

ALGORITMA DAN STRUKTUR DATA

Overview Struct



Oleh :

Fina Salsabila Pramudita (5223600006)

Program Studi Sarjana Terapan Teknologi Game

Departemen Teknologi Multimedia Kreatif

Politeknik Elektronika Negeri Surabaya

2024

Program 1

Mengisi field dari variable struktur kemudian menampilkannya

```
#include <iostream> /* input dan output */
#include <string> /* nama */
using namespace std;

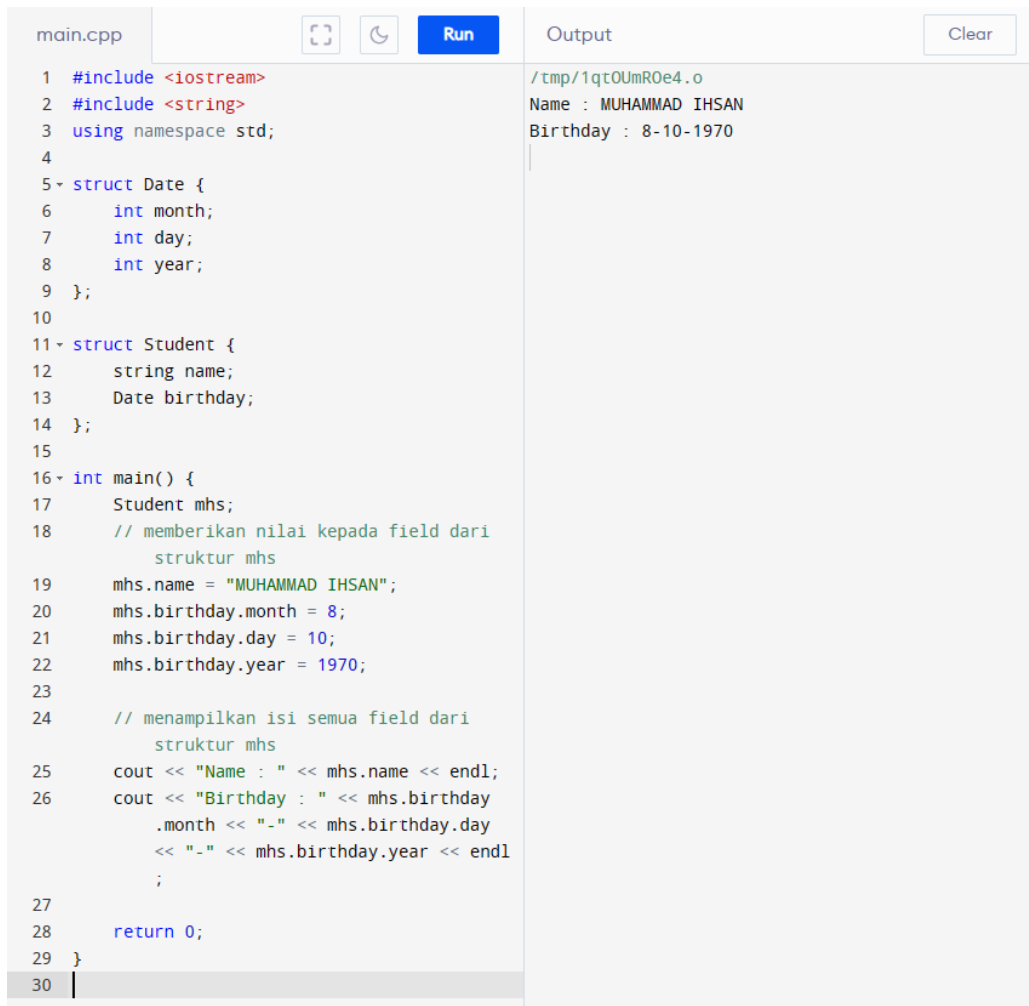
struct Date {
    int month;
    int day;
    int year;
};

struct Student {
    string name;
    Date birthday;
};

int main() {
    Student mhs;
    // memberikan nilai kepada field dari struktur 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 /* mencetak */ << "Name : " << mhs.name << endl;
    cout << "Birthday : " << mhs.birthday.month << "-" << mhs.birthday.day
    << "-" << mhs.birthday.year << endl;

    return 0;
}
```



```
main.cpp  Run  Output  Clear

1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  struct Date {
6      int month;
7      int day;
8      int year;
9  };
10
11 struct Student {
12     string name;
13     Date birthday;
14 };
15
16 int main() {
17     Student mhs;
18     // memberikan nilai kepada field dari
19     // struktur mhs
20     mhs.name = "MUHAMMAD IHSAN";
21     mhs.birthday.month = 8;
22     mhs.birthday.day = 10;
23     mhs.birthday.year = 1970;
24
25     // menampilkan isi semua field dari
26     // struktur mhs
27     cout << "Name : " << mhs.name << endl;
28     cout << "Birthday : " << mhs.birthday
29         .month << "-" << mhs.birthday.day
30         << "-" << mhs.birthday.year << endl;
31
32     return 0;
33 }
```

