LAPORAN PRAKTIKUM

MODUL 02

Disusun Oleh:

Faishal Arif Setiawan 2311104066

Dosen :

WAHYU ANDI SAPUTRA

PROGRAM STUDI S1 SOFTWARE ENGINEERING

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY

PURWOKERTO

2. TUJUAN

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

3.DASAR TEORI

ADT adalah TYPE dan sekumpulan PRIMITIF (operasi dasar) terhadap TYPE tersebut. Selain itu, dalam sebuah ADT yang lengkap, disertakan pula definisi invarian dari TYPE dan aksioma yang berlaku. ADT merupakan definisi STATIK.

TYPE diterjemahkan menjadi *type* terdefinisi dalam bahasa yang bersangkutan. Jika dalam bahasa C menggunakan struct PRIMITIF, dalam konteks procedural.PRIMITIF dikelompokan menjadi:

1. Konstruktor/Kreator, pemebentuk nilai *type*.Semua objek (variabel) bertype tersebut harus melalui konstruktor. Biasanya namanya diawali Make.

2. *Selector*, untuk mengakses tipe komponen(biasanya namanya diawali Get).

3. Prosedur pengubah nilai komponen (biasanya namanya diawali Get).

4. Tipe validator komponen, yang dipakai untuk mentest apakah dapat membentuk tipe sesuai dengan batasan.

5. Destruktor/Dealokator yaitu untuk “menghancurkan” nilai objek/variabel (sekaligus memori penyimpanannya).

6. Baca/Tulis, untuk interface dengan *input*/*output* device.

7. Operator relasional, terhadap tipe tersebut untuk mendefinisikan lebih besar, lebih kecil, sama dengan dan sebagainya.

8. Aritmatika terhadap tipe tersebut, karena biasanya aritmatika dalam bahasa C hanya terdefinisi untuk bilangan numerik.

9. Konversi dari tipe tersebut ke tipe dasar dan sebaliknya.

IV.GUIDED

```cpp
#include <iostream>

using namespace std;

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

void inputMhs(mahasiswa &m);
float rata2(mahasiswa m);

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


}

void inputMhs(mahasiswa &m){
     cout << "input nim= ";
     cin >> (m).nim;
     cout << "input nilai = ";
     cin >> (m).nilai1;
     cout << "input nilai = ";
     cin >> (m).nilai2;
}
float rata2(mahasiswa m){
   return(m.nilai1+m.nilai2)/2;
}
```

PENJELASAN:

Program ini menggunakan struch yang mendefinisikan tipe data mahasiswa,yang menyimpan NIM dan serta dua nilai mahasiswa

Output:

```
input nim= 2311104066
input nilai = 100
input nilai = 100
rata-rata = 100
Process returned 0 (0x0)   execution time : 10.385 s
Press any key to continue.
```

V.UNGUIDED

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

// array of siswa
struct student {
    string nama, nim, indeks;
    double tugas, uts, uas, hasil;
};


double nilaiakhir(int i, student siswa[]) {
    return (siswa[i].tugas * 0.3) +  + (siswa[i].uts * 0.3) + (siswa[i].uas * 0.4);
}

string nilaitertinggi(student siswa[], int jml) {
    string ket;
    int max = 0;
    for (int i = 0; i < jml; i++) {
        if (max < siswa[i].hasil) {
            max = siswa[i].hasil;
            ket = siswa[i].nim;
        }
    }
    return ket;
}

int main(){
    const int jumlah_siswa = 2;
    student siswa[jumlah_siswa];
    int i;
    double hasil;



    i = 0;
    do {

        cout << "\tDATA SISWA KE\t " << i+1 << endl;



        cout << "Masukkan NAMA siswa ke-" << i+1 << endl;
        cin >> siswa[i].nama;

        cout << "Masukkan NIM dari " << siswa[i].nama << ": ";
        cin >> siswa[i].nim;
        cout << endl;

        cout << "Masukkan NILAI TUGAS dari " << siswa[i].nama << ": ";
        cin >> siswa[i].tugas;
        cout << endl;


        cout << "Masukkan NILAI UTS dari " << siswa[i].nama << ": ";
        cin >> siswa[i].uts;
```

OUTPUT:

