**LAPORAN PRAKTIKUM MODUL 3**

**ABSTRACT DATA TYPE (ADT0**



**Disusun Oleh:**

**Dhiemas Tulus Ikhsan 2311104046 SE-07-02**

**Dosen :**

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

**PROGRAM STUDI S1 SOFTWARE ENGINEERING FAKULTAS INFORMATIKA**

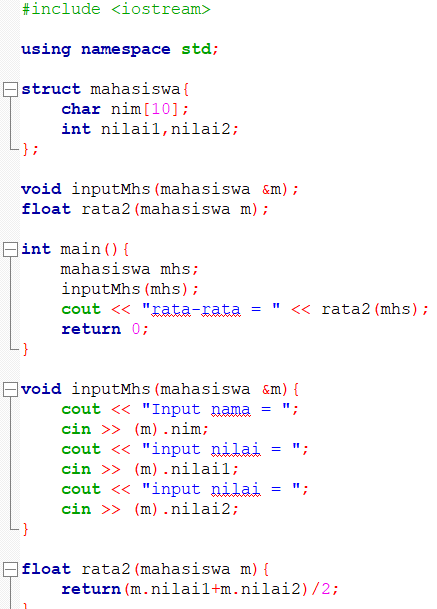
**TELKOM UNIVERSITY PURWOKERTO**

**2024**

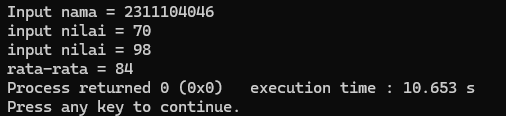
1. **TUJUAN**
   1. Memahami konsep *Abstract Data Type* (ADT) dan penggunaannya dalam pemrograman.
2. **LANDASAN TEORI**
3. ***Abstract Data Type***

ADT (Abstract Data Type) adalah tipe data yang mendefinisikan jenis data dan sekumpulan operasi dasar yang dapat dilakukan terhadapnya. ADT mencakup definisi statis dari tipe data, termasuk invarian (aturan yang harus dipenuhi) dan aksioma yang berlaku. Sebuah ADT dapat terdiri dari ADT lainnya, misalnya ADT waktu yang terdiri dari ADT JAM dan DATE, atau ADT garis yang terdiri dari dua buah ADT POINT. Dalam implementasi, ADT diterjemahkan menjadi tipe data yang sesuai dengan bahasa pemrograman, seperti `struct` dalam bahasa C. Operasi dasar atau primitif dalam ADT mencakup beberapa jenis, seperti konstruktor untuk membentuk objek, selector untuk mengakses komponen, prosedur pengubah nilai komponen, validator untuk memeriksa kesesuaian nilai, destruktor untuk menghapus objek, serta fungsi baca/tulis untuk menghubungkan dengan perangkat input/output. Selain itu, ADT juga dapat memiliki operator relasional untuk perbandingan, operasi aritmatika, dan konversi tipe. Implementasi ADT biasanya melibatkan dua modul utama: satu modul untuk spesifikasi tipe dan operasi dasar (dalam file `.h`), dan modul lain untuk realisasi atau implementasi dari operasi tersebut (dalam file `.c`). Spesifikasi ini menggambarkan deskripsi tipe dan operasi, sementara realisasi adalah kode program yang memanfaatkan selector dan konstruktor.

