**Overview Struktur**

Konversi C menjadi C++

**Oleh:**

Marlon Immanuel Sarapang / 5223600010

**Program Studi STr Teknologi Game**

**Departemen Teknologi Multimedia Kreatif**

**Politeknik Elektronika Negeri Surabaya**

**2024**

1. File program: student1.c dalam bentuk C++

**Source code**

#include <iostream>

#include <cstring>

using namespace std;

struct Date {

int month;

int day;

int year;

};

struct Student {

char name[30];

Date birthday;

};

Student mhs;

int main() {

// Memberikan nilai kepada field dari struktur mhs

strcpy(mhs.name, "MUHAMMAD IHSAN");

mhs.birthday.month = 8;

mhs.birthday.day = 10;

mhs.birthday.year = 1970;

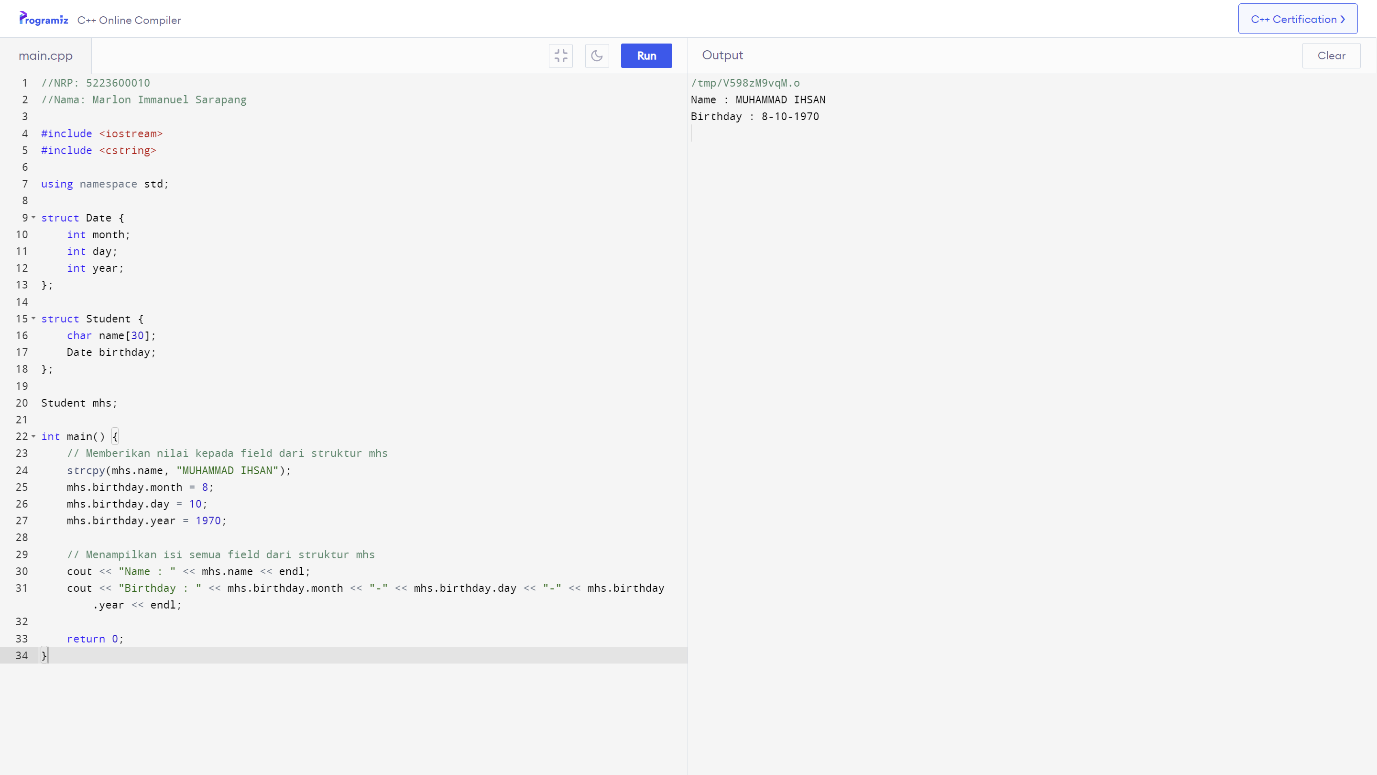
// Menampilkan isi semua field dari struktur mhs

