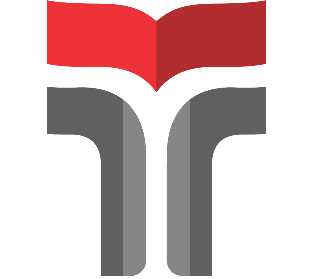
**LAPORAN PRAKTIKUM STRUKTUR DATA DAN ALGORITMA**

**MODUL I**

**TIPE DATA**



# Disusun Oleh :

NAMA : Aulia Radix Putra Winarko  
NIM : 2311102056

# Dosen

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

**PROGRAM STUDI S1 TEKNIK INFORMATIKA   
FAKULTAS INFORMATIKA**

**INSTITUT TEKNOLOGI TELKOM PURWOKERTO  
2024**

1. **Dasar Teori**

Tipe data adalah adalah sebuah pengklasifikasian data berdasarkan jenis data tersebut. Tipe data dibutuhkan agar kompiler dapat mengetahui bagaimana sebuah data akan digunakan. Adapun tipe data yang akan dipelajari, sebagai berikut :

1. Tipe data Primitif
2. Tipe data Abstrak
3. Tipe data Koleksi]

* Tipe data Primitif

Tipe data primitif adalah tipe data yang sudah ditentukan oleh sistem, tipe data primitif ini disediakan oleh banyak bahasa pemrograman, perbedaannya terletak pada jumlah bit yang dialokasikan untuk setiap bit pada tipe data primitif tergantung pada Bahasa pemrograman,compiler dan sistem operasinya.

* Tipe Data Abstrak

Tipe data abstrak atau yang biasa disebut Abstrak Data Tipe(ADT) merupakan tipe data yang dibentuk oleh programer itu sendiri. Pada tipe data abstrak bisa berisi banyak tipe data, jadi nilainya bisa lebih dari satu dan beragam tipe data. Fitur Class adalah fitur Object Oriented Program(OPP) pada bahasa C++ yang mirip dengan fitur data structures Struct pada bahasa C. Keduanya berfungsi untuk membungkus tipe data di dalamnya sebagai anggota. menurut learn.microsoft.com perbedaan antara Struct dan Class adalah pada akses defaultnya dimana Struct bersifat public dan Class bersifat private.

