**LAPORAN PRAKTIKUM**

**Modul 3**

**“Abstract Data Type (ADT)”**

****

Disusun Oleh:

Tiurma Grace Angelina 2311104042

SE-07-02

Dosen :

Wahyu Andy Saputra

**PROGRAM STUDI S1 SOFTWARE ENGINEERING**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY**

**PURWOKERTO**

**2024**

# TUJUAN

Memahami konsep Abstract Data Type (ADT) dan penggunaannya dalam

pemrograman.

# LANDASAN 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.

ADT terdiri dari dua komponen utama: data dan operasi. Data merepresentasikan

informasi yang disimpan, sementara operasi adalah fungsi-fungsi yang dapat

digunakan untuk memanipulasi data tersebut. Dengan menggunakan ADT,

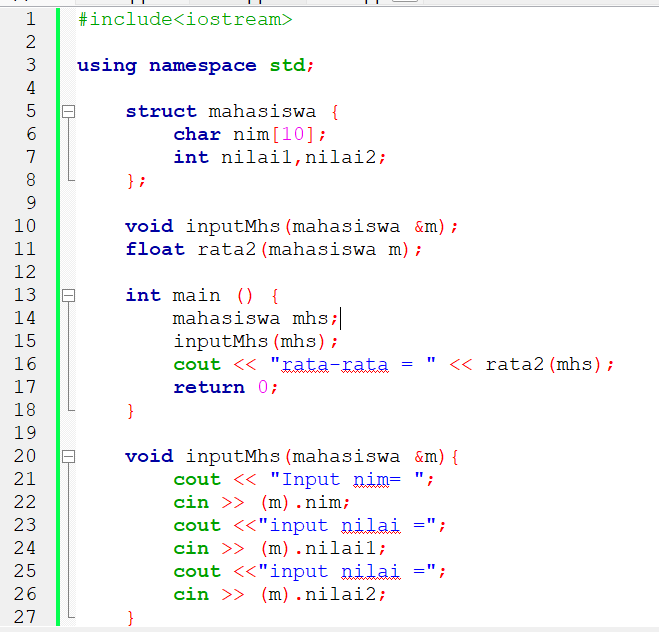
programmer dapat bekerja pada tingkat abstraksi yang lebih tinggi, yang

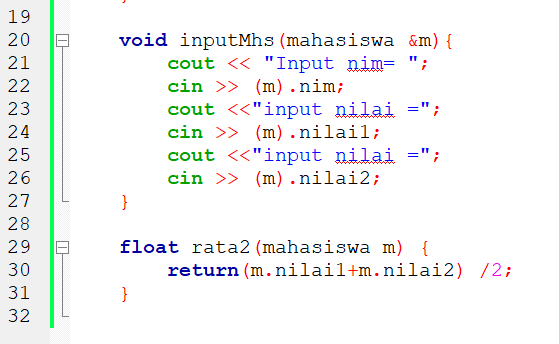
memungkinkan pengembangan software yang lebih modular, mudah dipelihara, dan

dapat digunakan kembali. Contoh umum dari ADT meliputi stack, queue, list, dan tree, yang masing-masing memiliki operasi spesifik yang terkait dengannya.

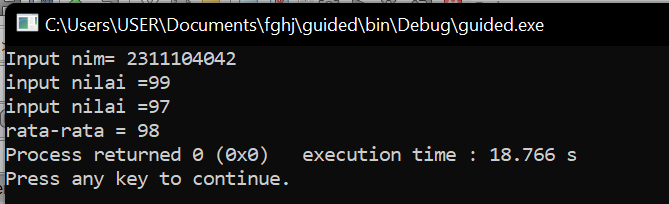
# GUIDED

CODE :





OUTPUT:



PENJELASAN:

1. Struct `mahasiswa`:

- Digunakan untuk menyimpan data mahasiswa, yaitu NIM, nilai pertama (`nilai1`), dan nilai kedua (`nilai2`).

