

**LAPORAN PRAKTIKUM
STRUKTUR DATA**

**MODUL 3
ABSTRACT DATA TYPE (ADT)**



Disusun Oleh :

NAMA : Raihan Dzaky Muflih

NIM : 103112430029

Dosen

FAHRUDIN MUKTI WIBOWO

**PROGRAM STUDI STRUKTUR DATA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025**

A. Dasar Teori

Abstract Data Type (ADT) atau tipe data abstrak adalah konsep yang menjelaskan bagaimana suatu data dapat digunakan tanpa perlu tahu bagaimana data itu disimpan atau diolah di dalamnya. ADT berisi kumpulan data dan operasi yang bisa dilakukan, seperti menambah, menghapus, atau mencari data. Tujuannya yaitu agar program lebih mudah untuk dipahami dan juga diubah.

B. Guided (berisi screenshot source code & output program disertai penjelasannya)

Guided 1

```
//mahasiswa.h
#ifndef MAHASISWA_H_INCLUDED
#define MAHASISWA_H_INCLUDED

struct mahasiswa
{
    char nim[10];
    int nilai1, nilai2;
};

void inputMhs(mahasiswa &m);
float rata2(mahasiswa m);
#endif
```

```
//mahasiswa.cpp
#include "mahasiswa.h"
#include <iostream>
using namespace std;

void inputMhs(mahasiswa &m) {
    cout << "input nama : ";
    cin >> (m).nim;
    cout << "masukan nilai 1 : ";
    cin >> (m).nilai1;
    cout << "masukan nilai 2 : ";
    cin >> (m).nilai2;
}
```

```
float rata2(mahasiswa m) {  
    return float(m.nilai1 + m.nilai2) / 2;  
}
```

```
//main.cpp  
#include <iostream>  
#include "mahasiswa.h"  
#include "mahasiswa.cpp"  
using namespace std;  
  
int main() {  
    mahasiswa mhs;  
    inputMhs(mhs);  
    cout << "rata rata : " << rata2(mhs);  
}
```

Screenshots Output

```
PS D:\C++> d:\C++\main  
input nama : raihan  
masukan nilai 1 : 98  
masukan nilai 2 : 89  
rata rata : 93.5  
PS D:\C++> 
```

Deskripsi:

Program diatas adalah program sederhana yang digunakan untuk menghitung rata-rata dari 2 nilai yang diinputkan oleh pengguna dan akan ditampilkan sebagai output.

D. Unguided/Tugas (berisi screenshot source code & output program disertai penjelasannya)

Unguided 1

```
//nilaiMahasiswa.h
#ifndef MAHASISWA_H
#define MAHASISWA_H

#include <string>
using namespace std;

struct Mahasiswa {
    string nama;
    string nim;
    float uts;
    float uas;
    float tugas;
    float nilaiAkhir;
};

float NilaiAkhir(float uts, float uas, float tugas);
void inputData(Mahasiswa &mhs);
void tampilData(const Mahasiswa mhs[], int jumlah);

#endif
```

```
//nilaiMahasiswa.cpp
#include <iostream>
#include "NilaiMahasiswa.h"
using namespace std;

float NilaiAkhir(float uts, float uas, float tugas) {
    return 0.3f * uts + 0.4f * uas + 0.3f * tugas;
}
```

```

void inputData(Mahasiswa &mhs) {
    cout << "Nama      : ";
    getline(cin, mhs.nama);
    cout << "NIM       : ";
    getline(cin, mhs.nim);
    cout << "nilai UTS    : ";
    cin >> mhs.uts;
    cout << "nilai UAS    : ";
    cin >> mhs.uas;
    cout << "nilai Tugas : ";
    cin >> mhs.tugas;
    cin.ignore();

    mhs.nilaiAkhir = NilaiAkhir(mhs.uts, mhs.uas, mhs.tugas);
}

void tampilData(const Mahasiswa mhs[], int jumlah) {
    for (int i = 0; i < jumlah; i++) {
        cout << "\nMahasiswa " << i + 1 << endl;
        cout << "Nama          : " << mhs[i].nama << endl;
        cout << "NIM           : " << mhs[i].nim << endl;
        cout << "UTS           : " << mhs[i].uts << endl;
        cout << "UAS           : " << mhs[i].uas << endl;
        cout << "Tugas         : " << mhs[i].tugas << endl;
        cout << "Nilai Akhir  : " << mhs[i].nilaiAkhir << endl;
    }
}