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

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

return 0;

}



***Gambar 1*** *(Hasil output)*

**Analisis dan Kesimpulan:**

1. **Penggunaan Fungsi printf dan cout:** Penggunaan cout di C++ adalah alternatif modern dan lebih fleksibel dibandingkan dengan fungsi printf di C. cout mendukung overloading dan memiliki kemampuan formatting yang lebih baik.
2. **Penggunaan strcpy:** Meskipun fungsi strcpy masih digunakan di C++, lebih disarankan untuk menggunakan std::string untuk manipulasi string karena lebih aman dan mudah digunakan.
3. **Fungsi main:** Fungsi main pada C++ harus memiliki tipe data int dan mengembalikan nilai 0 pada akhir program sebagai indikator keberhasilan.

Dalam keseluruhan, program tersebut telah diubah menjadi C++ dengan menggunakan namespace std dan beberapa penyesuaian sintaks. Kesimpulannya, perubahan ini meningkatkan kejelasan, keamanan, dan ekspresivitas program.

1. File program: zodiak.c dalam bentuk C++

**Source code**

#include <iostream>

using namespace std;

struct Zodiak {

char nama[11];

int tgl\_awal;

int bln\_awal;

int tgl\_akhir;

int bln\_akhir;

};

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

int main() {

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

char retry;

do {

cout << "Masukkan tgl lahir Anda (XX-XX-XXXX): ";

scanf("%d-%d-%d", &tgl\_lhr, &bln\_lhr, &thn\_lhr);

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;

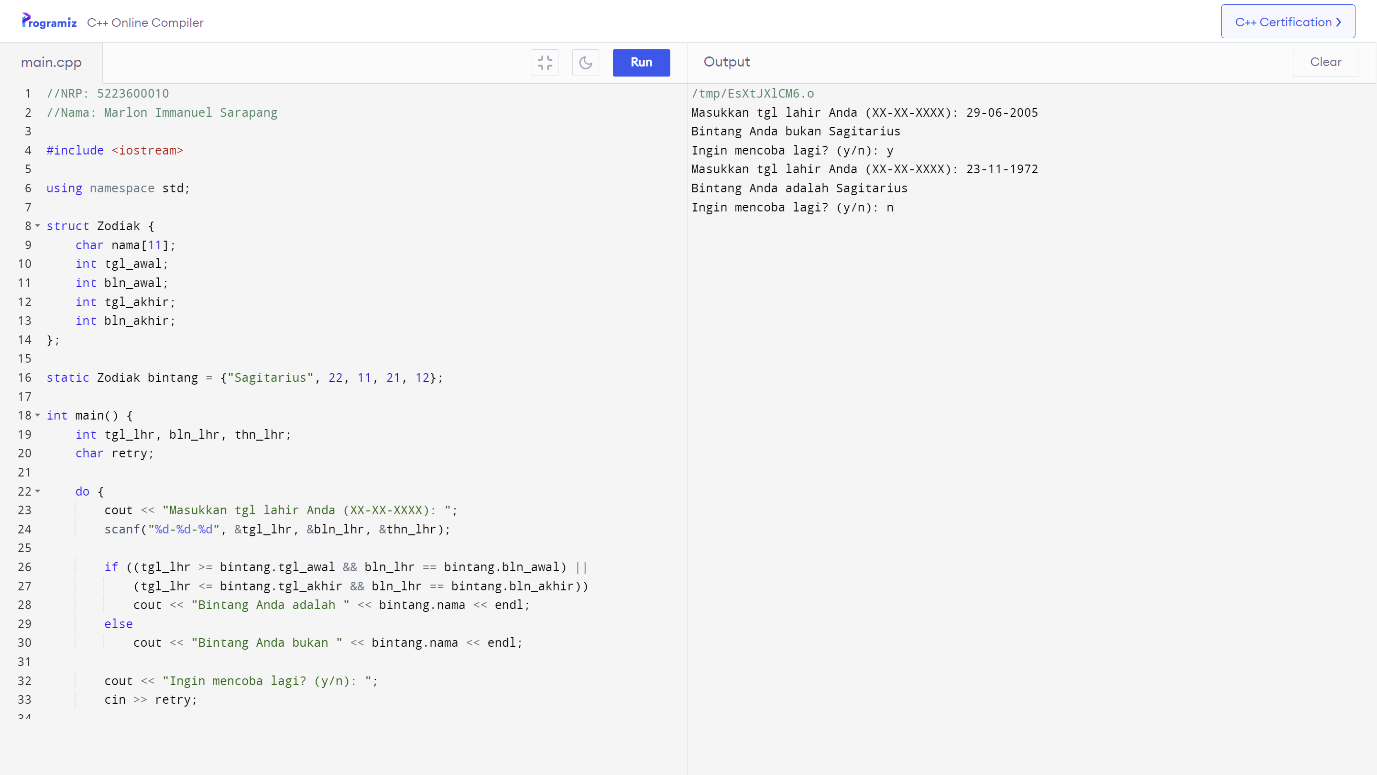
cout << "Ingin mencoba lagi? (y/n): ";