1. **GUIDED**



Output

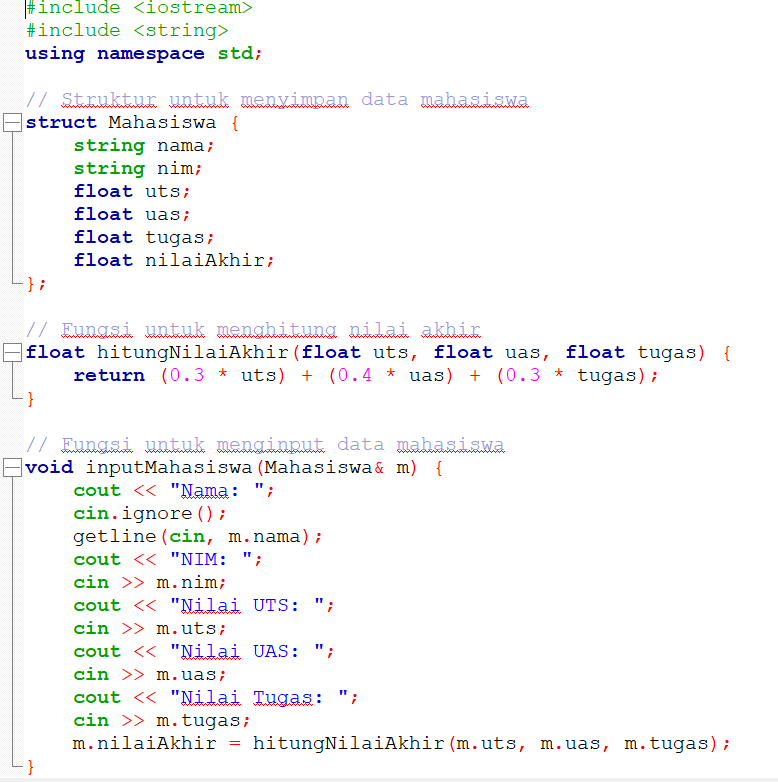


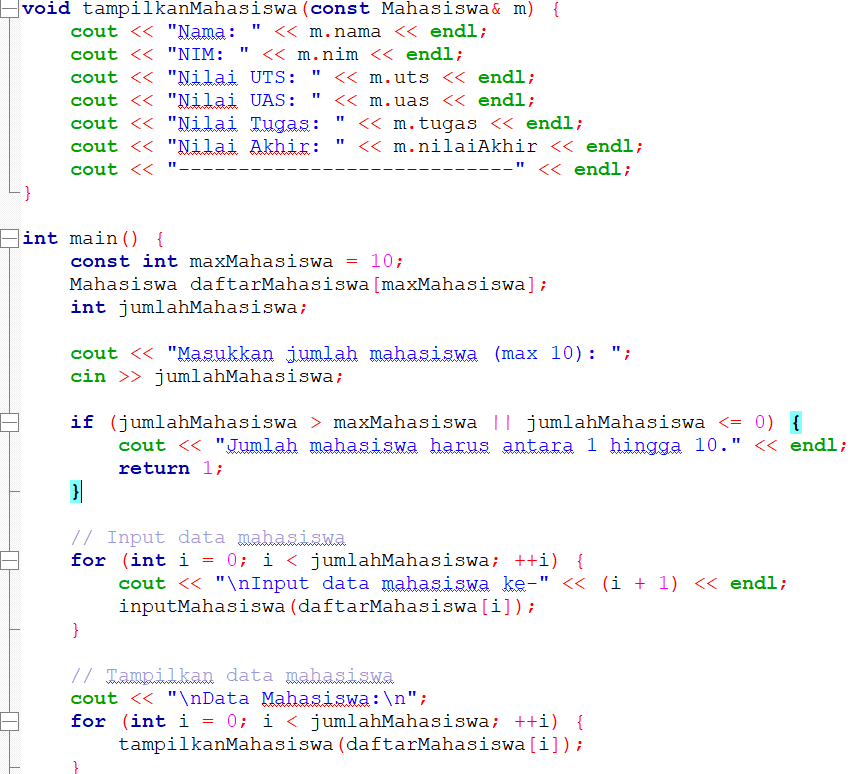
Program di atas menggunakan bahasa C++ untuk mengelola data seorang mahasiswa, dengan struktur data `mahasiswa` yang memiliki tiga anggota: `nim` (Nomor Induk Mahasiswa), `nilai1`, dan `nilai2`.

* Struct `mahasiswa’ : Mendefinisikan tipe data yang menyimpan `nim` sebagai karakter array (`char nim[10]`) dan dua nilai integer (`nilai1` dan `nilai2`).
* Fungsi `inputMhs` : Menerima referensi dari objek `mahasiswa` sebagai parameter dan meminta input dari pengguna untuk `nim`, `nilai1`, dan `nilai2`.
* Fungsi `rata2`: Menghitung rata-rata dari `nilai1` dan `nilai2` dari objek `mahasiswa` yang diterima sebagai parameter, dan mengembalikannya dalam bentuk `float`.

Di dalam fungsi `main`, program membuat objek `mhs` dari tipe `mahasiswa`, lalu memanggil `inputMhs` untuk memasukkan data. Setelah itu, program menghitung dan menampilkan rata-rata nilai dengan memanggil fungsi `rata2` dan mencetak hasilnya ke layar.

