

**IK141 Struktur Data**  
**Pendahuluan Struktur Data**  
**ADT (Abstrack Data Type)**



**Di Susun Oleh :**  
**Wendi Kardan, 2100016**

**PROGRAM STUDI PENDIDIKAN ILMU KOMPUTER**  
**FAKULTAS PENDIDIKAN MATEMATIKA DAN ILMU PENGETAHUAN ALAM**  
**UNIVERSITAS PENDIDIKAN INDONESIA**  
**19 Februari 2022**

## 1. Implementasi dan Hasil

### Implementasi dan Hasil

- a. Potongan kode dibawah ini adalah program yang digunakan untuk membuat Persegi. ADT dan Fungsi yang disimpan di file header adalah ADT Titik, ADT garis, ADT persegi, fungsi inputPersegi dan printPersegi.

File -> Persegi.h

```
1  #include<stdio.h>
2
3  typedef struct titik{
4      int X;
5      int Y;
6  }Titik;
7
8  typedef struct garis{
9      Titik awal;
10     Titik akhir;
11 }Garis;
12
13 typedef struct persegi{
14     Garis sisi1;
15     Garis sisi2;
16     Garis sisi3;
17     Garis sisi4;
18 }Persegi;
19
20
21 Persegi inputPersegi(Persegi p){
22     printf("Untuk sisi1 : \n");
23     scanf("%d %d %d %d", (&p.sisi1.awal.X), (&p.sisi1.awal.Y),(&p.sisi1.akhir.X), (&p.sisi1.akhir.Y));
24     printf("Untuk sisi2 : \n");
25     scanf("%d %d %d %d", (&p.sisi2.awal.X), (&p.sisi2.awal.Y),(&p.sisi2.akhir.X), (&p.sisi2.akhir.Y));
26     printf("Untuk sisi3 : \n");
27     scanf("%d %d %d %d", (&p.sisi3.awal.X), (&p.sisi3.awal.Y),(&p.sisi3.akhir.X), (&p.sisi3.akhir.Y));
28     printf("Untuk sisi4 : \n");
29     scanf("%d %d %d %d", (&p.sisi4.awal.X), (&p.sisi4.awal.Y),(&p.sisi4.akhir.X), (&p.sisi4.akhir.Y));
30     return p;
31 }
32
33 void printPersegi(Persegi p){
34     printf("Persegi yang dibentuk terdiri dari 4 sisi yang tersusun:\n");
35     printf("Sisi 1 dari titik (%d,%d) ke (%d,%d)\n",p.sisi1.awal.X,
36     p.sisi1.awal.Y, p.sisi1.akhir.X, p.sisi1.akhir.Y);
37     printf("Sisi 2 dari titik (%d,%d) ke (%d,%d)\n",p.sisi2.awal.X,
38     p.sisi2.awal.Y, p.sisi2.akhir.X, p.sisi2.akhir.Y);
39     printf("Sisi 3 dari titik (%d,%d) ke (%d,%d)\n",p.sisi3.awal.X,
40     p.sisi3.awal.Y, p.sisi3.akhir.X, p.sisi3.akhir.Y);
41     printf("Sisi 4 dari titik (%d,%d) ke (%d,%d)\n",p.sisi4.awal.X,
42     p.sisi4.awal.Y, p.sisi4.akhir.X, p.sisi4.akhir.Y);
43 }
```

File Persegi.h merupakan file header yang berisikan tipe data nested structure untuk membentuk persegi yang terdiri dari sisi1 – sisi4 dimana sisi-sisi tersebut terdiri dari titik awal dan titik akhir dengan koordinat x dan y yang dibuat menjadi typedata buatan Persegi. Setelah itu ada sebuah function dengan nama inputPersegi dengan tipe data Persegi yang menerima parameter p dengan

tipe data P yang akan mengisi data structure tersebut berdasarkan input dari user. Kemudian, terdapat prosedur printPersegi untuk mencetak data persegi ke dalam console.