cin >> retry;

} while (retry == 'y' || retry == 'Y');

return 0;

}



***Gambar 2*** *(Hasil output)*

**Analisis dan Kesimpulan:**

1. **Penggunaan Fungsi printf dan cout:** Sama seperti pada kasus sebelumnya, penggunaan cout di C++ lebih disukai daripada printf karena keamanan dan kemudahan penggunaan.
2. **Penggunaan Fungsi scanf:** Meskipun masih dapat menggunakan scanf di C++, lebih disarankan untuk menggunakan mekanisme input C++ seperti cin untuk memastikan keamanan dan menghindari masalah buffer overflow.
3. **Fungsi main:** Fungsi main harus memiliki tipe data int dan mengembalikan nilai 0 pada akhir program.
4. **Namespace std:** Penggunaan namespace std memungkinkan penggunaan fungsi dan objek dari Standard Template Library (STL) tanpa menambahkan std:: secara eksplisit.
5. File program: student2.c dalam bentuk C++

**Source code**

#include <iostream>

#include <string>

#define MAKS 20

using namespace std;

struct Date {

int month;

int day;

int year;

};

struct Student {

string name;

Date birthday;

};

Student data\_mhs[MAKS];

int main() {

int i = 0, jml;

char lagi;

// Memasukkan data

do {

cout << "Name : ";

getline(cin, data\_mhs[i].name);

cout << "Birthday (mm-dd-yyyy): ";

cin >> data\_mhs[i].birthday.month >> data\_mhs[i].birthday.day >> data\_mhs[i].birthday.year;

cout << endl;

i++;

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

do {

cin >> lagi;

} while (lagi != 'Y' && lagi != 'y' && lagi != 'T' && lagi != 't');

cin.ignore(); // Hapus sisa data dalam penampung keyboard

cout << endl;

} while (lagi == 'Y' || lagi == 'y');

jml = i;

// Menampilkan data

cout << "DATA SISWA" << endl;

