

**LAPORAN PRAKTIKUM STRUKTUR DATA  
MODUL 3  
ABSTRACT DATA TYPE (ADT)**



**Disusun Oleh:  
Dwi Candra Pratama – 2211104035**

**FAKULTAS iNFORMATIKA PROGRAM STUDI S1  
REKAYASA  
PERANGKAT LUNAK TELKOM UNIVERSITY  
PURWOKERTO**

# GUIDED

## 1. TUJUAN

Memahami konsep Abstract Data Type (ADT) dan penggunaannya dalam pemrograman.

- a) Abstract Data Type (ADT) dalam C++ adalah sebuah konsep yang memisahkan spesifikasi dari implementasi. ADT memungkinkan kita untuk mendefinisikan struktur data dan operasi-operasi yang terkait dengannya tanpa memikirkan bagaimana operasi tersebut diimplementasikan.
1. ADT Waktu: Bisa terdiri dari dua ADT lain, yaitu JAM dan DATE.
  2. Garis: Terdiri dari dua ADT POINT.
  3. Segi Empat: Terdiri dari dua pasangan POINT: (Top, Left) dan (Bottom, Right).

### PRIMITIF ADT

Primitif adalah operasi dasar yang digunakan dalam ADT. Dikelompokkan menjadi:

1. Konstruktor/Kreator: Fungsi pembentuk nilai objek bertipe ADT. Misalnya, fungsi Make untuk menciptakan objek.
2. Selector: Mengakses komponen dari ADT. Biasanya dengan nama awalan Get seperti GetHour() untuk ADT JAM.
3. Prosedur Pengubah: Mengubah nilai komponen ADT, sering kali juga diawali dengan Set.
4. Validator: Memeriksa apakah sebuah nilai memenuhi syarat untuk membentuk tipe ADT.
5. Destruktor/Dealokator: Untuk menghancurkan atau menghapus objek, termasuk memori yang digunakan.
6. Baca/Tulis: Mengelola input/output dari ADT.
7. Operator Relasional: Untuk membandingkan nilai dari ADT, seperti lebih besar atau sama dengan.
8. Operator Aritmatika: Untuk operasi matematika (misalnya penjumlahan, pengurangan) pada ADT.
9. Konversi Tipe: Mengonversi ADT menjadi tipe dasar atau sebaliknya.

### Implementasi ADT

ADT biasanya diimplementasikan dalam dua modul:

1. Header File (.h): Definisi spesifikasi tipe dan primitif, meliputi:
  - Spesifikasi Type: Mengikuti aturan bahasa (C++ dalam hal ini).
  - Spesifikasi Primitif:
    - Fungsi: Nama, domain (parameter), range (return type), prekondisi.
    - Prosedur: State awal dan akhir serta proses yang dilakukan.
2. Body File (.cpp): Implementasi dari primitif, termasuk realisasi fungsi dan prosedur. Biasanya, menggunakan selector dan konstruktor.

Berikut adalah implementasi dari program ADT pada saat praktikum:

```
#include <iostream>

using namespace std;

struct mahasiswa{
    char nim[10];
    int nilai1, nilai2;
};

Tabnine | Edit | Test | Explain | Document | Ask
void inputMhs(mahasiswa &m);
Tabnine | Edit | Test | Explain | Document | Ask
float rata2(mahasiswa m);

Tabnine | Edit | Test | Explain | Document | Ask
int main(){
    mahasiswa mhs;
    inputMhs(mhs);
    cout << "rata-rata = " << rata2(mhs);
    return 0;
}

Tabnine | Edit | Test | Explain | Document | Ask
void inputMhs(mahasiswa &m){
    cout << "Input nim = ";
    cin >> (m).nim;
    cout << "Input nilai 1 = ";
    cin >> (m).nilai1;
    cout << "Input nilai 2 = ";
    cin >> (m).nilai2;
}

Tabnine | Edit | Test | Explain | Document | Ask
float rata2(mahasiswa m){
    return(m.nilai1 + m.nilai2)/2;
}
```

```
PS D:\Semester 5\Struktur Data\Praktikum Struktur Data\Pertemuan 3> cd 'd:\Semester 5\Struktur Data\Praktikum Struktur Data\Pertemuan 3\output'
PS D:\Semester 5\Struktur Data\Praktikum Struktur Data\Pertemuan 3\output> & .\Modul3.exe
Input nim = 2211104035
Input nilai 1 = 97
Input nilai 2 = 87
rata-rata = 92
PS D:\Semester 5\Struktur Data\Praktikum Struktur Data\Pertemuan 3\output> |
```

## UNGUIDED

1. Buat program yang dapat menyimpan data mahasiswa (max. 10) ke dalam sebuah array dengan field nama, nim, uts, uas, tugas, dan nilai akhir. Nilai akhir diperoleh dari FUNGSI dengan rumus  $0.3 \times \text{uts} + 0.4 \times \text{uas} + 0.3 \times \text{tugas}$ .