2. Fungsi `inputMhs(mahasiswa &m)`:

- Mengambil input dari pengguna berupa NIM, nilai pertama, dan nilai kedua untuk disimpan di struct `mahasiswa`.

3. Fungsi `rata2(mahasiswa m)`:

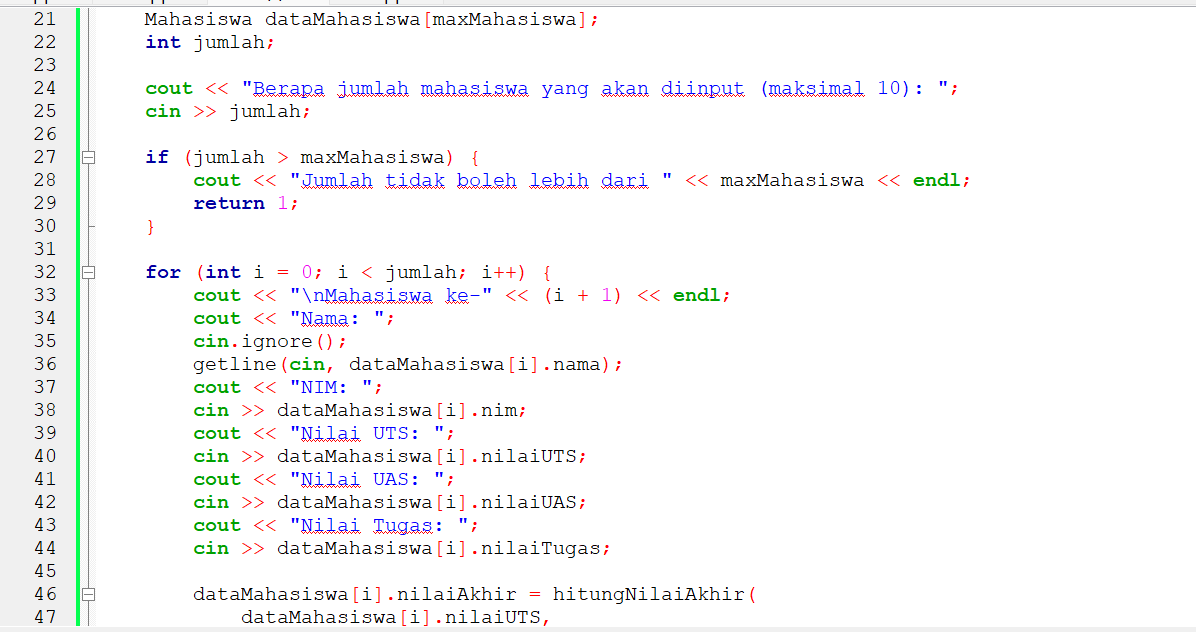
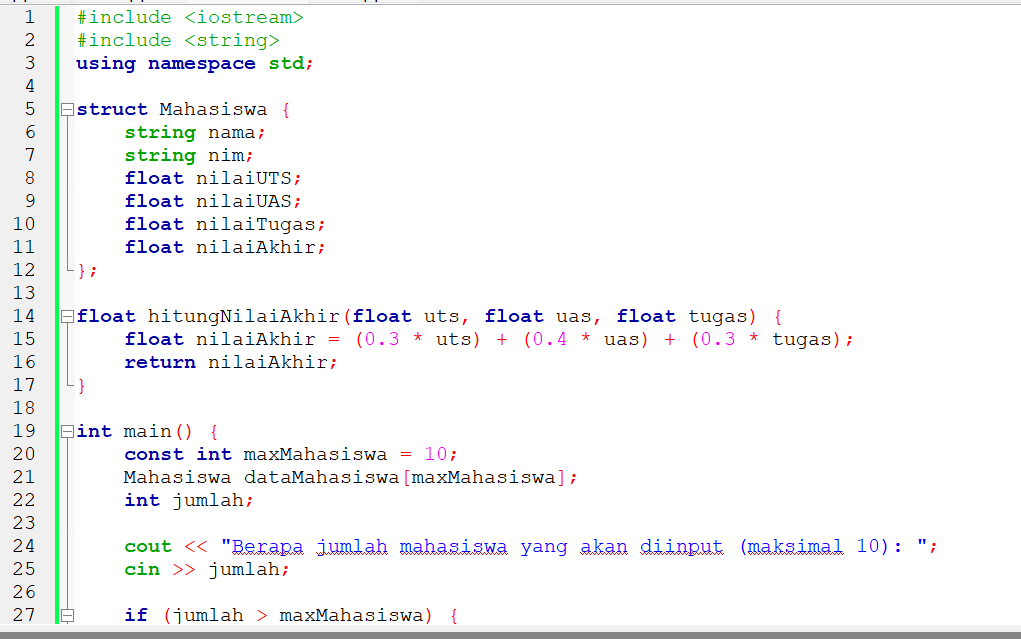
- Menghitung rata-rata dari dua nilai mahasiswa yang diberikan, kemudian mengembalikannya sebagai nilai float.