for (i = 0; i < jml; i++) {

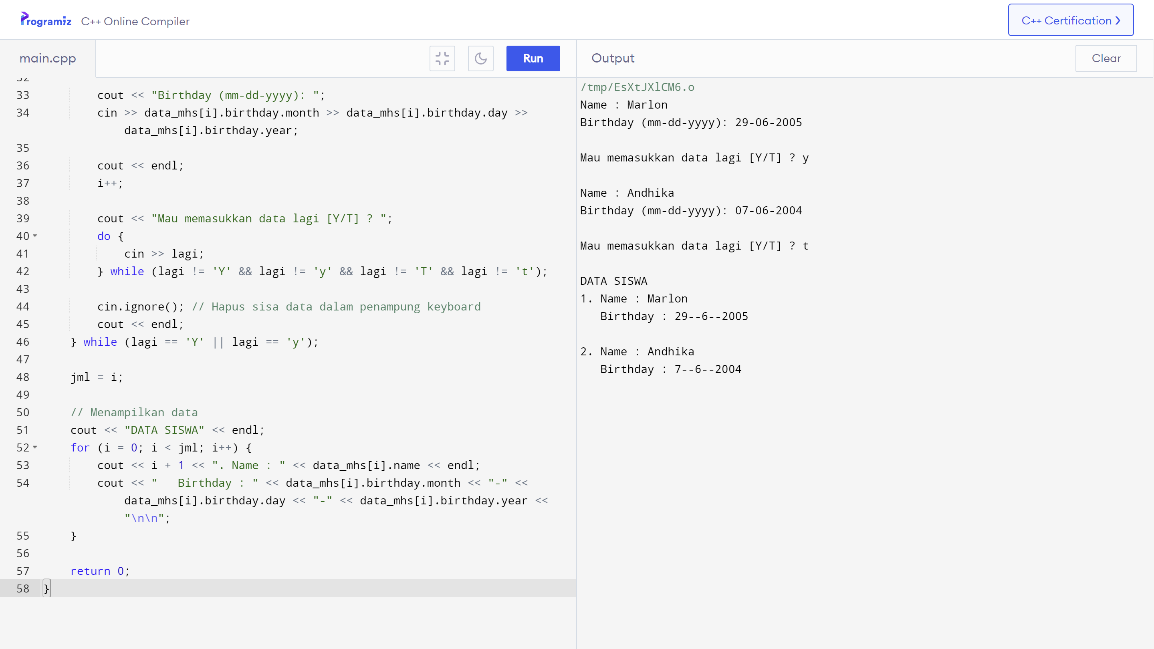
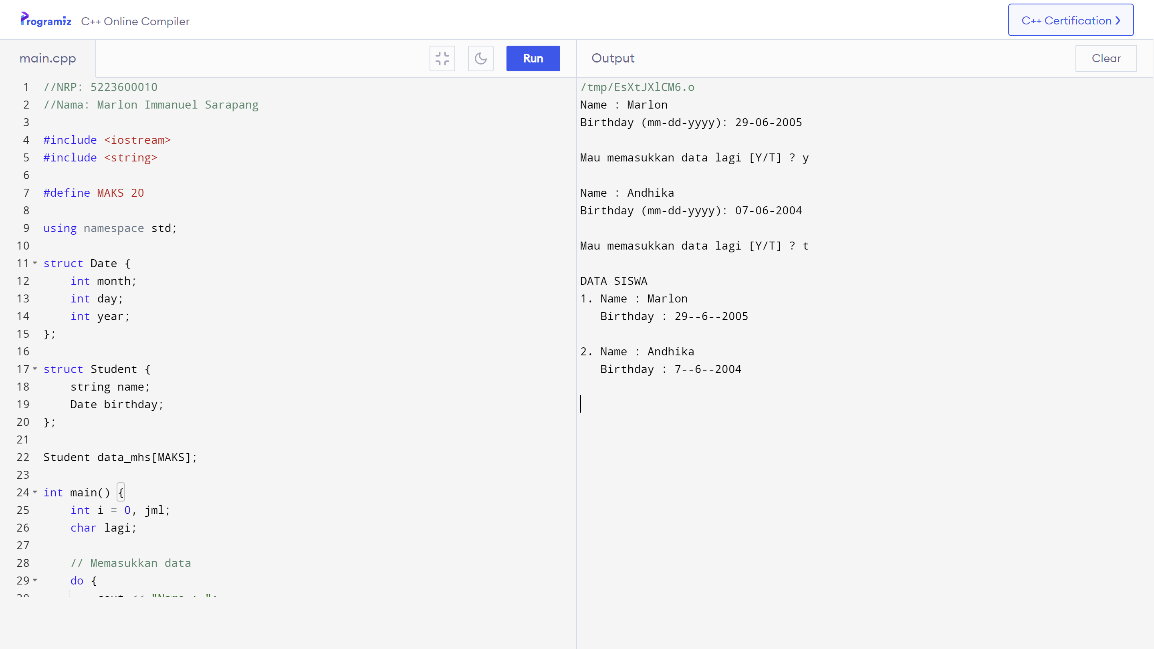
cout << i + 1 << ". Name : " << data\_mhs[i].name << endl;

cout << " Birthday : " << data\_mhs[i].birthday.month << "-" << data\_mhs[i].birthday.day << "-" << data\_mhs[i].birthday.year << "\n\n";

