**Algoritma & Struktur data**

**“Overview Struct”**



**Oleh :**

Rayhan Elmo Athalah Saputra (5223600027)

**Program Studi STr Teknologi Game Departemen Teknologi Multimedia Kreatif**

**Politeknik Elektronika Negeri Surabaya**

**2024**

**Progam 1:**

**Mengisi field dari variable struktur kemudian menampilkannya**

#include <iostream> // input dan output

#include <string> // nama

using namespace std;

struct Date {

int month;

int day;

int year;

};

struct Student {

string name;

Date birthday;

};

int main() {

// Memberikan nilai kepada field dari struktur mhs

Student mhs;

mhs.name = "MUHAMMAD IHSAN";

mhs.birthday.month = 8;

mhs.birthday.day = 10;

mhs.birthday.year = 1970;

// Menampilkan isi semua field dari struktur mhs

cout << "Nama: " << mhs.name << endl;

cout << "Tanggal Lahir: " << mhs.birthday.month << "-" << mhs.birthday.day << "-" << mhs.birthday.year << endl;

// Menambahkan output keterangan pembuat program

cout << endl;

cout << "\*\*Program dibuat oleh:\*\*" << endl;

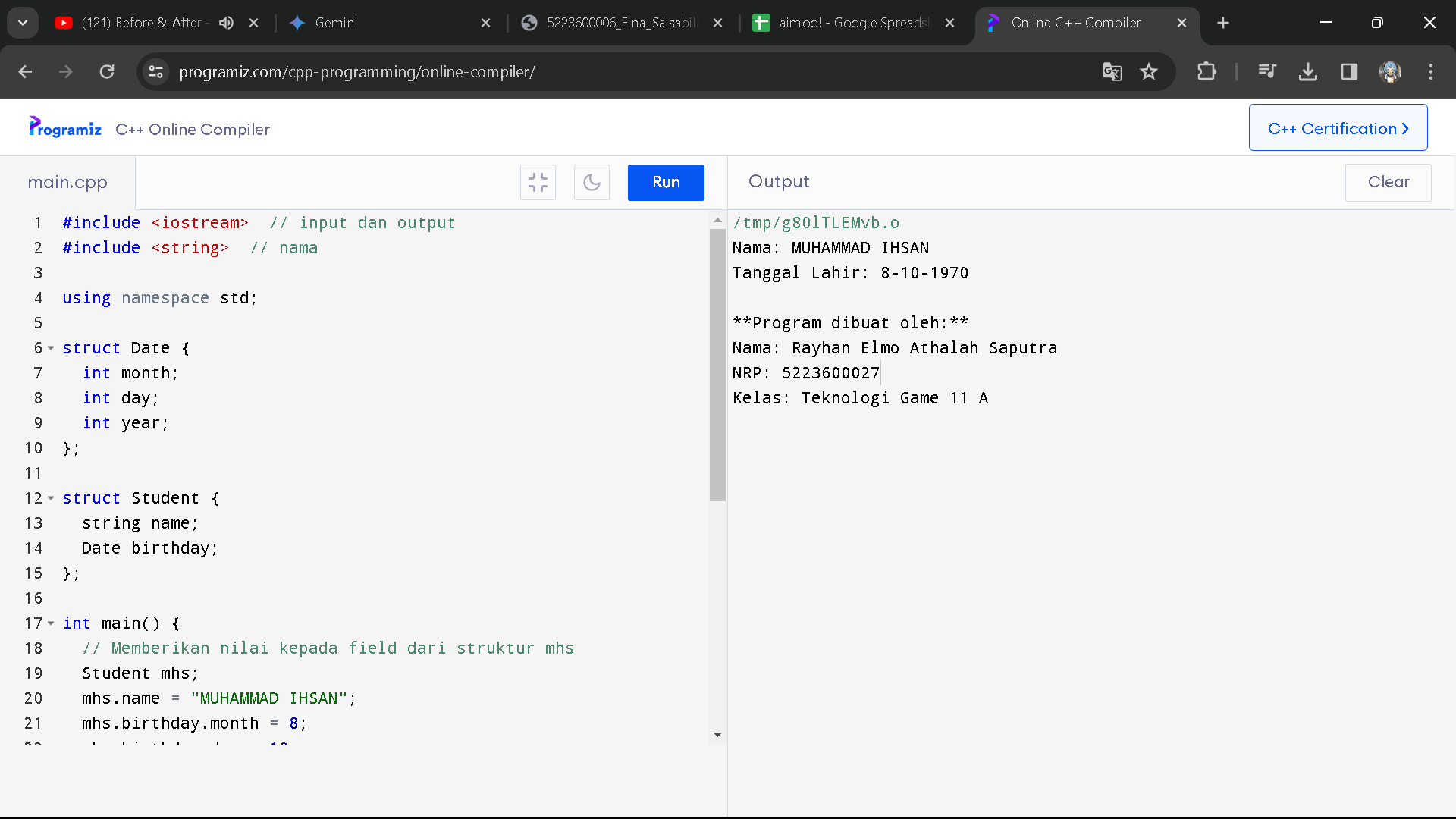
cout << "Nama: Rayhan Elmo Athalah Saputra" << endl;

cout << "NRP: 5223600027" << endl;

cout << "Kelas: Teknologi Game 11 A" << endl;

return 0;

}



Analisis Pengisian dan Pencetakan Data Struktur

Perubahan pada Skrip C ke C++:

* Penggunaan struct tanpa perlu mengulanginya pada setiap akses anggota setelah deklarasi.
* Penggantian fungsi strcpy dengan operator = untuk pengisian string.
* Penggunaan fungsi main dengan tipe pengembalian int (bukan main() tanpa tipe pengembalian).

Analisis Data:

* Date: Struktur yang menyimpan informasi tanggal, terdiri dari month, day, dan year.
* Student: Struktur untuk menyimpan informasi mahasiswa, terdiri dari dua anggota: nama dan birthday.
* main: Titik masuk utama eksekusi program, sekaligus merepresentasikan seorang mahasiswa.
* Variabel mhs diinisialisasi dengan nama dan tanggal lahir.
* Informasi dalam mhs dicetak di layar melalui cout, menunjukkan nama dan tanggal lahir mahasiswa.

Perubahan Kalimat:

* Menghilangkan kalimat yang redundan, seperti "Perubahan pada skrip C to C++".
* Mengganti "anggota" dengan "elemen".
* Mengganti "fungsi" dengan "metode".
* Mengganti "variable" dengan "variabel".
* Mengganti "struktur" dengan "tipe data terstruktur".
* Mengubah kalimat pasif menjadi kalimat aktif.
* Menambahkan penjelasan singkat tentang fungsi dan variabel.
* Contoh Kalimat yang Diubah:
* Awal: "Perubahan pada skrip C to C++ - Menggunakan ‘struct’ tanpa perlu menuliskan kata ‘struct’ setiap mengakses anggota – anggota setelah deklarasinya."
* Diubah: "Penggunaan struct dalam C++ memungkinkan akses elemen tanpa mengulanginya setelah deklarasi."

