

**LAPORAN PRAKTIKUM
STRUKTUR DATA**

**MODUL 1
PERKENALAN ADT**

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	ADT JAM.....	1
I.1.A.	Program boolean.h.....	1
I.1.A.1	source code.....	1
I.1.A.2	Hasil.....	2
I.1.A.3	Analisa.....	2
I.1.B.	Program jam.h.....	2
I.1.B.1	Source code.....	2
I.1.B.2	Hasil.....	5
I.1.B.3	Analisa.....	5
I.1.C.	Program jam.c.....	6
I.1.C.1	Source code.....	6
I.1.C.2	Hasil.....	11
I.1.C.3	Analisa.....	12
I.1.D.	Program main.c.....	12
I.1.D.1	Source code.....	12
I.1.D.2	Hasil.....	14
I.1.D.3	Analisa.....	15
BAB II.	TUGAS PRAKTIKUM.....	16
II.1	ADT POINT.....	16
II.1.A.	Program Boolean.h.....	16
II.1.A.1	Source code.....	16
II.1.A.2	Hasil.....	16

II.1.A.3 Analisa	17
II.1.B. Program point.h.....	17
II.1.B.1 Source code.....	17
II.1.B.2 Hasil.....	19
II.1.B.3 Analisa	19
II.1.C. Program point.c	20
II.1.C.1 Source code.....	20
II.1.C.2 Hasil.....	25
II.1.C.3 Analisa	25
II.1.D. Program main.c	26
II.1.D.1 Source code.....	26
II.1.D.2 Hasil.....	29
II.1.D.3 Analisa	30
BAB III. KESIMPULAN	31

DAFTAR GAMBAR

Gambar 1. 1 source code boolean.h	2
Gambar 1. 1 source code boolean.h	2
Gambar 1. 2 Source code dari jam.h	5
Gambar 1. 2 Source code dari jam.h	5
Gambar 1. 3 Source code jam.c	11
Gambar 1. 3 Source code jam.c	11
Gambar 1. 4 Source code jam.c	11
Gambar 1. 4 Source code jam.c	11
Gambar 1. 5 Hasil program main.c	14
Gambar 1. 5 Hasil program main.c	14
Gambar 1. 6 Source code boolean.h	16
Gambar 2. 1 Source code	16
Gambar 2. 2 Source code point.h.....	19
Gambar 2. 3 Sebagian Source code point.c.....	25
Gambar 2. 4 Sebagian Source code point.c.....	25
Gambar 2. 5 Hasil dari ADT POINT	30

BAB I. HASIL PRAKTIKUM

I.1 ADT JAM

Program ADT JAM merupakan program yang mendefinisikan sebuah type bentukan yaitu type JAM dan menjalankan sekumpulan Primitif (operasi) terhadap type dari JAM tersebut. Di dalam program ADT JAM terdapat sebuah deklarasi type JAM kemudian primitive yang dijalankan mulai dari konstruktor hingga konversi type. Pada program ini terdapat 3 file yaitu header file, body file, dan juga driver.

I.1.A. Program boolean.h

I.1.A.1 source code

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

I.1.A.2 Hasil

```
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
```

Gambar 1. 1 source code boolean.h

Gambar 1. 2 source code boolean.h

I.1.A.3 Analisa

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.

I.1.B. Program jam.h

I.1.B.1 Source code

```
/*
    program          : jam.h
    author           : 2350081008/Ammar  Bagus  Fathurrahman
Wantoro
    kelas           : A
    deskripsi       : file header dari prototype jam.h
    tanggal          : 3/14/2024

*/

#ifndef JAM_H
#define JAM_H
#include "boolean.h"
#define true 1
```

```

#define false 0

#define boolean unsigned char

typedef struct{

    int Hour;

    int Minute;

    int Second;

}JAM;

/*prototype ADT JAM*/

/*Konstruktor membentuk ADT JAM*/

void CreateJam(JAM *J, int HH, int MM, int SS );

/*Selektor nilai JAM*/

int GetHour(JAM J);

int GetMinute(JAM J);

int GetSecond(JAM J);

/*Set nilai Komponen*/

void SetHour(JAM *J, int NewHour);

void SetMinute(JAM *J, int NewMinute);

void SetSecond(JAM *J, int NewSecond);

/*Kelompok baca tulis*/

void ReadJam(JAM *J);

void PrintJam(JAM J);

/*kelompok validasi type*/

boolean IsValid(int H, int M, int S);

