

**LAPORAN PRAKTIKUM  
STRUKTUR DATA**

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



**Disusun Oleh :**

NAMA : Muhammad Luthfi Arrafi  
Ramadhani

NIM : 103112430043

**Dosen**

Fahrudin Mukti Wibowo

**PROGRAM STUDI S1 TEKNIK INFORMATIKA  
FAKULTAS INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO  
2025**

## A. Dasar Teori

Dasar teori pada modul ini membahas konsep Abstract Data Type (ADT), yaitu tipe data yang didefinisikan oleh pengguna beserta operasi-operasi dasarnya. ADT memisahkan antara definisi tipe data, implementasi fungsi, dan penggunaannya dalam program. Dalam penerapannya, ADT biasanya dibagi menjadi tiga file: file header (.h) yang berisi definisi struktur dan deklarasi fungsi, file implementasi (.cpp) yang berisi isi dari fungsi-fungsi tersebut, serta file utama (main.cpp) yang menjalankan program menggunakan fungsi dari ADT. Konsep ini membantu membuat program lebih terstruktur, mudah dibaca, dan memisahkan logika data dari alur utama program.

## 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
```

Deskripsi:

File mahasiswa.h ini berisi deklarasi fungsi. Struct mahasiswa menyimpan nim, nilai1, dan nilai2. Dua fungsi dideklarasikan: *inputMhs()* untuk input data, dan *rata2()* untuk menghitung rata-rata nilai. Terdapat baris *#ifndef*–*#endif* juga supaya mencegah file dibaca dua kali saat kompilasi.

### Guided 2

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

void inputMhs(mahasiswa &m)
{
    cout << "Input Nama = ";
    cin >> (m).nim;
    cout << "Input Nilai = ";
    cin >> (m).nilai1;
    cout << "Input Nilai2 = ";
    cin >> (m).nilai2;
}
```

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

Deskripsi:

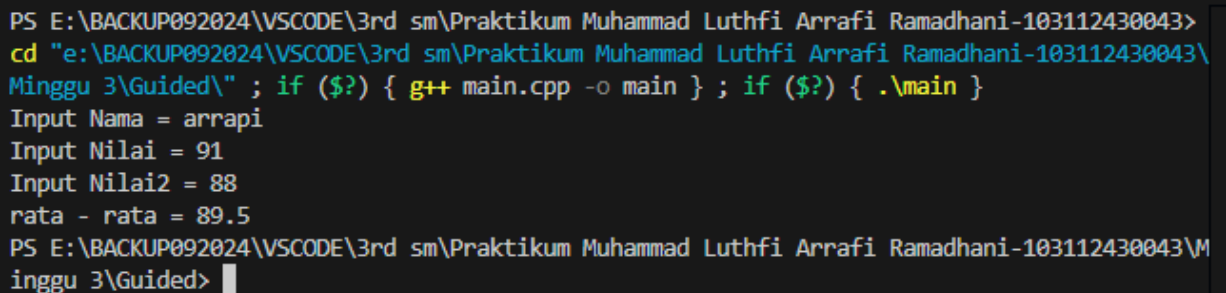
File mahasiswa.cpp ini berisi implementasi fungsi dari header. *inputMhs()* meminta user mengisi NIM dan dua nilai, lalu menyimpannya ke variabel *m.rata2()* menjumlahkan dua nilai mahasiswa dan membaginya dua untuk mendapat hasil rata-rata bertipe float.

### Guided 3

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

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

Screenshots Output



```
PS E:\BACKUP092024\VSCODE\3rd sm\Praktikum Muhammad Luthfi Arrafi Ramadhani-103112430043>
cd "e:\BACKUP092024\VSCODE\3rd sm\Praktikum Muhammad Luthfi Arrafi Ramadhani-103112430043\
Minggu 3\Guided\" ; if ($?) { g++ main.cpp -o main } ; if ($?) { .\main }
Input Nama = arrapi
Input Nilai = 91
Input Nilai2 = 88
rata - rata = 89.5
PS E:\BACKUP092024\VSCODE\3rd sm\Praktikum Muhammad Luthfi Arrafi Ramadhani-103112430043\M
inggu 3\Guided> |
```

Deskripsi:

File main.cpp ini merupakan program utama dari dua file yang telah dibuat sebelumnya (mahasiswa.h dan mahasiswa.cpp). Program membuat rata-rata dari dua nilai yang telah di inputkan. Variabel *mhs* memanggil *inputMhs(mhs)* untuk input data, lalu menampilkan hasil *rata2(mhs)* ke layar.

