**Laporan Hasil Praktikum**

Overview Sturktur

****

**Oleh :**

Tiyyasha Ananda Mufti Hanifah (5223600011)

**Program Studi D4 Teknologi Game**

**Departemen Teknologi Multimedia Kreatif**

**Politeknik Elektronika Negeri Surabaya**

**2024**

1. **Dasar Teori**

Struct dalam C++ adalah sebuah tipe data terstruktur yang memungkinkan kita untuk mengelompokkan beberapa variabel dengan tipe data yang berbeda menjadi satu kesatuan. Struct sering digunakan untuk mewakili data yang kompleks, seperti data mahasiswa, data karyawan, data buku, dan lain sebagainya.

Berikut adalah beberapa keuntungan menggunakan struct dalam C++:

1. Membuat kode lebih terstruktur dan mudah dibaca.
2. Memudahkan pengelolaan data yang kompleks.
3. Meningkatkan efisiensi memori dengan menggabungkan beberapa variabel menjadi satu kesatuan.
4. Memudahkan akses data dengan menggunakan nama anggota struct.
5. **Alat dan Bahan**

Alat yang digunakan pada percobaan menggambar kali ini adalah :

1. Laptop/PC
2. C++ compiler
3. Internet
4. **Analisa**
5. Mendefinisikan & Mendeklarasikan Struktur

struct date {

int month, day, year;

};

Analisa: Program untuk mendefinisikan struktur bernama date. Program ini memiliki tiga variable, yaitu:

* *month*: untuk menyimpan nilai bulan.
* *day*: untuk menyimpan nilai tanggal.
* *year*: untuk menyimpan nilai tahun.

struct student {

char name[30];

struct date birthday;

};

Analisa: Program ini mendefinisikan struktur bernama student. Program ini memiliki dua variable, yaitu:

* *name*: array dengan 30 karakter untuk menyimpan nama siswa.
* *date* birthday: variabel bertipe struck date untuk menyimpan informasi ulang tahun siswa.

struct student mhs; //deklarasi var mhs

Analisa: Program ini untuk mendeklarasikan variabel bernama mhs bertipe student. Ini untuk menyimpan informasi seorang siswa, termasuk nama dan tanggal lahir mereka.

1. Mengakses Elemen Struktur

#include <iostream>

#include <string>

using namespace std;

struct Date {

int month;

int day;

int year;

};

struct student {

string name;

date birthday;

};

student mhs;

int main() {

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

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

return 0;

}

Analisa: Program ini mendefinisikan dua struktur data, *date* dan *student*, dan menggunakannya untuk menampung informasi tentang seorang siswa di *main*. Program ini menunjukkan bagaimana cara mengisi dan mencetak nilai *field* dari struktur data.

1. Menginisialisasi Struktur

#include <iostream>

using namespace std;

struct rasi\_bintang {

char nama[11];

int tanggal\_awal;

int bulan\_awal;

int tanggal\_akhir;

int bulan\_akhir;

};

static struct rasi\_bintang zodiak = {"Sagitarius", 22, 11, 21, 12};

int main() {

int tanggal\_lahir, bulan\_lahir, tahun\_lahir;

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

cin >> tanggal\_lahir >> bulan\_lahir >> tahun\_lahir;

if ((tanggal\_lahir >= zodiak.tanggal\_awal && bulan\_lahir == zodiak.bulan\_awal) ||

(tanggal\_lahir <= zodiak.tanggal\_akhir && bulan\_lahir == zodiak.bulan\_akhir)) {

cout << "Zodiak Anda adalah " << zodiak.nama << endl;

} else {

cout << "Zodiak Anda bukan " << zodiak.nama << endl;

}

return 0;

}

Analisa: Program ini menggunakan satu struktur data, yaitu *rasi\_bintang*  yang berisi variable berikut:

* nama: variabel string dengan array panjang 11 karakter untuk menyimpan nama rasi bintang.
* tanggal\_awal: variabel integer untuk menyimpan tanggal awal periode rasi bintang.
* bulan\_awal: variabel integer untuk menyimpan bulan awal periode rasi bintang.
* tanggal\_akhir: variabel integer untuk menyimpan tanggal akhir periode rasi bintang.
* bulan\_akhir: variabel integer untuk menyimpan bulan akhir periode rasi bintang.