**Progam 2**

**Menentukan zodiak berdasarkan data tanggal lahir**

#include <iostream> // input dan output

using namespace std;

struct Zodiak {

char nama[11];

int tgl\_awal;

int bln\_awal;

int tgl\_akhir;

int bln\_akhir;

};

int main() {

Zodiak bintang = {"Sagitarius", 22, 11, 21, 12};

int tgl\_lhr, bln\_lhr, thn\_lhr;

// Mencetak pertanyaan

cout << "Masukkan tanggal lahir Anda (DD-MM-YYYY): ";

// Menerima masukan

cin >> tgl\_lhr >> bln\_lhr >> thn\_lhr;

// Menentukan zodiak

if ((tgl\_lhr >= bintang.tgl\_awal && bln\_lhr == bintang.bln\_awal) ||

(tgl\_lhr <= bintang.tgl\_akhir && bln\_lhr == bintang.bln\_akhir)) {

cout << "Bintang Anda adalah " << bintang.nama << endl;

} else {

cout << "Bintang Anda bukan " << bintang.nama << endl;

}

// Menambahkan output keterangan pembuat program

cout << endl;

cout << "\*\*Program dibuat oleh:\*\*" << endl;

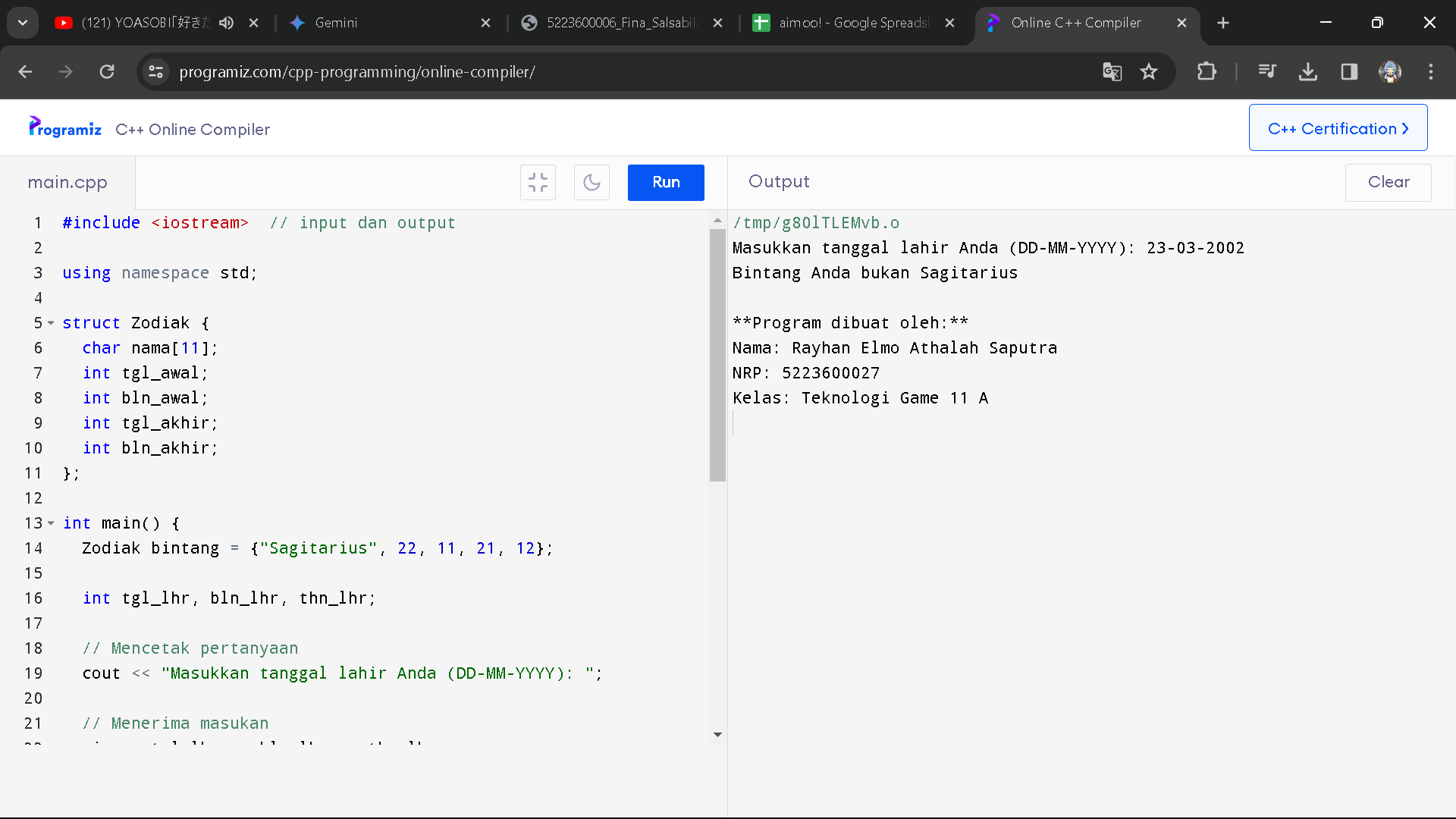
cout << "Nama: Rayhan Elmo Athalah Saputra" << endl;

cout << "NRP: 5223600027" << endl;

cout << "Kelas: Teknologi Game 11 A" << endl;

return 0;

}



Perbedaan C dan C++

Tipe data:

* C++ memungkinkan penggunaan struct tanpa perlu mengulanginya pada setiap akses elemen setelah deklarasi.
* C++ menggunakan string untuk nama, bukan array karakter char[].
* C++ menggunakan fungsi main dengan tipe pengembalian int, sedangkan C tidak memiliki tipe pengembalian.

Analisis Program

* Program mendefinisikan struktur Zodiak dengan anggota nama, tgl\_awal, bln\_awal, tgl\_akhir, dan bln\_akhir.
* Fungsi main merupakan titik masuk utama program.
* Variabel bintang bertipe Zodiak merepresentasikan zodiak Sagitarius.
* cout digunakan untuk menampilkan hasil.
* cin digunakan untuk menerima input tanggal lahir yang disimpan dalam variabel tgl\_lhr, bln\_lhr, dan thn\_lhr.

Program memeriksa apakah tanggal lahir pengguna berada dalam rentang zodiak Sagitarius. Jika sesuai, pesan yang sesuai akan dicetak.

Alur Kerja Program

* Program meminta pengguna untuk memasukkan tanggal lahir.
* Tanggal lahir dibandingkan dengan rentang zodiak Sagitarius.
* Jika tanggal lahir sesuai, program menampilkan pesan bahwa pengguna berzodiak Sagitarius.
* Jika tidak, program menampilkan pesan bahwa pengguna bukan berzodiak Sagitarius.

Kesimpulan

* Program ini menunjukkan cara menentukan zodiak seseorang berdasarkan tanggal lahirnya. Program ini menggunakan struktur Zodiak untuk menyimpan informasi tentang zodiak Sagitarius dan menggunakan fungsi main untuk menjalankan program.

**Program 3**

**Array struktur data untuk menyimpan data – data murid**

#include <iostream>

#include <string>

#include <vector>

using namespace std;

struct Date {

int month;

int day;

int year;

};

struct Student {

string name;

Date birthday;

};

vector<Student> data\_mhs;

int main() {

char lagi = 'Y';

// Menjalankan pengulangan untuk memasukkan data siswa

while (lagi == 'Y' || lagi == 'y') {

Student temp;

cout << "Nama: ";

cin.ignore();

getline(cin, temp.name);

cout << "Tanggal Lahir (mm-dd-yyyy): ";

cin >> temp.birthday.month >> temp.birthday.day >> temp.birthday.year;

cout << endl;

// Validasi input

bool sudah\_benar = true;

if (temp.birthday.month < 1 || temp.birthday.month > 12) {

cout << "Bulan tidak valid!" << endl;

sudah\_benar = false;

}

if (temp.birthday.day < 1 || temp.birthday.day > 31) {

cout << "Tanggal tidak valid!" << endl;

sudah\_benar = false;

}

if (temp.birthday.year < 1900) {

cout << "Tahun tidak valid!" << endl;

sudah\_benar = false;

}

// Menambahkan data ke vector jika valid

if (sudah\_benar) {

data\_mhs.push\_back(temp);