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

Unguided 1

```
// Muhammad Luthfi Arrafi Ramadhani
// 103112430043
// IF 12-06

#include <iostream>
using namespace std;

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

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

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

    cout << "Masukkan jumlah mahasiswa (max 10): ";
    cin >> n;

    for (int i = 0; i < n; i++) {
        cout << "\nData mahasiswa ke-" << i + 1 << endl;
        cout << "Nama : ";
        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 = hitungNilaiAkhir(mhs[i].uts, mhs[i].uas, mhs[i].tugas);
    }

    cout << "\n=== Data Mahasiswa ===\n";
    for (int i = 0; i < n; i++) {
        cout << "Nama: " << mhs[i].nama
            << ", NIM: " << mhs[i].nim
            << ", Nilai Akhir: " << mhs[i].nilaiAkhir << endl;
    }
}
```

```
    return 0;
}
```

## Screenshots Output

```
PS E:\BACKUP092024\VSCODE\3rd sm\Praktikum Muhammad Luthfi Arrafi Ramadhani-1031124
30043> cd "e:\BACKUP092024\VSCODE\3rd sm\Praktikum Muhammad Luthfi Arrafi Ramadhani
-103112430043\Minggu 3\Unguided\" ; if ($?) { g++ tugas1.cpp -o tugas1 } ; if ($?)
{ .\tugas1 }
Masukkan jumlah mahasiswa (max 10): 5

Data mahasiswa ke-1
Nama : Aldebarra
NIM : 105794
Nilai UTS : 84
Nilai UAS : 87
Nilai Tugas : 91

Data mahasiswa ke-2
Nama : Rangga
NIM : 108937
Nilai UTS : 77
Nilai UAS : 81
Nilai Tugas : 92

Data mahasiswa ke-3
Nama : Bagaskara
NIM : 106673
Nilai UTS : 75
Nilai UAS : 78
Nilai Tugas : 83

Data mahasiswa ke-4
Nama : Nalendra
NIM : 105982
Nilai UTS : 89
Nilai UAS : 87
Nilai Tugas : 94

Data mahasiswa ke-5
Nama : Adrian
NIM : 104653
Nilai UTS : 86
Nilai UAS : 81
Nilai Tugas : 84

=== Data Mahasiswa ===
Nama: Aldebarra, NIM: 105794, Nilai Akhir: 87.3
Nama: Rangga, NIM: 108937, Nilai Akhir: 83.1
Nama: Bagaskara, NIM: 106673, Nilai Akhir: 78.6
Nama: Nalendra, NIM: 105982, Nilai Akhir: 89.7
Nama: Adrian, NIM: 104653, Nilai Akhir: 83.4
PS E:\BACKUP092024\VSCODE\3rd sm\Praktikum Muhammad Luthfi Arrafi Ramadhani-103112430043\Minggu 3\Unguided> |
```

## Deskripsi:

Program tersebut merupakan program untuk memasukkan dan menampilkan data beberapa mahasiswa beserta nilai akhirnya. Di dalamnya terdapat *struct Mahasiswa* yang menyimpan nama, NIM, nilai UTS, UAS, tugas, dan nilai akhir. Fungsi *hitungNilaiAkhir()* dipakai untuk menghitung nilai akhir dengan rumus  $0.3 \times \text{uts} + 0.4 \times \text{uas} + 0.3 \times \text{tugas}$ . Pada bagian utama program, pengguna diminta memasukkan jumlah mahasiswa lalu mengisi data satu per satu. Setelah semua data dimasukkan, program menghitung nilai akhir setiap mahasiswa dan menampilkan hasilnya ke layar.

## Unguided 2

(pelajaran.h)

```
// Muhammad Luthfi Arrafi Ramadhani
// 103112430043
// IF 12-06

#ifndef PELAJARAN_H_INCLUDED
#define PELAJARAN_H_INCLUDED
#include <string>
using namespace std;

struct Pelajaran {
    string namaMapel;
    string kodeMapel;
};

Pelajaran create_pelajaran(string namapel, string kodepel);
void tampil_pelajaran(Pelajaran pel);
#endif
```

Deskripsi:

File pelajaran.h ini berisi definisi struct dan deklarasi fungsi. Di dalamnya terdapat *struct Pelajaran* yang menyimpan dua data yaitu namaMapel dan kodeMapel. Terdapat juga deklarasi dua fungsi yaitu *create\_pelajaran()* untuk membuat data pelajaran baru dan *tampil\_pelajaran()* untuk menampilkan data tersebut.

(pelajaran.cpp)

```
// Muhammad Luthfi Arrafi Ramadhani
// 103112430043
// IF 12-06

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

Pelajaran create_pelajaran(string namapel, string kodepel) {
    Pelajaran p;
    p.namaMapel = namapel;
    p.kodeMapel = kodepel;
    return p;
}

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

Deskripsi:

File pelajaran.cpp ini berisi implementasi fungsi yang telah dideklarasikan di pelajaran.h. Fungsi *create\_pelajaran()* digunakan untuk mengisi nilai namaMapel dan kodeMapel, kemudian mengembalikannya sebagai objek bertipe Pelajaran. Sedangkan *tampil\_pelajaran()* digunakan untuk menampilkan data pelajaran ke layar.

(main.cpp)