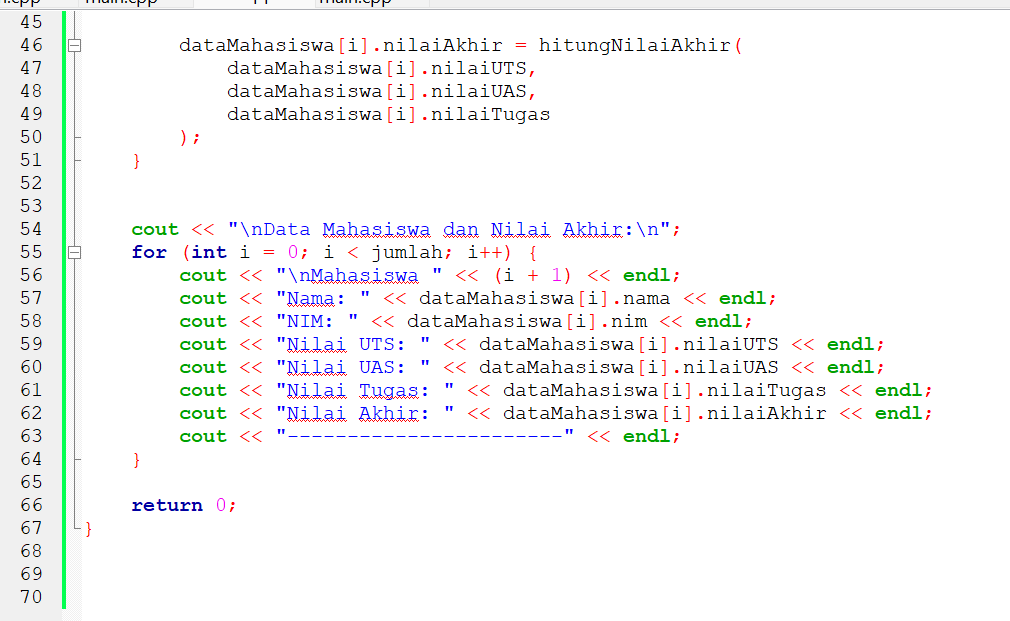
4. Fungsi `main()`:

- Memanggil fungsi `inputMhs()` untuk mengisi data mahasiswa, kemudian memanggil fungsi `rata2()` untuk menghitung rata-rata dan menampilkannya.