```

```
/*operator relasional*/

boolean JEQ(JAM J1,JAM J2);

boolean JLT(JAM J1,JAM J2);

boolean JGT(JAM J1,JAM J2);


/*operator aritmatika*/

void Reset(JAM *J);

JAM NextDetik(JAM J);

JAM NextNDetik(JAM J, int N);

long int Durasi(JAM JAw, JAM JAkH);


/*kelompok konversi terhadap type*/

long int JamToDetik(JAM J);


#endif
```


I.1.B.2 Hasil

```
1  /*
2      program      : jam.h
3      author       : 2350081008/Amar Bagas Fathurrahman Wantoro
4      kelas        : A
5      deskripsi    : file header dari prototype jam.h
6      tanggal     : 3/14/2024
7
8  */
9
10 #ifndef JAM_H
11 #define JAM_H
12 #include "boolean.h"
13 #define true 1
14 #define false 0
15 #define boolean unsigned char
16
17 typedef struct(
18     int Hour;
19     int Minute;
20     int Second;
21 )JAM;
22
23 /*prototype ADT JAM*/
24
25 /*Konstruktor membentuk ADT JAM*/
26 void CreateJam(JAM *J, int H, int M, int S);
27
28 /*Selektor nilai JAM*/
29 int GetHour(JAM J);
30 int GetMinute(JAM J);
31 int GetSecond(JAM J);
32
33 /*Set nilai Komponen*/
34 void SetHour(JAM *J, int NewHour);
35 void SetMinute(JAM *J, int NewMinute);
36 void SetSecond(JAM *J, int NewSecond);
37
38 /*Kelompok baca tulis*/
39 void ReadJam(JAM *J);
40 void PrintJam(JAM J);
41
42 /*kelompok validasi type*/
43 boolean IsValid(int H, int M, int S);
44
45 /*operator relasional*/
46 boolean JEQ(JAM J1, JAM J2);
47 boolean JLT(JAM J1, JAM J2);
48 boolean JGT(JAM J1, JAM J2);
49
50
51 /*operator aritmatika*/
52 void Reset(JAM *J);
53 JAM NextDetik(JAM J);
54 JAM NextMenit(JAM J, int H);
55 long int Durasi(JAM JAw, JAM JAk);
56
57 /*kelompok konversi terhadap type*/
58 long int JamToDetik(JAM J);
59
60 #endif
```

Gambar 1. 3 Source code dari jam.h

Gambar 1. 4 Source code dari jam.h

I.1.B.3 Analisa

Program jam.h merupakan program header dari prototype ADT JAM yang berisikan sebuah deklarasi dari type bentukan JAM yang di dalam nya terdapat variabel untuk menyatakan Jam(Hour), Menit(Minute), dan juga Detik(Second). Kemudian selain deklarasi dari type bentukan JAM terdapat juga deklarasi primitif dari prototype ADT

JAM di mulai dari konstruktor yang berupa procedure CreateJam yang berfungsi sebagai pembangun atau pembuat jam. Selanjut nya selector berupa 3 fuction yaitu GetMinute, GetHour, dan GetSecond. Kemudian ada pengubahan nilai yang berupa procedure yaitu SetHour, SetMinute, SetSecond. Lalu ada kelompok baca dan tulis berupa procedure yaitu ReadJam dan PrintJam. Selanjutnya ada kelompok validasi type dengan function IsValid. Lalu ada operator relasional yang menjalankan fungsi perbandingan dengan function yaitu JEQ, JLT, JGT. Kemudian ada operator aritmatika yang menjalankan operasi aritmatika yang berupa procedure dan function Reset, NextDetik, NextNDetik, Durasi. Lalu yang terakhir adalah konversti type dengan JamToDetik.