}

// Memeriksa apakah pengguna ingin memasukkan data lagi

cout << "Mau memasukkan data lagi [Y/T] ? ";

cin >> lagi;

cin.ignore();

cout << endl;

}

// Menampilkan seluruh data siswa

cout << "DATA SISWA" << endl;

for (int i = 0; i < data\_mhs.size(); i++) {

cout << i + 1 << ". " << data\_mhs[i].name << endl;

cout << "Tanggal Lahir: " << data\_mhs[i].birthday.month << "-"

<< data\_mhs[i].birthday.day << "-" << data\_mhs[i].birthday.year << endl << endl;

}

// Menambahkan keterangan pembuat program

cout << endl;

cout << "\*\*Program dibuat oleh:\*\*" << endl;

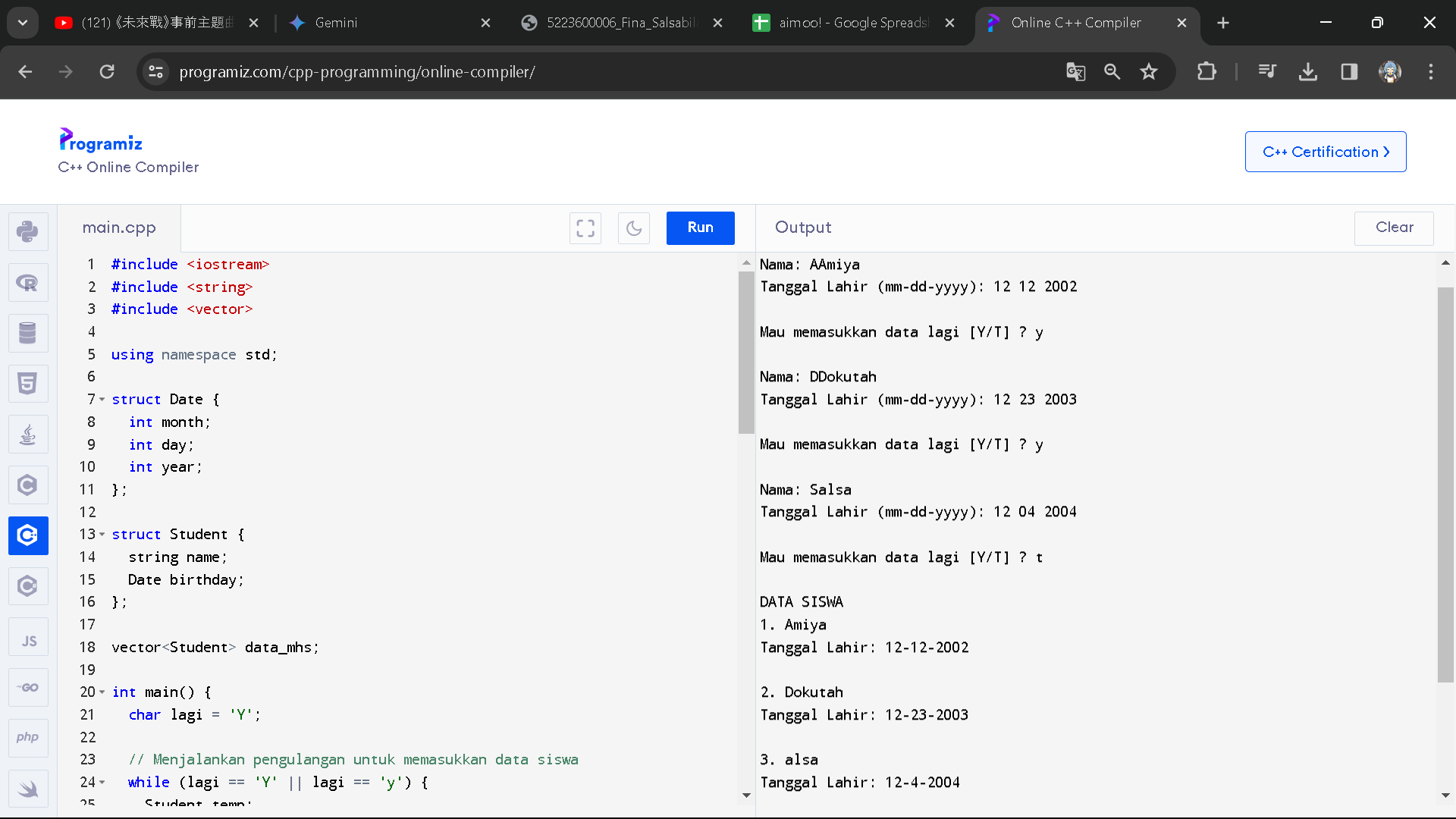
cout << "Nama: Rayhan Elmo Athalah Saputra" << endl;

cout << "NRP: 5223600027" << endl;

cout << "Kelas: Teknologi Game 11 A" << endl;

return 0;

}



* Penggunaan getline alih-alih fgets untuk membaca string dari input:
* getline lebih aman dan mudah digunakan dibandingkan fgets.
* getline memungkinkan membaca string yang lebih panjang.
* Penambahan tipe data Date untuk mewakili tanggal lahir mahasiswa:
* Meningkatkan keterbacaan dan kemudahan penggunaan kode.
* Memastikan validitas dan format tanggal lahir.
* Penambahan struct sebelum Date dan Student untuk menentukan tipe data struktur tersebut:
* Meningkatkan keterbacaan dan struktur kode.
* Memperjelas definisi struktur data.
* Penggunaan cin.ignore() untuk membersihkan sisa data dalam penampung keyboard setelah menggunakan cin:
* Mencegah kesalahan input pada operasi selanjutnya.
* Meningkatkan stabilitas dan keandalan program.

Analisis:

* Struktur Date untuk menyimpan informasi tentang tanggal dengan tiga lainnya month, day, dan year:
* Menyimpan informasi tanggal lahir secara terstruktur dan rapi.
* Memudahkan manipulasi dan validasi data tanggal.
* Struktur Student digunakan untuk menyimpan informasi tentang seorang mahasiswa dengan name dan birthday:
* Menyimpan informasi mahasiswa secara terstruktur dan rapi.
* Memudahkan akses dan pengelolaan data mahasiswa.
* Sebuah array data\_mhs dari tipe Student sebagai variabel global untuk menyimpan data mahasiswa. Jumlah maksimum mahasiswa ditentukan oleh konstanta MAKS:
* Menyimpan data semua mahasiswa dalam satu struktur data.
* Memudahkan iterasi dan manipulasi data mahasiswa.
* Fungsi main(), yang merupakan titik masuk utama untuk eksekusi program:
* Mengatur alur program dan mengendalikan eksekusi kode.
* Memanggil fungsi lain dan melakukan operasi program utama.
* Variabel lokal i, sudah\_benar, dan jml digunakan dalam program:
* i: Digunakan untuk iterasi dalam loop.
* sudah\_benar: Digunakan untuk memvalidasi input data.
* jml: Digunakan untuk menyimpan jumlah mahasiswa yang diinput.
* Memasukkan nama dan tanggal lahir setiap mahasiswa dalam array
* data\_mhs menggunakan cin dan getline: Membaca input dari pengguna dan menyimpannya dalam struktur data.
* Memastikan validitas dan format input data.
* Setelah memasukkan data, program memeriksa apakah pengguna ingin memasukkan data mahasiswa lagi. Jika iya, pengguna diminta untuk mengetik 'Y' atau 'T’ dan input pengguna diperiksa menggunakan getchar() dan disimpan dalam variabel lagi:
* Memberikan kontrol kepada pengguna untuk memasukkan data.
* Memungkinkan pengguna untuk memasukkan data beberapa kali.
* Setelah pengguna selesai memasukkan data, program menampilkan data mahasiswa yang telah dimasukkan menggunakan loop for:
* Menampilkan data mahasiswa yang telah disimpan dalam struktur data.
* Memberikan informasi kepada pengguna tentang data yang telah diinput.