```
Masukkan NILAI TUGAS dari FAHMI: 100

Masukkan NILAI UTS dari FAHMI: 80

Masukkan NILAI UAS dari FAHMI: 90

NAMA    NIM     TUGAS   UTS     UAS     NILAI AKHIR
FAHMI   231104074       100     80      90      90.00  (A)

        DATA SISWA KE    2
Masukkan NAMA siswa ke-2
ALVIN
Masukkan NIM dari ALVIN: 2311104070

Masukkan NILAI TUGAS dari ALVIN: 100

Masukkan NILAI UTS dari ALVIN: 75

Masukkan NILAI UAS dari ALVIN: 85

NAMA    NIM     TUGAS   UTS     UAS     NILAI AKHIR
ALVIN   2311104070      100     75      85      86.50  (A)

Nilai tertinggi yaitu: 231104074
```

2.MAIN CPP

```cpp
#include "pelajaran.h"

int main() {
    // Membuat objek pelajaran
    pelajaran p1 = create_pelajaran("Struktur data", "STD");

    // Menampilkan data pelajaran
    tampil_pelajaran(p1);

    return 0;
}
```

Pelajaran cpp

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

// Implementasi fungsi create_pelajaran
pelajaran create_pelajaran(string namapel, string kodepel) {
    pelajaran p;
    p.namaMapel = namapel;
    p.kodeMapel = kodepel;
    return p;
}

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

Pelajara.h

```cpp
#ifndef PELAJARAN_H
#define PELAJARAN_H

#include <string>
using namespace std;

struct pelajaran {
    string namaMapel;
    string kodeMapel;
};

// Deklarasi fungsi
pelajaran create_pelajaran(string namapel, string kodepel);

// Prosedur untuk menampilkan pelajaran
void tampil_pelajaran(pelajaran pel);

#endif
```

Output:

```
D:\struktur data pemograman\UNGUIDEDTP3NO2>program.exe
Nama Mata Pelajaran: Struktur data
Kode Mata Pelajaran: STD
```

3.

```
#include <iostream>

using namespace std;

void tampilkanArray2D(int array[3][3]) {
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            cout << array[i][j] << " ";
        }
        cout << endl;
    }
}

void tukarArray2D(int array1[3][3], int array2[3][3], int baris, int kolom) {
    if (baris >= 0 && baris < 3 && kolom >= 0 && kolom < 3) {
        int temp = array1[baris][kolom];
        array1[baris][kolom] = array2[baris][kolom];
        array2[baris][kolom] = temp;
    } else {
        cout << "Indeks di luar batas!" << endl;
    }
}

void tukarPointer(int *ptr1, int *ptr2) {
    int temp = *ptr1;
    *ptr1 = *ptr2;
    *ptr2 = temp;
}

int main() {

    int array1[3][3] = {
        {1, 2, 3},
        {4, 5, 6},
        {7, 8, 9}
    };
```

```
int array1[3][3] = {
    {1, 2, 3},
    {4, 5, 6},
    {7, 8, 9}
};

int array2[3][3] = {
    {9, 8, 7},
    {6, 0, 4},
    {3, 2, 1}
};

cout << "Isi Array 1:" << endl;
tampilkanArray2D(array1);

cout << "\nIsi Array 2:" << endl;
tampilkanArray2D(array2);

int baris = 1, kolom = 1;
cout << "\nMenukar posisi (" << baris << "," << kolom << ") antara Array 1 dan Array 2." << endl;
tukarArray2D(array1, array2, baris, kolom);

cout << "\nIsi Array 1 setelah penukaran:" << endl;
tampilkanArray2D(array1);

cout << "\nIsi Array 2 setelah penukaran:" << endl;
tampilkanArray2D(array2);

int a = 10, b = 20;
int *ptr1 = &a;
int *ptr2 = &b;

cout << "\nSebelum penukaran: a = " << a << ", b = " << b << endl;
tukarPointer(ptr1, ptr2);
cout << "Setelah penukaran: a = " << a << ", b = " << b << endl;
```

Output:

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

Isi Array 2:
9 8 7
6 0 4
3 2 1

Menukar posisi (1,1) antara Array 1 dan Array 2.

Isi Array 1 setelah penukaran:
1 2 3
4 0 6
7 8 9

Isi Array 2 setelah penukaran:
9 8 7
6 5 4
3 2 1

Sebelum penukaran: a = 10, b = 20
Setelah penukaran: a = 20, b = 10
```