus dengan Line 4
Garis Line 4 Tidak Tegak lurus dengan Line 1

Pengecekan garis apakah sejajar
Garis Line 1 tidak sejajar dengan Line 2
Garis Line 1 sejajar dengan Line 3
Garis Line 1 sejajar dengan Line 4

Pengecekan operator posisi
PS D:\kulia\semester 2\struktur data\praktikum\p2\ADT LINE>

```

Gambar 2. 8 Hasil ADT LINE pada main driver

### **II.1.E.3 Analisis**

Program mline.c merupakan program main driver dari keseluruhan ADT LINE. Dapat di lihat hasil dari operasi yang di lakukan dengan pertama – tama pengguna melakukan input pada nilai point awal dan point akhir garis kemudian dilakukan operasi aritmatika yang pertama yaitu GeserLine. Dengan operasi itu pengguna dapat menggeser titik awal dan titik akhir pada sebuah garis. Kemudian operasi yang dilakukan selanjutnya adalah operasi membandingkan antara 2 buah garis apakah sama atau tidak nya antara 2 garis, lalu apakah anatar 2 garis tersebut saling tegak lurus dan apakah antara 2 garis tersebut sejajar

### **BAB III. KESIMPULAN**

Kesimpulan pada praktikum kali ini kita dapat memahami bagaimana cara membuat sebuah ADT atau type bentukan dan beberapa struktur operasi dan apa saja yang dibutuhkan untuk membuat sebuah ADT kemudian primitif – primitif apa saja yang digunakan atau dioperasikan pada main driver. Kemudian bagaimana membuat sebuah ADT yang saling terhubung antara ADT LINE dan ADT POINT yang saling terhubung karena pembentukan ADT LINE membutuhkan beberapa operasi dari ADT POINT dan juga membutuhkan type bentukan point. Karena pada pembuatan type data bentukan line dibutuhkan point awal dan point akhir yang memiliki type data point.