I.1.C. Program jam.c

I.1.C.1 Source code

```
/*  
  
    program            : jam.c  
  
    author             : 2350081008/Ammar  Bagas  Fathurrahman  
Wantoro  
  
    kelas              : A  
  
    deskripsi          : program untuk pengoperasian atau menuliskan  
operasi dari prototype JAM  
  
    tanggal             : 3/14/2024  
  
*/  
  
#include "jam.h"  
#include <stdio.h>  
#include <conio.h>  
  
/*Konstruktor membentuk ADT JAM*/  
void CreateJam(JAM *J, int HH, int MM, int SS ){  
  
    (*J).Hour = HH;  
  
    (*J).Minute = MM;
```

```

        (*J).Second = SS;

    }

    /*Selektor nilali JAM*/
    int GetHour(JAM J){
        return J.Hour;
    }
    int GetMinute(JAM J){
        return J.Minute;
    }
    int GetSecond(JAM J){
        return J.Second;
    }

    /*Set nilai Komponen*/
    void SetHour(JAM *J, int NewHour){
        (*J).Hour = NewHour;
    }
    void SetMinute(JAM *J, int NewMinute){
        (*J).Minute = NewMinute;
    }
    void SetSecond(JAM *J, int NewSecond){
        (*J).Second = NewSecond;
    }

    /*Kelompok baca tulis*/
    void ReadJam(JAM *J){
        int hh,mm,ss;

        do{

```

```

        scanf("%d", &hh);

        scanf("%d", &mm);

        scanf("%d", &ss);

        }while(hh < 0 || hh > 23 || mm < 0 || mm > 59 || ss < 0 ||
ss > 59);

        CreateJam(&(*J),hh,mm,ss );

    }

    void PrintJam(JAM J){

        printf("%d:%d:%d",J.Hour,J.Minute,J.Second);

    }

    /*kelompok validasi type*/

    boolean IsValid(int H, int M, int S){

        if(H >= 0 && H <= 23 && M >= 0 && M <= 59 && S >= 0 && S
<= 59){

            return true;

        }else{

            return false;

        }

    }

    /*operator relasional*/

    boolean JEQ(JAM J1,JAM J2){

        if(J1.Hour == J2.Hour && J1.Minute == J2.Minute &&
J1.Second == J2.Second){

            return true;

        }else{

            return false;

        }

    }

```

```

        boolean JLT(JAM J1,JAM J2){

            if(J1.Hour < J2.Hour || J1.Minute < J2.Minute || J1.Second
< J2.Second){

                return true;

            }else{

                return false;

            }

        }

        boolean JGT(JAM J1,JAM J2){

            if(J1.Hour > J2.Hour || J1.Minute > J2.Minute ||
J1.Second > J2.Second){

                return true;

            }else{

                return false;

            }

        }

        /*operator aritmatika*/

        void Reset(JAM *J){

            (*J).Hour = 0;

            (*J).Minute = 0;

            (*J).Second = 0;

        }

        JAM NextDetik(JAM J){

            JAM JamBaru;

            JamBaru = J;

            JamBaru.Second = JamBaru.Second + 1;

            if(JamBaru.Second > 59){

                JamBaru.Second = 0;

```

```

        JamBaru.Minute = JamBaru.Minute + 1;

        if(JamBaru.Minute > 59){

            JamBaru.Minute = 0;

            JamBaru.Hour = JamBaru.Hour + 1;

            if(JamBaru.Hour > 23){

                JamBaru.Hour = 0;

            }

        }

    }

    return JamBaru;
}

JAM NextNDetik(JAM J, int N){

    int i;

    JAM JamBaru;

    JamBaru = J;

    if(N > 0 && N <= 86400){//86400 adalah second dari 24 jam

        for(i=1;i<=N;i++){

            JamBaru = NextDetik(JamBaru);

        }

    }

    return JamBaru;
}

long int Durasi(JAM JAw, JAM JAk){

    long int DetikAw, DetikAkh, result;

    DetikAw = JamToDetik(JAw);

    DetikAkh = JamToDetik(JAk);

    result = DetikAw - DetikAkh;