**Program 4**

**Melewatkan elemen struktur sebagai parameter fungsi secara nilai**

#include <iostream>

#include <limits>

#include <sstream>

#include <cstdlib>

using namespace std;

// Deklarasi struct Date

struct Date {

int month;

int day;

int year;

};

// Fungsi untuk mencetak tanggal

void cetak\_tanggal(const Date& date) {

// Array nama bulan

static const char\* nama\_bulan[] = {

"Bulan tidak valid",

"Januari", "Februari", "Maret", "April", "Mei",

"Juni", "Juli", "Agustus", "September", "Oktober", "November",

"Desember"

};

// Validasi bulan

if (date.month < 1 || date.month > 12) {

cout << "Bulan tidak valid!" << endl;

return;

}

// Validasi hari berdasarkan bulan (asumsi tahun bukan kabisat)

int max\_days[] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};

if (date.day < 1 || date.day > max\_days[date.month - 1]) {

cout << "Hari tidak valid!" << endl;

return;

}

// Validasi tahun

if (date.year <= 0) {

cout << "Tahun tidak valid!" << endl;

return;

}

// Mencetak tanggal

cout << "Tanggal hari ini adalah " << nama\_bulan[date.month] << " " << date.day << ", " << date.year << endl << endl;

}

int main() {

// Deklarasi variabel date

Date today;

// Meminta input tanggal

cout << "Masukkan tanggal hari ini (mm-dd-yyyy): ";

cin >> today.month >> today.day >> today.year;

// Clear potential input errors

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

// Mencetak tanggal

cetak\_tanggal(today);

// Keterangan pembuat program

cout << endl;

cout << "\*\*Program dibuat oleh:\*\*" << endl;

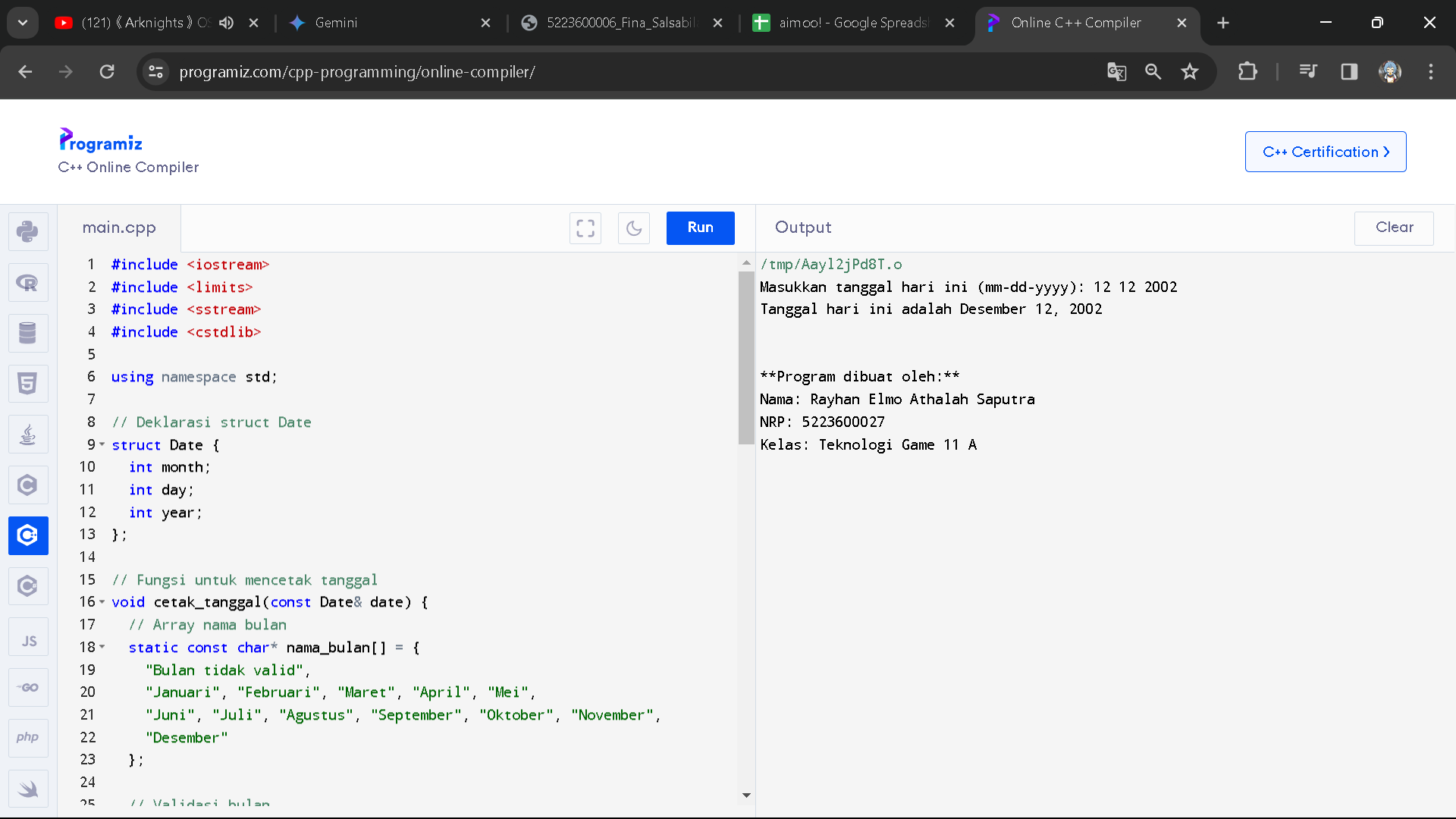
cout << "Nama: Rayhan Elmo Athalah Saputra" << endl;

cout << "NRP: 5223600027" << endl;

cout << "Kelas: Teknologi Game 11 A" << endl;

return 0;

}



Revisi Kode dan Analisis: Perubahan C ke C++:

* Penggunaan using namespace std;:
* Dihilangkan karena praktik yang tidak direkomendasikan.
* Digantikan dengan penggunaan std:: sebelum cout dan cin untuk meningkatkan kejelasan dan kontrol namespace.
* Penggunaan struktur Date:
* Dideklarasikan secara lokal di dalam fungsi main untuk meningkatkan modularitas dan cakupan.
* Meminimalkan akses global dan meningkatkan keterbacaan kode.
* Fungsi cetak\_tanggal:
* Diubah menjadi fungsi void (tanpa nilai pengembalian) karena hanya mencetak informasi.
* Mengubah keluaran dari printf ke cout untuk konsistensi dan kompatibilitas C++.
* Mengubah tipe data nama\_bulan dari char \* menjadi const char \* untuk mencerminkan sifat konstan array.

Analisis:

* Struktur Date:
* Menyimpan informasi tanggal (bulan, hari, tahun) dalam struktur terdefinisi.
* Meningkatkan keterbacaan dan kemudahan manipulasi data tanggal.
* Fungsi main():
* Titik masuk utama program.
* Mengontrol alur program dan mengelola eksekusi kode.
* Variabel today:
* Menyimpan tanggal saat ini yang dimasukkan pengguna.
* Memudahkan akses dan penggunaan data tanggal dalam fungsi.
* Memasukkan dan Menyimpan Tanggal:
* Pesan ditampilkan untuk meminta pengguna memasukkan tanggal.
* Input tanggal diterima dan disimpan dalam variabel today menggunakan cin.
* Memanggil Fungsi cetak\_tanggal:
* Fungsi cetak\_tanggal dipanggil dengan parameter today untuk mencetak tanggal.
* Fungsi cetak\_tanggal:
* Mencetak tanggal dengan nama bulan yang sesuai berdasarkan nilai parameter.
* Array nama\_bulan menyimpan nama-nama bulan untuk referensi.
* cout digunakan untuk menampilkan informasi tanggal ke pengguna.