File -> *mainProgram.c*

```
C mainProgram.c
1  #include<stdio.h>
2  #include "persegi.h"
3  int main(){
4      Persegi p1;
5      p1 = inputPersegi(p1);
6      printPersegi(p1);
7      return 0;
8  }
```

Di program mainnya akan memanggil file header yang sudah dibuat yaitu Persegi.h dimana function, structure, dan data type yang sudah dibuat di dalam file tersebut dapat di akses di file ini. Kemudian deklarasi sebuah variable bertipe data Persegi yang bernama p1 yang akan diisi oleh input user dan akan dicetak ke dalam console. Berikut adalah output yang dihasilkan setelah menerima input dari user.

```
Persegi yang dibentuk terdiri dari 4 sisi yang tersusun:
Sisi 1 dari titik (1,2) ke (3,4)
Sisi 2 dari titik (5,6) ke (7,8)
Sisi 3 dari titik (9,10) ke (11,12)
Sisi 4 dari titik (13,14) ke (15,16)
PS D:\UPI\SEMESTER 2\STRUKTUR DATA\Praktikum 2> []
```

## b. Restoran Coding

**Sources Code :**

```
1  #include <stdio.h>
2  #include <string.h>
3  #include <math.h>
4  #include <stdlib.h>
5
6  typedef struct makanan{
7      char nama[20];
8      int harga;
9      int total;
10 }Makanan;
11
12 int main() {
13
14     int total,i;
15     scanf("%d", &total);
16     Makanan listMakanan[total];
17
18     for(i = 0; i<total; i++){
19         fgets(listMakanan[i].nama,sizeof(listMakanan[i].nama), stdin);
20         gets(listMakanan[i].nama);
21         scanf("%d", &listMakanan[i].harga);
22         scanf("%d", &listMakanan[i].total);
23     }
24 }
```

```

24
25
26     int totalHarga = 0;
27     int perMenu;
28     for(i=0; i<total;i++){
29         perMenu = 0;
30         perMenu = listMakanan[i].harga * listMakanan[i].total;
31         totalHarga += perMenu;
32     }
33
34     printf("%d", totalHarga);
35
36
37     return 0;
38 }
39

```

Dalam program tersebut, diperlukan untuk membuat structure dengan tipe buatan baru yang dinamai **Makanan** yang didalamnya berisikan **nama(char)**, **harga(int)**, **total(int)** yang nantinya akan digunakan untuk menyimpan data makanan yang nanti akan dibeli oleh kostumer. Jumlah makanan akan diinputkan berdasar jumlah yang kostumer inginkan oleh karena itu perlu melakukan **scan data total(int)** yang nantinya akan diinisialisasikan sebagai array dengan **tipe data Makanan** yang panjang **array** tersebut sesuai dengan jumlah makanan yang nanti akan user beli. Lalu akan melakukan pengimputan data yang akan dilakukan oleh user, jumlah data yang diinput kan sesuai dengan jumlah makanan yang akan kostumer pesan dimana datanya itu terdiri dari **tipe data structure Makanan yang berisikan nama, harga, dan total**. Setelah diinput akan dikalkulasikan total yang harus dibayar dengan menjumlahkan harga makanan dikali dengan total tiap makanan. Hasil kalkulasi dari total makanan yang sudah dibeli akan ditampilkan ke dalam console. Hasil output dipaparkan di gambar di bawah ini.

#### **Output :**

```

5
Nasi goreng
40000
6
Sate ayam
60000
7
Soda gembira
45000
12
Es lemon
30000
4
Es krim
55000
8
1760000

```

### c. Jarak antara kamu dan dia

Buatlah program dalam bahasa C untuk membuat sebuah bidang dengan cara menyusun/menyambungkan N titik secara berurutan. Dari bidang yang dibangun tentukan 2 titik yang akan memotong bidang dengan titik potong paling panjang.

