

### TP Modul 3 Programming

#### *Object-Oriented Programming*

Pada TP ini akan ada dua soal. Masing-masing soal dikerjakan pada file .cpp yang berbeda. Soal satu diberi nama file TP3\_Soal1.cpp dan soal kedua diberi nama TP3\_Soal2.cpp. Pada kedua-dua soal ini, Anda diminta untuk melengkapi code yang diberi. Perhatikan komentar-komentarnya karena itu merupakan perintah yang Anda harus selesaikan.

#### 1. Object, Class, dan Operator Overloading

Buatlah sebuah class bernama LinearLine. Class ini memiliki empat floating point member data, yaitu x1, y1, x2, dan y2. Anda juga diminta untuk membuat method yang berisikan fungsi-fungsi untuk menghitung kemiringan, y-intercept (nilai y ketika  $x = 0$ ), dan fungsi lain-lainnya (cermati perintah pada komentar-komentar di code). Lengkapilah struktur class di bawah ini

```
#include <iostream>

class LinearLine
{
private:
    // empat float data type, yaitu x1, y1, x2, dan y2

public:
    LinearLine (float x1, float y1, float x2, float y2) {
        // Isilah definisi constructor ini dengan tujuan untuk
        // menginisialisasi data member
    }
    float gradient() {
        // hitunglah gradient berdasarkan 4 titik tersebut
    }
    float y_intercept() {
        // hitunglah nilai y ketika  $x = 0$ 
    }
    void operator + (float a) {
        // Buatlah operator overloading untuk menggeser garis linear
        // sejauh a pada arah sumbu x
    }
    void printEquation() {
        // print equation dengan format  $y = mx + c$ 
        // perhatikan  $m = \text{gradient}$  dan  $c = \text{y\_intercept}$ 
    }
}
```

```
void printPoints() {
    // print nilai member data dengan format
    // (x1, y1) = ( , )
    // (x2, y2) = ( , )
}

};

int main()
{
    LinearLine line(1.0f, 8.0f, 2.0f, 3.0f);    // inisialisasi objek line
    int a = 3;                                // nilai banyaknya penggeseran pada persamaan linear
    /*
    lengkapi fungsi main sehingga output program seperti pada contoh di bawah

    (x1, y1) = (1, 8)
    (x2, y2) = (2, 3)
    Sehingga didapatkan
    gradient = -5
    y_intercept = 13
    y = -5 x + 13

    Setelah digeser sejauh a, didapatkan persamaan baru:
    y = -5 x + 28
    */
}
```

## 2. Inheritance, Function Overriding, dan Virtual Function

Anda diberikan sebuah base class bernama Shape. Class tersebut memiliki turunan class, yaitu Rectangle. Class Rectangle memiliki member (float) length dan (float) width. Buatlah implementasi code-nya dan fokus utama dari class-nya adalah ingin menghitung area dan keliling dari objeknya

```
class Shape
{
private:
    float area;
    float keliling;

public:
    // gunakan virtual function
    virtual float calculateArea() { return 0; }
    float calculateKeliling() { return 0; }
};
```

```
// Buatlah class Rectangle sebagai turunan dari class Shape
// {
    // deklarasikan member data, (float) width dan (float) length

    // Buatlah constructor untuk menginisialisasi member!
    // Buatlah calculateArea()
    // Buatlah calculateKeliling()

// };

int main()
{
    Rectangle rect (5.0, 4.0);
    Shape* shape1 = &rect;

    // dengan menggunakan pointer shape1, panggilah
    // function calculateArea() dan calculateKeliling() pada class Rectangle
    // PERIKSA HASILNYA BENAR ATAU SALAH? Jika salah maka perbaikilah base class-nya

    return 0;
}
```