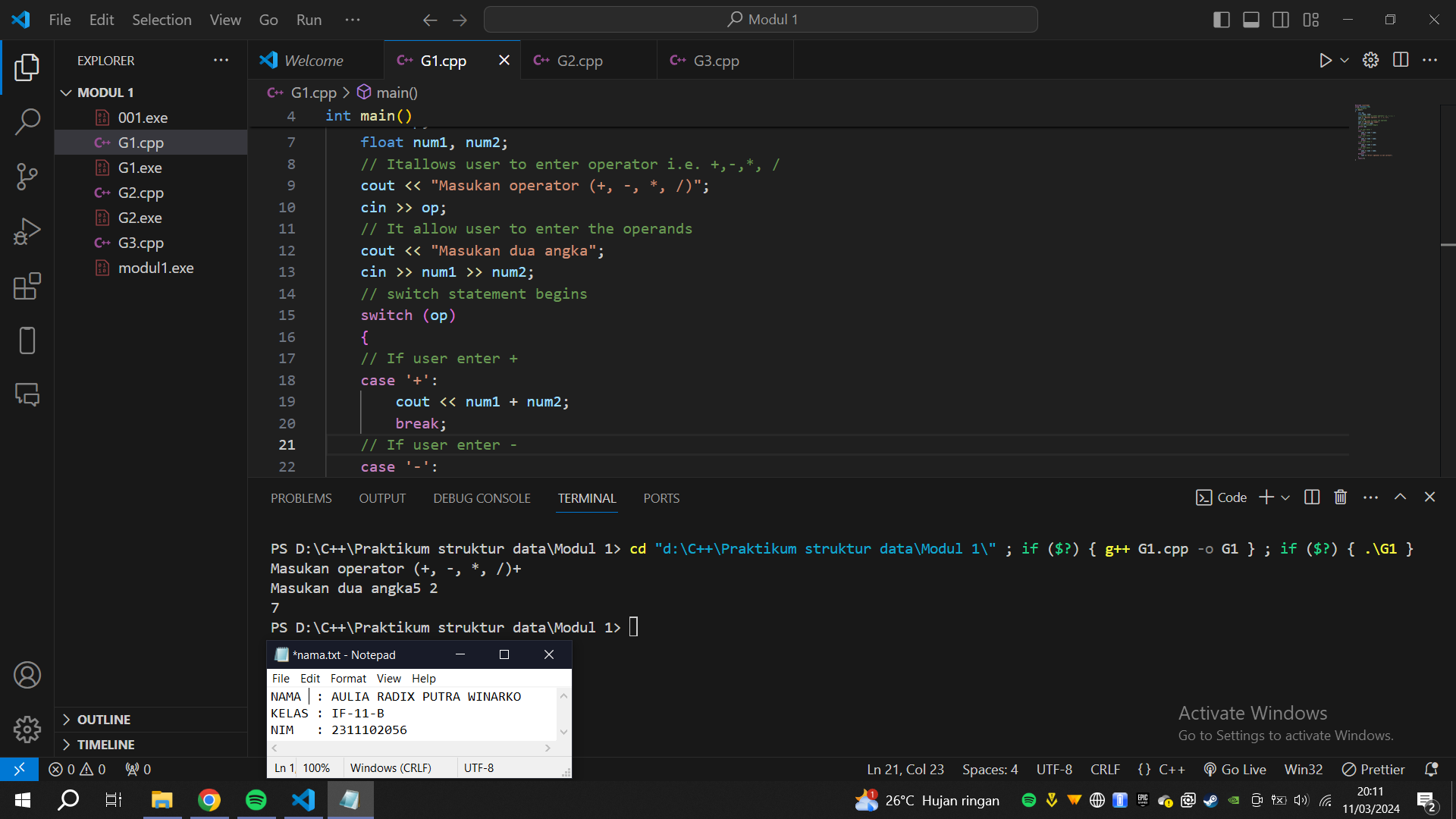
* Tipe Data Koleksi

Tipe data koleksi (Collection Data Type) adalah tipe data yang digunakan untuk mengelompokkan dan menyimpan beberapa nilai atau objek secara bersamaan. Tipe data koleksi memungkinkan Anda menyimpan, mengelola, dan mengakses sejumlah besar data dengan cara yang terstruktur.

1. **Guided** [

Guided 1

|  |
| --- |
| #include <iostream>  using namespace std;  // Main program  int main()  {      char op;      float num1, num2;      // Itallows user to enter operator i.e. +,-,\*, /      cout << "Masukan operator (+, -, \*, /)";      cin >> op;      // It allow user to enter