```

```

//main.cpp
#include <iostream>
#include "nilaiMahasiswa.h"
#include "nilaiMahasiswa.cpp"
using namespace std;

int main() {
    Mahasiswa data[10];
    int jumlah;

```

```
    cout << "Masukkan jumlah mahasiswa : ";
    cin >> jumlah;
    cin.ignore();

    if (jumlah > 10) jumlah = 10;

    for (int i = 0; i < jumlah; i++) {
        cout << "\nmahasiswa " << i + 1 << endl;
        inputData(data[i]);
    }

    tampilData(data, jumlah);
    return 0;
}
```

Screenshots Output

```
PS D:\C++> d:\C++\modul3\tugas1\main
Masukkan jumlah mahasiswa : 2
```

```
mahasiswa 1
Nama    : raihan
NIM     : 12345
nilai UTS : 89
nilai UAS : 86
nilai Tugas : 95
```

```
mahasiswa 2
Nama    : dzaky
NIM     : 54321
nilai UTS : 84
nilai UAS : 87
nilai Tugas : 93
```

```
Mahasiswa 1
Nama      : raihan
NIM       : 12345
UTS       : 89
UAS       : 86
Tugas    : 95
Nilai Akhir : 89.6
```

```
Mahasiswa 2
Nama      : dzaky
NIM       : 54321
UTS       : 84
UAS       : 87
Tugas    : 93
Nilai Akhir : 87.9
PS D:\C++> █
```

Deskripsi:

Program diatas meminta pengguna untuk memasukkan jumlah siswa yang akan dimasukkan datanya, setelah itu pengguna diminta untuk memasukkan nama, NIM, nilai UTS, UAS dan juga nilai tugas dan akan dihitung nilai akhirnya oleh program dan ditampilkan sebagai output.

- E. Unguided/Tugas (berisi screenshot source code & output program disertai penjelasannya)

Unguided 2

```
//pelajaran.h
#ifndef PELAJARAN_H
#define PELAJARAN_H

#include <iostream>
#include <string>
using namespace std;

struct pelajaran {
    string namaMapel;
    string kodeMapel;
};

pelajaran create_pelajaran(string namapel, string kodepel);

void tampil_pelajaran(pelajaran pel);

#endif
```

```
//pelajaran.cpp
#include "pelajaran.h"

pelajaran create_pelajaran(string namapel, string kodepel) {
    pelajaran pel;
    pel.namaMapel = namapel;
    pel.kodeMapel = kodepel;
    return pel;
}

void tampil_pelajaran(pelajaran pel) {
    cout << "nama pelajaran : " << pel.namaMapel << endl;
    cout << "kode pelajaran : " << pel.kodeMapel << endl;
}
```



```
//main.cpp
#include "pelajaran.h"
#include "pelajaran.cpp"
using namespace std;

int main() {
    string namapel = "Struktur Data";
    string kodepel = "STD";

    pelajaran pel = create_pelajaran(namapel, kodepel);

    tampil_pelajaran(pel);

    return 0;
}
```

Screenshots Output

```
PS D:\C++> d:\C++\modul3\tugas2\main
nama pelajaran : Struktur Data
kode pelajaran : STD
PS D:\C++>
```

Deskripsi:

Program ini dibuat untuk menampilkan data pelajaran seperti nama pelajaran dan juga kode pelajaran.