```

```

        if(result < 0){

            result = result * (-1);

        }

        return result;

    }

    /*kelompok konversi terhadap type*/

    long int JamToDetik(JAM J){

        long int Detik;

        Detik = (J.Hour * 3600) + (J.Minute * 60) + J.Second;

        return Detik;

    }

```

1.1.C.2 Hasil

```

1  /*
2  *   program   : jam.c
3  *   author    : 2108021006/Ammar Bagas Fathurrahman Alantora
4  *   kelas     : 4
5  *   deskripsi : program untuk pengoperasian atau menuliskan operasi dari prototype JAM
6  *   tanggal   : 3/04/2024
7  */
8
9
10 #include "jam.h"
11 #include <stdio.h>
12 #include <conio.h>
13
14 /*konstruktor membuat ADT JAM*/
15 void CreateJam(JAM *J, int H, int M, int S){
16     (*J).Hour = H;
17     (*J).Minute = M;
18     (*J).Second = S;
19 }
20
21 /*destruktor nilai JAM*/
22 void DestroyJam(JAM *J){
23     return J.Hour;
24 }
25 int dettHour(JAM J){
26     return J.Hour;
27 }
28 int dettMinute(JAM J){
29     return J.Minute;
30 }
31 int dettSecond(JAM J){
32     return J.Second;
33 }
34
35 /*Set nilai komponen*/
36 void SetHour(JAM *J, int NewHour){
37     (*J).Hour = NewHour;
38 }
39 void SetMinute(JAM *J, int NewMinute){
40     (*J).Minute = NewMinute;
41 }
42 void SetSecond(JAM *J, int NewSecond){
43     (*J).Second = NewSecond;
44 }
45
46 /*kelompok baca tulis*/
47 void ReadJam(JAM *J){
48     int H,M,S;
49     do{
50         scanf("%d", &H);
51         scanf("%d", &M);
52         scanf("%d", &S);
53     }while(H < 0 || H > 23 || M < 0 || M > 59 || S < 0 || S > 59);
54     CreateJam(J, H, M, S);
55 }
56
57 void PrintJam(JAM J){
58     printf("%d:%d:%d", J.Hour, J.Minute, J.Second);
59 }
60
61 /*kelompok validasi type*/
62 boolean isValid(int H, int M, int S){
63     if(H >= 0 && H <= 23 && M >= 0 && M <= 59 && S >= 0 && S <= 59){
64         return true;
65     }
66     return false;
67 }

```

Gambar 1. 7 Source code jam.c

```

67
68 /*operator relational*/
69 boolean is(JAM J1, JAM J2){
70     if(J1.Hour == J2.Hour && J1.Minute == J2.Minute && J1.Second == J2.Second){
71         return true;
72     }
73     return false;
74 }
75
76 boolean is1(JAM J1, JAM J2){
77     if(J1.Hour < J2.Hour || J1.Minute < J2.Minute || J1.Second < J2.Second){
78         return true;
79     }
80     return false;
81 }
82
83 boolean is2(JAM J1, JAM J2){
84     if(J1.Hour > J2.Hour || J1.Minute > J2.Minute || J1.Second > J2.Second){
85         return true;
86     }
87     return false;
88 }
89
90
91 /*operator aritmetika*/
92 void Reset(JAM *J){
93     (*J).Hour = 0;
94     (*J).Minute = 0;
95     (*J).Second = 0;
96 }
97
98 JAM NextDetik(JAM J){
99     JAM Jambaru;
100
101     Jambaru.Hour = J.Hour;
102     Jambaru.Second = J.Second + 1;
103     if(Jambaru.Second > 59){
104         Jambaru.Second = 0;
105         Jambaru.Minute = J.Minute + 1;
106         if(Jambaru.Minute > 59){
107             Jambaru.Minute = 0;
108             Jambaru.Hour = J.Hour + 1;
109             if(Jambaru.Hour > 23){
110                 Jambaru.Hour = 0;
111             }
112         }
113     }
114
115     return Jambaru;
116 }
117
118 JAM NextDetik(JAM J, int N){
119     int i;
120     JAM Jambaru;
121
122     Jambaru = J;
123     if(N > 0 && N <= 86400){ //86400 adalah second dari 24 jam
124         for(i=1; i<=N; i++){
125             Jambaru = NextDetik(Jambaru);
126         }
127     }
128     return Jambaru;
129 }
130
131 long int Durasi(JAM J1, JAM J2){
132     long int DetikJ1, DetikJ2, result;
133 }