the operands      cout << "Masukan dua angka";      cin >> num1 >> num2;      // switch statement begins      switch (op)      {      // If user enter +      case '+':          cout << num1 + num2;          break;      // If user enter -      case '-':          cout << num1 - num2;          break;      // If user enter \*      case '\*':          cout << num1 \* num2;          break;      // If user enter /      case '/':          cout << num1 / num2;          break;      default:          cout << "Error! operator is not correct";      }      return 0;  } |

Screenshots Output

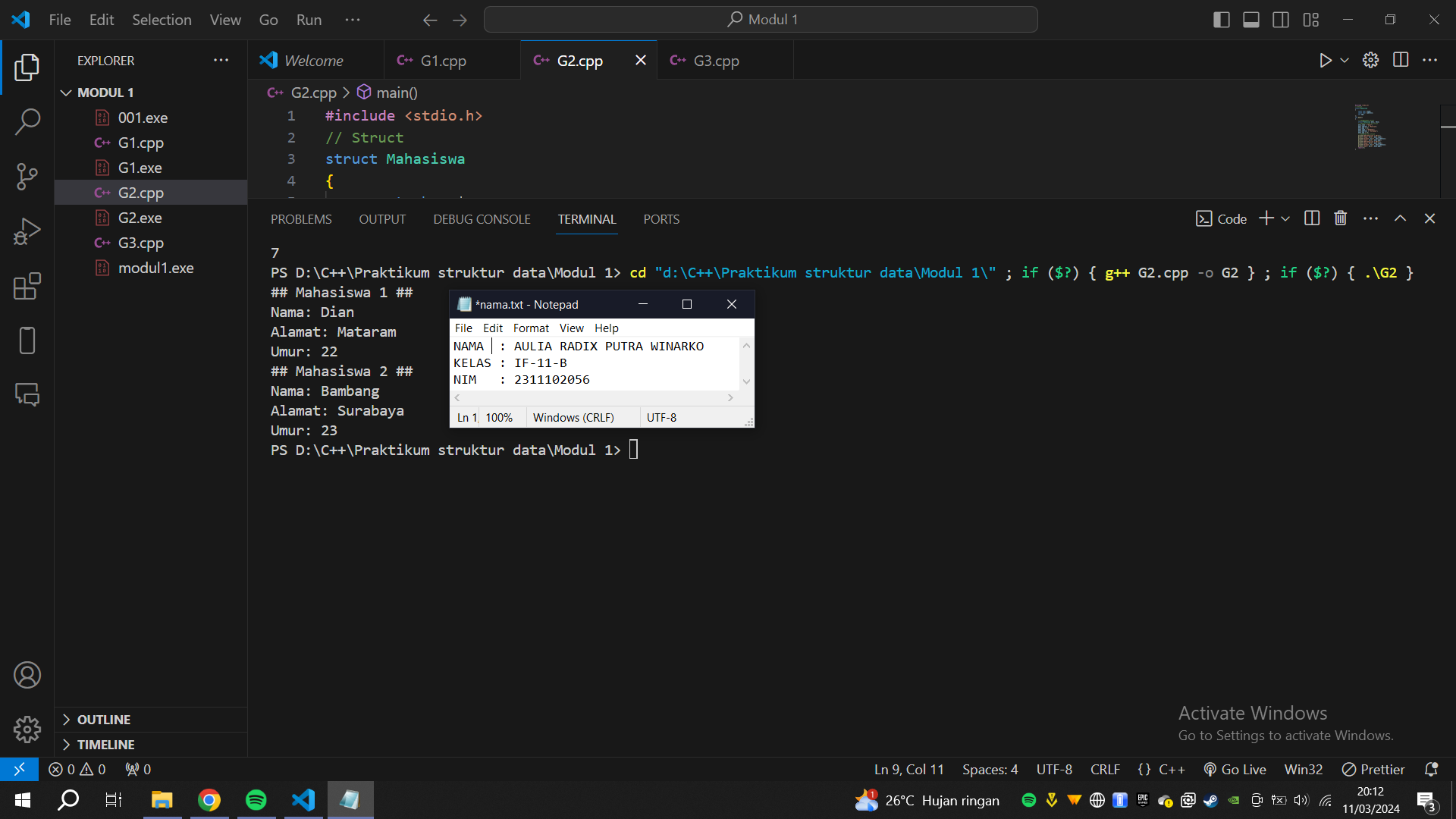
Deskripsi:

