

**LAPORAN PRAKTIKUM
STRUKTUR DATA**

**MODUL III
PENGENALAN BAHASA C++ (BAGIAN
KEDUA) ABSTRACT DATA TYPE (ADT)**



Disusun Oleh :

NAMA : ABYAN RAHMAN AL FARIZ

NIM : 103112430021

Dosen

FAHRUDIN MUKTI WIBOWO

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

A. Dasar Teori

ADT adalah TYPE dan sekumpulan PRIMITIF (operasi dasar) terhadap TYPE tersebut. Selain itu, dalam sebuah ADT yang lengkap, disertakan pula definisi invarian dari TYPE dan aksioma yang berlaku. ADT merupakan definisi STATIK. Definisi type dari sebuah ADT dapat mengandung sebuah definisi ADT lain. Misalnya :

- ADT waktu yang terdiri dari ADT JAM dan ADT DATE
- Garis terdiri dari dua buah ADT POINT
- SEGI4 yang terdiri dari pasangan dua buah POINT (Top,Left) dan (Bottom,Right)

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

Guided 1 (mahasiswa.cpp)

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

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

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

Guided 1 (mahasiswa.h)

```
#ifndef MAHASISWA_H_INCLUDED
#define MAHASISWA_H_INCLUDED

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

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

Guided 1 (main.cpp)

```
#ifndef MAHASISWA_H_INCLUDED
#define MAHASISWA_H_INCLUDED

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

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

Screenshots Output

```
PS D:\TelkomUniversity\Mata Kuliah\Semester 3\Struktur Data\Praktikum> & 'c:\Users\ACER\.vscode\extensions\ms-vscode.cpptools-1.27.7-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-qpnalkn0.s5e' '--stdout=Microsoft-MIEngine-Out-qwz1tgyf.1yy' '--stderr=Microsoft-MIEngine-Error-uwe10saq.axj' '--pid=Microsoft-MIEngine-Pid-ozcfgogb.ye3' '--dbgExe=C:\Users\ACER\mingw32\bin\gdb.exe' '--interpreter=mi'
input nama = Abyan
input nilai 1 = 90
input nilai 2 = 80
rata - rata = 85
PS D:\TelkomUniversity\Mata Kuliah\Semester 3\Struktur Data\Praktikum>
```

Deskripsi:

Program ini berfungsi untuk mengelola data mahasiswa dengan cara memasukkan nilai dan menghitung rata-rata dari nilai tersebut. Di dalamnya terdapat struktur data bernama mahasiswa yang menyimpan NIM serta dua nilai ujian. Fungsi

inputMhs bertugas menerima input berupa NIM dan nilai mahasiswa, sedangkan fungsi rata2 digunakan untuk menghitung rata-rata dari kedua nilai yang telah diinput. Secara keseluruhan, program ini serupa dengan versi sebelumnya, namun seluruh bagiannya digabung menjadi satu file utama.

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

Unguided 1

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

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

float rumus(float uts, float uas, float tugas) {
    return (0.3 * uts) + (0.4 * uas) + (0.3 * tugas);
}

void input(mahasiswa mhs[], int jumlah) {
    for (int i = 0; i < jumlah; i++) {
        cout << "\nMahasiswa ke-" << i + 1 << endl;

        cin.ignore();
        cout << "Nama: ";
        getline(cin, mhs[i].nama);

        cout << "NIM: ";
        cin >> mhs[i].nim;

        cout << "Nilai UTS: ";
        cin >> mhs[i].uts;

        cout << "Nilai UAS: ";
```

```

        cin >> mhs[i].uas;

        cout << "Nilai Tugas: ";
        cin >> mhs[i].tugas;

        mhs[i].nilaiAkhir = rumus(mhs[i].uts, mhs[i].uas,
mhs[i].tugas);
    }
}

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

int main() {
    int jumlah;
    mahasiswa mhs[10];

    cout << "Masukkan jumlah mahasiswa: ";
    cin >> jumlah;