#### Sources Code :

```
1  #include <stdio.h>
2  #include <string.h>
3  #include <math.h>
4  #include <stdlib.h>
5
6  typedef struct titik{
7      int x;
8      int y;
9  }Titik;
10
11 float checkDistance(int x1, int y1, int x2, int y2){
12     return sqrt((x2-x1)*(x2-x1) + (y2-y1)*(y2-y1));
13 }
14
15
16 int main() {
17     int total,i,j;
18     scanf("%d", &total);
19     Titik pointGroup[total];
20     for(i=0; i<total; i++){
21         scanf("%d %d", &pointGroup[i].x,&pointGroup[i].y);
22     }
23     float highestDistance = 0;
24     Titik selectedPoint[2];
25     selectedPoint[0].x = pointGroup[0].x;
26     selectedPoint[0].y = pointGroup[0].y;
27     selectedPoint[1].x = pointGroup[1].x;
28     selectedPoint[1].y = pointGroup[1].y;
29     float distance;
30     for(i=0; i<total; i++){
31         for(j = 0; j<total; j++){
32             distance = checkDistance(pointGroup[i].x, pointGroup[i].y, pointGroup[j].x, pointGroup[j].y);
33             if(distance>highestDistance){
34                 selectedPoint[0].x = pointGroup[i].x;
35                 selectedPoint[0].y = pointGroup[i].y;
36                 selectedPoint[1].x = pointGroup[j].x;
37                 selectedPoint[1].y = pointGroup[j].y;
38                 highestDistance = distance;
39             }
40         }
41     }
42
43     printf("%d %d\n%d %d", selectedPoint[0].x, selectedPoint[0].y, selectedPoint[1].x, selectedPoint[1].y);
44     return 0;
45 }
```

Terdapat structure dengan nama **titik** yang dideklarasikan dengan ADT Bernama **Titik**, yang nantinya akan digunakan sebagai tempat untuk menyimpan **data x dan y** titik tersebut. Dimana tipe data ADT tersebut nanti akan digunakan untuk **mendeklarasikan array** dengan panjang array tersebut **sesuai dengan banyaknya titik yang nantinya akan di inputkan oleh user** (disimpan dalam variable total). Kemudian program akan melakukan input yang nanti akan dimasukan **kedalam structure tersebut ke nilai x dan y** nya yang nanti akan diulangi sesuai dengan banyak titik yang user masukan. Setelah memasukan data, maka program akan **mencari distance** dari satu titik ke titik lainnya menggunakan **fungsi checkDistance yang menerima 4 parameter** yaitu titik 1 x dan y serta titik 2 x dan y, fungsi ini akan **mengembalikan nilai float** sesuai dengan kalkulasi dengan rumus :

$$\sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2}$$

Setelah ditemukan nilai dari rumus diatas, maka akan **dicek manakah 2 titik yang memiliki jarak diagonal yang paling tinggi**. 2 titik yang memiliki jarak paling tertinggi akan ditampilkan ke dalam console. Hasil dari output program ini dijelaskan di gambar di bawah ini.

**Output:**

```

5
2
1
3
4
5
3
4
1
5
2
2 1
5 3

```

```

4
1
3
1
5
2
3
2
1
1 5
2 1

```

```

3
1
2
2
2
2
3
1 2
2 3

```

#### d. Pointer

Source Code :

```
1  #include<stdio.h>
2
3  int main(){
4      int *a;
5      int b=2;
6      a = &b;
7      printf("%d\n", b);
8      printf("%d\n", &b);
9      printf("%d\n", a);
10     printf("%d\n", *a);
11     printf("%d\n", &a);
12 }
```

a. Alamat dari variable b

```
printf("%d\n", &b); -> 6487572
```

b. Alamat yang disimpan dipointer a

```
a = &b; printf("%d\n", a); -> 6487572
```

c. Nilai yang ditunjukkan oleh pointer a

```
printf("%d\n", *a); -> 2
```

d. Alamat dari pointer a

```
printf("%d\n", &a); -> 6487576
```

## 2. Kesimpulan

### Kesimpulan

Setelah mengikuti praktikum ini, dapat disimpulkan bahwa Abstract Data Type (ADT) merupakan sebuah tipe data yang dapat di buat dan didefinisikan oleh programmer itu sendiri berbeda dari tipe data primitive. Kata kunci yang biasa digunakan adalah **typedef**. Lalu terdapat structure dimana memungkinkan dapat menyimpan banyak dan berbagai tipe data dalam satu variable sesuai dengan kebutuhan programmer. Kata kunci yang digunakan untuk membuat structure adalah **struct**. Kemudian dalam membangun sebuah ADT juga bisa dikombinasikan dengan menggunakan **Struct**. Setelah mempelajari ADT dan struct, dilanjutkan dengan mempelajari pointer dimana pointer merupakan suatu variable yang dapat digunakan untuk menampung suatu alamat dari variable tertentu. Untuk membuat variable pointer dapat menggunakan symbol \* pada awal variable tersebut.