```

Gambar 1. 5 Source code jam.c

Gambar 1. 8 Source code jam.c

Gambar 1. 6 Source code jam.c

I.1.C.3 Analisa

Program jam.c merupakan body file pada struktur ADT JAM yang di dalam nya terdapat operasi atau isi dari primitif-primitif yang sudah di deklarasikan pada file header sebelum nya yaitu file atau program jam.h. Program jam.c ini sangat penting bagi keseluruhan project karena jika tidak ada maka primitif yang sudah di deklarasikan pada file header jam.h tidak akan berjalan karena belum memiliki isi atau operasi di dalam nya.

I.1.D. Program main.c

I.1.D.1 Source code

```
/*  
    program           : main.c  
    author            : 2350081008/Ammar  Bagus  Fathurrahman  
Wantoro  
    kelas             : A  
    deskripsi         : main driver untuk menjalankan prototype JAM  
    tanggal           : 3/14/2024  
  
*/  
  
#include <stdio.h>  
#include <stdlib.h>  
#include "jam.c"  
#include "jam.h"  
  
int main () {  
    JAM J1, J2, J3, J4;  
    int tambahanDetik;  
    ReadJam(&J1);  
    printf("Nilai J1 ="); PrintJam(J1);  
    printf("\n");  
}
```



```

J2 = NextDetik(J1);

printf("Nilai J2 =");PrintJam(J2);printf("\n");

printf("Masukan nilai tambahan detik (0...59) =");
scanf("%d",&tambahanDetik);

J3 = NextNDetik(J1,tambahanDetik);
printf("Nilai J3 =");PrintJam(J3);printf("\n");

J4 = J1;
printf("Nilai J4 =");PrintJam(J4);printf("\n");

printf("\npengecekan operator relasional\n");
if(JEQ(J1,J2)){
    printf("\nNilai J1 sama dengan J2");
}
if(JLT(J1,J2)){
    printf("\nNilai J1 lebih kecil dari J2");
}
if(JGT(J1,J2)){
    printf("\nNilai J1 lebih besar dari J2");
}

if(JEQ(J3,J1)){
    printf("\nNilai J3 sama dengan J1");
}
if(JLT(J3,J1)){
    printf("\nNilai J3 lebih kecil dari J1");
}
if(JGT(J3,J1)){
    printf("\nNilai J3 lebih besar dari J1");
}

```

```

    }

    if(JEQ(J1,J4)){

        printf("\nNilai J1 sama dengan J4");

    }

    if(JLT(J1,J4)){

        printf("\nNilai J1 lebih kecil dari J4");

    }

    if(JGT(J1,J4)){

        printf("\nNilai J1 lebih besar dari J4");

    }

    Reset(&J4);

    printf("Nilai J4 =");PrintJam(J4);printf("\n");

    return 0;

}

```

I.1.D.2 Hasil

```

10
30
20
Nilai J1 =10:30:20
Nilai J2 =10:30:21
Masukan nilai tambahan detik (0...59) =3600
Nilai J3 =11:30:20
Nilai J4 =10:30:20

pengecekan operator relasional

Nilai J1 lebih kecil dari J2
Nilai J3 lebih besar dari J1
Nilai J1 sama dengan J4
Nilai J4 =0:0:0

-----
Process exited after 8.744 seconds with return value 0
Press any key to continue . . . |

```

Gambar 1. 9 Hasil program main.c

Gambar 1. 10 Hasil program main.c

I.1.D.3 Analisa

Program main.c merupakan program main driver dari ADT JAM yang menjalankan semua primitif atau operasi dari prototype JAM dengan menggunakan atau mengimplementasikan operasi – operasi yang telah di deklarasikan pada jam.h dan isi pada jam.c main driver menjalankan semua struktur atau program ADT JAM yang telah dibuat.