    if (jumlah >= 1 && jumlah <= 10) {
        input(mhs, jumlah);
        tampilkanData(mhs, jumlah);
    } else {
        cout << "Jumlah mahasiswa harus antara 1 sampai 10" <<
endl;
    }

    return 0;
}

```

Screenshots Output

```
PS D:\TelkomUniversity\Mata Kuliah\Semester 3\Struktur Data\Praktikum> & 'c:\Users\ACER\.vscode\extensions\ms-vscode.cpptools-1.27.7-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-ktkauavj.y4x' '--stdout=Microsoft-MIEngine-Out-yjvxy22v.5yq' '--stderr=Microsoft-MIEngine-Error-k1wi11oi.2kq' '--pid=Microsoft-MIEngine-Pid-z1npka3w.x0s' '--dbgExe=C:\Users\ACER\mingw32\bin\gdb.exe' '--interpreter=mi'
Masukkan jumlah mahasiswa: 3

Mahasiswa ke-1
Nama: Abyan
NIM: 103112430021
Nilai UTS: 95
Nilai UAS: 98
Nilai Tugas: 94

Mahasiswa ke-2
Nama: Rahman
NIM: 103112430022
Nilai UTS: 92
Nilai UAS: 97
Nilai Tugas: 99

Mahasiswa ke-3
Nama: Fariz
NIM: 103112430023
Nilai UTS: 98
Nilai UAS: 93
Nilai Tugas: 97

Data Mahasiswa ke-1
Nama: Abyan
NIM: 103112430021
Nilai UTS: 95
Nilai UAS: 98
```

Data Mahasiswa ke-1

Nama: Abyan
NIM: 103112430021
Nilai UTS: 95
Nilai UAS: 98
Nilai Tugas: 94
Nilai Akhir: 95.9

Data Mahasiswa ke-2

Nama: Rahman
NIM: 103112430022
Nilai UTS: 92
Nilai UAS: 97
Nilai Tugas: 99
Nilai Akhir: 96.1

Data Mahasiswa ke-3

Nama: Fariz
NIM: 103112430023
Nilai UTS: 98
Nilai UAS: 93
Nilai Tugas: 97
Nilai Akhir: 95.7

Deskripsi:

Program ini berfungsi untuk mengelola data mahasiswa dengan cara memasukkan nilai dan menghitung nilai akhir berdasarkan bobot tertentu. Di dalam program terdapat struktur data mahasiswa yang digunakan untuk menyimpan informasi seperti nama, NIM, serta nilai UTS, UAS, dan tugas. Perhitungan nilai akhir dilakukan menggunakan fungsi rumus(), yang menerapkan formula $0.3 \times \text{UTS} + 0.4 \times \text{UAS} + 0.3 \times \text{Tugas}$.

Proses input data dilakukan melalui prosedur input(), di mana pengguna diminta mengisi nama, NIM, dan ketiga nilai untuk setiap mahasiswa. Setelah semua data dimasukkan, fungsi rumus() akan dipanggil untuk menghitung nilai akhir masing-masing mahasiswa. Data yang telah diinput kemudian ditampilkan melalui prosedur tampilkanData(), yang menampilkan seluruh informasi mahasiswa secara berurutan, termasuk hasil nilai akhirnya.

Unguided 2 (pelajaran.h)

```
#ifndef PELAJARAN_H_INCLUDED
#define PELAJARAN_H_INCLUDED

#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
```

Unguided 2 (pelajaran.cpp)

```
#include "pelajaran.h"

// Fungsi untuk membuat data pelajaran
Pelajaran create_pelajaran(string namapel, string kodepel) {
    Pelajaran p;
    p.namaMapel = namapel;
    p.kodeMapel = kodepel;
    return p;
}

void tampil_pelajaran(Pelajaran pel) {
    cout << "=== Data Pelajaran ===" << endl;
    cout << "Nama Mata Kuliah : " << pel.namaMapel << endl;
    cout << "Kode Mata Kuliah : " << pel.kodeMapel << endl;
}
```

Unguided 2 (main.cpp)