1. **UNGUIDED**
2. Task 1

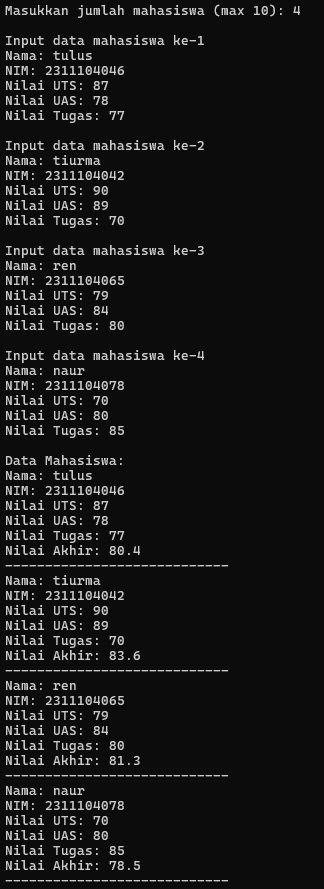
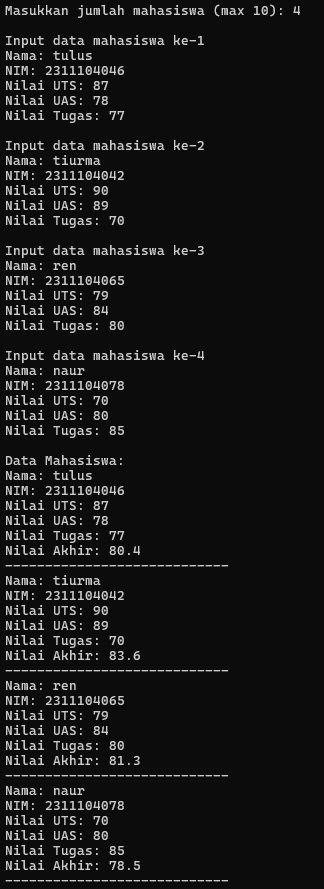




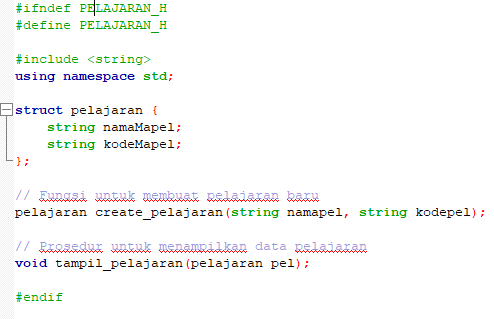
Penjelasan :

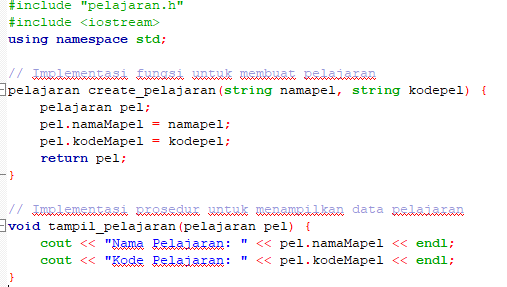
* **Struct Mahasiswa**: Menyimpan informasi nama, nim, uts, uas, tugas, dan nilaiAkhir.
* **Fungsi hitungNilaiAkhir**: Menghitung nilai akhir berdasarkan rumus 0.3 \* uts + 0.4 \* uas + 0.3 \* tugas.
* **Fungsi inputMahasiswa**: Meminta input pengguna untuk mengisi data mahasiswa, lalu menghitung nilaiAkhir.
* **Fungsi tampilkanMahasiswa**: Menampilkan data dari satu mahasiswa.
* **main Function**:
  + Meminta input jumlah mahasiswa (maksimal 10).
  + Memasukkan data mahasiswa menggunakan fungsi inputMahasiswa.
  + Menampilkan seluruh data mahasiswa menggunakan tampilkanMahasiswa.

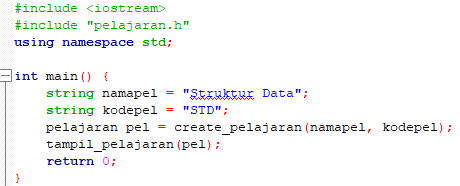
Output :