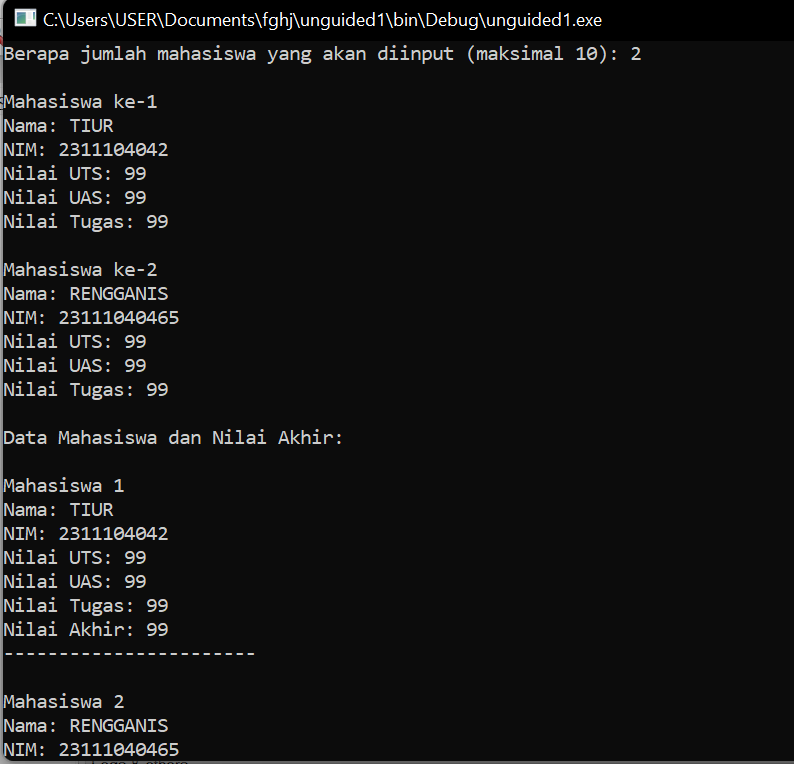
1. **UNGUIDED**
2. Buat program yang dapat menyimpan data mahasiswa (max. 10) ke dalam sebuah *array* dengan field nama, nim, uts, uas, tugas, dan nilai akhir. Nilai akhir diperoleh dari FUNGSI dengan rumus 0.3\*uts+0.4\*uas+0.3\*tugas.

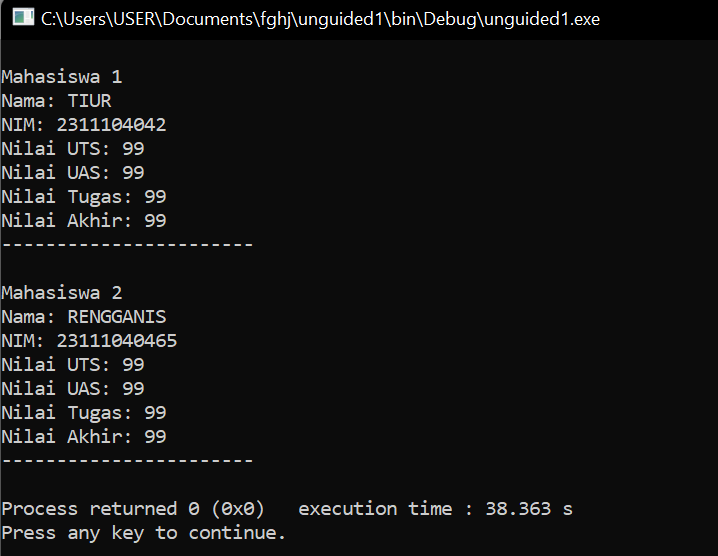
CODE :





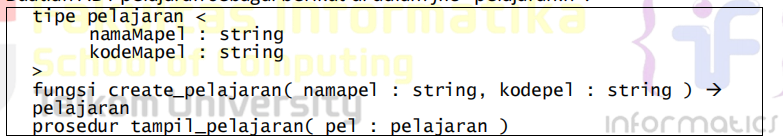
OUTPUT:





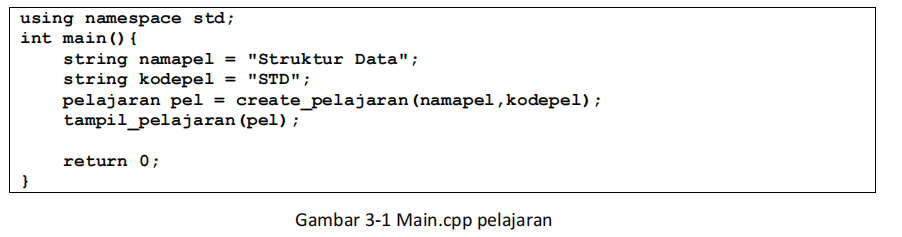
Penjelasan :

