**LAPORAN PRAKTIKUM**

**STRUKTUR DATA**

**MODUL 2**

**ADT LINE**

**DISUSUN OLEH :**

Ammar Bagas Fathurrahman Wantoro 2350081008

Logo

Description automatically generated

**PROGRAM STUDI INFORMATIKA**

**FAKULTAS SAINS DAN INFORMATIKA**

**UNIVERSITAS JENDERAL ACHMAD YANI**

**TAHUN 2024**

DAFTAR ISI

[BAB I. HASIL PRAKTIKUM 1](#_Toc162083253)

[I.1 Line.h 1](#_Toc162083254)

[I.1.A. Source Code 1](#_Toc162083255)

[I.1.B. Hasil 3](#_Toc162083256)

[I.1.C. Analisis 3](#_Toc162083257)

[BAB II. TUGAS PRAKTIKUM 4](#_Toc162083258)

[II.1 ADT LINE 4](#_Toc162083259)

[II.1.A. Point.h 4](#_Toc162083260)

[II.1.A.1 Source code 4](#_Toc162083261)

[II.1.A.2 Hasil 6](#_Toc162083262)

[II.1.A.3 Analisis 6](#_Toc162083263)

[II.1.B. Point.c 7](#_Toc162083264)

[II.1.B.1 Source code 7](#_Toc162083265)

[II.1.B.2 Hasil 12](#_Toc162083266)

[II.1.B.3 Analisis 12](#_Toc162083267)

[II.1.C. Line.c 12](#_Toc162083268)

[II.1.C.1 Source code 12](#_Toc162083269)

[II.1.C.2 Hasil 18](#_Toc162083270)

[II.1.C.3 Analisis 18](#_Toc162083271)

[II.1.D. Boolean.h 18](#_Toc162083272)

[II.1.D.1 Source code 18](#_Toc162083273)

[II.1.D.2 Hasil 19](#_Toc162083274)

[II.1.D.3 Analisis 19](#_Toc162083275)

[II.1.E. mline.c 19](#_Toc162083276)

[II.1.E.1 Source code 19](#_Toc162083277)

[II.1.E.2 Hasil 25](#_Toc162083278)

[II.1.E.3 Analisis 26](#_Toc162083279)

[BAB III. KESIMPULAN 27](#_Toc162083280)

**DAFTRA GAMBAR**

[Gambar 1. 1 Hasil line.h 3](#_Toc162083320)

[Gambar 2. 1 Hasil point.c 6](#_Toc162083328)

[Gambar 2. 2 Hasil Point.c 12](#_Toc162083329)

[Gambar 2. 3 Hasil Point.c 12](#_Toc162083330)

[Gambar 2. 4 Hasil line.c 18](#_Toc162083331)

[Gambar 2. 5 Hasil line.c 18](#_Toc162083332)

[Gambar 2. 6 Hasil boolean.h 19](#_Toc162083333)

[Gambar 2. 7 Hasil ADT LINE pada main driver 25](#_Toc162083334)

[Gambar 2. 8 Hasil ADT LINE pada main driver 25](#_Toc162083335)

# HASIL PRAKTIKUM

## Line.h

### 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

### Hasil

Gambar 1. 1 Hasil line.h

### 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.

# TUGAS PRAKTIKUM

## ADT LINE

### Point.h

#### 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 jarakO(POINT P);

void GeserKeSbX(POINT \*P);

void GeserKeSbY(POINT \*P);

#endif

#### Hasil

Gambar 2. 1 Hasil point.c

#### 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.

### Point.c

#### 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);

}

}

#### Hasil

Gambar 2. 2 Hasil Point.c

Gambar 2. 3 Hasil Point.c

#### 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.

### Line.c

#### 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.PAkhir.X = L.PAkhir.X + deltaX;

newL.PAkhir.Y = L.PAkhir.Y + deltaY;

return newL;

}

double Gradien(LINE L)

{

return (double)(L.PAkhir.Y - L.PAwal.Y) / (double)(L.PAkhir.X - L.PAwal.X);

}

#### Hasil

Gambar 2. 4 Hasil line.c

Gambar 2. 5 Hasil line.c

#### Analisis

Line.c merupakan prorgam body file yang berisikan semua operasi dari primitif – primitif ADT LINE yang sudah di deklarasikan pada file header. Program line.c tidak akan berfungsi jika kita tidak memasukan atau mengaitkan file dari point.c ataupun point.h. Karena pada operasi pembentukan ADT LINE atau konstruktor pada ADT LINE dibutuhkan sebuha operasi pada ADT POINT yaitu dengan menggunakan sebuah operasi konstruktor untuk membuat 2 buah titik. Dari titik awal sampai dengan titik akhir. Kemudian operasi yang lain nya seperti BacaLine membutuhkan juga salah satu operasi dari ADT POINT, dan beberapa operasi lain nya pada line.c membutuhkan kaitan dari ADT POINT.

### Boolean.h

#### Source code

/\*

Program file boolean.h

header file boolean

\*/

#ifndef boolean\_H

#define true 1

#define false 0

#define boolean unsigned char

#endif

#### Hasil

Gambar 2. 6 Hasil boolean.h

#### 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.

### mline.c

#### 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");

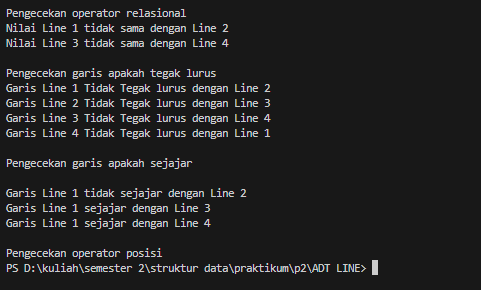
}

return 0;

}

#### Hasil

Gambar 2. 7 Hasil ADT LINE pada main driver



Gambar 2. 8 Hasil ADT LINE pada main driver

#### 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

# KESIMPULAN

Kesimpulan pada praktikum kali ini kita dapat memahami bagaiamana 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 di operasikan 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.