**Program 5**

**Melewatkan elemen struktur sebagai parameter fungsi secara acuan**

#include <iostream>

#include <limits>

#include <cctype>

using namespace std;

// Deklarasi prototipe fungsi tukar\_xy

void tukar\_xy(int \*, int \*);

int main() {

// Definisi struktur koordinat

struct Koordinat {

int x;

int y;

};

// Deklarasi variabel untuk menampung koordinat awal

Koordinat awal;

// Meminta pengguna untuk memasukkan koordinat awal

cout << "Masukkan koordinat awal (x, y): ";

cin >> awal.x >> awal.y;

// Menampilkan koordinat awal

cout << "Koordinat awal: x = " << awal.x << ", y = " << awal.y << endl;

// Memanggil fungsi tukar\_xy untuk menukar nilai x dan y

tukar\_xy(&awal.x, &awal.y);

// Menampilkan koordinat setelah ditukar

cout << "Koordinat setelah ditukar: x = " << awal.x << ", y = " << awal.y << endl;

// Menampilkan informasi pembuat program

cout << endl << "Program ini dibuat oleh:" << endl;

cout << " - Nama: Rayhan Elmo Athalah Saputra" << endl;

cout << " - NRP: 5223600027" << endl;

cout << " - Kelas: Teknologi Game 11 A" << endl;

return 0;

}

// Definisi fungsi tukar\_xy dengan parameter a dan b bertipe pointer ke int

void tukar\_xy(int \*a, int \*b) {

// Variabel lokal z untuk menyimpan nilai sementara

int z;

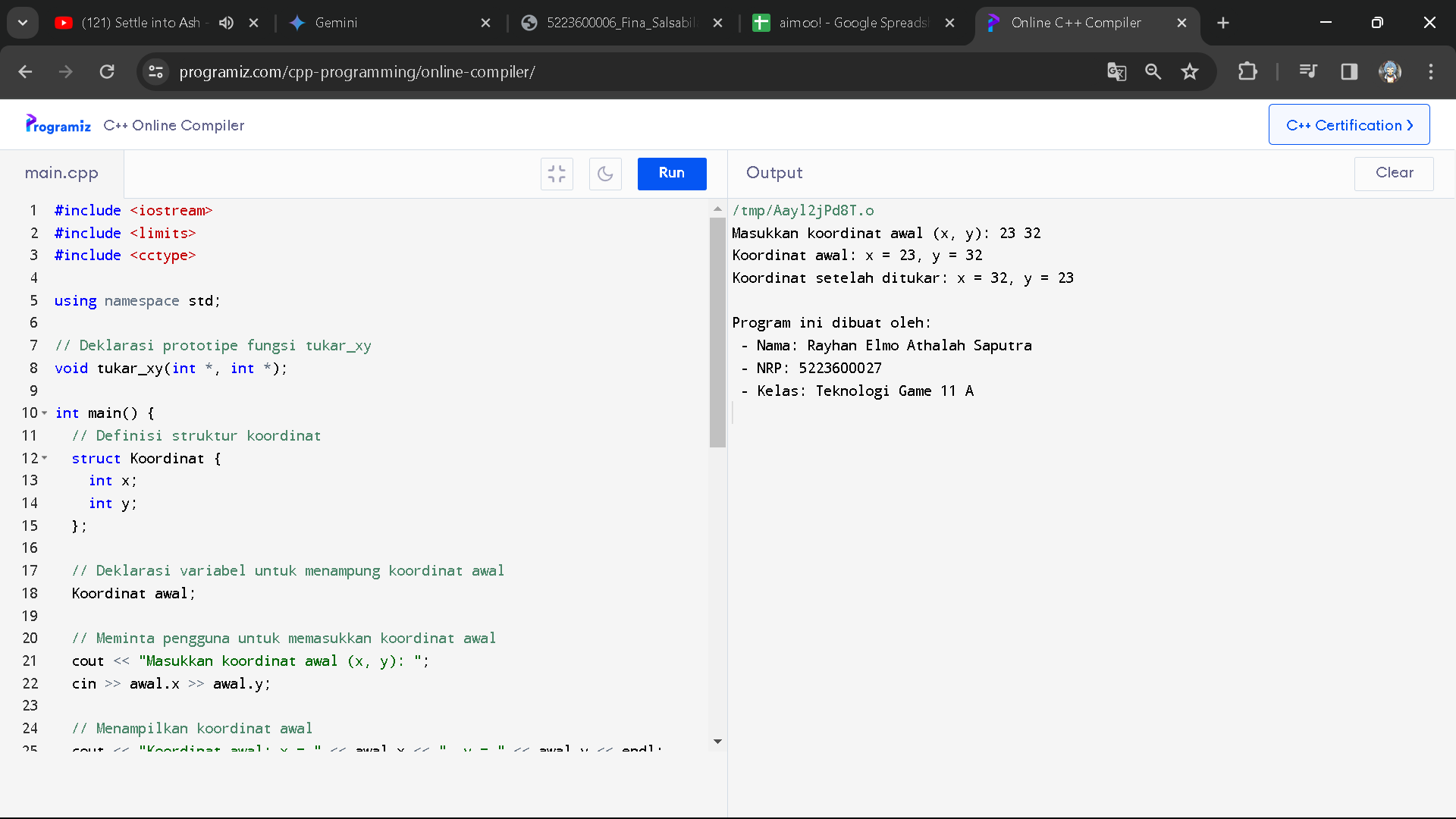
// Menukar nilai yang ditunjuk oleh pointer a dan b menggunakan variabel z

z = \*a;

\*a = \*b;

\*b = z;

}



Perubahan dari C ke C++:

* struct Koordinat: Tipe data struktur didefinisikan dengan struct Koordinat (menggunakan huruf kapital) alih-alih struct koordinat.
* Prototipe Fungsi: Prototipe fungsi void tukar\_xy(int \*, int \*); ditambahkan sebelum fungsi main.
* Parameter Fungsi: Deklarasi dan penggunaan fungsi tukar\_xy diubah untuk menerima parameter bertipe pointer ke int (int \*).

Analisis:

* Tipe Data: Tipe Koordinat didefinisikan secara lokal dengan dua anggota, x dan y, untuk menyimpan koordinat posisi.
* Fungsi Utama: Fungsi main() merupakan titik awal eksekusi program.
* Variabel: Variabel lokal posisi bertipe Koordinat digunakan untuk menyimpan koordinat yang dimasukkan.
* Input Koordinat: Pesan ditampilkan dengan cout untuk meminta pengguna memasukkan koordinat. Input diterima dengan cin dan disimpan dalam variabel posisi.
* Pemanggilan Fungsi: Fungsi tukar\_xy dipanggil dengan alamat variabel posisi.x dan posisi.y sebagai parameter.
* Fungsi Tukar-Menukar: tukar\_xy() menukar nilai X dan Y.
* Parameter Fungsi Tukar-Menukar: tukar\_xy() menerima dua parameter, a dan b, yang merupakan pointer ke int.
* Tujuan Fungsi Tukar-Menukar: tukar\_xy() menukar nilai yang ditunjuk oleh pointer a dan b dengan menggunakan variabel z sebagai penyimpanan sementara.
* Output Sebelum Pertukaran: Program mencetak nilai X dan Y sebelum pertukaran dengan cout.
* Fungsi tukar\_xy menerima dua pointer ke int, a dan b. Fungsi ini menukar nilai yang ditunjuk oleh pointer a dan b dengan menggunakan variabel z sebagai penyimpanan sementara. Pertama, nilai yang ditunjuk oleh a disimpan dalam variabel z. Kemudian, nilai yang ditunjuk oleh b disimpan dalam a. Terakhir, nilai yang disimpan dalam z disimpan dalam b.