1. **Struct Mahasiswa**: Menyimpan data mahasiswa dengan atribut nama, nim, uts, uas, tugas, dan nilaiAkhir.
2. **Fungsi hitungNilaiAkhir**: Menghitung nilai akhir mahasiswa berdasarkan rumus yang diberikan: 0.3\*uts + 0.4\*uas + 0.3\*tugas.
3. **Array mahasiswa[10]**: Menyimpan data maksimal 10 mahasiswa.
4. **Input dan Output**: Program meminta input data dari pengguna dan menampilkan data serta nilai akhir masing-masing mahasiswa. Program ini terbatas pada 10 mahasiswa, tetapi dapat dimodifikasi sesuai kebutuhan.
5. Buatlah ADT pelajaran sebagai berikut di dalam *file* “pelajaran.h”:

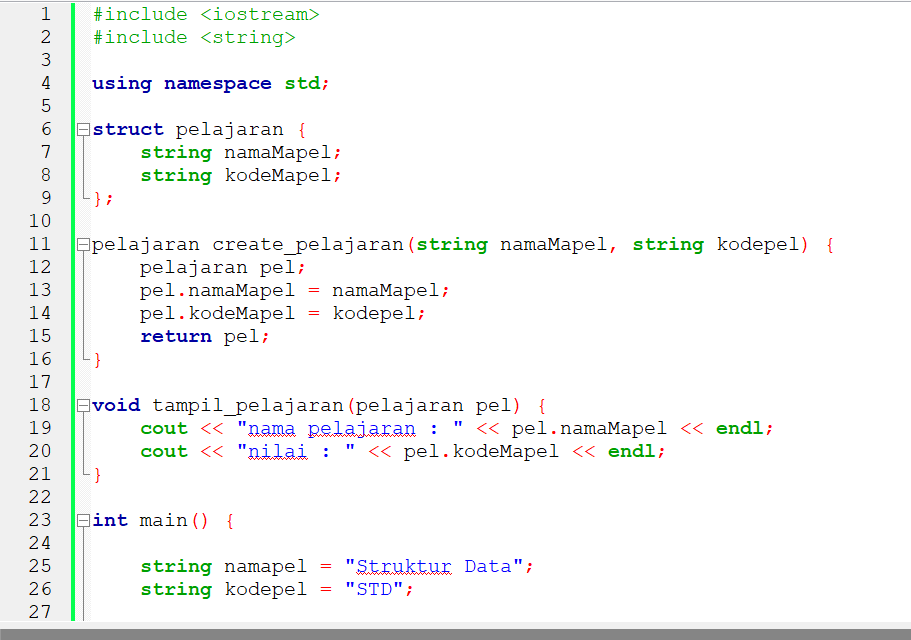


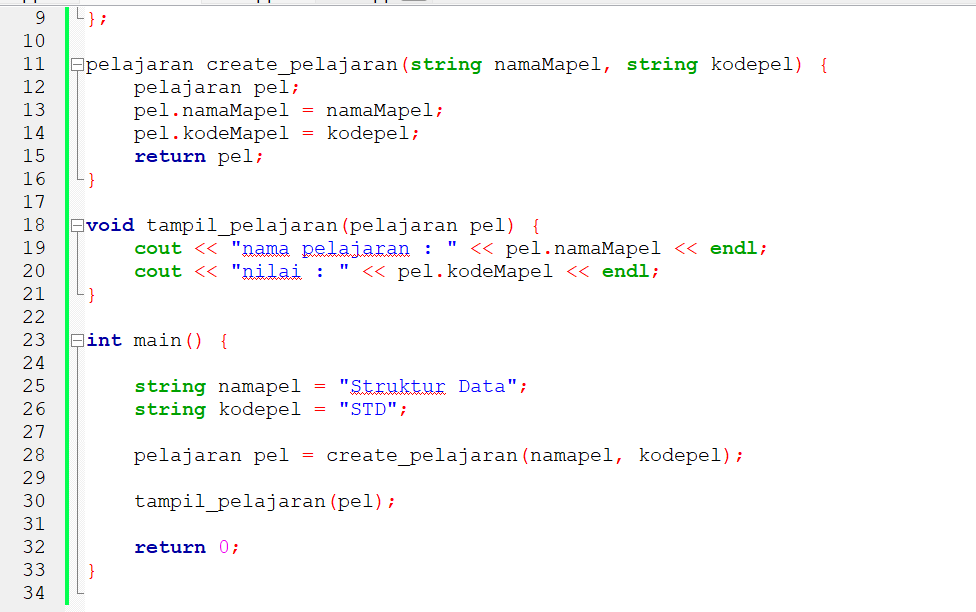
Buatlah implementasi ADT pelajaran pada *file* “pelajaran.cpp”