}

return 0;

}



***Gambar 3 dan 4*** *(Hasil output)*

**Analisis dan Kesimpulan:**

Beberapa perubahan yang dilakukan melibatkan penggunaan string untuk variabel nama, penggunaan getline untuk membaca string dari input, serta penyesuaian pada input tanggal dan penggunaan cin.ignore() untuk membersihkan sisa data dalam penampung keyboard. Program ini juga menggunakan namespace std untuk menyederhanakan penggunaan fungsi-fungsi dari C++ Standard Library.

1. File program: cetak1.c dalam bentuk C++

**Source code**

#include <iostream>

#include <string>

using namespace std;

struct Date {

int month;

int day;

int year;

};

void cetak\_tanggal(int mm, int dd, int yy);

int main() {

Date today;

cout << "Enter the current date (mm-dd-yyyy): ";

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

cetak\_tanggal(today.month, today.day, today.year);

return 0;

}

void cetak\_tanggal(int mm, int dd, int yy) {

static string nama\_bulan[] = {

"Wrong month", "January", "February", "March",

"April", "May", "June", "July", "August",

"September", "October", "November", "December"

};

cout << "Today's date is " << nama\_bulan[mm] << " " << dd << ", " << yy << "\n\n";

}



***Gambar 5*** *(Hasil output)*

**Analisis dan Kesimpulan:**

Perubahan yang dilakukan termasuk penggantian array karakter untuk nama bulan menjadi array string (string nama\_bulan[]), penggunaan cout dan cin sebagai pengganti printf dan scanf, serta penggunaan tipe data string untuk nama bulan. Program ini juga menggunakan namespace std untuk memberikan akses ke fungsi-fungsi dari C++ Standard Library.

1. File program: posisi1.c dalam bentuk C++

**Source code**

#include <iostream>

using namespace std;

struct Koordinat {

int x;

int y;

};

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

int main() {

Koordinat posisi;

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

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

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

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

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

return 0;