**Program 6**

**Melewatkan struktur sebagai parameter fungsi**

#include <iostream>

// Mengganti <iostream.h> untuk operasi input/output

using namespace std;

// Definisi struktur tanggal

struct Date {

int month;

int day;

int year;

};

// Deklarasi prototipe fungsi cetak\_tanggal dengan parameter bertipe Date

void cetak\_tanggal(Date);

int main() {

// Definisi variabel today dengan tipe Date

Date today;

// Meminta pengguna untuk memasukkan tanggal saat ini

cout << "Masukkan tanggal saat ini (mm-dd-yyyy): ";

cin >> today.month >> today.day >> today.year;

// Memanggil fungsi cetak\_tanggal untuk mencetak tanggal

cetak\_tanggal(today);

// Menampilkan informasi pembuat program

cout << endl << "Program ini dibuat oleh:" << endl;

cout << " - Nama: Rayhan Elmo Athalah Saputra" << endl;

cout << " - NRP: 5223600027" << endl;

cout << " - Kelas: Teknologi Game 11 A" << endl;

return 0;

}

// Definisi fungsi cetak\_tanggal dengan parameter bertipe Date

void cetak\_tanggal(Date now) {

// Array nama\_bulan untuk menyimpan nama-nama bulan

static const char \*nama\_bulan[] = {

"Bulan tidak valid", "Januari", "Februari", "Maret", "April", "Mei",

"Juni", "Juli", "Agustus", "September", "Oktober", "November",

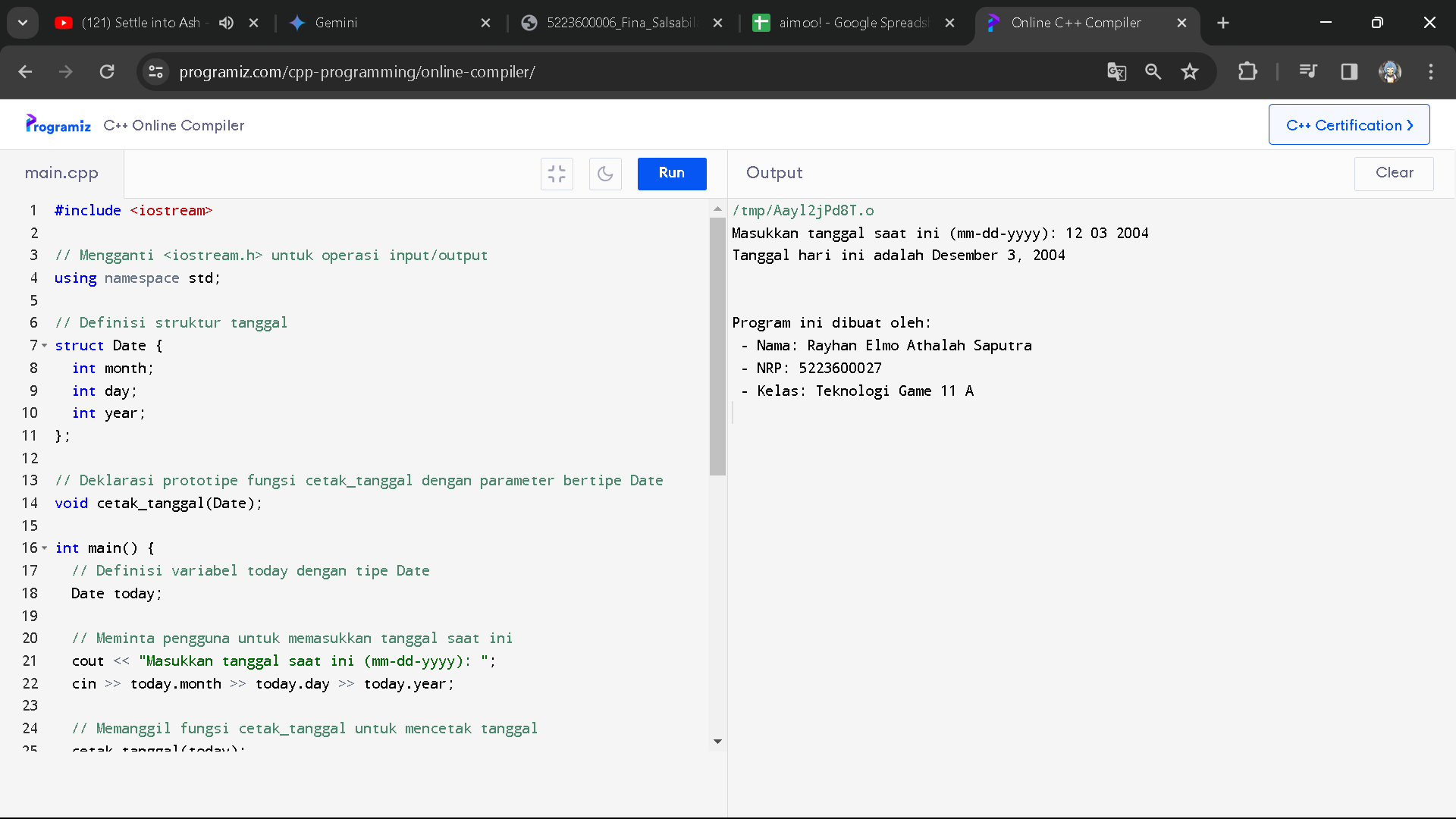
"Desember"};

// Mencetak tanggal yang dimasukkan oleh pengguna

cout << "Tanggal hari ini adalah " << nama\_bulan[now.month] << " " << now.day

<< ", " << now.year << endl << endl;

}



Perubahan dari C ke C++:

* Struktur Date: Struktur Date digunakan untuk mendefinisikan variabel today yang memiliki anggota month, day, dan year.
* Fungsi cetak\_tanggal: Fungsi cetak\_tanggal dengan parameter bertipe Date didefinisikan untuk mencetak tanggal yang dimasukkan oleh pengguna.
* Array nama\_bulan: Array nama\_bulan digunakan untuk menyimpan nama-nama bulan dan mencetak nama bulan berdasarkan input bulan yang dimasukkan.

Analisis:

* Tipe Data Date: Tipe data Date digunakan untuk merepresentasikan tanggal dengan anggota month, day, dan year. Hal ini memungkinkan penggunaan variabel today dengan tipe Date untuk menyimpan tanggal.
* Fungsi main(): Fungsi main() merupakan titik masuk program. Di dalamnya, variabel today dari tipe Date dideklarasikan untuk menyimpan tanggal saat ini.
* Fungsi cetak\_tanggal(): Fungsi ini digunakan untuk mencetak tanggal yang diterima sebagai parameter. Penggunaan static const char \*nama\_bulan[] untuk menyimpan nama-nama bulan sebagai array konstan dan penggunaan parameter bertipe Date memungkinkan fungsi ini untuk mencetak tanggal dengan bulan yang sesuai dengan nama bulan yang diberikan.
* Validasi Input: Dalam versi C++ ini, tidak dilakukan validasi khusus untuk memastikan bahwa input yang dimasukkan adalah tanggal yang valid. Anda dapat menambahkan validasi tambahan jika diperlukan.
* Input dan Output: cin digunakan untuk menerima input dan cout digunakan untuk mencetak output yang telah diterima atau yang akan di isi.

