

**LAPORAN PRAKTIKUM
STRUKTUR DATA**

**MODUL III
PENGENALAN ABSTRACT DATA TYPE**



Disusun Oleh :

NAMA : Sinta Sintiani

NIM : 103112430047

Dosen

FAHRUDIN MUKTI WIBOWO

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

A. Dasar Teori

Abstract Data Type (ADT) adalah konsep dalam pemrograman yang digunakan untuk mendefinisikan suatu tipe data beserta operasi-operasi yang dapat dilakukan terhadap data tersebut, tanpa memperhatikan bagaimana data itu disimpan atau diimplementasikan. ADT berfungsi untuk memisahkan antara definisi logika (konseptual) dan implementasi fisik dari struktur data, sehingga program menjadi lebih mudah dipahami, dimodifikasi, dan dipelihara.

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

mahasiswa.cpp

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

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

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

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
```

main.cpp

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

int main()
{
    mahasiswa mhs;
    inputMhs(mhs);
    cout << "rata - rata = " << rata2(mhs);
    return 0;
}
```

Screenshots Output

```
PS D:\SEM 3\STRUKTUR DATA\mod 3> cd "d:\SEM 3\STRUKTUR DATA\mod 3\" ; if ($?) { g++ main.cpp -o main } ; if ($?) { .\main }
input nama = sinta
input nilai1 = 90
input nilai2 = 80
rata - rata = 85
```

Deskripsi:

Program ini untuk mengelola data mahasiswa yang terdiri dari NIM, nilai1, dan nilai2. Struktur data dan fungsi didefinisikan dalam file header (mahasiswa.h), sedangkan implementasinya ada di mahasiswa.cpp. Fungsi inputMhs() digunakan untuk memasukkan data mahasiswa, dan rata2() untuk menghitung rata-rata dua nilai. Program utama memanggil kedua fungsi tersebut, lalu menampilkan hasil rata-rata ke layar.

Soal 1

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

float hitungNilaiAkhir(float uts, float uas, float tugas) {
```

```

        return 0.3 * uts + 0.4 * uas + 0.3 * tugas;
    }

int main() {
    string nama[10], nim[10];
    float uts[10], uas[10], tugas[10], akhir[10];
    int n;

    cout << "Jumlah mahasiswa (maks 10): ";
    cin >> n;

    for (int i = 0; i < n; i++) {
        cout << "\nMahasiswa ke-" << i + 1 << endl;
        cout << "Nama      : ";
        cin >> nama[i];
        cout << "NIM       : ";
        cin >> nim[i];
        cout << "UTS       : ";
        cin >> uts[i];
        cout << "UAS       : ";
        cin >> uas[i];
        cout << "Tugas     : ";
        cin >> tugas[i];

        akhir[i] = hitungNilaiAkhir(uts[i], uas[i], tugas[i]);
    }

    cout << "\n=====\\n";
    cout << "Data Nilai Akhir Mahasiswa\\n";
    cout << "=====\\n";
    for (int i = 0; i < n; i++) {
        cout << i + 1 << ". " << nama[i] << " (" << nim[i] << ") = " << akhir[i] << endl;
    }

    return 0;
}

```

Screenshots Output

```

PS D:\SEM 3\STRUKTUR DATA\Modul 3> cd "d:\SEM 3\STRUKTUR DATA\Modul 3\" ; if ($?) { g++ soal1.cpp -o soal1 } ; if ($?) { .\soal1 }
Jumlah mahasiswa: 3

Mahasiswa ke-1
Nama : Sinta
NIM : 103112430047
UTS : 80
UAS : 90
Tugas : 90

Mahasiswa ke-2
Nama : Chilya
NIM : 103112430053
UTS : 80
UAS : 85
Tugas : 90

Mahasiswa ke-3
Nama : Rere
NIM : 103112430031
UTS : 90
UAS : 90
Tugas : 90

=====
Data Nilai Akhir Mahasiswa
=====
1. Sinta (103112430047) = 87
2. Chilya (103112430053) = 85
3. Rere (103112430031) = 90

```

Deskripsi:

Program ini berfungsi untuk menghitung dan menampilkan nilai akhir mahasiswa berdasarkan nilai UTS, UAS, dan tugas. Data maksimal yang dapat dimasukkan adalah 10 mahasiswa. Program meminta pengguna menginput jumlah mahasiswa, lalu untuk setiap mahasiswa diminta memasukkan nama, NIM, nilai UTS, UAS, dan tugas.

Soal 2

pelajaran.h