}

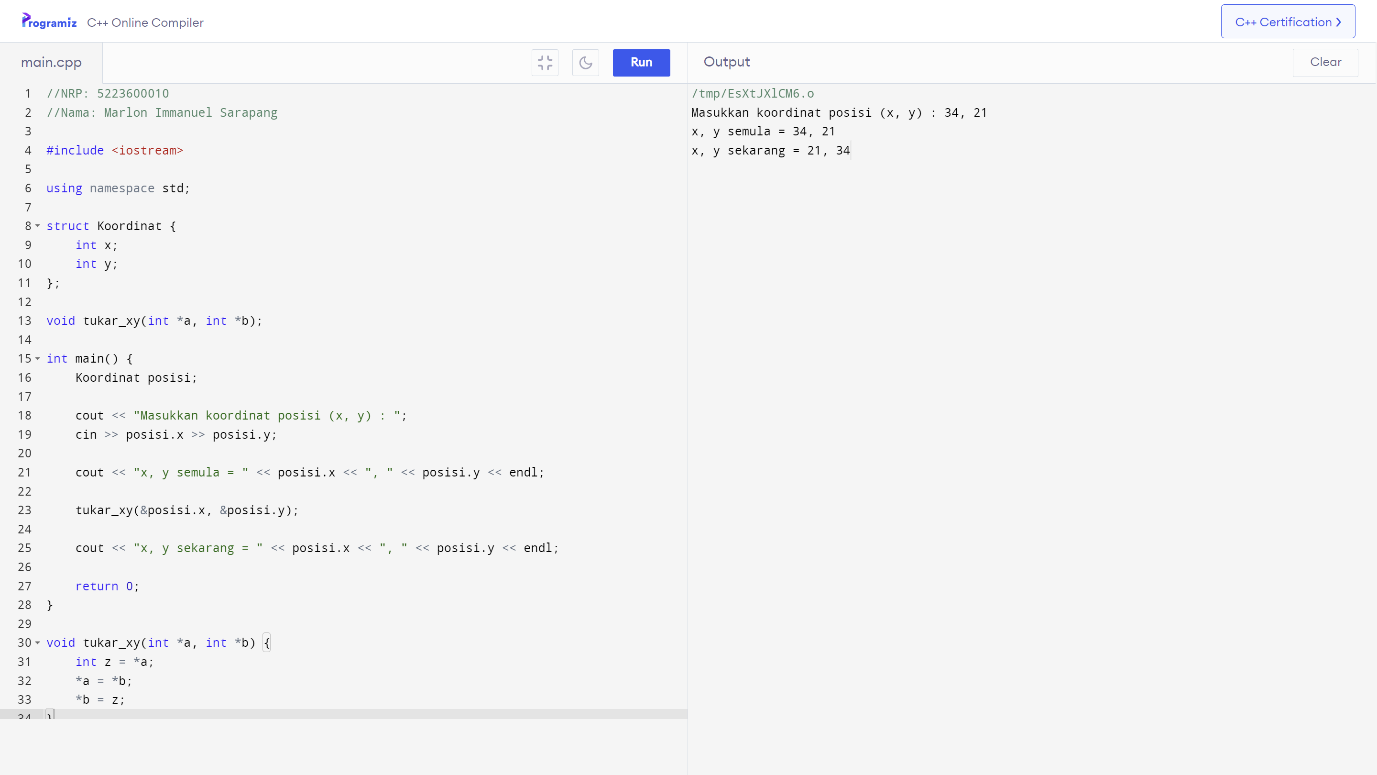
void tukar\_xy(int \*a, int \*b) {

int z = \*a;

\*a = \*b;

\*b = z;

}



***Gambar 6*** *(Hasil output)*

**Analisis dan Kesimpulan:**

Perubahan yang dilakukan termasuk penggunaan tipe data struct dalam bentuk struct Koordinat, penggantian printf dan scanf dengan cout dan cin, serta penggunaan namespace std untuk memberikan akses ke fungsi-fungsi dari C++ Standard Library. Fungsi tukar\_xy tetap menggunakan pointer untuk menukar nilai x dan y. Program ini juga memanfaatkan mekanisme string stream dari C++ untuk memudahkan input dan output.

1. File program: cetak2.c dalam bentuk C++

**Source code**

#include <iostream>

#include <string>

using namespace std;

struct Date {

int month;

int day;

int year;

};

void cetak\_tanggal(Date);

int main() {

Date today;

cout << "Enter the current date (mm-dd-yyyy): ";

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

cetak\_tanggal(today);

return 0;

