

**LAPORAN PRAKTIKUM
STRUKTUR DATA**

MODUL 3



Disusun Oleh :

NAMA : IBTIDA ZADA UTOMO

NIM : 103112430037

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 merupakan suatu konsep dalam pemrograman yang mendefinisikan tipe data beserta operasi-operasi yang dapat dilakukan terhadapnya tanpa memperhatikan bagaimana implementasinya dilakukan. Dalam bahasa C++, ADT biasanya diimplementasikan menggunakan struktur data seperti *struct* atau *class* yang dilengkapi dengan fungsi-fungsi pendukung untuk mengelola data tersebut. Pendekatan ini membantu memisahkan antara definisi data dan logika program, sehingga kode menjadi lebih terorganisir, modular, dan mudah untuk dikembangkan.

Pada praktikum ini, konsep ADT diterapkan untuk mengelola berbagai jenis data menggunakan beberapa contoh program. ADT pertama digunakan untuk mengelola data mahasiswa, seperti nama, NIM, serta nilai UTS, UAS, dan tugas, kemudian menghitung nilai akhir berdasarkan bobot tertentu. ADT kedua digunakan untuk mengelola data pelajaran yang berisi nama mata pelajaran dan kode pelajaran. Sedangkan pada ADT ketiga, digunakan struktur dan fungsi untuk memahami manipulasi data menggunakan array dua dimensi serta pointer, termasuk proses pertukaran nilai antar-elemen array dan antar-variabel melalui pointer.

Dengan menerapkan ADT, program menjadi lebih mudah dipahami dan dirawat karena setiap bagian memiliki tanggung jawab tersendiri. Hal ini juga mendukung prinsip *modularity* dalam pemrograman, di mana file header (*.h*) digunakan untuk mendeklarasikan struktur dan fungsi, sedangkan file implementasi (*.cpp*) berisi definisi dari fungsi-fungsi tersebut.

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

Guided 1

```
#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
```

```
#include "guided1modul3.h"
#include <iostream>
using namespace std;

void inputMhs(mahasiswa &m) {
    cout << "Input NIM: ";
    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;
}
```

```
#include <iostream>
#include "guided1modul3.h"
using namespace std;

int main() {
    mahasiswa mhs;
    inputMhs(mhs);
    cout << "Rata-rata = " << rata2(mhs) << endl;
}
```

Screenshots Output

```
PS D:\C++> d:\C++\main
input nama : Zada
masukan nilai 1 : 101
masukan nilai 2 : 102
rata rata : 101.5
PS D:\C++> 
```

deskripsi : program diatas adalah program untuk menginput data mahasiswa dan menghitung rata rata nilai mahasiswa menggunakan konsep adt dengan 3 file terpisah nantinya program ini meminta pengguna memasukan nim, lalu nilai tugas, dan program akan menghitung rata rata dari kedua nilai tersebut

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

Unguided 1

mahasiswa.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 hitungNilaiAkhir(float uts, float uas, float tugas);

Mahasiswa createMahasiswa(string nama, string nim, float uts,
float uas, float tugas);

void tampilMahasiswa(Mahasiswa mhs);

#endif
```

mahasiswa.cpp

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

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

Mahasiswa createMahasiswa(string nama, string nim, float uts,
float uas, float tugas) {
    Mahasiswa mhs;
    mhs.nama = nama;
    mhs.nim = nim;
    mhs.uts = uts;
    mhs.uas = uas;
    mhs.tugas = tugas;
    mhs.nilaiAkhir = hitungNilaiAkhir(uts, uas, tugas);
    return mhs;
}

void tampilMahasiswa(Mahasiswa mhs) {
    cout << "Nama          : " << mhs.nama << endl;
    cout << "NIM            : " << mhs.nim << endl;
    cout << "Nilai UTS       : " << mhs.uts << endl;
    cout << "Nilai UAS       : " << mhs.uas << endl;
    cout << "Nilai Tugas     : " << mhs.tugas << endl;
    cout << "Nilai Akhir     : " << mhs.nilaiAkhir << endl;
}
```

main.cpp

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

int main() {
    Mahasiswa daftar[10];
    int n;

    cout << "Masukkan jumlah mahasiswa (maks 10): ";
    cin >> n;
    if (n > 10) n = 10;

    for (int i = 0; i < n; i++) {
        string nama, nim;
        float uts, uas, tugas;
        cin.ignore();
        cout << "\nData mahasiswa ke-" << i + 1 << endl;
        cout << "Nama    : ";
        getline(cin, nama);
        cout << "NIM     : ";
        getline(cin, nim);
        cout << "UTS     : ";
        cin >> uts;
        cout << "UAS     : ";
        cin >> uas;
        cout << "Tugas   : ";
        cin >> tugas;

        daftar[i] = createMahasiswa(nama, nim, uts, uas, tugas);
    }
    cout << "\n=== DATA MAHASISWA ===\n";
    for (int i = 0; i < n; i++) {
        cout << "\nMahasiswa ke-" << i + 1 << endl;
        tampilMahasiswa(daftar[i]);
    }

    return 0;
}
```

Screenshots Output

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\struktur data\MODUL 3\UNDIUDED\soal1modul3> cd "d:\struktur data\MODUL 3\UNDIUDED\soal1modul3\" ; if ($?) { g++ main.cpp -o main } ; if ($?) { .\main }
Masukkan jumlah mahasiswa (maks 10): 2

Data mahasiswa ke-1
Nama : zada
NIM : 103112430037
UTS : 98
UAS : 100
Tugas : 89

Data mahasiswa ke-2
Nama : slamet
NIM : 12345678
UTS : 90
UAS : 80
Tugas : 70

=== DATA MAHASISWA ===

Mahasiswa ke-1
Nama : zada
NIM : 103112430037
Nilai UTS : 98
Nilai UAS : 100
Nilai Tugas : 89
Nilai Akhir : 96.1

Mahasiswa ke-2
Nama : slamet
NIM : 12345678
Nilai UTS : 90
Nilai UAS : 80
Nilai Tugas : 70
Nilai Akhir : 80
PS D:\struktur data\MODUL 3\UNDIUDED\soal1modul3>
```