Output

```
/tmp/1qt0UmR0e4.o
Name : MUHAMMAD IHSAN
Birthday : 8-10-1970
```

Gambar 1. Mengisi field dari variable struktur kemudian menampilkannya

Perubahan pada skrip C to C++

- Menggunakan 'struct' tanpa perlu menuliskan kata 'struct' setiap mengakses anggota – anggota setelah deklarasinya.
- Mengganti fungsi 'strcpy' dengan pengisian langsung menggunakan operator '=' untuk string
- Menggunakan fungsi 'main' dengan tipe pengembalian 'int' alih alih 'main()' tanpa tipe pengembalian

Analisis data

- 'Date' : struktur yang menyimpan informasi tentang tanggal dan tiga lainnya yaitu 'month', 'day', dan 'year'
- 'Student' : struktur untuk menyimpan informasi tentang mahasiswa dengan dua anggota 'nama' dan 'birthday'
- 'main' : sebagai titik masuk utama eksekusi pemrograman. Selain itu digunakan untuk merepresentasikan seorang mahasiswa.

- 'name' dan 'birthday' dari variable 'mhs' diisi dengan nilai yang sesuai
- Informasi didalam variable 'mhs' akan di cetak di layar melalui 'cout' yang berisi nama dan tanggal lahir mahasiswa.

Program 2

Menentukan zodiac berdasarkan data tanggal lahir masukan

```
#include <iostream> /*input dan output*/
using namespace std;

struct Zodiac {
    char nama[11];
    int tgl_awal;
    int bln_awal;
    int tgl_akhir;
    int bln_akhir;
};

int main() {
    Zodiac bintang = {"Sagitarious", 22, 11, 21, 12};

    int tgl_lhr, bln_lhr, thn_lhr;
    cout /*mencetak hasil*/ << "Masukkan tgl lahir Anda (XX-XX-XXXX): ";
    cin /*menerima masukan*/ >> 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;

    return 0;
}
```

```

main.cpp
1 #include <iostream> /*input dan output*/
2 using namespace std;
3
4 struct Zodiak {
5     char nama[11];
6     int tgl_awal;
7     int bln_awal;
8     int tgl_akhir;
9     int bln_akhir;
10 };
11
12 int main() {
13     Zodiak bintang = {"Sagittarius", 22, 11,
14                       21, 12};
15
16     int tgl_lhr, bln_lhr, thn_lhr;
17     cout /*mencetak hasil*/ << "Masukkan
18         tgl lahir Anda (XX-XX-XXXX): ";
19     cin /*menerima masukan*/ >> tgl_lhr >>
20         bln_lhr >> thn_lhr;
21
22     if ((tgl_lhr >= bintang.tgl_awal &&
23         bln_lhr == bintang.bln_awal) ||
24         (tgl_lhr <= bintang.tgl_akhir &&
25         bln_lhr == bintang.bln_akhir))
26         cout << "Bintang Anda adalah " <<
27             bintang.nama << endl;
28     else
29         cout << "Bintang Anda bukan " <<
30             bintang.nama << endl;
31
32     return 0;
33 }

```

Output

```

/tmp/hI1cUbLA7Y.o
Masukkan tgl lahir Anda (XX-XX-XXXX): 23-03-2002
Bintang Anda bukan Sagittarius

```

Gambar 2. Menentukan zodiac berdasarkan data tanggal lahir masukan

Perubahan C ke C++

- Menggunakan 'struct' tanpa perlu menuliskan kata kunci 'struct' setiap kali mengakses anggotanya setelah deklarasinya.
- Menggunakan 'string' daripada array karakter 'char[]' untuk nama.
- Menggunakan fungsi 'main' dengan tipe pengembalian int dibanding 'main()' tanpa tipe pengembalian.

