PERTEMUAN 3 ABSTRACT DATA TYPE



Nama:

Yehuda Melvin Sugiarto (2311104055)

Dosen:

Wahyu Andi Saputra S.Pd., M.Eng.

PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK FAKULTAS INFORMATIKA TELKOM UNIVERSITY PURWOKERTO 2024

I. TUJUAN

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

II. TOOL

- 1. Code::Block
- 2. C++
- 3. Laptop

III. DASAR TEORI

ADT (Abstract Data Type) adalah abstraksi dari tipe data yang mencakup TYPE dan operasioperasi dasar (primitif) terhadapnya. ADT berfungsi sebagai definisi statik yang meliputi invarian dan aksioma yang berlaku.

- TYPE: Struktur data yang didefinisikan, yang bisa terdiri dari ADT lain (contoh: ADT Waktu terdiri dari ADT JAM dan ADT DATE, atau Garis terdiri dari dua ADT POINT).
- Dalam bahasa pemrograman seperti C, TYPE diterjemahkan sebagai struct, dan operasi dasar diterjemahkan menjadi fungsi atau prosedur.

Primitif ADT meliputi:

- 1. Konstruktor/Kreator: Membentuk nilai tipe (biasanya dengan awalan "Make").
- 2. Selector: Mengakses komponen tipe (biasanya dengan awalan "Get").
- 3. Prosedur Pengubah: Mengubah nilai komponen.
- 4. Validator: Memvalidasi tipe agar sesuai dengan batasan.
- 5. Destruktor/Dealokator: Menghapus objek atau variabel.
- 6. Baca/Tulis: Untuk interaksi dengan perangkat input/output.
- 7. Operator Relasional: Membandingkan nilai (lebih besar, lebih kecil, dll.).
- 8. Aritmatika: Operasi aritmatika untuk tipe khusus.
- 9. Konversi Tipe: Mengubah tipe menjadi tipe dasar dan sebaliknya.

IV. GUIDE

1. Syntax

```
using namespace std;

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

void inputMhs(mahasiswa em);

float rata2(mahasiswa m);

int main()

{
    mahasiswa mhs;
    inputMhs(mhs);
    cout « "Bata rata = " « rata2(mhs);
    return 0;

}

void inputMhs (mahasiswa em) {
    cout « "Input NIM : ";
    cin >> (m).nim;
    cout « "Input NiAsi 1 : ";
    cin >> (m).nilail;
    cout « "Input NiAsi 2 : ";
    cin >> (m).nilail;
    cout « "Input NiAsi 2 : ";
    cin >> (m).nilail;
    cout « "Input NiAsi 2 : ";
    cin >> (m).nilail;
    cout « "Input NiAsi 2 : ";
    cin >> (m).nilail;
    cout « "Input NiAsi 2 : ";
    cin >> (m).nilail;
    return (m.nilail+m.nilail2)/2;
}
```

2. Output

```
Input NIM : 2311104055
Input Nilai 1 : 95
Input Nilai 2 : 89
Rata rata = 92
Process returned 0 (0x0) execution time
Press any key to continue.
```

V. UNGUIDED

1. Task 1

Syntax

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

struct Mhs

{
    string nama;
    string nim;
    double uts;
    double uts;
    double tugas;
    double nilaiakhir;
};

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

| int main() {
    const int maksMhs = 10;
    Mhs mahasiswa[maksMhs];
    int jmlMhs = 0;

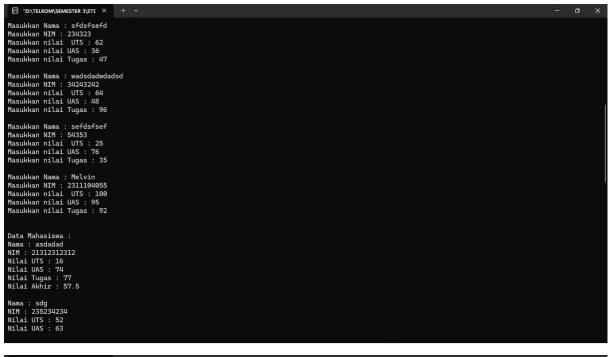
| while (jmlMhs < maksMhs) {
    cout << "Masukkan Nama : ";
    cin >> mahasiswa[jmlMhs].nama;
```

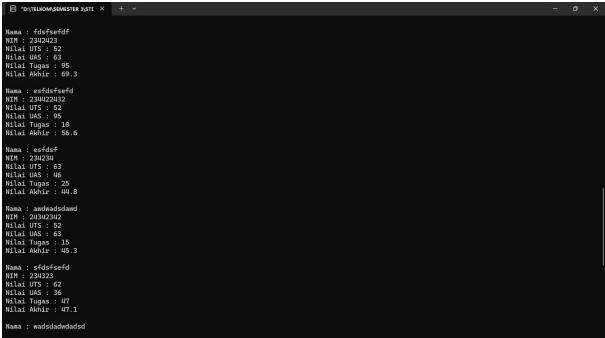
```
if (mahasiswa[imlMhs].nama == "Selesai") {
    cout << "Masukkan NIM : ";
    cin >> mahasiswa[jmlMhs].nim;
    cout << "Masukkan nilai UTS : ";
   cin >> mahasiswa[jmlMhs].uts;
    cout << "Masukkan nilai UAS : ";
    cin >> mahasiswa[jmlMhs].uas;
    cout << "Masukkan nilai Tugas : ";
   cin >> mahasiswa[jmlMhs].tugas;
    cout << "\n";
   mahasiswa[jmlMhs].nilaiakhir =hitungNilaiAkhir(mahasiswa[jmlMhs].uts,
    mahasiswa[jmlMhs].uas, mahasiswa[jmlMhs].tugas);
cout << "\nData Mahasiswa :\n";
for (int i = 0; i < jmlMhs; i++) {
  cout << "Nama : " << mahasiswa[i].nama << endl;
  cout << "NIM : " << mahasiswa[i].nim << endl;
    cout << "Nilai UTS : " << mahasiswa[i].uts << endl;
    cout << "Nilai UAS : " << mahasiswa[i].uas << endl;</pre>
   cout << "Nalal Ingas : " << mahasiswa[i].tugas << endl;
cout << "Nalai akhir : " << mahasiswa[i].nilaiakhir << endl;</pre>
 cout << "\nData Mahasiswa :\n";
 for (int i = 0; i < jmlMhs; i++) {
       cout << "Nama : " << mahasiswa[i].nama << endl;
       cout << "NIM : " << mahasiswa[i].nim << endl;</pre>
       cout << "Nilai UTS : " << mahasiswa[i].uts << endl;</pre>
       cout << "Nilai UAS : " << mahasiswa[i].uas << endl;
       cout << "Nilai Tugas : " << mahasiswa[i].tugas << endl;</pre>
       cout << "Nilai Akhir : " << mahasiswa[i].nilaiakhir << endl;</pre>
       cout << "\n";
```

Output

}

return 0;