```
#include <iostream>
#include "pelajaran.h"
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:\TelkomUniversity\Mata Kuliah\Semester 3\Struktur Data\Praktikum\Modul 3\Unguided\Unguided2> g++ main.cpp pelajaran.c
pp -o main
PS D:\TelkomUniversity\Mata Kuliah\Semester 3\Struktur Data\Praktikum\Modul 3\Unguided\Unguided2> ./main
=== Data Pelajaran ===
Nama Mata Kuliah : Struktur Data
Kode Mata Kuliah : STD
PS D:\TelkomUniversity\Mata Kuliah\Semester 3\Struktur Data\Praktikum\Modul 3\Unguided\Unguided2> |
```

Deskripsi:

Program ini merupakan implementasi konsep Abstract Data Type (ADT) pada bahasa C++ untuk mengelola data mata kuliah. Program dibagi menjadi tiga file utama, yaitu pelajaran.h, pelajaran.cpp, dan main.cpp, guna memisahkan antara spesifikasi tipe data, implementasi fungsi, dan program utama (driver).

Pada file pelajaran.h, didefinisikan struktur Pelajaran yang memiliki dua atribut, yaitu namaMapel dan kodeMapel. File ini juga mendeklarasikan dua primitif ADT berupa fungsi create_pelajaran() untuk membuat objek pelajaran baru dan prosedur tampil_pelajaran() untuk menampilkan data pelajaran ke layar.

File pelajaran.cpp berisi realisasi (implementasi) dari fungsi dan prosedur yang telah dideklarasikan. Fungsi create_pelajaran() digunakan untuk membentuk sebuah objek pelajaran berdasarkan input nama dan kode mata kuliah, sedangkan prosedur tampil_pelajaran() menampilkan informasi tersebut dalam format yang rapi.

Terakhir, file main.cpp berperan sebagai driver program, yaitu bagian utama yang memanggil fungsi dan prosedur dari ADT Pelajaran. Di sini, program membuat objek pelajaran bernama pel dengan nilai "Struktur Data" sebagai nama mata kuliah dan "STD" sebagai kode mata kuliah, kemudian menampilkannya menggunakan tampil_pelajaran().

Unguided 3

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

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

```

void tukarElemen(int arrA[3][3], int arrB[3][3], int barisA, int
kolomA, int barisB, int kolomB) {
    int temp = arrA[barisA][kolomA];
    arrA[barisA][kolomA] = arrB[barisB][kolomB];
    arrB[barisB][kolomB] = temp;
}

void tukarDataPointer(int *p1, int *p2) {
    int sementara = *p1;
    *p1 = *p2;
    *p2 = sementara;
}

int main() {
    int matriks1[3][3] = {
        {11, 12, 13},
        {14, 15, 16},
        {17, 18, 19}
    };
    int matriks2[3][3] = {
        {91, 92, 93},
        {94, 95, 96},
        {97, 98, 99}
    };

    cout << "=== Matriks 1 (Awal) ===" << endl;
    tampilArray(matriks1);
    cout << "=== Matriks 2 (Awal) ===" << endl;
    tampilArray(matriks2);

    tukarElemen(matriks1, matriks2, 0, 1, 2, 2);

    cout << "=== Setelah Penukaran Elemen ===" << endl;
    cout << "Matriks 1:" << endl;
    tampilArray(matriks1);
    cout << "Matriks 2:" << endl;
    tampilArray(matriks2);

    int x = 50, y = 100;
    int *ptrX = &x;
    int *ptrY = &y;

```

```

        cout << "Sebelum tukar pointer: x = " << x << ", y = " << y <<
endl;
        tukarDataPointer(ptrX, ptrY);
        cout << "Sesudah tukar pointer: x = " << x << ", y = " << y <<
endl;

        return 0;
}

```

Screenshots Output

```

PS D:\TelkomUniversity\Mata Kuliah\Semester 3\Struktur Data\Praktikum> & 'c:\Users\ACER\.vscode\extensions\ms-vscode.cpptools-1.27.7-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-cfgbjfsf.zmm' '--stdout=Microsoft-MIEngine-Out-z3x1sg00.lzt' '--stderr=Microsoft-MIEngine-Error-iir2zakw.y4q' '--pid=Microsoft-MIEngine-Pid-idi05rht.g1z' '--dbgExe=C:\Users\ACER\mingw32\bin\gdb.exe' '--interpreter=mi'
=== Matriks 1 (Awal) ===
11 12 13
14 15 16
17 18 19

