**TUGAS PEMROGRAMAN BERORIENTASI OBJEK**



**Disusun oleh:**

**Nama: Haniza Kurnia Dwi Putri**

**NIM: 22091397070**

**Kelas: 2022B**

**UNIVERSITAS NEGERI SURABAYA**

**2023**

1. **Class Circle**

* Source code

#include <iostream>

// Deklarasi kelas Circle

class Circle {

private:

double radius;

public:

// Konstruktor untuk inisialisasi objek Circle

Circle(double r) {

radius = r;

}

// Metode untuk menghitung luas lingkaran

double calculateArea() {

return 3.14159 \* radius \* radius;

}

// Metode untuk menghitung keliling lingkaran

double calculateCircumference() {

return 2 \* 3.14159 \* radius;

}

// Metode untuk mengubah nilai radius

void setRadius(double r) {

radius = r;

}

// Metode untuk mendapatkan nilai radius

double getRadius() {

return radius;

}

};

int main() {

// Membuat objek Circle

Circle lingkaran(5.0);

// Menggunakan metode-metode dari objek Circle

std::cout << "Radius Lingkaran: " << lingkaran.getRadius() << std::endl;

std::cout << "Luas Lingkaran: " << lingkaran.calculateArea() << std::endl;

std::cout << "Keliling Lingkaran: " << lingkaran.calculateCircumference() << std::endl;

// Mengubah nilai radius

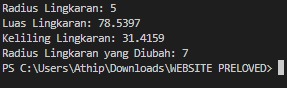
lingkaran.setRadius(7.0);

std::cout << "Radius Lingkaran yang Diubah: " << lingkaran.getRadius() << std::endl;

return 0;

}

* Program running



1. **Class Date**

* Source code

#include <iostream>

class Date {

private:

int day;

int month;

int year;

public:

// Konstruktor untuk inisialisasi objek Date

Date(int d, int m, int y) {

day = d;

month = m;

year = y;

}

// Metode untuk menampilkan tanggal

void displayDate() {

std::cout << day << "/" << month << "/" << year << std::endl;

}

// Metode untuk mengatur tanggal

void setDate(int d, int m, int y) {

day = d;

month = m;

year = y;

}

// Metode untuk mendapatkan nilai hari

int getDay() {

return day;

}

// Metode untuk mendapatkan nilai bulan

int getMonth() {

return month;

}

// Metode untuk mendapatkan nilai tahun

int getYear() {

return year;

}

};

int main() {

// Membuat objek Date

Date today(3, 10, 2023);

// Menampilkan tanggal

std::cout << "Tanggal Hari Ini: ";

today.displayDate();

// Mengubah tanggal

today.setDate(4, 10, 2023);

std::cout << "Tanggal Setelah Diubah: ";

today.displayDate();

// Mengakses nilai hari, bulan, dan tahun

std::cout << "Hari: " << today.getDay() << std::endl;

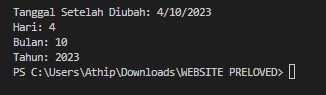
std::cout << "Bulan: " << today.getMonth() << std::endl;

std::cout << "Tahun: " << today.getYear() << std::endl;

return 0;

}

* Program running



1. **Class Time**

* Source code

#include <iostream>

class Time {

private:

int hour;

int minute;

int second;

public:

// Konstruktor untuk inisialisasi objek Time

Time(int h, int m, int s) {

hour = h;

minute = m;

second = s;

}

// Metode untuk menampilkan waktu

void displayTime() {

std::cout << hour << ":" << minute << ":" << second << std::endl;

}

// Metode untuk mengatur waktu

void setTime(int h, int m, int s) {

hour = h;

minute = m;

second = s;

}

// Metode untuk mendapatkan nilai jam

int getHour() {

return hour;

}

// Metode untuk mendapatkan nilai menit

int getMinute() {

return minute;

}

// Metode untuk mendapatkan nilai detik

int getSecond() {

return second;

}

};

int main() {

// Membuat objek Time

Time currentTime(15, 30, 45);