Cobalah hasil implementasi ADT pada *file* “main.cpp”

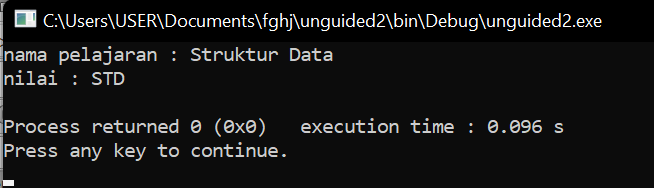


CODE:





OUTPUT:



PENJELASAN:

* **#include <iostream> dan #include <string>**: Digunakan untuk mengakses fungsi input-output (cout, cin) dan tipe data string.
* **Struct pelajaran**: Mendefinisikan tipe data pelajaran yang menyimpan namaMapel dan kodeMapel.
* **Fungsi create\_pelajaran**: Mengembalikan objek pelajaran yang berisi nama dan kode pelajaran.
* **Fungsi tampil\_pelajaran**: Mencetak nama dan kode pelajaran ke layar.
* **Fungsi main()**: Kode utama yang menjalankan program. Program membuat objek pelajaran, kemudian menampilkan informasinya.

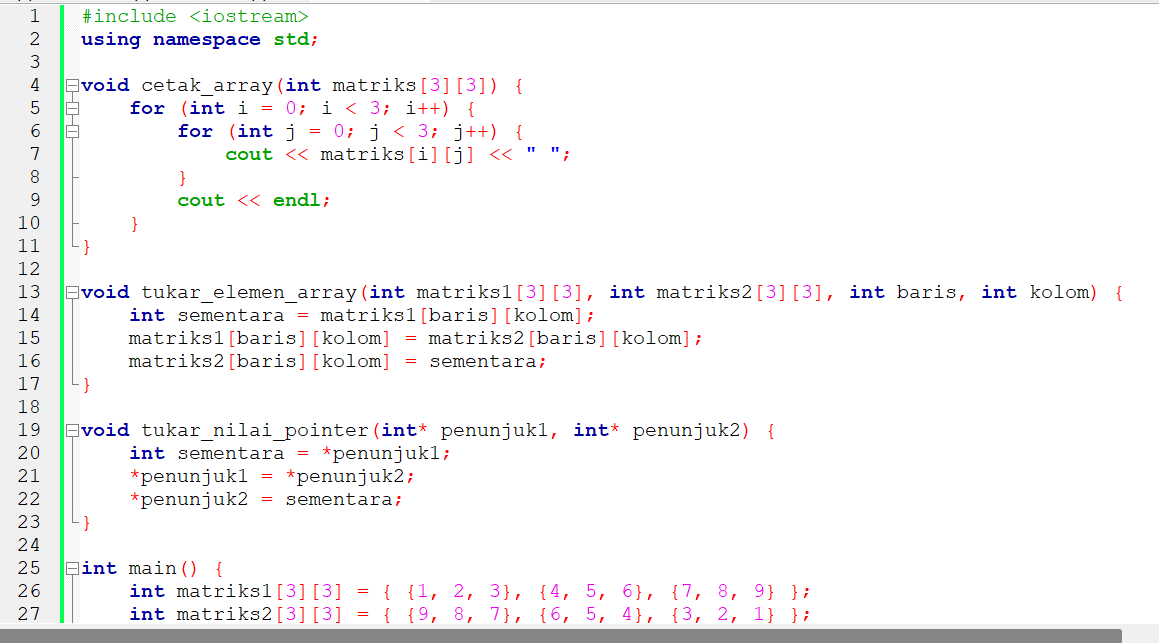
1. Buatlah program dengan ketentuan :