}

void cetak\_tanggal(Date now) {

static string nama\_bulan[] = {

"Wrong month", "January", "February", "March",

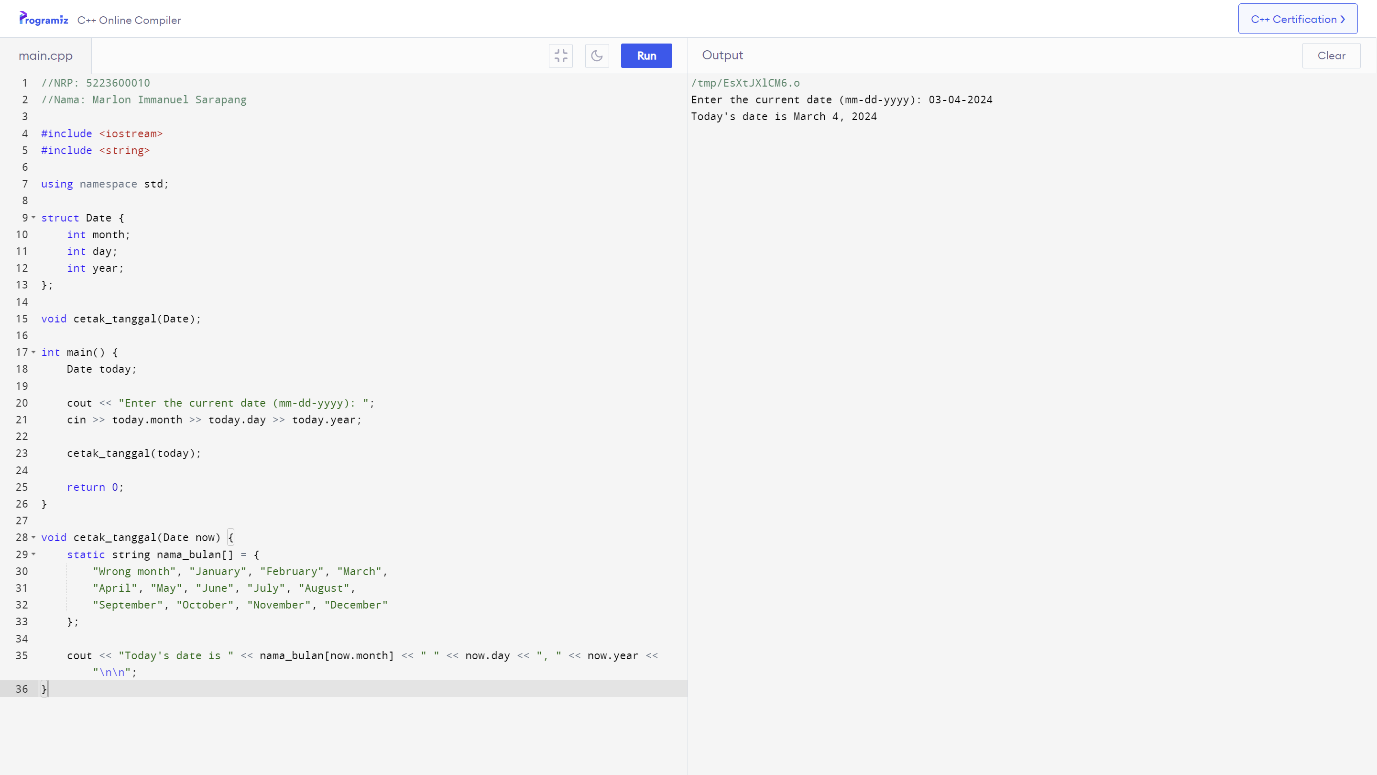
"April", "May", "June", "July", "August",

"September", "October", "November", "December"

};

cout << "Today's date is " << nama\_bulan[now.month] << " " << now.day << ", " << now.year << "\n\n";

}



***Gambar 7*** *(Hasil output)*

**Analisis dan Kesimpulan:**

Perubahan yang dilakukan termasuk penggantian printf dan scanf dengan cout dan cin, serta penggunaan tipe data struct dalam bentuk struct Date. Program ini juga menggunakan namespace std untuk memberikan akses ke fungsi-fungsi dari C++ Standard Library. Fungsi cetak\_tanggal tetap menerima parameter dari tipe struct Date.

1. File program: posisi2.c dalam bentuk C++

**Source code**

#include <iostream>

using namespace std;

struct Koordinat {

int x;

int y;

};

void tukar\_xy(Koordinat\*);

int main() {

Koordinat posisi;

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

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

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

tukar\_xy(&posisi);

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

return 0;