// Menampilkan waktu

std::cout << "Waktu Saat Ini: ";

currentTime.displayTime();

// Mengubah waktu

currentTime.setTime(16, 45, 10);

std::cout << "Waktu Setelah Diubah: ";

currentTime.displayTime();

// Mengakses nilai jam, menit, dan detik

std::cout << "Jam: " << currentTime.getHour() << std::endl;

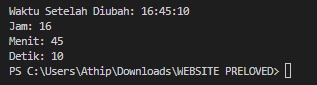
std::cout << "Menit: " << currentTime.getMinute() << std::endl;

std::cout << "Detik: " << currentTime.getSecond() << std::endl;

return 0;

}

* Program running



1. **Class Point**

* Source code

#include <iostream>

class Point {

private:

double x;

double y;

public:

// Konstruktor untuk inisialisasi objek Point

Point(double xCoord, double yCoord) {

x = xCoord;

y = yCoord;

}

// Metode untuk menampilkan koordinat titik

void displayPoint() {

std::cout << "(" << x << ", " << y << ")" << std::endl;

}

// Metode untuk mengatur koordinat titik

void setPoint(double xCoord, double yCoord) {

x = xCoord;

y = yCoord;

}

// Metode untuk mendapatkan nilai koordinat x

double getX() {

return x;

}

// Metode untuk mendapatkan nilai koordinat y

double getY() {

return y;

}

};

int main() {

// Membuat objek Point

Point titikA(2.5, 3.0);

// Menampilkan koordinat titik

std::cout << "Koordinat Titik A: ";

titikA.displayPoint();

// Mengubah koordinat titik

titikA.setPoint(4.0, 5.5);

std::cout << "Koordinat Titik A yang Diubah: ";

titikA.displayPoint();

// Mengakses nilai koordinat x dan y

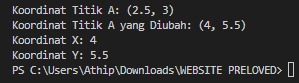
std::cout << "Koordinat X: " << titikA.getX() << std::endl;

std::cout << "Koordinat Y: " << titikA.getY() << std::endl;

return 0;

}

* Program running



1. **Class Account**

* Source code

#include <iostream>

#include <string>

class Account {

private:

std::string accountNumber;

std::string accountHolder;

double balance;

public:

// Konstruktor untuk inisialisasi objek Account

Account(const std::string &number, const std::string &holder, double initialBalance) {

accountNumber = number;

accountHolder = holder;

balance = initialBalance;

}

// Metode untuk menampilkan informasi akun

void displayAccountInfo() {

std::cout << "Nomor Akun: " << accountNumber << std::endl;

std::cout << "Pemilik Akun: " << accountHolder << std::endl;

std::cout << "Saldo Akun: $" << balance << std::endl;

}

// Metode untuk setor dana ke akun

void deposit(double amount) {

if (amount > 0) {

balance += amount;

std::cout << "Setoran berhasil. Saldo baru: $" << balance << std::endl;

} else {

std::cout << "Setoran gagal. Jumlah setoran harus lebih dari 0." << std::endl;

}

}

// Metode untuk tarik dana dari akun

void withdraw(double amount) {

if (amount > 0 && amount <= balance) {

balance -= amount;

std::cout << "Penarikan berhasil. Saldo baru: $" << balance << std::endl;

} else {

std::cout << "Penarikan gagal. Jumlah penarikan tidak valid atau saldo tidak mencukupi." << std::endl;

}

}

// Metode untuk mendapatkan saldo akun

double getBalance() {

return balance;

}

};

int main() {

// Membuat objek Account

Account myAccount("12345", "John Doe", 1000.0);

// Menampilkan informasi akun

std::cout << "Informasi Akun Awal:" << std::endl;

myAccount.displayAccountInfo();

// Melakukan setoran dan penarikan

myAccount.deposit(500.0);

myAccount.withdraw(200.0);

// Menampilkan saldo akun setelah transaksi

std::cout << "Saldo Akun Setelah Transaksi: $" << myAccount.getBalance() << std::endl;

return 0;

}

* Program running