=== Matriks 2 (Awal) ===
91 92 93
94 95 96
97 98 99

=== Setelah Penukaran Elemen ===
Matriks 1:
11 99 13
14 15 16
17 18 19

Matriks 2:
91 92 93
94 95 96
97 98 12

Sebelum tukar pointer: x = 50, y = 100
Sesudah tukar pointer: x = 100, y = 50
PS D:\TelkomUniversity\Mata Kuliah\Semester 3\Struktur Data\Praktikum>

```

Deskripsi:

Program ini berfungsi untuk memanipulasi dua buah array dua dimensi (matriks) dan dua pointer integer. Program menunjukkan cara kerja fungsi dan pointer dalam menukar nilai baik di dalam array maupun antar variabel.

Pertama, terdapat fungsi tampilArray() yang digunakan untuk menampilkan isi dari sebuah array dua dimensi berukuran 3×3 dengan format yang rapi. Kemudian, fungsi tukarElemen() digunakan untuk menukar elemen antara dua array 2D pada posisi tertentu yang ditentukan melalui parameter indeks baris dan kolom. Proses penukaran dilakukan dengan menyimpan sementara nilai dari elemen pertama ke dalam variabel temp, kemudian mengganti nilainya dengan elemen dari array kedua.

Selanjutnya, fungsi `tukarDataPointer()` digunakan untuk menukar nilai dua variabel integer melalui pointer. Fungsi ini menggunakan teknik dereferensi (`*p1` dan `*p2`) untuk mengakses nilai yang ditunjuk pointer, lalu menukar keduanya dengan variabel sementara sementara.

Pada bagian `main()`, dua array (`matriks1` dan `matriks2`) diinisialisasi dengan nilai berbeda. Program menampilkan isi kedua matriks sebelum dan sesudah proses penukaran elemen tertentu. Selain itu, terdapat dua variabel (`x` dan `y`) yang nilainya juga ditukar menggunakan pointer.

E. Kesimpulan

Secara keseluruhan, Abstract Data Type (ADT) merupakan salah satu konsep fundamental dalam pemrograman yang menitikberatkan pada fungsi dan perilaku suatu tipe data tanpa memperhatikan detail implementasinya secara internal. Melalui penerapan ADT, tipe data beserta operasi dasarnya — seperti pembuatan, pembacaan, modifikasi, dan penampilan data — didefinisikan secara terpisah dari realisasi kodenya. Pendekatan ini menjadikan struktur program lebih modular, terorganisasi, serta mudah untuk dikembangkan dan dipelihara.

Dalam implementasinya pada bahasa pemrograman C++, konsep ADT umumnya direalisasikan menggunakan `struct`, fungsi, serta file header (`.h`) sebagai tempat pendefinisian tipe dan operasi dasar, sedangkan file `.cpp` digunakan untuk mengimplementasikan logika fungsionalnya. Pemisahan antara spesifikasi dan implementasi ini memberikan manfaat berupa kemudahan dalam proses pengembangan perangkat lunak berskala besar, karena setiap modul dapat dikembangkan atau dimodifikasi secara independen tanpa memengaruhi keseluruhan sistem.

F. Referensi

<https://id.scribd.com/document/411113387/Abstract-Data-Type-ADT-1>

<https://brilliant.org/wiki/abstract-data-types/>

<https://www.scholarhat.com/tutorial/datastructures/abstract-data-type>