-

Analisis

- Pendefinisian struct 'Zodiak' dengan anggota 'nama', 'tgl_awal', 'bln_awal', 'tgl_akhir', dan 'bln_akhir'
- Fungsi 'main()', yang merupakan titik masuk utama untuk eksekusi program.
- Variabel 'bintang' dari tipe 'Zodiak' untuk merepresentasikan zodiak Sagittarius.
- Meampilkan hasil menggunakan 'cout'
- Sedangkan input tanggal lahir diterima menggunakan 'cin' dan disimpan dalam variabel 'tgl_lhr', 'bln_lhr', dan 'thn_lhr'

- Sebuah kondisi diterapkan untuk memeriksa apakah tanggal lahir pengguna berada dalam rentang zodiak Sagitarius atau tidak. Jika sesuai, maka pesan yang sesuai dengan zodiak tersebut akan dicetak menggunakan 'cout'

Program 3

Array struktur data untuk menyimpan data – data student

```
#include <iostream> // Menggantikan <stdio.h> untuk operasi input/output
using namespace std;

#define MAKS 20 // Penggunaan konstanta MAKS

struct Date { // Definisi global dari tipe Date
    int month;
    int day;
    int year;
};

struct Student { // Definisi global dari tipe Student
    char name[30];
    Date birthday;
};

Student data_mhs[MAKS]; // Deklarasi global dari array student

int main() {
    int i = 0, sudah_benar, jml;
    char lagi;

    // Memasukkan data
    do {
        cout << "Name : ";
        cin.getline(data_mhs[i].name, sizeof(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++;
    } while (sudah_benar == 0);
    jml = i;
    while (jml > 0) {
        cout << "Data ke-" << jml << " : ";
        cout << data_mhs[jml-1].name << " ";
        cout << data_mhs[jml-1].birthday.month << "-";
        cout << data_mhs[jml-1].birthday.day << "-";
        cout << data_mhs[jml-1].birthday.year << endl;
        jml--;
    }
    cout << "Program selesai. Terima kasih." << endl;
    return 0;
}
```

```

        // Memeriksa apakah pengguna ingin memasukkan data lagi
        cout << "Mau memasukkan data lagi [Y/T] ? ";
        do {
            lagi = getchar(); // Membaca tombol
            sudah_benar = (lagi == 'Y') || (lagi == 'y') || (lagi == 'T')
|| (lagi == 't');
            } while (!sudah_benar);

        // Menghapus sisa data dalam penampung keyboard
        cin.ignore();
        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;
            cout << " Birthday : " << data_mhs[i].birthday.month << "-" <<
data_mhs[i].birthday.day << "-" << data_mhs[i].birthday.year << endl <<
endl;
            }

        return 0;
    }

```

```

main.cpp
1 #include <iostream> // Menggantikan <stdio.h> untuk operasi input/output
2 using namespace std;
3
4 #define MAKS 20 // Penggunaan konstanta MAKS
5
6 struct Date { // Definisi global dari tipe Date
7     int month;
8     int day;
9     int year;
10 };
11
12 struct Student { // Definisi global dari tipe Student
13     char name[30];
14     Date birthday;
15 };
16
17 Student data_mhs[MAKS]; // Deklarasi global dari array student
18
19 int main() {
20     int i = 0, sudah_benar, jml;
21     char lagi;
22
23     // Memasukkan data
24     do {
25         cout << "Name : ";
26         cin.getline(data_mhs[i].name, sizeof(data_mhs[i].name));
27         cout << "Birthday (mm-dd-yyyy): ";
28         cin >> data_mhs[i].birthday.month >> data_mhs[i].birthday.day >> data_mhs[i].birthday.year;
    } while (i < MAKS);
    cout << "DATA SISWA\n";
    for (i = 0; i < MAKS; i++) {
        cout << i + 1 << ". Name : " << data_mhs[i].name << " Birthday : " << data_mhs[i].birthday.month << "-" << data_mhs[i].birthday.day << "-" << data_mhs[i].birthday.year << "\n";
    }
    return 0;
}

```

Output

```

/tmp/hilcUblA7Y.o
Name : Fina Salsabila Pramudita
Birthday (mm-dd-yyyy): 03-12-2004
Mau memasukkan data lagi [Y/T] ? y
Name : Wildan
Birthday (mm-dd-yyyy): 4-16-1974
Mau memasukkan data lagi [Y/T] ? t
DATA SISWA
1. Name : Fina Salsabila Pramudita Birthday : 3--12--2004
2. Name : Wildan Birthday : 4--16--1974

```

Gambar 3. Array struktur data untuk menyimpan data – data student

Perubahan C ke C++

- Penggunaan 'getline' alih-alih 'fgets' untuk membaca string dari input.
- Penambahan tipe data 'Date' untuk mewakili tanggal lahir mahasiswa.
- Penambahan struct sebelum 'Date' dan 'Student' untuk menentukan tipe data struktur tersebut.
- Penggunaan 'cin.ignore()' untuk membersihkan sisa data dalam penampung keyboard setelah menggunakan 'cin'

Analisis

- Struktur 'Date' untuk menyimpan informasi tentang tanggal dengan tiga lainnya 'month', 'day', dan 'year'.
- Struktur 'Student' digunakan untuk menyimpan informasi tentang seorang mahasiswa dengan 'name' dan 'birthday'
- Sebuah array 'data_mhs' dari tipe 'Student' sebagai variabel global untuk menyimpan data mahasiswa. Jumlah maksimum mahasiswa ditentukan oleh konstanta 'MAKS'.
- Fungsi 'main()', yang merupakan titik masuk utama untuk eksekusi program.