BAB II. TUGAS PRAKTIKUM

II.1 ADT POINT

Program ADT POINT merupakan program yang mendefinisikan sebuah type bentukan yaitu type POINT dan menjalankan sekumpulan Primitif (operasi) terhadap type dari POINT tersebut. Di dalam program ADT POINT terdapat sebuah deklarasi type POINT kemudian primitive yang dijalankan mulai dari konstruktor hingga konversi type. Pada program ini terdapat 3 file yaitu header file, body file, dan juga main driver.

II.1.A. Program Boolean.h

II.1.A.1 Source code

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

II.1.A.2 Hasil

```
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
```

Gambar 2. 1 Source code

II.1.A.3 Analisa

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.B. Program point.h

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  
  
*/  
  
#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.B.2 Hasil

```

1  /*
2      program      : point.h
3      author       : 2350081008/Amar Bagas Fathurrahman Wantoro
4      kelas        : A
5      deskripsi    : header file dari prototype point
6      tanggal     : 3/16/2024
7
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     int X;
21     int Y;
22 }POINT;
23
24 /*konstruktor POINT*/
25 void CreatePoint(POINT *P);
26 void CreatePoint2(POINT *P,int XBaru,int YBaru);
27
28 /*selektor POINT*/
29 int GetAbsis(POINT P);
30 int GetOrdinat(POINT P);
31
32 /*set nilai POINT*/
33 void SetAbsis(POINT *P,int XBaru);
34 void SetOrdinat(POINT *P,int YBaru);
35
36 /*kelompok interaksi dengan I/O device, Baca/Tulis*/
37 void BacaPoint(POINT *P);
38 void CetakPoint(POINT *P);
39
40 /*kelompok operasi relasional terhadap POINT*/
41 boolean EQ(POINT P1, POINT P2);
42 boolean NEQ(POINT P1,POINT P2);
43 boolean LT(POINT P1,POINT P2);
44 boolean MT(POINT P1,POINT P2);
45
46 /*kelompok menentukan dimana P berada*/
47 boolean IsOrigin(POINT P);
48 boolean IsOnSbX(POINT P);
49 boolean IsOnSbY(POINT P);
50 int Kuadran(POINT P);
51
52 /*kelompok operasi lain terhadap type*/
53 POINT MirrorOf(POINT P, boolean SbX, boolean SbY);
54 float jarakO(POINT P);
55 void GeserKeSbX(POINT *P);
56 void GeserKeSbY(POINT *P);

```

Gambar 2. 2 Source code point.h

II.1.B.3 Analisa

Program point.h merupakan header file dari ADT POINT di dalam nya terdapat deklarasi type bentukan POINT yang berisikan absis X dan ordinat Y. Kemudian 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.C. Program point.c