Kode yang Anda berikan adalah program C++ yang mengimplementasikan fungsi sederhana untuk melakukan operasi matematika dasar (+, -, \*, /) pada dua angka yang diinputkan oleh pengguna.

Guided 2

|  |
| --- |
| #include <stdio.h>  // Struct  struct Mahasiswa  {      const char \*name;      const char \*address;      int age;  };  int main()  {      // menggunakan struct      struct Mahasiswa mhs1, mhs2;      // mengisi nilai ke struct      mhs1.name = "Dian";      mhs1.address = "Mataram";      mhs1.age = 22;      mhs2.name = "Bambang";      mhs2.address = "Surabaya";      mhs2.age = 23;      // mencetak isi struct      printf("## Mahasiswa 1 ##\n");      printf("Nama: %s\n", mhs1.name);      printf("Alamat: %s\n", mhs1.address);      printf("Umur: %d\n", mhs1.age);      printf("## Mahasiswa 2 ##\n");      printf("Nama: %s\n", mhs2.name);      printf("Alamat: %s\n", mhs2.address);      printf("Umur: %d\n", mhs2.age);      return 0;  } |

Screenshots Output



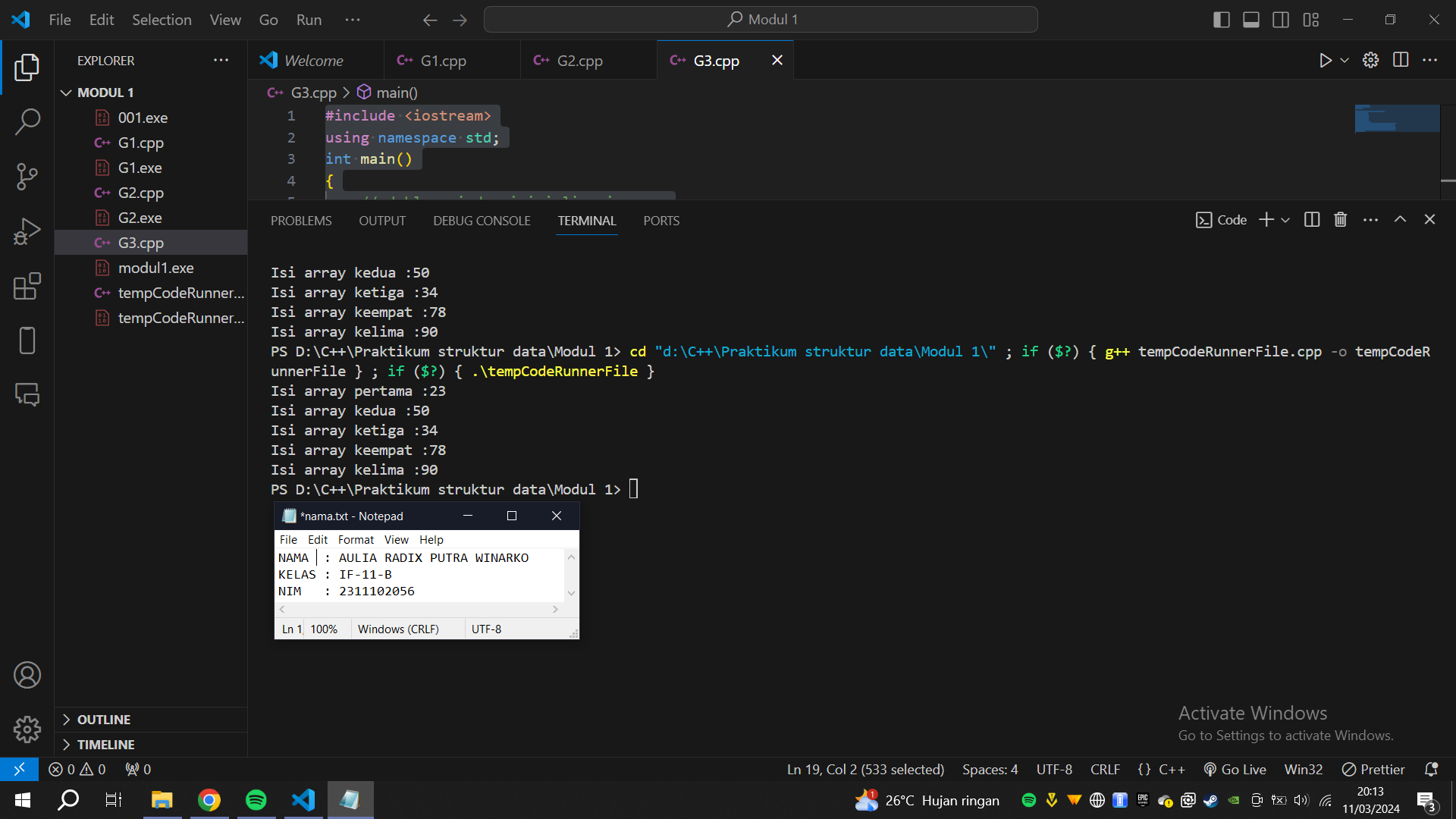
Deskripsi:

kode tersebut menggunakan struct Mahasiswa untuk menyimpan informasi tentang mahasiswa, yaitu nama, alamat, dan umur. Kemudian, user mengisi nilai elemen dari mhs1 dan mhs2 dengan informasi mahasiswa yang telah ditentukan. Selanjutnya, mencetak informasi mahasiswa yang telah di simpan dalam variabel mhs1 dan mhs2 menggunakan perintah printf dengan format yang sesuai.

Guided 3

|  |
| --- |
| #include <iostream>  using namespace std;  int main()  {      // deklarasi dan inisialisasi array      int nilai[5];      nilai[0] = 23;      nilai[1] = 50;      nilai[2] = 34;      nilai[3] = 78;      nilai[4] = 90;      // mencetak array      cout << "Isi array pertama :" << nilai[0] << endl;      cout << "Isi array kedua :" << nilai[1] << endl;      cout << "Isi array ketiga :" << nilai[2] << endl;      cout << "Isi array keempat :" << nilai[3] << endl;      cout << "Isi array kelima :" << nilai[4] << endl;      return 0;  } |

Screenshots Output



Deskripsi:

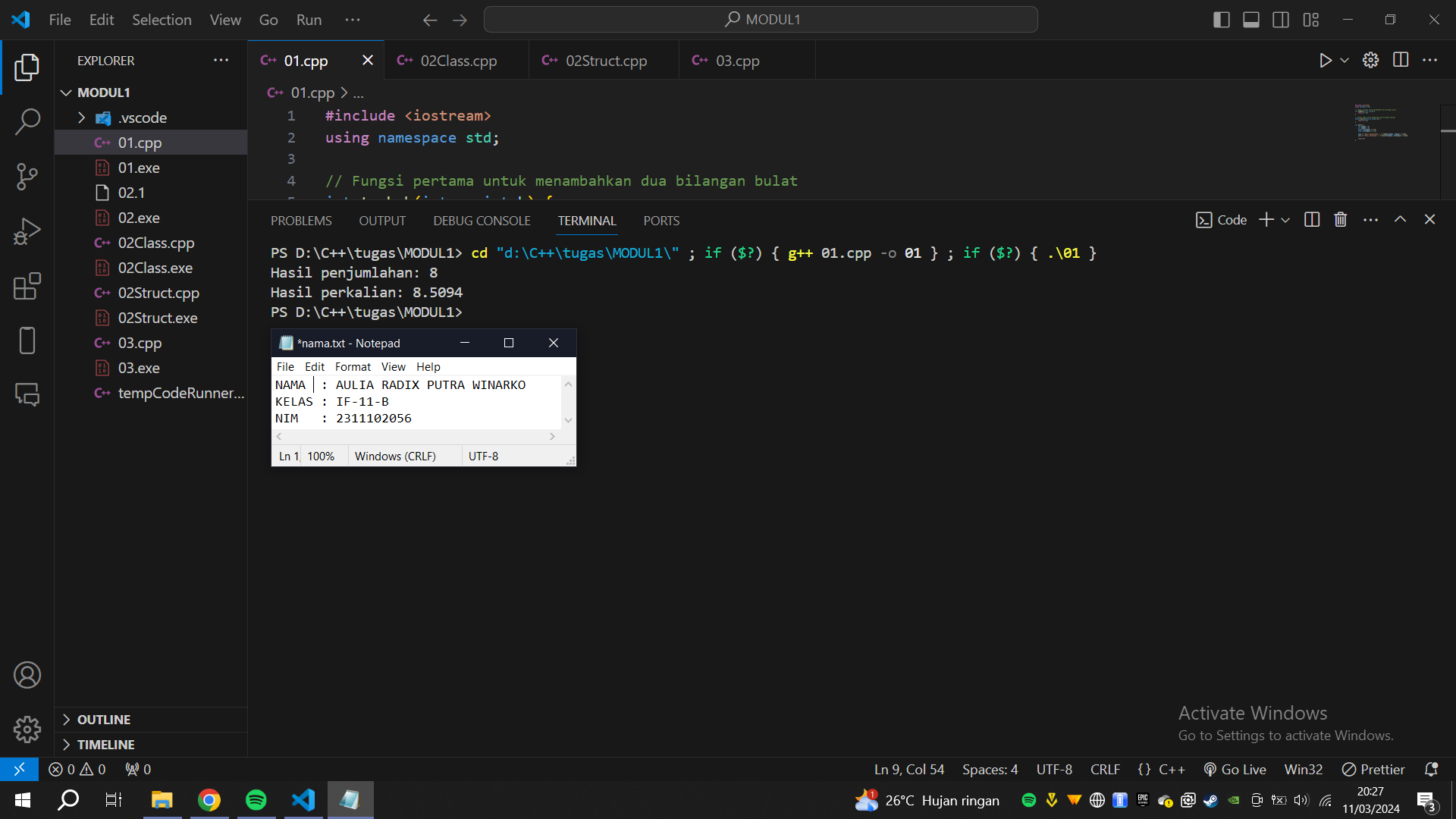
Kode tersebut, menggunakan array nilai untuk menyimpan beberapa nilai integer. Kemudian, user mengisi nilai elemen dari nilai dengan beberapa nilai integer yang telah kita tentukan. Selanjutnya, mencetak nilai elemen dari nilai menggunakan perintah cout dengan format yang sesuai.