- Variabel lokal `i`, `sudah_benar`, dan `jml` digunakan dalam program.
- Memasukkan nama dan tanggal lahir setiap mahasiswa dalam array `data_mhs` menggunakan `cin` dan `getline`
- 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`.
- Setelah pengguna selesai memasukkan data, program menampilkan data mahasiswa yang telah dimasukkan menggunakan loop `for`

Program 4

Melewatkan elemen struktur sebagai parameter fungsi secara nilai

```
#include <iostream> // Menggantikan <stdio.h> untuk operasi input/output
using namespace std;

// Deklarasi fungsi cetak_tanggal
void cetak_tanggal(int, int, int);

int main() {
    // Definisi lokal dari tipe date
    struct Date {
        int month;
        int day;
        int year;
    } today;

    // Meminta pengguna memasukkan tanggal saat ini
    cout << "Enter the current date (mm-dd-yyyy): ";
    cin >> today.month >> today.day >> today.year;

    // Memanggil fungsi cetak_tanggal dengan melewati elemen struktur
    today sebagai parameter
    cetak_tanggal(today.month, today.day, today.year);

    return 0;
}

// Definisi fungsi cetak_tanggal
```

```

void cetak_tanggal(int mm, int dd, int yy) {
    // Array nama_bulan untuk menyimpan nama bulan
    static const char *nama_bulan[] = {
        "Wrong month", "January", "February", "March",
        "April", "May", "June", "July", "August",
        "September", "October", "November", "December"
    };

    // Mencetak tanggal saat ini dengan nama bulan yang sesuai
    cout << "Today's date is " << nama_bulan[mm] << " " << dd << ", " << yy
    << endl << endl;
}

```

The screenshot shows a C++ IDE with a file named `main.cpp`. The code defines a `struct Date` with `month`, `day`, and `year` members. In the `main` function, it prompts the user to enter the current date (mm-dd-yyyy), reads the input, and then calls the `cetak_tanggal` function, passing the `month`, `day`, and `year` members of the `today` struct as arguments. The `cetak_tanggal` function (defined at the bottom) uses a static array of month names to print the date in a formatted string.

```

main.cpp
1 #include <iostream> // Menggantikan <stdio.h> untuk operasi input/output
2 using namespace std;
3
4 // Deklarasi fungsi cetak_tanggal
5 void cetak_tanggal(int, int, int);
6
7 int main() {
8     // Definisi lokal dari tipe date
9     struct Date {
10         int month;
11         int day;
12         int year;
13     } today;
14
15     // Meminta pengguna memasukkan tanggal saat ini
16     cout << "Enter the current date (mm-dd-yyyy): ";
17     cin >> today.month >> today.day >> today.year;
18
19     // Memanggil fungsi cetak_tanggal dengan melewati elemen struktur today sebagai parameter
20     cetak_tanggal(today.month, today.day, today.year);
21
22     return 0;
23 }
24
25 // Definisi fungsi cetak_tanggal
26 void cetak_tanggal(int mm, int dd, int yy)
27 {
    // Array nama_bulan untuk menyimpan nama bulan

```

Output

```

/tmp/hIlcUblA7Y.o
Enter the current date (mm-dd-yyyy): 3 4 2024
Today's date is March 4, 2024

```

Gambar 4. Melewatkan elemen struktur sebagai parameter fungsi secara nilai

Perubahan C ke C++

- Penambahan 'using namespace std;' telah dihilangkan karena telah diubah menjadi penggunaan 'std::' sebelum 'cout' dan 'cin'
- Penggunaan struktur 'Date' telah dideklarasikan secara lokal di dalam fungsi 'main'

- Fungsi 'cetak_tanggal' telah diubah menjadi fungsi 'void' (tanpa nilai pengembalian).
- Perubahan dalam fungsi 'cetak_tanggal' untuk menggunakan 'cout' alih-alih 'printf', dan juga untuk menggantikan nama_bulan dari tipe 'char *' menjadi 'const char *', sesuai dengan tipe data array yang konstan.