II.1.C.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.C.2 Hasil

```

1  /*
2  *   program      : point.h
3  *   author       : 2308021001/Amer Bagus Fathurrahman Wanto
4  *   kelas        : A
5  *   deskripsi    : header file dari prototype point
6  *   tanggal      : 3/10/2024
7  */
8
9
10 #include "point.h"
11 #include <stdio.h>
12 #include <conio.h>
13 #include <math.h>
14
15 /*struktur POINT*/
16 void CreatePoint (POINT *P){
17     (*P).x = 0;
18     (*P).y = 0;
19 }
20 void CreatePoint2 (POINT *P, int XBaru, int YBaru){
21     (*P).x = XBaru;
22     (*P).y = YBaru;
23 }
24
25 /*selektor POINT*/
26 int GetAbsis (POINT P){
27     return P.x;
28 }
29 int GetOrdinat (POINT P){
30     return P.y;
31 }
32
33 /*test nilai POINT*/
34 void SetAbsis (POINT *P, int XBaru){
35     (*P).x = XBaru;
36 }
37 void SetOrdinat (POINT *P, int YBaru){
38     (*P).y = YBaru;
39 }
40
41 /*Melompok interaksi dengan I/O device, Baca/tulis*/
42 void BacaPoint (POINT *P){
43     int x, y;
44     printf ("Masukkan nilai Absis X : "); scanf ("%d", &x);
45     printf ("Masukkan nilai Ordinat Y : "); scanf ("%d", &y);
46     CreatePoint2 (&(*P), x, y);
47 }
48 void CetakPoint (POINT *P){
49     printf ("Nilai Point adalah (%d,%d)", (*P).x, (*P).y);
50 }
51
52 /*Melompok operasi relasional terhadap POINT*/
53 boolean Is (POINT P1, POINT P2){
54     if (GetAbsis (P1) == GetAbsis (P2) && GetOrdinat (P1) == GetOrdinat (P2)){
55         return true;
56     }
57     else{
58         return false;
59     }
60 }
61 boolean NEq (POINT P1, POINT P2){
62     if (GetAbsis (P1) != GetAbsis (P2) && GetOrdinat (P1) != GetOrdinat (P2)){
63         return false;
64     }

```

Gambar 2. 4 Sebagian Source code point.c

```

64     }
65     else{
66         return true;
67     }
68 }
69 boolean IsOrdin (POINT P){
70     if (GetOrdinat (P) == 0){
71         return true;
72     }
73     else{
74         return false;
75     }
76 }
77 boolean IsOrdin (POINT P){
78     if (GetOrdinat (P) == 0){
79         return true;
80     }
81     else{
82         return false;
83     }
84 }
85 int Kuadrat (POINT P){
86     if (GetAbsis (P) > 0 && GetOrdinat (P) > 0){
87         return 1;
88     }
89     else if (GetAbsis (P) < 0 && GetOrdinat (P) > 0){
90         return 2;
91     }
92     else if (GetAbsis (P) < 0 && GetOrdinat (P) < 0){
93         return 3;
94     }
95     else if (GetAbsis (P) > 0 && GetOrdinat (P) < 0){
96         return 4;
97     }
98     else{
99         return 0;
100    }
101 }
102
103 /*Melompok operasi lain terhadap type*/
104 POINT MirrorOri (POINT P, boolean SbX, boolean SbY){
105     if (SbX == true){
106         P.x = GetAbsis (P) * -1;
107     }
108     else if (SbY == true){
109         P.y = GetOrdinat (P) * -1;
110     }
111     return P;
112 }
113 float Jarak (POINT P){
114     float Jarak;
115     Jarak = sqrt (pow ((GetAbsis (P) - 0), 2) + pow ((GetOrdinat (P) - 0), 2));
116 }
117 void GeserKeSbX (POINT *P){
118     if (GetOrdinat ((*P)) != 0){
119         SetOrdinat (&(*P), 0);
120     }
121 }
122 void GeserKeSbY (POINT *P){
123     if (GetAbsis ((*P)) != 0){
124         SetAbsis (&(*P), 0);
125     }
126 }

```

Gambar 2. 3 Sebagian Source code point.c

II.1.C.3 Analisa

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.

II.1.D. Program main.c

II.1.D.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 <stdio.h>  
#include <stdlib.h>  
#include "point.h"  
  
int main() {  
    POINT P1,P2;  
  
    /*konstruktor*/  
    CreatePoint(&P1);  
    CreatePoint(&P2);  
  
    CreatePoint2(&P1,0,5);  
    CreatePoint2(&P2,-1,3);  
  
    /*selektor*/  
    printf("\nNilai Absis dari P1 :%d",GetAbsis(P1));  
    printf("\nNilai Ordinat dari P1 :%d",GetOrdinat(P1));  
  
    /*Set nilai POINT*/
```

```

SetAbsis(&P1,-3);

