

**LAPORAN PRAKTIKUM  
STRUKTUR DATA**

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



**Disusun Oleh :**  
NAMA : Zidane Aji Noegroho  
NIM : 103112430006

**Dosen**  
FAHRUDIN MUKTI WIBOWO

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

## A. Dasar Teori

Algoritma merupakan fondasi dalam pembuatan program komputer. Secara sederhana, algoritma adalah serangkaian langkah logis dan sistematis yang disusun untuk menyelesaikan suatu masalah. C++, berfungsi sebagai alat untuk mengimplementasikan algoritma tersebut agar dapat dimengerti dan dieksekusi oleh komputer. C++ sering digunakan sebagai bahasa pengantar untuk mempelajari konsep pemrograman dasar karena strukturnya yang terorganisir dan kemampuannya untuk menangani operasi tingkat rendah.

Untuk membangun logika dalam program sesuai dengan algoritma yang dirancang, C++ menyediakan struktur kontrol, yang terbagi menjadi dua jenis utama:

Struktur Percabangan (Conditional): Digunakan untuk pengambilan keputusan, di mana program akan menjalankan blok kode tertentu jika suatu kondisi terpenuhi. Struktur ini mencakup if-else untuk mengevaluasi kondisi boolean dan switch-case untuk memilih blok kode berdasarkan nilai dari sebuah variabel.

Struktur Perulangan (Looping): Digunakan untuk mengeksekusi blok kode yang sama secara berulang kali selama kondisi tertentu masih terpenuhi. C++ menyediakan tiga jenis perulangan utama: for, while, dan do-while, yang masing-masing memiliki karakteristik penggunaan yang spesifik dalam implementasi algoritma.

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

### Guided 1

Mahasiswa.cpp

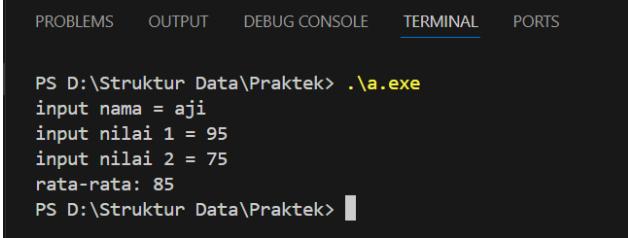
```
#include "mahasiswa.h"
#include <iostream>
using namespace std;

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 (m.nilai1 + m.nilai2) / 2;  
}
```

## Screenshots Output



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS  
  
PS D:\Struktur Data\Praktek> .\a.exe  
input nama = aji  
input nilai 1 = 95  
input nilai 2 = 75  
rata-rata: 85  
PS D:\Struktur Data\Praktek>
```

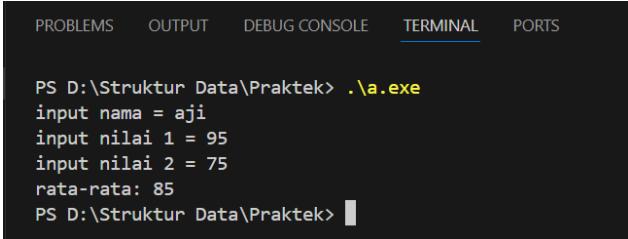
Deskripsi:

berfungsi sebagai file implementasi dari fungsi yang dideklarasikan di header mahasiswa.h. Di dalamnya terdapat fungsi inputMhs yang meminta pengguna untuk memasukkan data mahasiswa berupa NIM, nilai1, dan nilai2. Nilai-nilai ini kemudian disimpan dalam variabel bertipe mahasiswa yang dikirim lewat referensi. Fungsi rata2 menerima data mahasiswa sebagai parameter dan mengembalikan hasil perhitungan rata-rata dari dua nilai yang dimasukkan sebelumnya.

## 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
```

## Screenshots Output



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS  
  
PS D:\Struktur Data\Praktek> .\a.exe  
input nama = aji  
input nilai 1 = 95  
input nilai 2 = 75  
rata-rata: 85  
PS D:\Struktur Data\Praktek>
```

Deskripsi:

merupakan file header yang mendefinisikan struktur mahasiswa dan mendeklarasikan dua fungsi, yaitu inputMhs dan rata2. Struktur mahasiswa memiliki tiga anggota data:

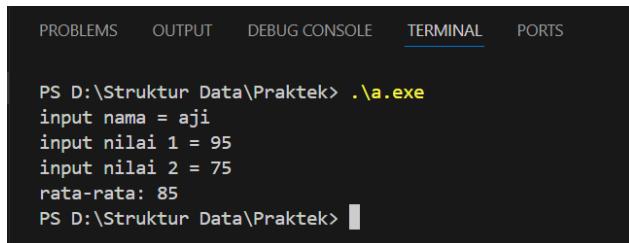
nim bertipe karakter array untuk menyimpan nomor induk mahasiswa, serta nilai1 dan nilai2 yang bertipe integer. File ini digunakan untuk memberikan deklarasi agar bisa dipanggil di file lain tanpa menduplikasi kode.

### Main.cpp

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

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

### Screenshots Output



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\Struktur Data\Praktek> .\a.exe
input nama = aji
input nilai 1 = 95
input nilai 2 = 75
rata-rata: 85
PS D:\Struktur Data\Praktek>
```