```
// Muhammad Luthfi Arrafi Ramadhani
// 103112430043
// IF 12-06

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

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

    Pelajaran pel = create_pelajaran(namelapel, kodepel);
    tampil_pelajaran(pel);

    return 0;
}
```

Screenshots Output

```
PS E:\BACKUP092024\VSCODE\3rd sm\Praktikum Muhammad Luthfi Arrafi Ramadhani-103112430043\Minggu 3\Unguided> g++ main.cpp pelajaran.cpp -o main
PS E:\BACKUP092024\VSCODE\3rd sm\Praktikum Muhammad Luthfi Arrafi Ramadhani-103112430043\Minggu 3\Unguided> ./main
Nama Mata Kuliah : Struktur Data
Kode Mata Kuliah : STD
PS E:\BACKUP092024\VSCODE\3rd sm\Praktikum Muhammad Luthfi Arrafi Ramadhani-103112430043\Minggu 3\Unguided> |
```

Deskripsi:

File main.cpp ini merupakan program utama yang menjalankan keseluruhan proses sebelumnya. Program membuat variabel *pel* bertipe *Pelajaran*, lalu memanggil fungsi *create\_pelajaran()* untuk mengisi data dan *tampil\_pelajaran()* untuk menampilkannya.

### Unguided 3

```
// Muhammad Luthfi Arrafi Ramadhani
// 103112430043
// IF 12-06

#include <iostream>
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 tukarArray(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 *a, int *b) {
    int temp = *a;
    *a = *b;
    *b = temp;
}

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

    int B[3][3] = {
        {10, 11, 12},
        {13, 14, 15},
        {16, 17, 18}
    };

    int x = 5, y = 10;
    int *p1 = &x;
    int *p2 = &y;

    cout << "Array A sebelum ditukar:\n";
    tampilArray(A);
    cout << "\nArray B sebelum ditukar:\n";
    tampilArray(B);
```



```

    tukarArray(A, B, I, I);

    cout << "\nSetelah pertukaran elemen [1][1]:\n";
    cout << "Array A:\n";
    tampilArray(A);
    cout << "Array B:\n";
    tampilArray(B);

    cout << "\nNilai sebelum tukar pointer: x = " << x << ", y = " << y << endl;
    tukarPointer(p1, p2);
    cout << "Nilai setelah tukar pointer: x = " << x << ", y = " << y << endl;

    return 0;
}

```

### Screenshots Output

```

PS E:\BACKUP092024\VSCODE\3rd sm\Praktikum Muhammad Luthfi Arrafi Ramadhani-103112430043> cd "e:\BACKUP092024\VSCODE
\3rd sm\Praktikum Muhammad Luthfi Arrafi Ramadhani-103112430043\Minggu 3\Unguided\" ; if ($?) { g++ tugas3.cpp -o tu
gas3 } ; if ($?) { .\tugas3 }
Array A sebelum ditukar:
1 2 3
4 5 6
7 8 9

Array B sebelum ditukar:
10 11 12
13 14 15
16 17 18

Setelah pertukaran elemen [1][1]:
Array A:
1 2 3
4 14 6
7 8 9
Array B:
10 11 12
13 5 15
16 17 18

Nilai sebelum tukar pointer: x = 5, y = 10
Nilai setelah tukar pointer: x = 10, y = 5
PS E:\BACKUP092024\VSCODE\3rd sm\Praktikum Muhammad Luthfi Arrafi Ramadhani-103112430043\Minggu 3\Unguided>

```

### Deskripsi:

Program di atas berfungsi untuk menampilkan dan menukar isi dua array 2D berukuran 3×3 serta menukar nilai dua variabel dengan pointer. Array A dan B diisi langsung di program, lalu fungsi *tampilArray()* digunakan untuk menampilkan isinya, *tukarArray()* untuk menukar elemen pada posisi tertentu, dan *tukarPointer()* untuk menukar nilai dua variabel dengan pointer. Setelah dijalankan, program menampilkan hasil sebelum dan sesudah pertukaran, baik untuk array maupun variabel x dan y.

#### D. Kesimpulan

Dari ketiga program di atas dapat disimpulkan bahwa penerapan Abstract Data Type (ADT) membantu membuat program lebih terstruktur dan mudah dipahami. Pada program pertama, konsep struktur dan fungsi digunakan untuk mengolah data mahasiswa dan menghitung nilai akhir. Program kedua menerapkan ADT secara lengkap dengan pemisahan file header, implementasi, dan program utama pada contoh data pelajaran. Sementara itu, program ketiga memperdalam pemahaman tentang fungsi, array 2D, dan pointer melalui proses penukaran data. Secara keseluruhan, ketiga program ini menunjukkan pentingnya penggunaan ADT dan fungsi terpisah dalam membuat program yang rapi, efisien, dan mudah dikelola.

#### E. Referensi

- Meyer, B. (2022). *Abstract data types*. In *Handbook of Requirements and Business Analysis* (pp. 205–224). Springer. [https://link.springer.com/chapter/10.1007/978-3-031-06739-6\\_10](https://link.springer.com/chapter/10.1007/978-3-031-06739-6_10)
- Malek, J. (2024). *An implementation and experimental comparison of dynamic ordered sets (arXiv:2411.01090)*. arXiv. <https://arxiv.org/abs/2411.01090>