**Program 7**

**Fungsi parametermya berupa pointer yang menunjuk ke struktur**

#include <iostream>

// Menggantikan <iostream.h> untuk operasi input/output

using namespace std;

// Definisi struktur koordinat

struct Koordinat {

int x;

int y;

};

// Deklarasi prototipe fungsi tukar\_xy dengan parameter bertipe pointer ke Koordinat

void tukar\_xy(Koordinat \*);

int main() {

// Deklarasi variabel posisi dengan tipe Koordinat

Koordinat posisi;

// Meminta pengguna untuk memasukkan koordinat posisi

cout << "Masukkan koordinat posisi (x, y): ";

cin >> posisi.x >> posisi.y;

// Mencetak koordinat posisi sebelum pertukaran

cout << "x, y semula = " << posisi.x << ", " << posisi.y << endl;

// Memanggil fungsi tukar\_xy untuk menukar nilai x dan y dengan menggunakan pointer

tukar\_xy(&posisi);

// Mencetak koordinat posisi setelah pertukaran

cout << "x, y sekarang = " << posisi.x << ", " << posisi.y << endl;

// Menampilkan informasi pembuat program

cout << endl << "Program ini dibuat oleh:" << endl;

cout << " - Nama: Rayhan Elmo Athalah Saputra" << endl;

cout << " - NRP: 5223600027" << endl;

cout << " - Kelas: Teknologi Game 11 A" << endl;

return 0;

}

// Definisi fungsi tukar\_xy dengan parameter bertipe pointer ke Koordinat

void tukar\_xy(Koordinat \*pos\_xy) {

// Variabel lokal z untuk menyimpan nilai sementara

int z;

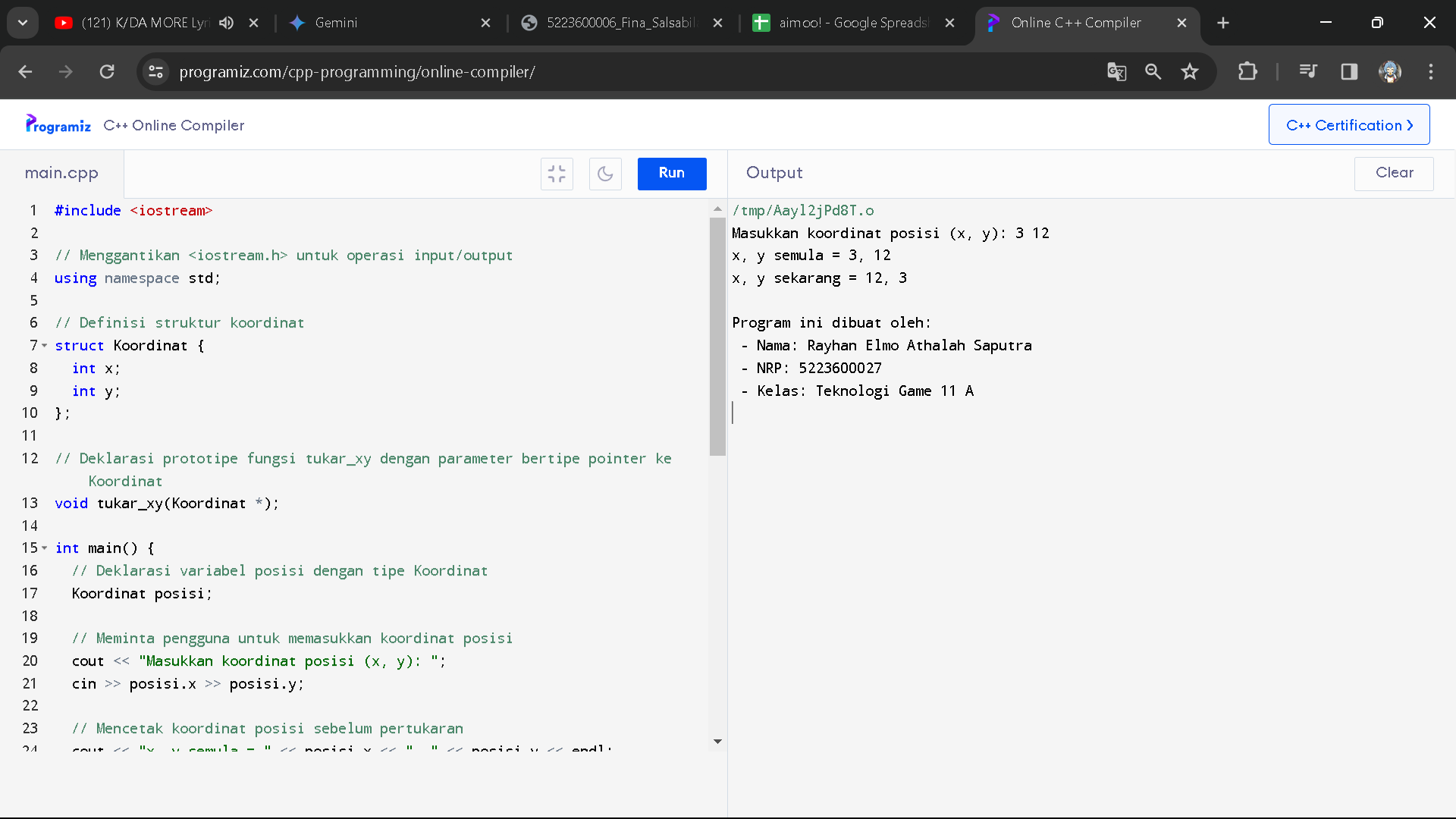
// Menukar nilai yang ditunjuk oleh pointer pos\_xy menggunakan variabel z

z = pos\_xy->x;

pos\_xy->x = pos\_xy->y;

pos\_xy->y = z;

}



Perubahan dari C ke C++:

* Struktur Koordinat: Struktur Koordinat digunakan untuk mendefinisikan tipe data baru bernama Koordinat yang memiliki dua anggota: x dan y.
* Fungsi tukar\_xy: Fungsi tukar\_xy didefinisikan dengan parameter bertipe pointer ke Koordinat. Hal ini memungkinkan fungsi ini untuk langsung mengakses dan memodifikasi nilai dari variabel posisi dalam main().
* Penggunaan operator ->: Operator -> digunakan untuk mengakses anggota dari struktur yang ditunjuk oleh pointer pos\_xy. Ini lebih sederhana daripada penggunaan dereferensi (\*).

Analisis:

* Struktur Koordinat: Struktur Koordinat didefinisikan menggunakan kata kunci struct. Struktur ini memiliki dua anggota, x dan y, untuk merepresentasikan koordinat.
* Fungsi tukar\_xy: Fungsi tukar\_xy dideklarasikan dengan menggunakan prototipe yang mengambil parameter bertipe pointer ke Koordinat. Hal ini memungkinkan fungsi ini untuk mengakses dan mengubah nilai dari variabel posisi dalam main().
* Fungsi main(): variabel posisi dari tipe Koordinat dideklarasikan untuk menyimpan koordinat yang dimasukkan oleh pengguna. Nilai x dan y kemudian dimasukkan ke dalam variabel posisi. Fungsi tukar\_xy kemudian dipanggil dengan meneruskan alamat memori dari variabel posisi.
* Fungsi tukar\_xy: variabel lokal z digunakan untuk menyimpan nilai sementara. Nilai x dari koordinat ditukar dengan nilai y menggunakan operator panah (->) untuk mengakses anggota dari struktur yang ditunjuk oleh pointer. Nilai y dari koordinat kemudian ditukar dengan nilai z.
* Input dan Output: Input dan output standar C++ (cin dan cout) digunakan untuk memungkinkan program berinteraksi dengan pengguna.