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

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

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

int main() {
    Mahasiswa mhs[10];
    int jumlah_mhs = 0;

    while (jumlah_mhs < 10) {
        cout << "Masukkan data mahasiswa ke-" << jumlah_mhs + 1 << ": " << endl;
        cout << "Nama: ";
        getline(cin, mhs[jumlah_mhs].nama);
        cout << "NIM: ";
        getline(cin, mhs[jumlah_mhs].nim);
        cout << "Nilai UTS: ";
        cin >> mhs[jumlah_mhs].uts;
        cout << "Nilai UAS: ";
        cin >> mhs[jumlah_mhs].uas;
        cout << "Nilai Tugas: ";
        cin >> mhs[jumlah_mhs].tugas;

        mhs[jumlah_mhs].nilai_akhir = hitungNilaiAkhir(mhs[jumlah_mhs].uts, mhs[jumlah_mhs].uas, mhs[jumlah_mhs].tugas);

        jumlah_mhs++;

        char lanjut;
        cout << "Lanjut input data? (y/n): ";
        cin >> lanjut;
        cin.ignore();
        if (lanjut != 'y' && lanjut != 'Y') break;
    }

    cout << "\nData Mahasiswa: " << endl;
    for (int i = 0; i < jumlah_mhs; i++) {
        cout << i+1 << ". " << mhs[i].nama << " (NIM: " << mhs[i].nim << ") " << endl;
        cout << "    Nilai Akhir: " << mhs[i].nilai_akhir << endl;
    }

    return 0;
}
```

```
PS D:\Semester 5\Struktur Data\Praktikum Struktur Data\Pertemuan 3> cd 'd:\Semester 5\Struktur Data\Prak
PS D:\Semester 5\Struktur Data\Praktikum Struktur Data\Pertemuan 3\UNGUIDED\output> & .\Latihan1.exe'
Masukkan data mahasiswa ke-1:
Nama: Anak Lanang
NIM: 2211104035
Nilai UTS: 87
Nilai UAS: 90
Nilai Tugas: 85
Lanjut input data? (y/n): y
Masukkan data mahasiswa ke-2:
Nama: Mas Can
NIM: 2211107017
Nilai UTS: 90
Nilai UAS: 88
Nilai Tugas: 76
Lanjut input data? (y/n): y
Masukkan data mahasiswa ke-3:
Nama: kaji ishaq
NIM: 4422201063
Nilai UTS: 77
Nilai UAS: 86
Nilai Tugas: 91
Lanjut input data? (y/n): y
Masukkan data mahasiswa ke-4:
Nama: Entus
NIM: 5533307082
Nilai UTS: 77
Nilai UAS: 89
Nilai Tugas: 84
Lanjut input data? (y/n): n

Data Mahasiswa:
1. Anak Lanang (NIM: 2211104035)
   Nilai Akhir: 87.6
2. Mas Can (NIM: 2211107017)
   Nilai Akhir: 85
3. kaji ishaq (NIM: 4422201063)
   Nilai Akhir: 84.8
4. Entus (NIM: 5533307082)
   Nilai Akhir: 83.9
PS D:\Semester 5\Struktur Data\Praktikum Struktur Data\Pertemuan 3\UNGUIDED\output>
```

Dari data tersebut bahwa output aktual akan tergantung pada data yang dimasukkan oleh pengguna. Program ini memungkinkan pengguna untuk memasukkan data hingga 10 mahasiswa, tetapi dalam hal ini saya hanya memasukkan data untuk 4 mahasiswa. Karena agar bisa di Screenshoot.

2. Buatlah ADT pelajaran sebagai berikut di dalam file “pelajaran.h”:

```
tipe pelajaran <
    namaMapel : string
    kodeMapel : string
>
fungsi create_pelajaran( namapel : string, kodepel : string ) → pelajaran
prosedur tampil_pelajaran( pel : pelajaran )
```

Buatlah implementasi ADT pelajaran pada file “pelajaran.cpp”  
Cobalah hasil implementasi ADT pada file “main.cpp”

```
using namespace std;
int main(){
    string namapel = "Struktur Data";
    string kodepel = "STD";
    pelajaran pel = create_pelajaran(namapel,kodepel);
    tampil_pelajaran(pel);
    return 0;
}
```

Contoh *output* hasil:

```
nama pelajaran : Struktur Data
nilai : STD
```

Main.cpp

```
#include "Pelajaran.h"
#include <iostream>
#include "Pelajaran.cpp"

using namespace std;