}

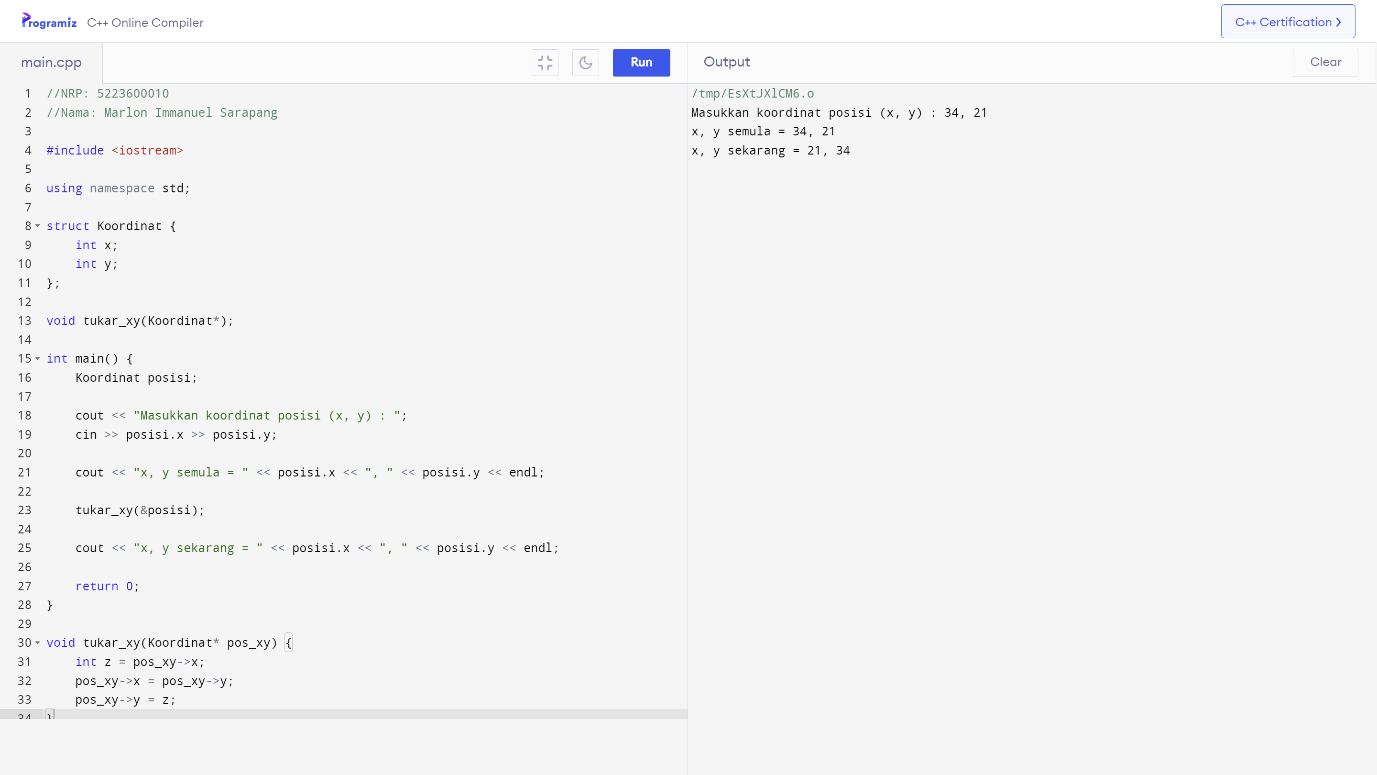
void tukar\_xy(Koordinat\* pos\_xy) {

int z = pos\_xy->x;

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

pos\_xy->y = z;

}



***Gambar 8*** *(Hasil output)*

**Analisis dan Kesimpulan:**

Perubahan yang dilakukan termasuk penggantian printf dan scanf dengan cout dan cin, serta penggunaan tipe data struct dalam bentuk struct Koordinat. Program ini juga menggunakan namespace std untuk memberikan akses ke fungsi-fungsi dari C++ Standard Library. Fungsi tukar\_xy tetap menerima parameter dari tipe pointer ke struct Koordinat. Dalam C++, operator -> digunakan untuk mengakses anggota dari struktur yang ditunjuk oleh pointer.

**Soal**

1. Berikut adalah definisi struktur (record) dengan 3 field

// Definisi struktur (record)

struct Record {

int loop; // Field bertipe integer

char word[5]; // Field bertipe array karakter dengan 5 elemen

float sum; // Field bertipe float

};

1. Pendeklarasian sebuah variabel dengan tipe struktur tersebut adalah sebagai berikut.

// Deklarasi variabel struktur

struct Record sample;

Dengan cara ini, variabel sample telah dideklarasikan sebagai variabel bertipe struktur Record yang memiliki tiga field sesuai dengan definisi yang diberikan. Anda dapat menggunakan variabel sample untuk menyimpan data sesuai dengan struktur yang telah didefinisikan.