Analisis

- Definisi lokal dari tipe 'Date', yang memiliki tiga diantaranya, 'month', 'day', dan 'year' untuk menyimpan tanggal.
- Fungsi 'main()', yang merupakan titik masuk utama untuk eksekusi program.
- Variabel lokal 'today' dari tipe 'Date' untuk menyimpan tanggal saat ini yang dimasukkan.
- Memasukkan tanggal saat ini melalui pesan yang ditampilkan menggunakan 'cout'.
- Input tanggal diterima menggunakan 'cin' dan disimpan dalam variabel 'today'.
- Setelah menerima input, fungsi 'cetak_tanggal' dipanggil dengan melewati tanggal saat ini.
- 'cetak_tanggal()' Ini adalah fungsi yang digunakan untuk mencetak tanggal dengan nama bulan yang sesuai.
- Menerima tiga parameter: 'mm', 'dd', dan 'yy'.
- Array 'nama_bulan' untuk menyimpan nama-nama bulan.
- Menggunakan 'cout' untuk mencetak tanggal dengan nama bulan yang sesuai.
- Array 'nama_bulan' ini digunakan untuk menyimpan nama-nama bulan.

Program 5

Melewatkan elemen struktur sebagai parameter fungsi secara acuan

```
#include <iostream> // Menggantikan <stdio.h> untuk operasi input/output
using namespace std;

// Deklarasi prototipe fungsi tukar_xy
void tukar_xy(int *, int *);

int main() {
    // Definisi struktur koordinat
    struct Koordinat {
        int x;
        int y;
    } 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
tukar_xy(&posisi.x, &posisi.y);

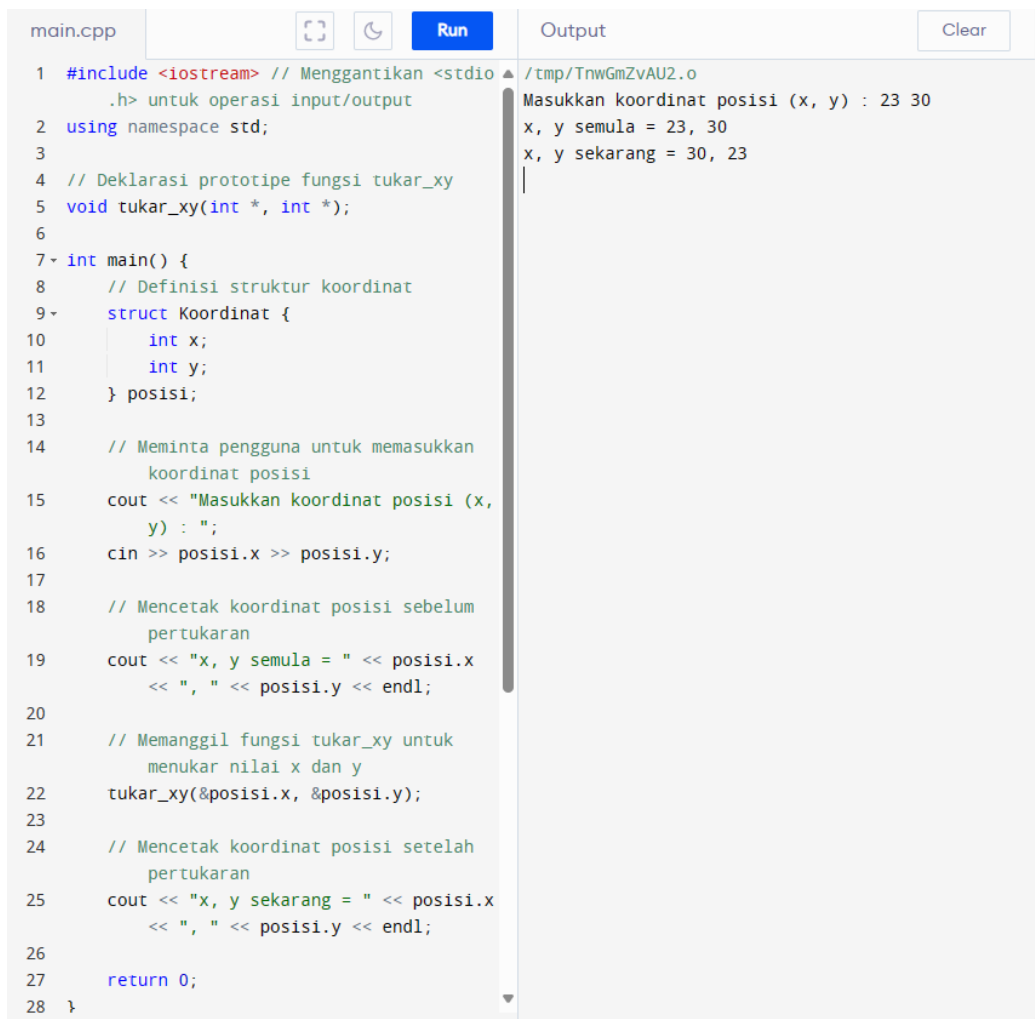
// Mencetak koordinat posisi setelah pertukaran
cout << "x, y sekarang = " << posisi.x << ", " << posisi.y << endl;

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;
}