- F. Unguided/Tugas (berisi screenshot source code & output program disertai penjelasannya)

Unguided 3

```
//pointer.h
#ifndef POINTER_H
#define POINTER_H

void tampilArray(int arr[3][3]);

void tukarIsiArray(int arr1[3][3], int arr2[3][3], int baris, int kolom);

void tukarPointer(int *p1, int *p2);

#endif
```

```
//pointer.cpp
#include <iostream>
#include "pointer.h"
using namespace std;

void tampilArray(int arr[3][3]) {
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            cout << arr[i][j] << " ";
        }
        cout << endl;
    }
}

void tukarIsiArray(int arr1[3][3], int arr2[3][3], int baris, int kolom) {
    int temp = arr1[baris][kolom];
    arr1[baris][kolom] = arr2[baris][kolom];
    arr2[baris][kolom] = temp;
}
```

```

}

void tukarPointer(int *p1, int *p2) {
    int temp = *p1;
    *p1 = *p2;
    *p2 = temp;
}

```

```

//main.cpp
#include <iostream>
#include "pointer.h"
#include "pointer.cpp"
using namespace std;

int main() {
    int array1[3][3] = {
        {1, 2, 3},
        {4, 5, 6},
        {7, 8, 9}
    };

    int array2[3][3] = {
        {9, 8, 7},
        {6, 5, 4},
        {3, 2, 1}
    };

    int x = 10, y = 20;
    int *p1 = &x;
    int *p2 = &y;

    cout << "sebelum ditukar" << endl;
    cout << "array 1:" << endl;
    tampilArray(array1);
    cout << "array 2:" << endl;
    tampilArray(array2);

    tukarIsiArray(array1, array2, 0, 0);
}

```

```
cout << "\nsetelah pertukaran" << endl;
cout << "array 1:" << endl;
tampilArray(array1);
cout << "array 2:" << endl;
tampilArray(array2);

cout << "\nSebelum tukar pointer:" << endl;
cout << "x = " << x << ", y = " << y << endl;

tukarPointer(p1, p2);

cout << "Setelah tukar pointer:" << endl;
cout << "x = " << x << ", y = " << y << endl;

return 0;
}
```

Screenshots Output

```
PS D:\C++> d:\C++\modul3\tugas3\main
sebelum ditukar
array 1:
1 2 3
4 5 6
7 8 9
array 2:
9 8 7
6 5 4
3 2 1

setelah pertukaran
array 1:
9 2 3
4 5 6
7 8 9
array 2:
1 8 7
6 5 4
3 2 1

Sebelum tukar pointer:
x = 10, y = 20
Setelah tukar pointer:
x = 20, y = 10
PS D:\C++> 
```

Deskripsi:

Program tersebut adalah program yang melakukan pertukaran nilai dalam array berbentuk 2D, dan juga pertukaran pointer yang nantinya akan dikeluarkan sebagai output dalam bentuk sebelum dan sesudah pertukaran.

G. Kesimpulan

ADT membantu kita membuat program yang teratur dan mudah dikelola karena memisahkan antara cara data digunakan dan bagaimana data itu bekerja di dalamnya. Dengan cara ini program jadi lebih mudah diperbaiki atau dikembangkan tanpa harus mengubah banyak bagian lain.

H. Referensi

[Liskov, Barbara](#); Zilles, Stephen (1974). "Programming with abstract data types". *Proceedings of the ACM SIGPLAN Symposium on Very High Level Languages*. SIGPLAN Notices. Vol. 9. pp. 50–59. [CiteSeerX](#) [10.1.1.136.3043](#). doi:[10.1145/800233.807045](#).

Dale, Nell; Walker, Henry M. (1996). *Abstract Data Types: Specifications, Implementations, and Applications*. Jones & Bartlett Learning. [ISBN 978-0-66940000-7](#).

Mitchell, John C.; Plotkin, Gordon (July 1988). "[Abstract Types Have Existential Type](#)" (PDF). *ACM Transactions on Programming Languages and Systems*. 10 (3).