```

#ifndef PELAJARAN_H
#define PELAJARAN_H

#include <string>
using namespace std;

struct pelajaran {
    string namaMapel;
    string kodeMapel;
};

pelajaran buat_pelajaran(string namaPelajaran, string
kodePelajaran);

void tampil_pelajaran(pelajaran pel);

```

```
#endif
```

pelajaran.cpp

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

pelajaran buat_pelajaran(string namaPelajaran, string
kodePelajaran) {
    pelajaran p;
    p.namaMapel = namaPelajaran;
    p.kodeMapel = kodePelajaran;
    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 namaPelajaran = "Struktur Data";
    string kodePelajaran = "STD";

    pelajaran pel = buat_pelajaran(namaPelajaran, kodePelajaran);

    tampil_pelajaran(pel);

    return 0;
}
```

Screenshots Output

```
PS D:\SEM 3\STRUKTUR DATA\Modul 3> cd "d:\SEM 3\STRUKTUR DATA\Modul 3\" ; if ($?) { g++ main.cpp -o main } ; if ($?) { .\main }
nama pelajaran : Struktur Data
nilai : STD
```

Deskripsi:

Program ini merupakan implementasi ADT (Abstract Data Type) bernama pelajaran yang digunakan untuk menyimpan dan menampilkan data mata pelajaran. File pelajaran.h berisi deklarasi struct dan prototype fungsi, sedangkan pelajaran.cpp berisi implementasi fungsi dan prosedurnya. Pada file main.cpp, data nama dan kode pelajaran dimasukkan ke dalam variabel, kemudian diproses melalui fungsi buat_pelajaran() untuk membuat objek pel, dan hasilnya ditampilkan menggunakan prosedur tampil_pelajaran().

Soal 3

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

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

void tukarArray(int A[3][3], int B[3][3], int i, int j) {
    int temp = A[i][j];
    A[i][j] = B[i][j];
    B[i][j] = temp;
}

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

int main() {
```

```
int A[3][3] = { {1,2,3}, {4,5,6}, {7,8,9} };
int B[3][3] = { {9,8,7}, {6,5,4}, {3,2,1} };
int *p1, *p2;
int x = 10, y = 20;

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

tukarArray(A, B, 1, 1);

cout << "\nSetelah tukar elemen:" << endl;
cout << "Array A:" << endl;
tampil(A);
cout << "Array B:" << endl;
tampil(B);

p1 = &x;
p2 = &y;
tukarPointer(p1, p2);

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

return 0;
}
```

Screenshots Output


```

PS D:\SEM 3\STRUKTUR DATA\Modul 3> cd "d:\SEM 3\STRUKTUR DATA\Modul 3\" ; if ($?) { g++ soal3.cpp -o soal3 } ; if ($?) { .\soal3 }
Array A:
1 2 3
4 5 6
7 8 9
Array B:
9 8 7
6 5 4
3 2 1

Setelah tukar elemen:
Array A:
1 2 3
4 5 6
7 8 9
Array B:
9 8 7
6 5 4
3 2 1

Setelah tukar pointer:
x = 20, y = 10

```

Deskripsi:

Program ini dibuat untuk menunjukkan penggunaan array 2 dimensi dan pointer dalam bahasa C++. Di dalamnya terdapat dua buah array berukuran 3×3 serta dua pointer yang menunjuk ke variabel bertipe integer. Program memiliki tiga prosedur utama, yaitu `tampil()` untuk menampilkan isi array, `tukarArray()` untuk menukar elemen tertentu antara dua array pada posisi yang sama, dan `tukarPointer()` untuk menukar nilai dari dua variabel yang ditunjuk oleh pointer. Melalui program ini, pengguna dapat melihat bagaimana data dalam array maupun nilai variabel dapat saling bertukar dengan memanfaatkan konsep fungsi, prosedur, dan pointer secara sederhana dan efisien.

C. Kesimpulan

Dari praktikum ini dapat disimpulkan bahwa penggunaan array dua dimensi dan pointer dalam bahasa C++ memungkinkan pengelolaan data menjadi lebih fleksibel dan efisien. Melalui implementasi fungsi dan prosedur seperti `tampil()`, `tukarArray()`, dan `tukarPointer()`, dapat dipahami cara menampilkan, menukar elemen antar-array, serta menukar nilai variabel menggunakan pointer. Praktikum ini juga memperkuat pemahaman mengenai konsep dasar struktur data, terutama dalam hal manipulasi data dan penerapan fungsi yang modular dalam pemrograman C++.

D. Referensi

https://en.wikipedia.org/wiki/Abstract_data_type

<https://www.cs.odu.edu/~zeil/cs330/latest/Public/implementingADTS/index.html>