```
© TD:\TELKOM\SEMESTER 3\STE × + ∨
Nilai UTS : 62
Nilai UAS : 36
Nilai Tugas : 47
Nilai Akhir : 47.1
Nama : wadsdadwdadsd
NIM : 34243242
Nilai UTS : 64
Nilai UAS : 48
Nilai Tugas : 96
Nilai Akhir: 67.2
Nama : sefdsfsef
NIM : 54353
Nilai UTS : 25
Nilai UAS : 76
Nilai Tugas : 35
Nilai Akhir : 48.4
Nama : Melvin
NIM : 2311104055
Nilai UTS : 100
Nilai UAS : 95
Nilai Tugas : 92
Nilai Akhir: 95.6
Process returned 0 (0x0) execution time : 100.267 s
Press any key to continue.
```

2. Task 2

Syntax

```
main.cpp X main.cpp X taskl.cpp X pelajaran.h X task2.cpp X

#include <istring>
#include <string>
#include <istring kodeMapel;
#include <istring kodeMapel;
#include <istring kodeMapel) #include #includ
```

Output

```
Nama Pelajaran : Struktur Data
Nilai : STD

Process returned 0 (0x0) execution time : 0.040 s

Press any key to continue.
```

3. Task 3

Syntax

```
wing namespace std;

void menampilkanArray(int arr[3][3]) {
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            cout << arr[i][j] << " ";
        }
        cout << end1;
    }
}

void menukarArray(int arr[3][3], int arr2[3][3], int posisi1[2], int posisi2[2]) {
    int temp = arr1[posisi1[0]][posisi1[1]];
        arr1[posisi1[0]][posisi1[1]] = arr2[posisi2[0]][posisi2[1]];
        arr2[posisi2[0]][posisi2[1]] = temp;
}

void menukarPointer(int* ptrl, int* ptr2) {
    int temp = *ptrl;
    *ptrl = *ptr2;
    *ptr2 = temp;
}</pre>
```

```
lint main() {
    int arr1[3][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
    int arr2[3][3] = {{10, 11, 12}, {13, 14, 15}, {16, 17, 18}};

int varl = 20;
    int var2 = 30;
    int vpr1 = vvar1;
    int* ptr2 = svar2;

cout << "Lgg, Array 1 : " << end1;
    menampilkanArray(arr1);
    cout << "Lgd, Array 2 : " << end1;
    menampilkanArray(arr2);
    cout << "N << end1;
    int posisi1[2] = {1,1};
    int posisi2[2] = {2,2};
    menukarArray(arr1, arr2, posisi1, posisi2);

cout << "Lgd, Nilad, Array 1 (UPDATED):" << end1;
    menampilkanArray(arr1);
    cout << "Lgd, Nilad, Array 2 (UPDATED):" << end1;
    menampilkanArray(arr2);
    cout << "N << end2;
    menampilkanArray(arr2);
    cout << "N </pre>
```

```
int posisi1[2] = {1,1};
int posisi2[2] = {2,2};
menukarArray(arr1, arr2, posisi1, posisi2);

cout << "Isi Nilai Array 1 (UPDATED):" << endl;
menampilkanArray(arr1);
cout << "Isi Nilai Array 2 (UPDATED):" << endl;
menampilkanArray(arr2);
cout << "\n" << endl;

menukarPointer(ptr1, ptr2);

cout << "Isi Nilai Variabel 1 (UPDATED): " << *ptr1 << endl;
cout << "Isi Nilai Variabel 2 (UPDATED): " << *ptr2 << endl;
cout << "\n" << endl;
return 0;
}</pre>
```

Output

VI. KESIMPULAN

Di Praktikum kali ini kita dapat mengetahui definisi dan strukturnya, dan juga cara pengimplementasian dari ADT itu sendiri. Yang dimana kita juga diberi beberapa latihan untuk menggabungkan modul kali ini dengan materi yang sudah dipelajari di modul sebelumnya.