1. Unguided/Tugas (berisi screenshot source code & output program disertai penjelasannya)

Unguided 1

|  |
| --- |
| #include <iostream>  using namespace std;  // Fungsi pertama untuk menambahkan dua bilangan bulat  int tambah(int a, int b) {      return a + b;  }  // Fungsi kedua untuk mengalikan dua bilangan desimal  double kali(double a, double b) {      return a \* b;  }  int main() {      int angka1 = 5;      int angka2 = 3;      double bilangan1 = 3.14;      double bilangan2 = 2.71;      cout << "Hasil penjumlahan: " << tambah(angka1, angka2) << endl;      cout << "Hasil perkalian: " << kali(bilangan1, bilangan2) << endl;      return 0;  } |

Screenshots Output



Deskripsi:

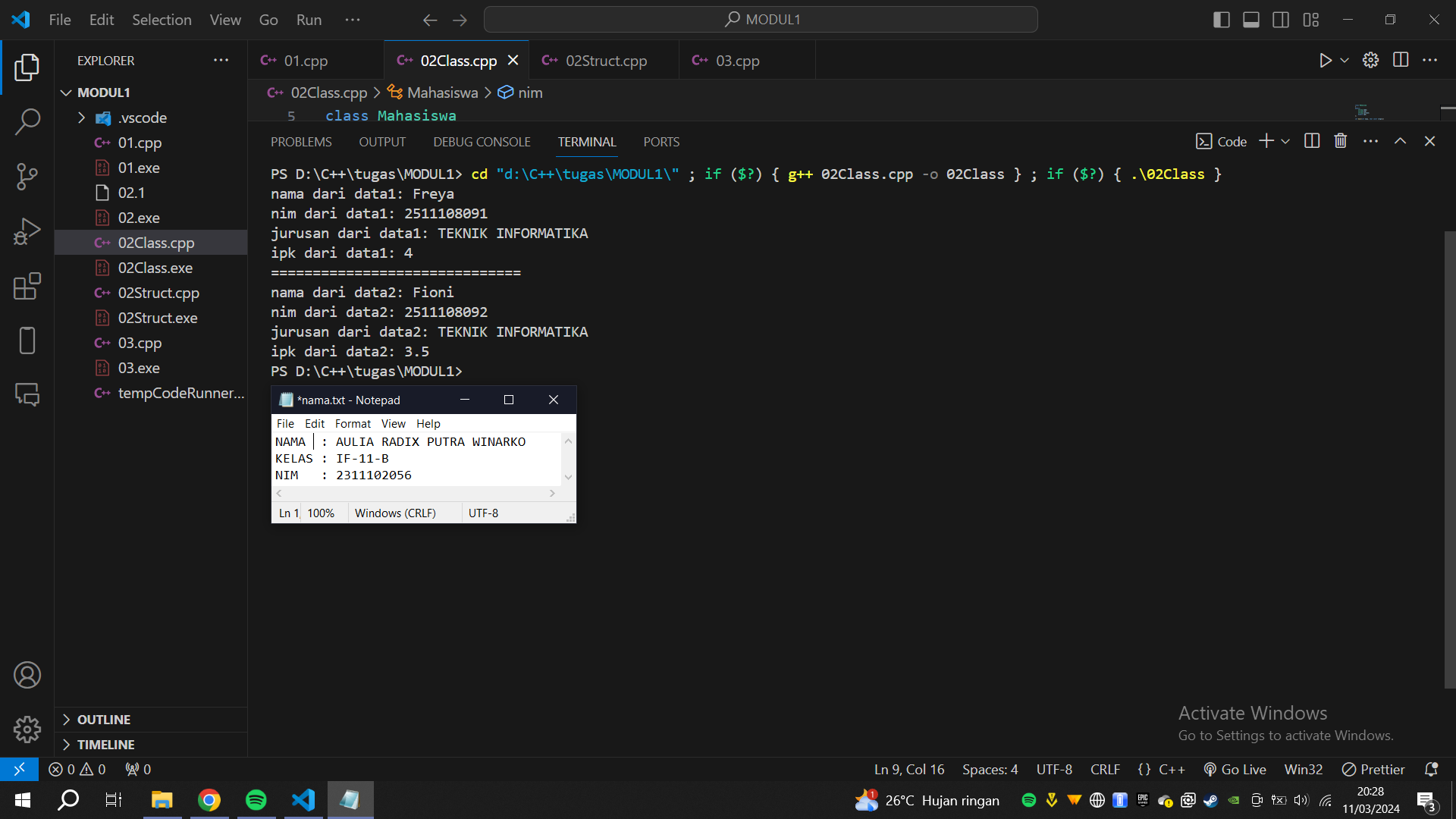
Kode tersebut, menggunakan dua fungsi, yaitu tambah dan kali, untuk melakukan operasi matematika dasar. Kemudian, mendeklarasikan beberapa variabel dan menginisialisasikannya dengan beberapa nilai yang telah ditentukan. Selanjutnya, menggunakan fungsi tambah dan kali untuk melakukan operasi matematika terhadap variabel tersebut. Hasil dari operasi tersebut kemudian dicetak menggunakan perintah cout dengan format yang sesuai.