Tabnine | Edit | Test | Explain | Document | Ask
int main()
{
    string namapel = "Struktur Data";
    string kodepel = "STD";
    pelajaran pel = create_pelajaran(namapel, kodepel);
    tampil_pelajaran(pel);

    return 0;
}
```

## Pelajaran.cpp

```
#include "Pelajaran.h"
#include <iostream>

Tabnine | Edit | Test | Explain | Document | Ask
pelajaran create_pelajaran(std::string namapel, std::string kodepel) {
    pelajaran pel;
    pel.namaMpel = namapel;
    pel.kodeMpel = kodepel;
    return pel;
}

Tabnine | Edit | Test | Explain | Document | Ask
void tampil_pelajaran(pelajaran pel) {
    std::cout << "nama pelajaran : " << pel.namaMpel << std::endl;
    std::cout << "kode : " << pel.kodeMpel << std::endl;
}
```

## Pelajaran.h

```
#ifndef PELAJARAN_H
#define PELAJARAN_H

#include <string>

struct pelajaran {
    std::string namaMpel;
    std::string kodeMpel;
};

Tabnine | Edit | Test | Explain | Document | Ask
pelajaran create_pelajaran(std::string namapel, std::string kodepel);
Tabnine | Edit | Test | Explain | Document | Ask
void tampil_pelajaran(pelajaran pel);

#endif // PELAJARAN_H
```

Outputnya:

```
PS D:\Semester5\StrukturData\PraktikumStrukturData\Pertemuan3\UNGUIEDED\No2\output> & .\'main.exe'
nama pelajaran : Struktur Data
kode : STD
```

3. Buatlah program dengan ketentuan :

- 2 buah array 2D integer berukuran 3x3 dan 2 buah pointer integer
- fungsi/prosedur yang menampilkan isi sebuah array integer 2D
- fungsi/prosedur yang akan menukarkan isi dari 2 array integer 2D pada posisi tertentu
- fungsi/prosedur yang akan menukarkan isi dari variabel yang ditunjuk oleh 2 buah Pointer

SourceCode:

```

#include <iostream>
using namespace std;

// Fungsi untuk menampilkan isi array 2D
Tabnine | Edit | Test | Explain | Document | Ask
void tampilkanArray(int arr[3][3]) {
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            cout << arr[i][j] << " ";
        }
        cout << endl;
    }
    cout << endl;
}

// Fungsi untuk menukar isi dari 2 array integer 2D pada posisi tertentu
Tabnine | Edit | Test | Explain | Document | Ask
void tukarArrayPosisi(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;
}

// Fungsi untuk menukar isi dari variabel yang ditunjuk oleh 2 buah pointer
Tabnine | Edit | Test | Explain | Document | Ask
void tukarPointer(int* ptr1, int* ptr2) {
    int temp = *ptr1;
    *ptr1 = *ptr2;
    *ptr2 = temp;
}

Tabnine | Edit | Test | Explain | Document | Ask
int main() {
    // Membuat 2 buah array 2D integer berukuran 3x3
    int array1[3][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
    int array2[3][3] = {{10, 11, 12}, {13, 14, 15}, {16, 17, 18}};

    // Membuat 2 buah pointer integer
    int a = 100, b = 200;
    int *ptr1 = &a, *ptr2 = &b;

    cout << "Array 1 awal:" << endl;
    tampilkanArray(array1);

    cout << "Array 2 awal:" << endl;
    tampilkanArray(array2);

    cout << "Menukar elemen pada posisi [1][1]" << endl;
    tukarArrayPosisi(array1, array2, 1, 1);

    cout << "Array 1 setelah penukaran:" << endl;
    tampilkanArray(array1);

    cout << "Array 2 setelah penukaran:" << endl;
    tampilkanArray(array2);

    cout << "Nilai pointer awal: *ptr1 = " << *ptr1 << ", *ptr2 = " << *ptr2 << endl;
    tukarPointer(ptr1, ptr2);
    cout << "Nilai pointer setelah penukaran: *ptr1 = " << *ptr1 << ", *ptr2 = " << *ptr2 << endl;

    return 0;
}

```

## Outputnya:

```

PS D:\Semester5\StrukturData\PraktikumStrukturData\Pertemuan3> cd 'd:\Semester5\StrukturData\PraktikumStrukturData\Pertemuan3\UNGUIEDED\No3\output'
PS D:\Semester5\StrukturData\PraktikumStrukturData\Pertemuan3\UNGUIEDED\No3\output> .\latihan3.exe
Array 1 awal:
1 2 3
4 5 6
7 8 9

Array 2 awal:
10 11 12
13 14 15
16 17 18

Menukar elemen pada posisi [1][1]
Array 1 setelah penukaran:
1 2 3
4 14 6
7 8 9

Array 2 setelah penukaran:
10 11 12
13 5 15
16 17 18

Nilai pointer awal: *ptr1 = 100, *ptr2 = 200
Nilai pointer setelah penukaran: *ptr1 = 200, *ptr2 = 100
PS D:\Semester5\StrukturData\PraktikumStrukturData\Pertemuan3\UNGUIEDED\No3\output>

```