SetOrdinat(&P1,2);


/*kelompok interaksi dengan I/O device, Baca/Tulis*/

BacaPoint(&P1);

CetakPoint(&P1);

CetakPoint(&P2);


/*kelompok operasi relasional terhadap POINT*/

if(EQ(P1, P2) == true){

    printf("\nNilai POINT sama");

    printf("(True)");

}else{

    printf("\nNilai POINT berbeda");

    printf("(False)");

}


if(NEQ(P1,P2) == true){

    printf("\nNilai POINT berbeda");

    printf("(True)");

}else{

    printf("\nNilai POINT sama");

    printf("(False)");

}


if(LT(P1,P2) == true){

    printf("\nPOINT P1 lebih kecil dari P2");

    printf("(True)");

}else{

    printf("\nPOINT P1 tidak lebih kecil dari P2");

```

```

        printf("(False)");
    }

    if(MT(P1,P2) == true){
        printf("\nPOINT P1 lebih besar dari P2");
        printf("(True)");
    }else{
        printf("\nPOINT P1 tidak lebih besar dari P2");
        printf("(False)");
    }

    /*kelompok menentukan dimana P berada*/
    if(IsOrigin(P1) == true){
        printf("\nPOINT berada di titik (0,0)");
        printf("(True)");
    }else{
        printf("\nPOINT berada di titik selain (0,0)");
        printf("(False)");
    }

    if(IsOnSbX(P1) == true){
        printf("\nPOINT berada di sumbu X");
        printf("(True)");
    }else{
        printf("\nPOINT tidak berada di sumbu X");
        printf("(False)");
    }

    if(IsOnSbY(P1) == true){
        printf("\nPOINT berada di sumbu Y");

```



```

        printf("(True)");

    }else{

        printf("\nPOINT tidak berada di sumbu Y");

        printf("(False)");

    }

    printf("\nPOINT    P1    berada    pada    kuadran    :    %d",
Kuadran(P1));

    printf("\nPOINT    P2    berada    pada    kuadran    :    %d",
Kuadran(P2));

    /*kelompok operasi lain terhadap type*/
    MirrorOf(P1,IsOnSbX(P1) , IsOnSbY(P1));

    printf("\nJarak POINT dari titik (0,0) : %d", jarakO(P1));

    GeserKeSbX(&P1);

    GeserKeSbY(&P1);

    return 0;

}

```

II.1.D.2 Hasil

```

Nilai Absis dari P1 :0
Nilai Ordinat dari P1 :5
Masukan nilai Absis X :-1
Masukan nilai Ordinat Y :3

Nilai Point adalah (-1,3)
Nilai Point adalah (-1,3)
Nilai POINT sama(True)
Nilai POINT sama(False)
POINT P1 tidak lebih kecil dari P2(False)
POINT P1 tidak lebih besar dari P2(False)
POINT berada di titik selain (0,0)(False)
POINT tidak berada di sumbu X(False)
POINT tidak berada di sumbu Y(False)
POINT P1 berada pada kuadran : 2
POINT P2 berada pada kuadran : 2
Jarak POINT dari titik (0,0) : 1610612736
-----
Process exited after 11.3 seconds with return value 0
Press any key to continue . . . |

```

II.1.D.3 Analisa

Program main.c adalah main driver dari struktur ADT POINT yang berisikan implementasi primitif point yang sudah di deklarasikan pada program header point.h dan juga operasi yang telah di buat pada program body file point.c dapat di lihat juga hasil dari operasi – operasi tersebut pada bagian hasil.

Gambar 2. 5 Hasil dari ADT POINT

BAB III. KESIMPULAN

Jadi kesimpulan pada praktikum kali ini kita dapat memahami bagaimana struktur dari sebuah ADT karena ADT harus di buat menjadi spesifikasi. Kemudian kita juga menjadi paham bagaimana membuat sebuah program ADT dalam Bahasa C yang terdiri dari header file untuk deklarasi dan body file untuk operasi tiap primitif nya dan main driver untuk menjalankan ADT nya.