```



```
main.cpp [Run] Output [Clear]
1 #include <iostream> // Menggantikan <stdio.h> untuk operasi input/output
2 using namespace std;
3
4 // Deklarasi prototipe fungsi tukar_xy
5 void tukar_xy(int *, int *);
6
7 int main() {
8     // Definisi struktur koordinat
9     struct Koordinat {
10         int x;
11         int y;
12     } posisi;
13
14     // Meminta pengguna untuk memasukkan koordinat posisi
15     cout << "Masukkan koordinat posisi (x, y) : ";
16     cin >> posisi.x >> posisi.y;
17
18     // Mencetak koordinat posisi sebelum pertukaran
19     cout << "x, y semula = " << posisi.x << ", " << posisi.y << endl;
20
21     // Memanggil fungsi tukar_xy untuk menukar nilai x dan y
22     tukar_xy(&posisi.x, &posisi.y);
23
24     // Mencetak koordinat posisi setelah pertukaran
25     cout << "x, y sekarang = " << posisi.x << ", " << posisi.y << endl;
26
27     return 0;
28 }
```

Output: /tmp/TnwGmZvAU2.o
Masukkan koordinat posisi (x, y) : 23 30
x, y semula = 23, 30
x, y sekarang = 30, 23

Gambar 5. Melewatkan elemen struktur sebagai parameter fungsi secara acuan

Perubahan C ke C++

- `struct Koordinat` alih-alih `struct koordinat` untuk mendefinisikan tipe data struktur.
- Menambahkan prototipe fungsi `void tukar_xy(int *, int *);` sebelum fungsi `main`.
- Merubah deklarasi dan penggunaan fungsi `tukar_xy` untuk menerima parameter bertipe pointer ke int.

Analisis

- Definisi lokal dari tipe `Koordinat`, yang memiliki dua anggota: `x` dan `y` digunakan untuk menyimpan koordinat posisi.
- Fungsi `main()`, yang merupakan titik masuk utama untuk eksekusi program.
- Variabel lokal `posisi` dari tipe `Koordinat` digunakan untuk menyimpan koordinat posisi yang dimasukkan.
- Memasukkan koordinat posisi melalui pesan yang ditampilkan menggunakan `cout`.
- Input koordinat posisi diterima menggunakan `cin` dan disimpan dalam variabel `posisi`.

- Setelah menerima input, fungsi `tukar_xy` dipanggil dengan melewati alamat variabel `posisi.x` dan `posisi.y` sebagai parameter.
- `tukar_xy()` Ini adalah fungsi yang digunakan untuk menukar nilai x dan y.
- `tukar_xy()` Menerima dua parameter: `a` dan `b`, yang merupakan pointer ke int.
- `tukar_xy()` ini bertujuan untuk menukar nilai yang ditunjuk oleh pointer `a` dan `b` menggunakan variabel `z` sebagai penyimpanan sementara.
- Program mencetak nilai x dan y sebelum pertukaran menggunakan `cout`.

Program 6

Melewatkan struktur sebagai parameter fungsi

```
#include <iostream> // Menggantikan <stdio.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 << "Enter the current date (mm-dd-yyyy): ";
    cin >> today.month >> today.day >> today.year;

    // Memanggil fungsi cetak_tanggal untuk mencetak tanggal
    cetak_tanggal(today);

    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[] = {
        "Wrong month", "January", "February", "March",
        "April", "May", "June", "July", "August",
        "September", "October", "November", "December"
    };

    // Mencetak tanggal yang dimasukkan oleh pengguna
    cout << "Today's date is " << nama_bulan[now.month] << " " << now.day
    << ", " << now.year << endl << endl;
}