Unguided 2

* Class

|  |
| --- |
| #include <iostream>  #include <string>  using namespace std;  class Mahasiswa  {  public:      string nama;      string nim;      string jurusan;      double ipk;  };  int main(int argc, char const \*argv[])  {      Mahasiswa data1;      data1.nama = "Freya";      data1.nim = "2511108091";      data1.jurusan = "TEKNIK INFORMATIKA";      data1.ipk = 4;      Mahasiswa data2;      data2.nama = "Fioni";      data2.nim = "2511108092";      data2.jurusan = "TEKNIK INFORMATIKA";      data2.ipk = 3.5;      cout << "nama dari data1: " << data1.nama << endl;      cout << "nim dari data1: " << data1.nim << endl;      cout << "jurusan dari data1: " << data1.jurusan << endl;      cout << "ipk dari data1: " << data1.ipk << endl;      cout << "==============================" << endl;      cout << "nama dari data2: " << data2.nama << endl;      cout << "nim dari data2: " << data2.nim << endl;      cout << "jurusan dari data2: " << data2.jurusan << endl;      cout << "ipk dari data2: " << data2.ipk << endl;      return 0;  } |

Screenshots Output



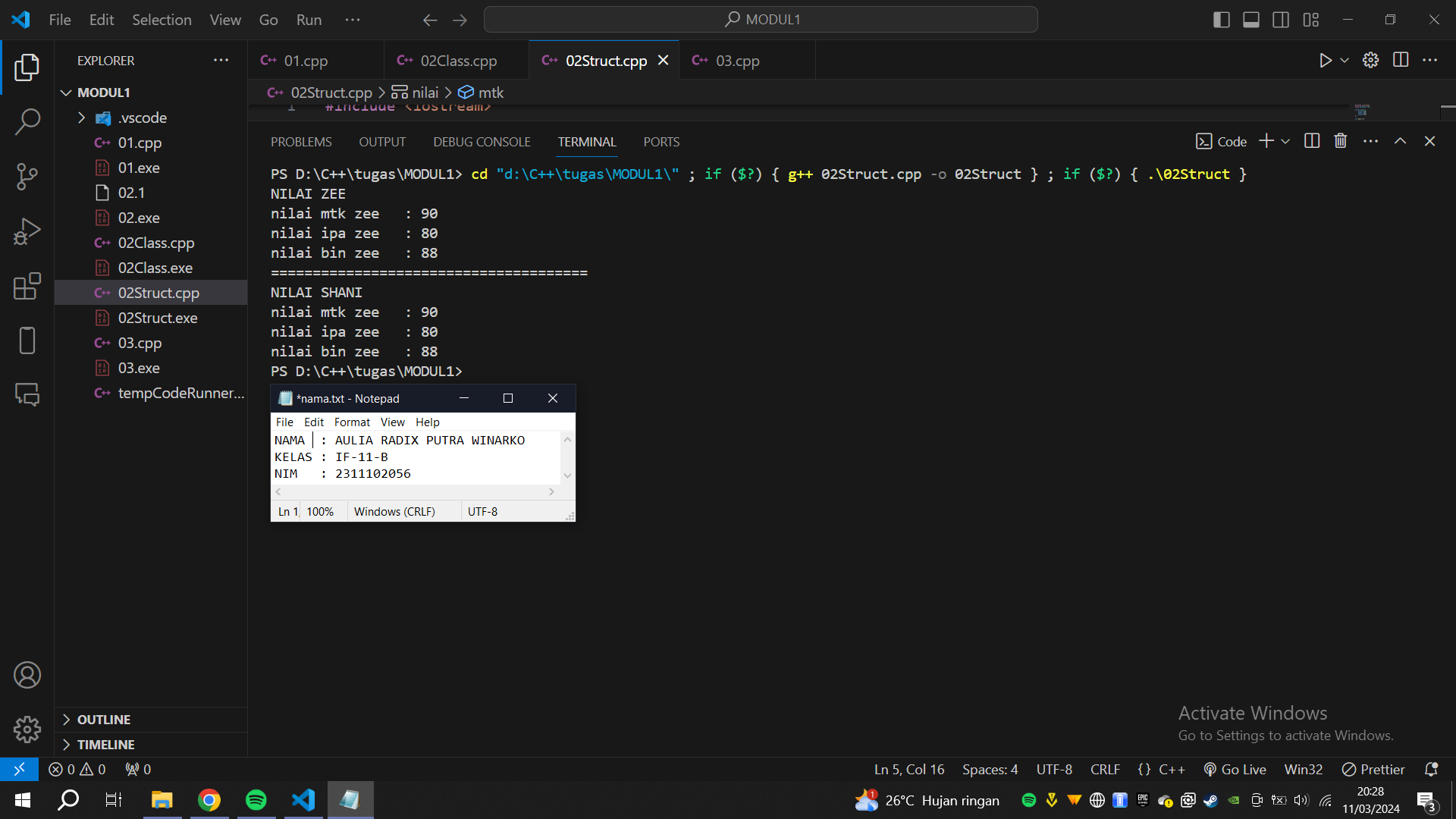
Deskripsi:

Program tersebut menggambarkan penggunaan class untuk merepresentasikan objek-objek dalam bahasa C++. Dalam kasus ini, class Mahasiswa digunakan untuk merepresentasikan data mahasiswa, dan objek-objek data1 dan data2 merepresentasikan dua mahasiswa yang berbeda dengan atribut-atribut yang berbeda pula.

* Struct

|  |
| --- |
| #include <iostream>  using namespace std;  struct nilai {      double mtk;      double ipa;      double bin;  };  int main (){      nilai zee;      nilai shani;      zee.mtk = 90;      zee.ipa = 80;      zee.bin = 88;      shani.mtk = 100;      shani.ipa = 100;      shani.bin = 100;        cout << "NILAI ZEE" << endl;      cout << "nilai mtk zee \t: " << zee.mtk << endl;      cout << "nilai ipa zee \t: " << zee.ipa << endl;      cout << "nilai bin zee \t: " << zee.bin << endl;      cout << "======================================" << endl;      cout << "NILAI SHANI" << endl;      cout << "nilai mtk zee \t: " << zee.mtk << endl;      cout << "nilai ipa zee \t: " << zee.ipa << endl;      cout << "nilai bin zee \t: " << zee.bin << endl;  }; |