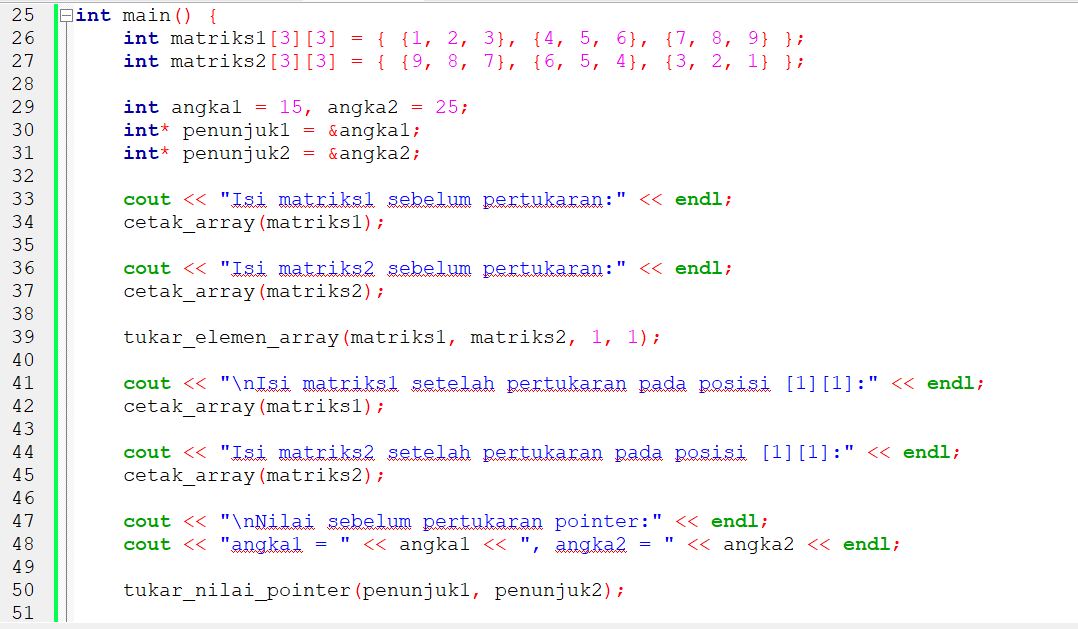
- 2 buah *array* 2D *integer* berukuran 3x3 dan 2 buah *pointer integer*

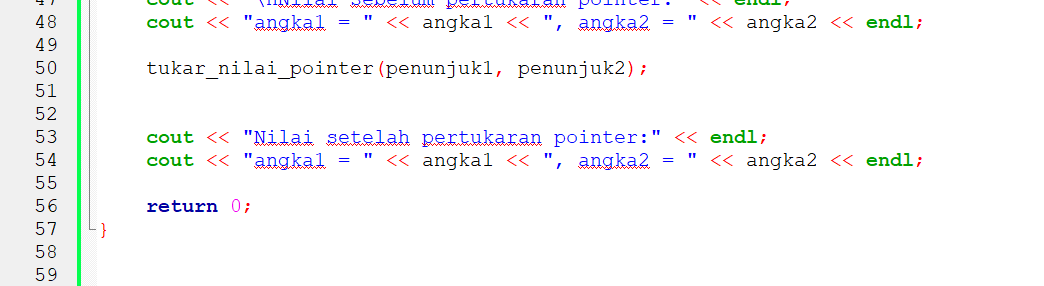
- fungsi/prosedur yang menampilkan isi sebuah *array integer* 2D