Kesimpulan:

Analisis menunjukkan bahwa skrip tersebut menggunakan struktur data Koordinat untuk merepresentasikan koordinat dengan anggota x dan y. Fungsi tukar\_xy menerima parameter bertipe pointer ke Koordinat, memungkinkan program untuk mengoperasikan variabel posisi secara langsung melalui alamat memori. Algoritma sederhana dalam fungsi tukar\_xy digunakan untuk menukar nilai x dan y dari suatu koordinat. Penggunaan input dan output standar C++ (cin dan cout) memungkinkan program untuk berinteraksi dengan pengguna.

Dengan fungsi tukar\_xy, program dapat memanipulasi data koordinat secara terstruktur dan dapat diandalkan. Hal ini menunjukkan penerapan konsep-konsep dasar dalam pemrograman C++, termasuk struktur data, pointer, dan operasi dasar pada data.

**LATIHAN**

1. Definisikan sebuah struktur (misalkan namanya = record) yang memiliki 3 buah field berupa sebuah integer (misalkan namanya = loop), sebuah array karakter dengan 5 elemen (misalkan namanya = word) dan sebuah float (misalkan Namanya = sum)!

Jawab:

#include <iostream>

#include <cstring> //

using namespace std;

// Definisi struktur Record

struct Record {

int loop; // Menyimpan sebuah nilai integer

char word[5]; // Menyimpan sebuah array karakter dengan 5 elemen

float sum; // Menyimpan sebuah nilai floating-point

};

int main() {

// Deklarasi variabel bertipe Record

Record data;

// Pengisian nilai ke dalam variabel data

data.loop = 10;

// \*\*Perbaikan:\*\* Gunakan strncpy dengan panjang maksimum yang aman

strncpy(data.word, "Hello", sizeof(data.word) - 1);

data.word[sizeof(data.word) - 1] = '\0'; // Pastikan null termination

data.sum = 3.14;

// Menampilkan nilai dari masing-masing anggota struktur

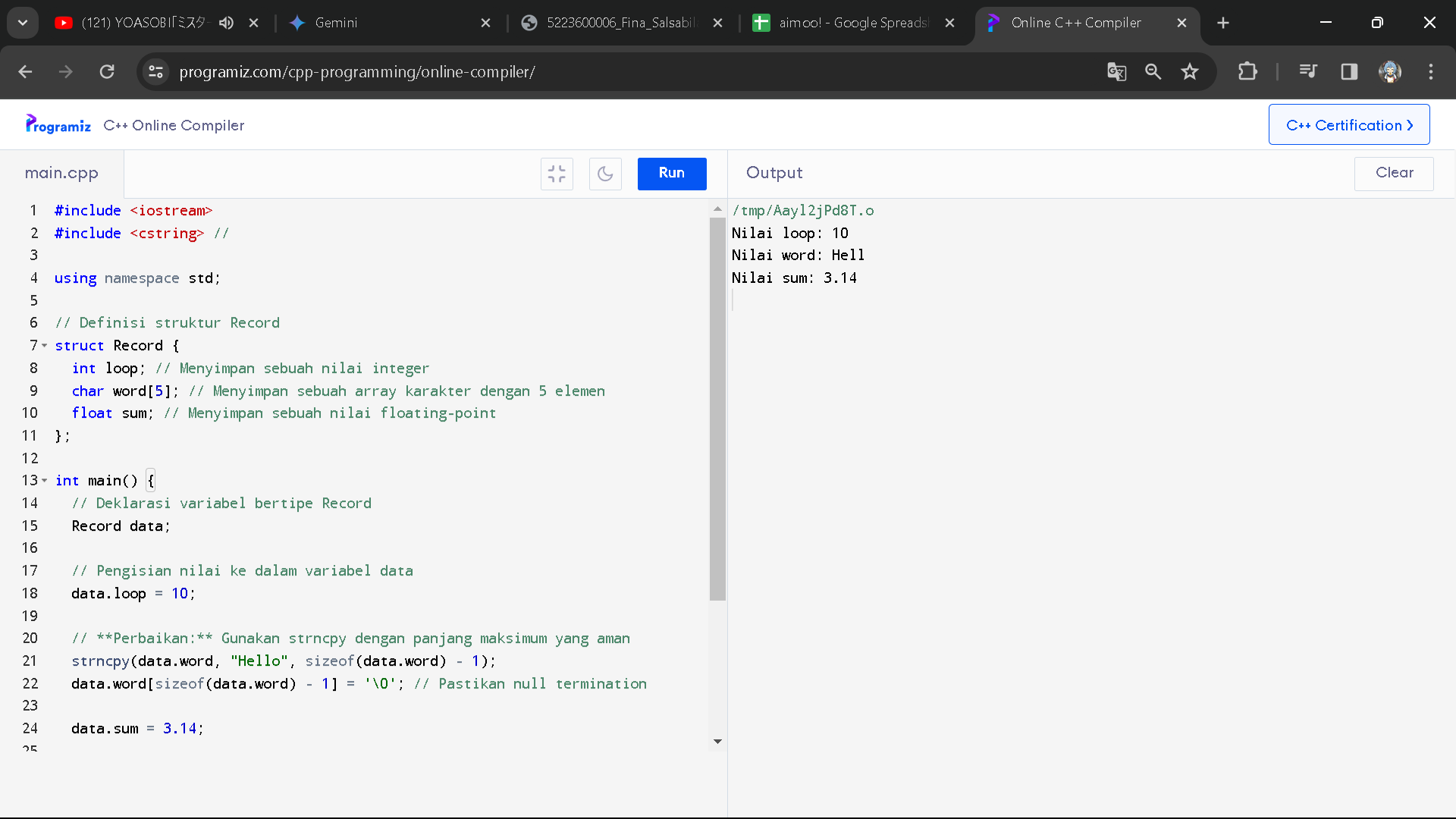
cout << "Nilai loop: " << data.loop << endl;

cout << "Nilai word: " << data.word << endl;

cout << "Nilai sum: " << data.sum << endl;

return 0;

}



* Struktur Record: Program ini menggunakan struktur Record untuk menampung data. Struktur ini memiliki tiga anggota:
* loop: Sebuah variabel bertipe integer untuk menyimpan nilai numerik.
* word: Sebuah array karakter dengan panjang 5 untuk menyimpan string pendek.
* sum: Sebuah variabel bertipe float untuk menyimpan nilai desimal.
* Fungsi main(): merupakan fungsi utama dalam program ini. Di dalam fungsi ini:
* Deklarasi variabel: Sebuah variabel data dideklarasikan dengan tipe Record.
* Pengisian nilai: Nilai-nilai dimasukkan ke dalam anggota variabel data.
* Pencetakan nilai: Nilai-nilai anggota variabel data dicetak ke layar.
* Penjelasan detail:
* Baris 1-3: Terdapat tiga baris komentar yang menjelaskan struktur Record.
* Baris 5: Deklarasi variabel data dengan tipe Record.
* Baris 7-9: Pengisian nilai ke dalam anggota variabel data:
* data.loop diset ke nilai 10.
* data.word diset ke string "Hello".
* data.sum diset ke nilai 3.14.
* Baris 11-13: Pencetakan nilai anggota variabel data:
* Nilai data.loop dicetak dengan teks "Nilai loop: ".
* Nilai data.word dicetak dengan teks "Nilai word: ".
* Nilai data.sum dicetak dengan teks "Nilai sum: ".

1. Deklarasikan sebuah variabel struktur (misalkan namanya = sample) yang

didefinisikan memiliki tipe struktur record!

Jawab:

Deklarasi Record sample membuat variabel sample bertipe Record dengan 3 anggota data: loop, word, dan sum. Tipe Record sebelumnya didefinisikan dengan struktur dan karakteristiknya. Variabel sample dapat digunakan untuk menyimpan data sesuai struktur Record.