Screenshots Output



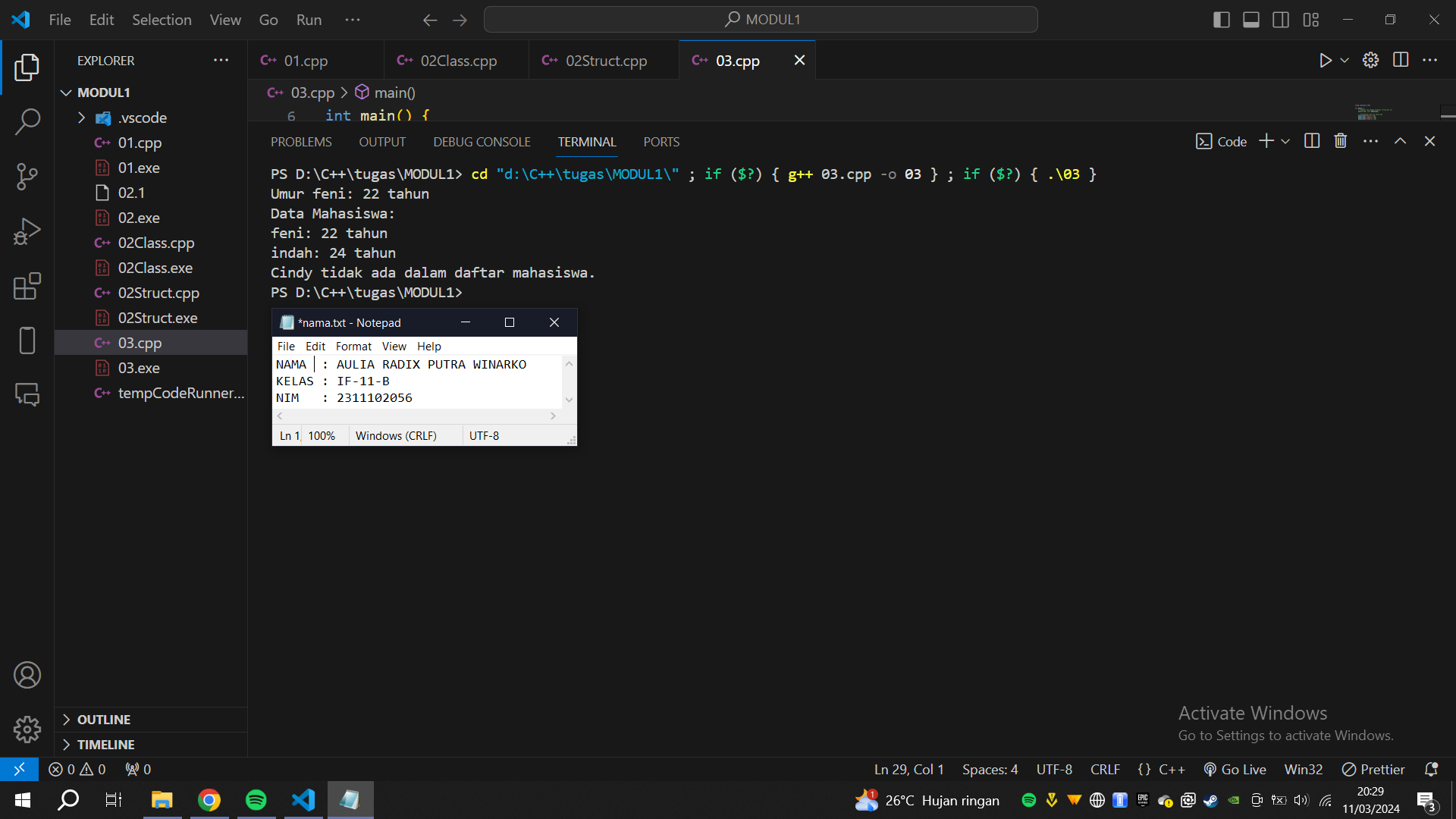
Deskripsi:

Program di atas adalah contoh penggunaan struktur (struct) dalam C++ untuk menyimpan nilai-nilai mata pelajaran dari dua siswa, yaitu "Zee" dan "Shani". Setiap siswa memiliki nilai untuk mata pelajaran matematika (mtk), ilmu pengetahuan alam (ipa), dan bahasa Indonesia (bin).

Unguided 3

|  |
| --- |
| #include <iostream>  #include <map>  #include <string>  using namespace std;  int main() {      // Membuat map dengan pasangan string dan int      map<string, int> mahasiswa;      // Menambahkan data ke dalam map      mahasiswa["feni"] = 22;      mahasiswa["indah"] = 23;      mahasiswa["Cindy"] = 21;      // Mengakses data dalam map      cout << "Umur feni: " << mahasiswa["feni"] << " tahun" << endl;      // Mengubah data dalam map      mahasiswa["indah"] = 24;      // Menghapus data dalam map      mahasiswa.erase("Cindy");      // Iterasi melalui semua elemen map      cout << "Data Mahasiswa:" << endl;      for (const auto& pair : mahasiswa) {          cout << pair.first << ": " << pair.second << " tahun" << endl;      }      // Mengecek apakah sebuah kunci ada dalam map      if (mahasiswa.find("Cindy") != mahasiswa.end()) {          cout << "Cindy ada dalam daftar mahasiswa." << endl;      } else {          cout << "Cindy tidak ada dalam daftar mahasiswa." << endl;      }      return 0;  } |

Screenshots Output



Deskripsi:

Kode tersebut merupakan contoh penggunaan struktur data map dalam C++. map adalah struktur data yang mengimplementasikan asosiasi berupa pasangan kunci-nilai yang terurut berdasarkan kunci. Dalam contoh ini, kunci-kunci adalah string yang mewakili nama mahasiswa, dan nilainya adalah integer yang mewakili usia mahasiswa.

1. Kesimpulan

Kesimpulan dari data dan algoritma adalah bahwa mereka menjadi dasar penting. Data struct, seperti array dan stack, memungkinkan kita untuk mengorganisir dan mengelola data dengan efisiensi tinggi, sementara algoritma menentukan langkah-langkah yang diperlukan untuk mengatasi masalah tertentu.

1. Referensi[Bob

Karumanchi, N. (2016). *Data Structures and algorithms made easy: Concepts, problems, Interview Questions*. CareerMonk Publications.

TylerMSFT. (n.d.). Collections (C++/CX). diakses dari https://learn.microsoft.com/en-us/cpp/cppcx/collections-c-cx?view=msvc-170ot Penilaian sebesar 15%