- fungsi/prosedur yang akan menukarkan isi dari 2 *array integer* 2D pada posisi tertentuSTRUKTUR DATA 46

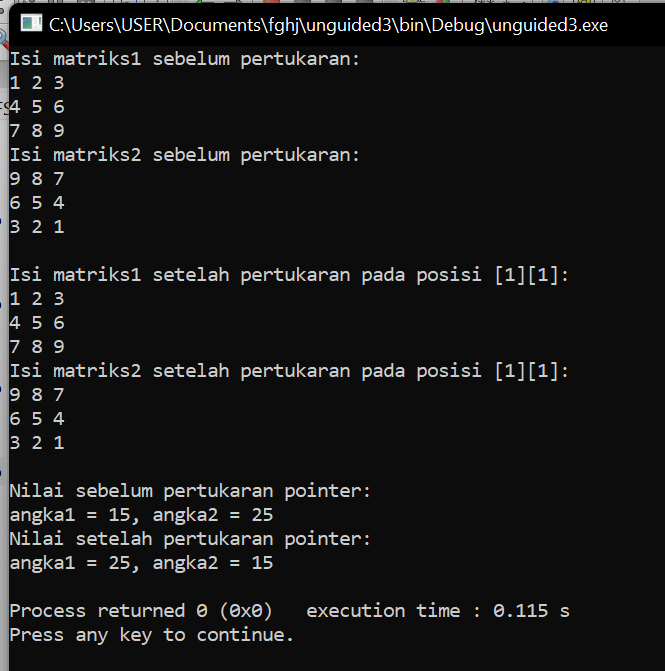
- fungsi/prosedur yang akan menukarkan isi dari variabel yang ditunjuk oleh 2 buah *pointer*







OUTPUT:



PENJELASAN :

**Nama variabel**:

* array1 dan array2 diganti menjadi matriks1 dan matriks2 untuk memberikan nama yang lebih spesifik.
* ptr1 dan ptr2 diganti menjadi penunjuk1 dan penunjuk2 agar lebih jelas sebagai pointer dalam konteks bahasa Indonesia.
* Variabel a dan b diganti menjadi angka1 dan angka2.

**Nama fungsi**:

* tampil\_array diubah menjadi cetak\_array untuk memberi nuansa nama fungsi yang lebih sesuai.
* tukar\_array diubah menjadi tukar\_elemen\_array untuk lebih menjelaskan bahwa fungsi ini menukar elemen di posisi tertentu dari dua array.
* tukar\_pointer diubah menjadi tukar\_nilai\_pointer agar lebih jelas bahwa fungsi ini menukar nilai yang ditunjuk oleh pointer.

# KESIMPULAN

# 1. Pemahaman tentang Array 2D dan Pointer:

# - Program ini menunjukkan cara kerja array 2D dan penggunaan pointer dalam menukar nilai-nilai variabel. Dua array 2D digunakan untuk menyimpan angka-angka dan ada beberapa fungsi yang menangani pertukaran elemen-elemen di posisi tertentu.

# 2. Penggunaan Fungsi/Prosedur:

# - Program ini mengimplementasikan fungsi untuk menampilkan isi array 2D, menukar elemen-elemen di dalam array pada posisi yang spesifik, dan menukar nilai yang ditunjuk oleh dua pointer.

# - Fungsi `cetak\_array()` berfungsi untuk mencetak isi array 2D ke layar, sedangkan `tukar\_elemen\_array()` bertanggung jawab untuk menukar elemen antara dua array 2D di indeks yang sama.

# 3. Manfaat Pointer dalam Pertukaran Nilai:

# - Dengan pointer, program ini dapat menukar nilai-nilai antara dua variabel secara langsung di dalam memori, yang memperlihatkan pentingnya pemahaman tentang pointer dalam pemrograman C++.

# 4. Abstraksi yang Lebih Modular:

# - Program ini mengimplementasikan abstraksi sederhana yang membuat kode menjadi lebih modular. Setiap fungsi memiliki tanggung jawab yang jelas, sehingga program lebih mudah dikelola dan dikembangkan lebih lanjut.