```

```

main.cpp
1 #include <iostream> // Menggantikan <stdio
  .h> untuk operasi input/output
2 using namespace std;
3
4 // Definisi struktur tanggal
5 struct Date {
6     int month;
7     int day;
8     int year;
9 };
10
11 // Deklarasi prototipe fungsi
    cetak_tanggal dengan parameter bertipe
    Date
12 void cetak_tanggal(Date);
13
14 int main() {
15     // Definisi variabel today dengan tipe
    Date
16     Date today;
17
18     // Meminta pengguna untuk memasukkan
    tanggal saat ini
19     cout << "Enter the current date (mm-dd
    -yyyy): ";
20     cin >> today.month >> today.day >>
    today.year;
21
22     // Memanggil fungsi cetak_tanggal
    untuk mencetak tanggal
23     cetak_tanggal(today);
24
25     return 0;
26 }
27
28 // Definisi fungsi cetak tanggal dengan

```

Output

```

/tmp/TnwGmZvAU2.o
Enter the current date (mm-dd-yyyy): 03 05 2024
Today's date is March 5, 2024

```

Gambar 6. Melewatkan struktur sebagai parameter fungsi

Perubahan C ke C++

- Struktur `Date` digunakan untuk mendefinisikan variabel `today`, yang memiliki anggota `month`, `day`, dan `year`.

- Fungsi `cetak_tanggal` dengan parameter bertipe `Date`. Fungsi `cetak_tanggal` didefinisikan untuk mencetak tanggal yang dimasukkan oleh pengguna.
- Penggunaan array `nama_bulan` untuk menyimpan nama-nama bulan dan mencetak nama bulan berdasarkan input bulan yang dimasukkan.

Analisis

- `Date` digunakan untuk merepresentasikan tanggal dengan anggota `month`, `day`, dan `year`. Memungkinkan penggunaan variabel `today` dengan tipe `Date` untuk menyimpan tanggal.
- Fungsi `main()` digunakan sebagai 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.
- Dalam versi C++ ini, tidak dilakukan validasi khusus untuk memastikan bahwa input yang dimasukkan adalah tanggal yang valid. Namun, bisa menambahkan validasi tambahan jika diperlukan.
- Menggunakan `cin` untuk menerima input dan `cout` untuk mencetak output yang telah diterima atau yang akan di isi

Program 7

Fungsi parameter berupa pointer yang menunjuk ke struktur

```
#include <iostream> // Menggantikan <stdio.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);

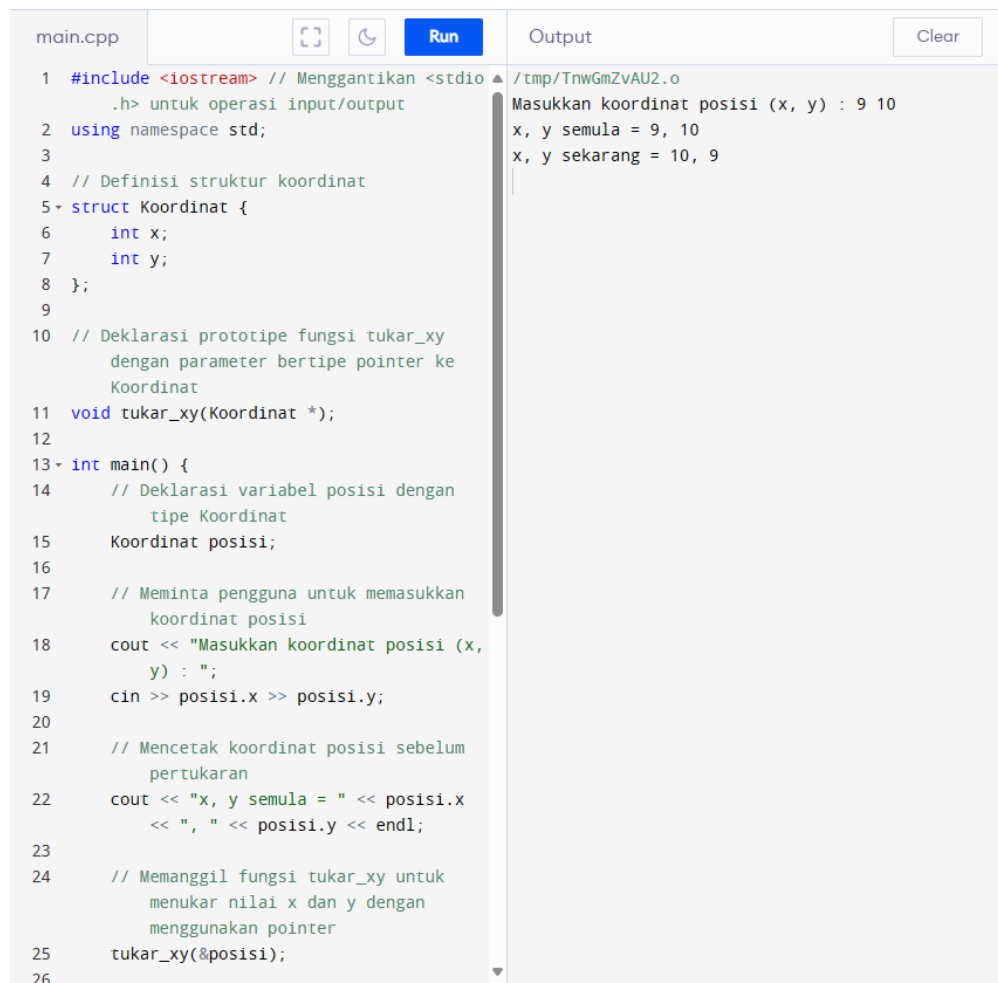
    // Mencetak koordinat posisi setelah pertukaran
    cout << "x, y sekarang = " << posisi.x << ", " << posisi.y << endl;

    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;
}

