

**LAPORAN PRAKTIKUM
STRUKTUR DATA**

**MODUL III
PENGENALAN CODE BLOCKS**



Disusun Oleh :

NAMA : Rikza Nur Zaki

NIM : 103112430030

Dosen

Fahrudin

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

A. Dasar Teori

Abstract Data Type (ADT) merupakan suatu model atau tipe data yang didefinisikan oleh perilaku (operasi-operasi yang dapat dilakukan terhadapnya), bukan oleh implementasinya. ADT berfokus pada apa yang dilakukan suatu tipe data, bukan bagaimana cara melakukannya. Type (tipe data), yaitu struktur data atau himpunan nilai yang mungkin dimiliki. Primitive operations (operasi dasar), yaitu fungsi dan prosedur yang dapat diterapkan pada tipe tersebut.

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

Guided 1

```
C mahasiswa.h > ...
1  #ifndef MAHASISWA_H_INCLUDED
2  #define MAHASISWA_H_INCLUDED
3  struct mahasiswa
4  {
5      char nim[10];
6      int nilai1, nilai2;
7  };
8  void inputMahasiswa(mahasiswa &m);
9  float rerata(mahasiswa m);
10 #endif
```

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

void inputMahasiswa(mahasiswa &m)
{
    cout << "Input Nama: ";
    cin >> (m.nim);
    cout << "Input Nilai 1: ";
    cin >> (m.nilai1);
    cout << "Input Nilai 2: ";
    cin >> (m.nilai2);
}

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

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

✓ int main()
  {
    mahasiswa mhs;
    inputMahasiswa(mhs);
    cout << "Rata-rata: " << rerata(mhs) << endl;
    return 0;
  }
```

Screenshots Output

```
Input Nama: Zaki
Input Nilai 1: 80
Input Nilai 2: 70
Rata-rata: 75
```

Deskripsi:

Program diatas menggunakan konsep ADT, pada bagian mahasiswa.h bertipe struct mahasiswa dan prototipe fungsi yang berguna untuk 2 file yang lain, pada pelajaran.cpp untuk memasukan nama dan nilai 1 dan 2, kemudian main.cpp untuk menampilkan data tersebut yaitu nama nilai 1, 2 dan rata rata .

Unguided 1

```
#include <iostream>
#include <string>
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: ";
    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 << "UTS     : "; cin >> mhs[i].uts;
        cout << "UAS     : "; cin >> mhs[i].uas;
        cout << "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 << mhs[i].nama << " (" << mhs[i].nim << ")"
            << " - Nilai Akhir: " << mhs[i].nilaiAkhir << endl;
    }
    return 0;
}
```

Screenshots Output

```
Masukkan jumlah mahasiswa: 3

Data mahasiswa ke-1
Nama : Rikza
NIM : 1034
UTS : 80
UAS : 75
Tugas : 77

Data mahasiswa ke-2
Nama : Nur
NIM : 1035
UTS : 85
UAS : 90
Tugas : 88

Data mahasiswa ke-3
Nama : Zaki
NIM : 1036
UTS : 90
UAS : 93
Tugas : 99

Data Mahasiswa:
Rikza (1034) - Nilai Akhir: 77.1
Nur (1035) - Nilai Akhir: 87.9
Zaki (1036) - Nilai Akhir: 93.9
```

Deskripsi:

Program diatas digunakan untuk mengelola mahasiswa dengan maksimal 10 mahasiswa, dengan memasukan nama, nim, nilai uts (30%) uas (40%) dan tugas (30%), kemudian di rata ratakan untuk dimasukan ke nilai akhir.

Unguided 2

```

#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

```

```

#include "pelajaran.h"
#include <iostream>
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;
}

#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

```
Nama pelajaran : Struktur Data  
Nilai : STD
```

Deskripsi:

Program diatas menggunakan konsep ADT juga, pada bagian pelajaran.h bertipe struct pelajaran dan berprototipe untuk konstruktor create_pelajaran dan tulis tampil_pelajaran, pada pelajaran.cpp untuk merealisasikan kedua fungsi tersebut, dan main.cpp untuk menampilkan data tersebut.

Unguided 3

```

#include <iostream>
using namespace std;

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

void tukarArray(int A[3][3], int B[3][3], int baris, int kolom) {
    int i = baris - 1;
    int j = kolom - 1;
    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;

    cout << "\nSebelum tukar angka: \n";
    cout << "Array A:\n";
    tampilArray(A);
    cout << "\nArray B:\n";
    tampilArray(B);

    tukarArray(A, B, 1, 1);

    cout << "\nSetelah tukar angka: \n";
    cout << "Array A:\n";
    tampilArray(A);
    cout << "\nArray B:\n";
    tampilArray(B);

    int x = 30, y = 50;
    p1 = &x;
    p2 = &y;

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

    tukarPointer(p1, p2);

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

    return 0;
}

```

Screenshots Output


```

Sebelum tukar angka:
Array A:
1      2      3
4      5      6
7      8      9

Array B:
9      8      7
6      5      4
3      2      1

Setelah tukar angka:
Array A:
9      2      3
4      5      6
7      8      9

Array B:
1      8      7
6      5      4
3      2      1

Sebelum tukar pointer:
x = 30, y = 50

Setelah tukar pointer:
x = 50, y = 30

```

Deskripsi:

Program diatas mengelola 2 array 2D int berukuran 3 x 3 dan dua poin integer, menggunakan fungsi seperti menampilkan isi array, menukar angka posisi, dan menukar nilai pada pointer.

C. Kesimpulan

Praktikum ini mengenalkan dan mengimplementasikan konsep Abstract Data Type (ADT) pada C++. Konsep ADT yaitu dengan memisahkan kode menjadi file spesifikasi (.h) yang berisi definisi tipe data (struct) dan prototipe fungsi (primitif) dan file realisasi (.cpp) yang berisi implementasi fungsi tersebut, serta file driver (main.cpp).

D. Referensi

Nugroho, Adi. *Struktur Data Menggunakan C++*. Bandung: Informatika, 2017.

Weiss, Mark Allen. *Data Structures and Algorithm Analysis in C++*. Pearson Education, 2014.