1. Array dan Struktur

#include <iostream>

#include <string>

using namespace std;

#define MAX\_STUDENT 20

struct date {

int month;

int day;

int year;

};

struct StudentInfo {

string name;

date dateOfBirth;

};

// Deklarasi global dari variabel student

StudentInfo studentData[MAX\_STUDENT];

int main() {

int currentIndex = 0, isCorrect, totalStudents;

char continueInput;

// Memasukkan data

do {

cout << "Nama: ";

getline(cin, studentData[currentIndex].name);

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

cin >> studentData[currentIndex].dateOfBirth.month >>

studentData[currentIndex].dateOfBirth.day >>

studentData[currentIndex].dateOfBirth.year;

cout << endl;

currentIndex++;

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

do {

continueInput = getchar(); // Baca tombol

isCorrect = (continueInput == 'Y') || (continueInput == 'y') ||

(continueInput == 'T') || (continueInput == 't');

} while (!isCorrect);

// Hapus sisa data dalam penampung keyboard

cin.ignore();

cout << endl;

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

totalStudents = currentIndex;

// Menampilkan data

cout << "DATA SISWA" << endl;

for (currentIndex = 0; currentIndex < totalStudents; currentIndex++) {

cout << currentIndex + 1 << ". Nama: " << studentData[currentIndex].name;

cout << " Tanggal Lahir: " << studentData[currentIndex].dateOfBirth.month <<

"-" << studentData[currentIndex].dateOfBirth.day << "-" <<

studentData[currentIndex].dateOfBirth.year << endl << endl;

}

return 0;

};

Analisa: Program ini menggunakan dua struktur data *date* dan *StudentInfo* dan menggunakan perulangan *while* yang memungkinkan untuk memasukkan nama dan tanggal lahir hingga 20 siswa.

1. Melewatkan Elemen Struktur ke dalam Fungsi

#include <iostream>

using namespace std;

void tukar\_p\_q(int\*, int\*);

int main() {

struct titik {

int p;

int q;

};

titik lokasi;

cout << "Masukkan koordinat titik (p, q): ";

cin >> lokasi.p >> lokasi.q;

cout << "p, q semula = " << lokasi.p << ", " << lokasi.q << endl;

tukar\_p\_q(&lokasi.p, &lokasi.q);

cout << "p, q sekarang = " << lokasi.p << ", " << lokasi.q << endl;

return 0;

}

void tukar\_p\_q(int\* x, int\* y) {

int z;

z = \*x;

\*x = \*y;

\*y = z;

}

Analisa: Program ini menunjukkan cara menukar nilai dua variabel integer yang disimpan dalam struct dengan menggunakan fungsi dan pointer.

1. Melewatkan Struktur ke dalam Fungsi

#include <iostream>

#include <string>

using namespace std;

struct Tanggal { // Nama struct diubah menjadi Tanggal

int bulan; // Nama member diubah menjadi bulan, hari, dan tahun

int hari;

int tahun;

};

void tampilkanTanggal(Tanggal tanggalSaatIni); // Nama fungsi diubah menjadi tampilkanTanggal

int main() {

Tanggal tanggalHariIni; // Nama variabel diubah menjadi tanggalHariIni

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

cin >> tanggalHariIni.bulan >> skipws >> tanggalHariIni.hari >> skipws >> tanggalHariIni.tahun;

tampilkanTanggal(tanggalHariIni);

return 0;

}

void tampilkanTanggal(Tanggal tanggalSaatIni) { // Nama parameter fungsi diubah menjadi tanggalSaatIni

static const string namaBulan[] = {

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

"April", "Mei", "Juni", "Juli", "Agustus",

"September", "Oktober", "November", "Desember"

};

if (tanggalSaatIni.bulan < 1 || tanggalSaatIni.bulan > 12) {

cout << "Bulan yang Anda masukkan tidak valid." << endl;

return;

}

cout << "Tanggal hari ini adalah " << namaBulan[tanggalSaatIni.bulan] << " " << tanggalSaatIni.hari << ", " << tanggalSaatIni.tahun << endl;

}

Analisa: Program ini menunjukkan cara mendeklarasikan struct, memasukkan data ke dalam struct, dan menampilkan data struct dengan format yang lebih mudah dibaca.

1. Struktur dan Pointer (Pointer ke Struktur)

#include <iostream>

using namespace std;

struct koordinat {

int p;

int q;

};

void tukar\_pq(struct koordinat \*pos\_pq) {

int ;r

r = pos\_pq->p;

pos\_pq->p = pos\_pq->q;

pos\_pq->q = r;

}

int main() {

struct koordinat posisi;

cout << "Masukkan koordinat posisi (p, q) : ";

cin >> posisi.p >> posisi.q;

cout << "p,q semula = " << posisi.p << ", " << posisi.q << endl;

tukar\_pq(&posisi);

cout << "p,q sekarang = " << posisi.p << ", " << posisi.q << endl;

return 0;

}

Analisa: Program C++ ini menunjukkan cara menukar nilai p dan q dalam sebuah struct menggunakan fungsi dan pointer. Dengan memasukkan nilai p dan q, kemudian program menukar nilai tersebut dan mencetaknya kembali.

1. **Latihan Soal**
2. 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).

* struct record {

int loop;

char word[5];

float sum;

};

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

* struct record {

int id;

std::string nama;

float usia;

};

record sample;

1. **Kesimpulan**

Dalam C++, *struct* (struktur) adalah sebuah tipe data bentukan yang memungkinkan pengelompokkan variabel-variabel terkait di bawah satu nama. Variabel-variabel penyusun *struct* dapat memiliki tipe data yang berbeda-beda, sehingga program dapat lebih terorganisir dan mudah dipahami ketika menangani data yang kompleks.