```



```
main.cpp  Run  Output  Clear

1 #include <iostream> // Menggantikan <stdio
  .h> untuk operasi input/output
2 using namespace std;
3
4 // Definisi struktur koordinat
5 struct Koordinat {
6     int x;
7     int y;
8 };
9
10 // Deklarasi prototipe fungsi tukar_xy
    dengan parameter bertipe pointer ke
    Koordinat
11 void tukar_xy(Koordinat *);
12
13 int main() {
14     // Deklarasi variabel posisi dengan
        tipe Koordinat
15     Koordinat posisi;
16
17     // Meminta pengguna untuk memasukkan
        koordinat posisi
18     cout << "Masukkan koordinat posisi (x,
        y) : ";
19     cin >> posisi.x >> posisi.y;
20
21     // Mencetak koordinat posisi sebelum
        pertukaran
22     cout << "x, y semula = " << posisi.x
        << ", " << posisi.y << endl;
23
24     // Memanggil fungsi tukar_xy untuk
        menukar nilai x dan y dengan
        menggunakan pointer
25     tukar_xy(&posisi);
26
```

Output

```

Masukkan koordinat posisi (x, y) : 9 10
x, y semula = 9, 10
x, y sekarang = 10, 9

```

Gambar 7. Fungsi parameter berupa pointer yang menunjuk ke struktur

Perubahan C ke C++

- `Koordinat` berfungsi `tukar_xy()` didefinisikan dengan parameter bertipe pointer ke `Koordinat`, yang memungkinkan fungsi ini untuk langsung mengakses dan memodifikasi nilai dari variabel `posisi` dalam `main()`.
- Penggunaan operator `->` digunakan untuk mengakses anggota dari struktur yang ditunjuk oleh pointer `pos_xy`. Ini lebih sederhana daripada penggunaan dereferensi (*).

Analisis

- Struktur `Koordinat` didefinisikan menggunakan kata kunci `struct`, yang memiliki dua anggota, `x` dan `y`, untuk merepresentasikan koordinat.
- Fungsi `tukar_xy` fungsi ini dideklarasikan dengan menggunakan prototipe yang mengambil parameter bertipe pointer ke `Koordinat`, sehingga fungsi ini dapat mengakses dan mengubah nilai dari variabel `posisi` dalam `main()`

- Fungsi 'main()' variabel 'posisi' dari tipe 'Koordinat' dideklarasikan untuk menyimpan koordinat yang dimasukkan oleh pengguna.
- Memasukkan nilai 'x' dan 'y'.
- Fungsi 'tukar_xy' 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 ditukar dengan nilai 'z'
- Input menggunakan 'cin' dan output menggunakan 'cout'.

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, 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)!

```
#include <iostream>
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;
    strcpy(data.word, "Hello");
    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` didefinisikan dengan tiga anggota: `loop` bertipe integer, `word` bertipe array karakter dengan panjang 5, dan `sum` bertipe float.
- Dalam fungsi `main()`, variabel `data` dari tipe `Record` dideklarasikan.
- Nilai-nilai dimasukkan ke dalam variabel `data`.
- Nilai-nilai tersebut kemudian dicetak ke layar.

2. Deklarasikan sebuah variabel struktur (misalkan namanya = sample) yang didefinisikan memiliki tipe struktur record.!

Deklarasi ini membuat variabel `sample` bertipe `Record`, sehingga variabel ini memiliki tiga anggota data: `loop`, `word`, dan `sum`, sesuai dengan definisi struktur `Record`. Variabel ini dapat digunakan untuk menyimpan data yang sesuai dengan struktur yang telah ditentukan sebelumnya.