1. Task 2







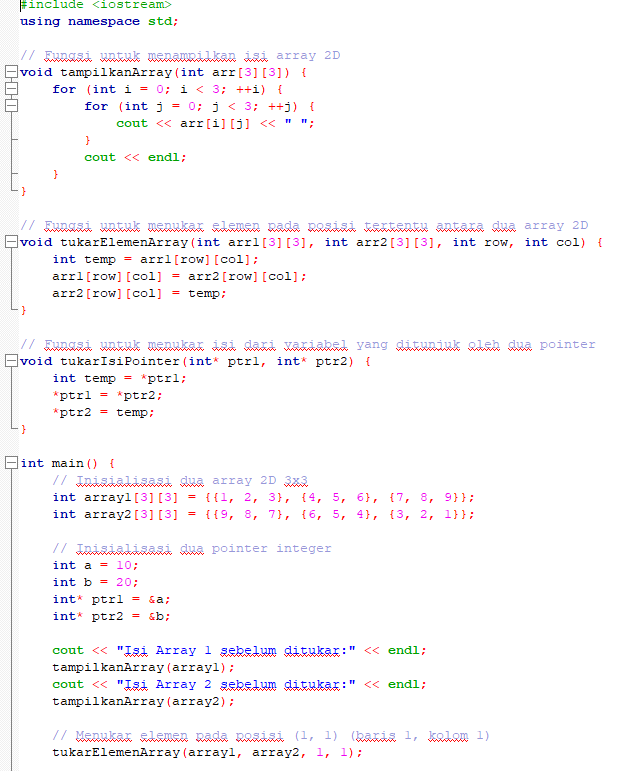
Penjelasan Program:

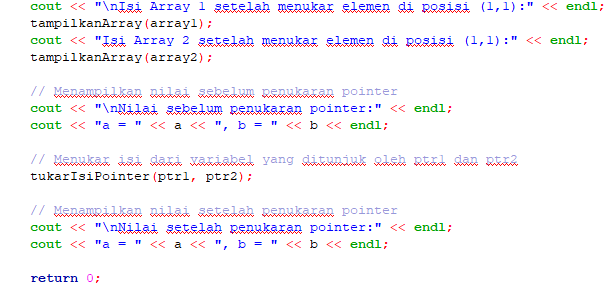
* pelajaran.h:
* Mendefinisikan tipe data pelajaran yang terdiri dari namaMapel dan kodeMapel (keduanya bertipe string).
* Deklarasi fungsi create\_pelajaran untuk membuat objek pelajaran.
* Deklarasi prosedur tampil\_pelajaran untuk menampilkan data pelajaran.
* pelajaran.cpp:
* Mengimplementasikan fungsi create\_pelajaran, yang menerima nama pelajaran dan kode pelajaran sebagai parameter, lalu mengembalikan objek pelajaran yang berisi data tersebut.
* Mengimplementasikan prosedur tampil\_pelajaran untuk menampilkan nama dan kode pelajaran ke layar.
* main.cpp:
* Membuat objek pelajaran dengan nama "Struktur Data" dan kode "STD" menggunakan fungsi create\_pelajaran.
* Menampilkan informasi pelajaran tersebut dengan memanggil prosedur tampil\_pelajaran.

Output Program :



1. Task 3





Penjelasan Program:

1. Array 2D: array1 dan array2 masing-masing berukuran 3x3, diinisialisasi dengan nilai-nilai berbeda.
2. Pointer: ptr1 menunjuk ke a (bernilai 10), dan ptr2 menunjuk ke b (bernilai 20).
3. Fungsi tampilkanArray: Mencetak isi dari array 2D yang diberikan.
4. Fungsi tukarElemenArray: Menukar elemen di posisi tertentu (row, col) antara array1 dan array2.
5. Fungsi tukarIsiPointer: Menukar nilai dari dua variabel yang ditunjuk oleh ptr1 dan ptr2.
6. main Function:

* Menampilkan isi array1 dan array2 sebelum dan sesudah penukaran elemen pada posisi (1, 1).
* Menampilkan nilai a dan b sebelum dan sesudah penukaran menggunakan pointer.

1. **KESIMPULAN**

Pada praktikum ini, kami mengimplementasikan beberapa konsep ADT (Abstract Data Type) dalam program C++:

1. Penggunaan Struktur Data sebagai ADT:

Kami menggunakan struktur Mahasiswa untuk menyimpan data (nama, NIM, nilai UTS, UAS, dan tugas), serta menghitung nilai akhir dengan fungsi. Ini menunjukkan bagaimana ADT dapat mengelola data secara terstruktur.

1. Manipulasi Array 2D:

Kami menggunakan dua array 2D untuk menyimpan data dalam bentuk matriks dan menukarkan elemen di posisi tertentu. Ini menggambarkan bagaimana ADT bekerja dengan data yang lebih kompleks seperti matriks.

1. Operasi dengan Pointer:

Dengan dua pointer integer, kami menukar nilai dari dua variabel. Ini menekankan fleksibilitas pointer dalam manipulasi data dinamis, penting dalam pengelolaan memori. Secara keseluruhan, praktikum ini mengajarkan pentingnya ADT untuk membuat program lebih modular dan mudah dikelola, dengan memisahkan data dan operasi yang dapat dilakukan terhadap data tersebut.