Praktikum 7

Pemrograman Berbasis Obyek (PBO)

(Polimorfisme)

A. Tujuan

1. Mahasiswa menerapkan konsep Polimorfisme pada pemrograman PBO

B. Dasar Teori

Polymorphism (polimorfisme) adalah kemampuan untuk mempunyai beberapa bentuk class yang berbeda. Polimorfisme merupakan suatu konsep yang menyatakan bahwa sesuatu yang sama dapat memiliki berbagai bentuk dan perilaku yang berbeda. Dalam hal ini polimorfisme merupakan suatu sifat menyandarkan pada kesamaan nama dalam program. Pengenal data, instans, dan metode, bahkan nama fungsi dapat dibuat dengan nama yang sama untuk kegunaan yang berbeda. Contoh polimorfisme adalah pada mobil yang ada di pasaran terdiri atas berbagai tipe dan merk, tetapi semuanya memiliki interface yang sama, seperti: stir, tongkat transmisi, pedal gas dan rem. Jika seseorang dapat mengemudikan satu jenis mobil saja dari satu merk tertentu, maka orang itu akan dapat mengemudikan hampir semua jenis mobil yang ada, karena semua mobil tersebut menggunakan interface yang sama. Contoh lainnya adalah pada program untuk menghitung suatu luasan atau keliling pada area tertentu. Persegi panjang, lingkaran, segitiga merupakan suatu bentuk(*shape*), masing-masing dari bentuk tersebut mempunyai input dan cara perhitungan yang berbeda, sehingga dalam pembuatan dan akses program memungkinkan terjadi perulangan-perulangan pada program, dengan menerapkan konsep polimorfisme maka dapat mengurangi duplikasi kode program dan membuta program lebih terstruktur. Berikut contoh programnya.

```
#include <iostream>
class persegipanjang : public shape {
private:
    float panjang;
    float lebar;
public:
    persegipanjang(float dpanjang, float dlebar)
    {
        panjang = dpanjang;
        lebar = dlebar;
    }
    float getLuas()
    {
        return panjang * lebar;
    }
```

```
    float getKeliling()
    {
         return (2 * (panjang + lebar));
    }
};
class lingkaran : public shape{
private :
    float radius;
public:
    lingkaran (float dradius)
    {
        radius = dradius;
    }
    float getLuas()
    {
        return 3.14 * radius * radius;
    }
    float getKeliling ()
    {
        return 2 * 3.14 * radius;
    }
};

int main(int argc, const char * argv[]) {

    persegipanjang pp(10,5);
    std::cout << "luas: " << pp.getLuas() << std:: endl;
    std::cout << "keliling : " << pp.getKeliling() << std:: endl;


    lingkaran ling(5);
    std::cout << "luas: " << ling.getLuas() << std:: endl;
    std::cout << "keliling : " << ling.getKeliling() << std:: endl;
    return 0;
}
```

Pada program diatas terdapat class persegipanjang dan lingkaran yang merupakan suatu bentuk/*shape* yang menghasikan output data luas dan keliling dengan input dan proses perhitungan yang berbeda, pada program tersebut dalam penggunaan atau akses program untuk Luas(getLuas) dan Keliling(getKeliling) harus dideklaraasikan masing-masing (contoh pada pp.getLuas(), ling.getLuas()). Dengan menerapkan konsep polomorfisme maka dapat membuat program lebih mudah dalam akses member fungtion yang ada didalam masing-masing class dengan menambahkan program sebagai berikut.

```
#include <iostream>
class shape {
public:
    virtual float getLuas() =0;
    virtual float getKeliling() =0;
};
class persegipanjang : public shape {
private:
    float panjang;
    float lebar;
public:
    persegipanjang(float dpanjang, float dlebar)
    {
        panjang = dpanjang;
        lebar = dlebar;
    }
    float getLuas()
    {
        return panjang * lebar;
    }
    float getKeliling()
    {
        return (2 * (panjang + lebar));
    }
};
class lingkaran : public shape{
private :
    float radius;
public:
    lingkaran (float dradius)
    {
        radius = dradius;
    }
    float getLuas()
    {
        return 3.14 * radius * radius;
    }
    float getKeliling ()
    {
        return 2 * 3.14 * radius;
    }
};

void printshape(shape &bentuk)
{
    std::cout << "Luas      : " << bentuk.getLuas() << std:: endl;
    std::cout << "Keliling  : " << bentuk.getKeliling() << std:: endl;
}

int main(int argc, const char * argv[]) {

    persegipanjang pp(10,5);
    lingkaran ling(5);

    printshape(pp);
    printshape(ling);

    return 0;
}
```

Program diatas menerapkan konses polimorfisme dengan menjadikan fungsi virtual**.** Cara menjadikan fungsi virtual menjadi fungsi virtual adalah dengan menambahkan kerword **virtual** dan menghilangkan body fungsi, seperti terlihat pada potongan code di bawah ini.

```
class Shape
{
    protected:
        char nama[20];
    public:
      //fungsi virtual
       virtual void hitung() = 0;
};
```

C. Percobaan

Buatlah program berdasarkan berdasarkan studi kasus dibawah ini dengan menerapakan konsep polimorfisme pad PBO.

Sebuah perusahaan membutuhkan simulasi program untuk menghitung data gaji karyawan setiap bulannya. Karyawan pada perusahaan tersebut terdiri dari direktur, manager, teknisi dan operator. Variabel perhitunagn Gaji dari dari masing-masing karyawan adalah sebagai berikut :

Karyawan	Gaji Pokok	Uang Makan / Hari	Uang Transport/hari	Uang Lembur / Jam
Direktur	IDR5.400.000	IDR 120.000	IDR 50.000	IDR 50.000
Manager	IDR4.500.000	IDR 100.000	IDR 30.000	IDR 40.000
Teknisi	IDR3.200.000	IDR 60.000	IDR 20.000	IDR 15.000
Operator	IDR2.400.000	IDR 60.000	IDR 20.000	IDR 10.000