Deskripsi: program diatas adalah program untuk memasukan data mahasiswa nanti akan diminta berapa jumlah mahasiswa maksimal 10 dan akan diminta memasukan nama mahasiswa lalu nim dan nilai uts uas dan tugas, nantinya program akan menyimpan data tersebut, program diatas menggunakan adt

Unguided 2

pelajaran.h

```
#ifndef PELAJARAN_H
#define PELAJARAN_H

#include <string>
using namespace std;

struct pelajaran {
    string namaMapel;
    string kodeMapel;
};

pelajaran create_pelajaran(string nama, string kode);

void tampil_pelajaran(pelajaran pel);
```

```
#endif
```

pelajaran.cpp

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

pelajaran create_pelajaran(string nama, string kode) {
    pelajaran p;
    p.namaMapel = nama;
    p.kodeMapel = kode;
    return p;
}

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

main.cpp

```
#include <iostream>
#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:\struktur data\MODUL 3\UNGUIDED\soal1modul3> cd "d:\struktur data\MODUL 3\UNGUIDED\soal2modul3\" ; if ($?) { g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
nama pelajaran : Struktur Data
nilai : STD
PS D:\struktur data\MODUL 3\UNGUIDED\soal2modul3> █
```

Deskripsi: program diatas adalah program untuk membuat dan menampilkan data sebuah pelajaran, program memanfaatkan file header dan file implementasi untuk mendefinisikan struktur data serta fungsi fungsi yang berkaitan dengan pelajaran tersebut

Unguided 3

array.h

```
#ifndef ARRAY2D_H
#define ARRAY2D_H

#include <iostream>
using namespace std;

const int N = 3;

void tampilArray(int arr[N][N]);
void tukarArrayPosisi(int arr1[N][N], int arr2[N][N], int baris,
int kolom);
void tukarPointer(int *p1, int *p2);

#endif
```

array.cpp

```
#include "array.h"

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

```

void tukarArrayPosisi(int arr1[N][N], int arr2[N][N], 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 "array.h"
#include "array.cpp"

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

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

    cout << "Array A:" << endl;
    tampilkanArray(A);
    cout << "Array B:" << endl;
    tampilkanArray(B);

    int baris = 1, kolom = 2;
    tukarArrayPosisi(A, B, baris, kolom);
}

```

```

        cout << "\nSetelah menukar posisi [" << baris << "]" << kolom
<< "]" << endl;
        cout << "Array A:" << endl;
        tampilArray(A);
        cout << "Array B:" << endl;
        tampilArray(B);

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

        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

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\struktur data\MODUL 3\UNQUIDED\soal2modul3> cd "d:\struktur data\MODUL 3\UNQUIDED\soal3modul3\"; if ($?) { g++ main.cpp -o main }; if ($?) { .\main }
Array A:
1 2 3
4 5 6
7 8 9
Array B:
9 8 7
6 5 4
3 2 1

Setelah menukar posisi [1][2]:
Array A:
1 2 3
4 5 4
7 8 9
Array B:
9 8 7
6 5 6
3 2 1

Sebelum tukar pointer:
x = 10, y = 20
Setelah tukar pointer:
x = 20, y = 10
PS D:\struktur data\MODUL 3\UNQUIDED\soal3modul3>

```

Deskripsi: program diatas adalah program untuk mendemonstrasikan dua cara dalam menukar nilai, menggunakan array dan pointer program ini memiliki 2 tujuan utama yaitu menukar nilai pada posisi tertentu di antara dua array dua dimensi, Menukar nilai dari dua variabel integer dengan menggunakan pointer.

C. Kesimpulan

Dari hasil praktikum, dapat disimpulkan bahwa penggunaan ADT dalam pemrograman C++ membantu mengorganisir program menjadi lebih modular, mudah dibaca, dan efisien.

Dengan memisahkan definisi struktur dan implementasi fungsi dalam file yang berbeda, programmer dapat lebih mudah melakukan perbaikan dan pengembangan program tanpa mengubah keseluruhan kode.

Selain itu, pemahaman terhadap pointer dan array juga penting dalam pengelolaan data pada level memori yang lebih rendah.

D. Referensi

- Harry. H. Chaudhary (28 July 2014). "[Cracking The Java Programming Interview :: 2000+ Java Interview Que/Ans](#)". Archived from the original on 27 May 2021. Retrieved 29 May 2016.
- Roger Poon (1 May 2017). "[Scaling JS++: Abstraction, Performance, and Readability](#)". Archived from the original on 11 May 2020. Retrieved 21 April 2020.
- "[The evolution of an extension language: a history of Lua](#)". www.lua.org. Retrieved 4 January 2023.