Deskripsi:

program utama yang menjalankan keseluruhan logika. File ini meng-include mahasiswa.h untuk mengenali struktur dan fungsi yang digunakan. Di dalam fungsi main, dibuat sebuah variabel mhs bertipe mahasiswa, lalu dipanggil fungsi inputMhs(mhs) untuk mengisi data mahasiswa. Setelah itu, fungsi rata2(mhs) dipanggil untuk menghitung rata-rata nilai dan hasilnya ditampilkan ke layar menggunakan cout.

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

### Unguided 1 (Main.cpp)

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

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

```

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

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

    if (n > 10) n = 10;

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

    cout << "\nData Mahasiswa:\n";
    for (int i = 0; i < n; i++) {
        cout << "\nNama : " << mhs[i].nama;
        cout << "\nNIM : " << mhs[i].nim;
        cout << "\nUTS : " << mhs[i].uts;
        cout << "\nUAS : " << mhs[i].uas;
        cout << "\nTugas : " << mhs[i].tugas;
        cout << "\nNilai Akhir : " << mhs[i].nilaiAkhir << endl;
    }
    return 0;
}

```

Unguided 1 (ugd.cpp)

```

#include "ugd.h"

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

```

Unguided 1 (ugd.h)

```
#ifndef FUNGSI_H
#define FUNGSI_H
float hitungNilaiAkhir(float uts, float uas, float tugas);
#endif
```

## Screenshots Output

```
Data mahasiswa ke-1
Nama : aji
NIM : 183112430006
Nilai UTS : 98
Nilai UAS : 85
Nilai Tugas : 80

Data mahasiswa ke-2
Nama : zidane
NIM : 183112430009
Nilai UTS : 95
Nilai UAS : 75
Nilai Tugas : 90

Data Mahasiswa:
Nama : aji
NIM : 183112430006
UTS : 98
UAS : 85
Tugas : 80
Nilai Akhir : 85

Nama : zidane
NIM : 183112430009
UTS : 95
UAS : 75
Tugas : 90
Nilai Akhir : 85.5
PS D:\Struktur Data\Praktek\modul 3\UNGUIDED1>
```

## Deskripsi:

Untuk menghitung dan menampilkan nilai akhir mahasiswa. File header ugd.h berisi deklarasi fungsi hitungNilaiAkhir, sedangkan file implementasinya mendefinisikan fungsi tersebut yang menghitung nilai akhir berdasarkan bobot UTS 30%, UAS 40%, dan tugas 30%. File utama berisi program main yang menggunakan fungsi tersebut. Di dalamnya, dibuat array Mahasiswa untuk menampung data beberapa mahasiswa, lalu program meminta input berupa nama, NIM, nilai UTS, UAS, dan tugas. Setelah itu, nilai akhir tiap mahasiswa dihitung dengan memanggil fungsi hitungNilaiAkhir, dan hasilnya ditampilkan ke layar. Ketiga bagian bekerja bersama: header untuk deklarasi, implementasi untuk logika perhitungan, dan main untuk menjalankan program.

## Unguided 2 (main.cpp)

```
#include <iostream>
#include "ugd2.h"
#include "ugd2.cpp"
using namespace std;

int main() {
    string namapel = "Struktur Data";
    string kodemapel = "STD";
    pelajaran pel = create_pelajaran(namapel, kodemapel);
    tampil_pelajaran(pel);

    return 0;
}
```

```
}
```

### Unguided 2 (ugd2.cpp)

```
#include <iostream>
#include "ugd2.h"
using namespace std;

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

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

### Unguided 2 (ugd.h)

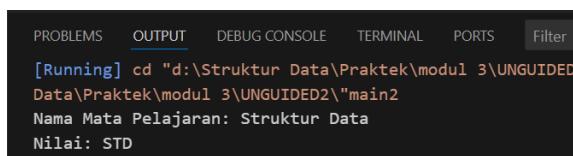
```
#include <string>
#ifndef PELAJARAN_H_INCLUDED
#define PELAJARAN_H_INCLUDED
using namespace std;

struct pelajaran {
    string namaMapel;
    string kodeMapel;
};

pelajaran create_pelajaran(string namapel, string kodemapel);

void tampil_pelajaran(pelajaran pel);
#endif
```

### Screenshots Output



```
[Running] cd "d:\Struktur Data\Praktek\modul 3\UNGUIDED Data\Praktek\modul 3\UNGUIDED2\"main2
Nama Mata Pelajaran: Struktur Data
Nilai: STD
```

### Deskripsi:

program yang membuat dan menampilkan data mata pelajaran. File header ugd2.h berisi definisi struktur pelajaran dengan atribut namaMapel dan kodeMapel, serta deklarasi fungsi create\_pelajaran dan tampil\_pelajaran. File implementasi ugd2.cpp berisi definisi kedua fungsi tersebut, di mana create\_pelajaran membuat objek pelajaran

berdasarkan input nama dan kode mata pelajaran, sedangkan tampil\_pelajaran menampilkan data tersebut ke layar. File utama memanggil kedua fungsi itu: pertama membuat objek pelajaran dengan nilai yang sudah ditentukan, lalu menampilkan hasilnya. Semua file bekerja bersama — header untuk deklarasi, implementasi untuk logika fungsi, dan main untuk menjalankan program.

### Unguided 3 (main3.cpp)

```
#include <iostream>
#include "ugd3.h"
using namespace std;

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 *ptr1, *ptr2;

    cout << "Array A awal:" << endl;
    tampilArray(A);
    cout << "\nArray B awal:" << endl;
    tampilArray(B);

    tukarPosisi(A, B, 1, 2);

    cout << "\nSetelah tukar posisi [1][2]:" << endl;
    cout << "Array A:" << endl; tampilArray(A);
    cout << "\nArray B:" << endl;
    tampilArray(B);

    int x = 100, y = 200;
    ptr1 = &x;
    ptr2 = &y;

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

    tukarPointer(ptr1, ptr2);
```

```

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

    return 0;
}

```

### Unguided 3 (ugd3.cpp)

```

#include <iostream>
#include "ugd3.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] << "\t";
        }
        cout << endl;
    }
}

void tukarPosisi(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;
}

```

### Unguided 3 (ugd3.h)

```

#ifndef FUNGSI_H
#define FUNGSI_H

void tampilArray(int arr[3][3]);
void tukarPosisi(int arr1[3][3], int arr2[3][3], int baris, int kolom);
void tukarPointer(int *p1, int *p2);
#endif

```

### Screenshots Output

```
Array A awal:
1      2      3
4      5      6
7      8      9

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

Setelah tukar posisi [1][2]:
Array A:
1      2      3
4      5      15
7      8      9

Array B:
10     11     12
13     14     6
16     17     18

Sebelum tukar pointer:
x = 100, y = 200
Setelah tukar pointer:
x = 200, y = 100
PS D:\Struktur Data\Praktek\modul 3> █
```

Deskripsi:

program yang menampilkan isi array dua dimensi, menukar elemen antar array, dan menukar nilai menggunakan pointer. File header ugd3.h berisi deklarasi tiga fungsi, yaitu tampilArray untuk menampilkan isi array  $3 \times 3$ , tukarPosisi untuk menukar elemen array pada posisi tertentu antara dua array, dan tukarPointer untuk menukar nilai yang ditunjuk oleh dua pointer. File implementasi mendefinisikan ketiga fungsi tersebut sesuai dengan deklarasinya. File utama berisi program main yang mendeklarasikan dua array  $3 \times 3$  (A dan B), menampilkan nilainya, lalu menukar elemen tertentu menggunakan tukarPosisi. Setelah itu, program juga mendemonstrasikan pertukaran nilai dua variabel menggunakan pointer melalui fungsi tukarPointer.

#### D. Kesimpulan

melalui praktikum Modul III ini saya dapat memahami konsep Abstract Data Type (ADT) serta penerapannya dalam pemrograman C++ melalui pemisahan antara deklarasi dan implementasi menggunakan file header dan file sumber. Dari kegiatan praktikum yang saya lakukan, saya belajar bagaimana membangun program yang terstruktur dengan memanfaatkan fungsi, struktur, array, dan pointer untuk mengelola serta memanipulasi data secara efisien. Praktikum ini memberikan saya pemahaman tentang pentingnya ADT dalam membuat program yang modular, mudah dikelola, dan mendukung penerapan prinsip pemrograman yang bersih serta terorganisir.

#### E. Referensi

Kaswar, A. B., & Zain, S. G. (2021). Mudah Belajar Pemrograman Dasar C++. Syiah Kuala University Press.

Hanief, S., Jepriana, I. W., & Kom, S. (2020). Konsep Algoritme dan Aplikasinya dalam Bahasa Pemrograman C++. Penerbit Andi.

Imamuddin, A., & Sobarnas, M. A. (2021). PEMBELAJARAN JARAK JAUH PEMROGRAMAN DASAR MENGGUNAKAN BAHASA C++ UNTUK UMUM:

SEBUAH PROGRAM PENGABDIAN KEPADA MASYARAKAT. BEMAS: Jurnal  
Bermasyarakat, 1(2), 59-67.