

LAPORAN SEDERHANA  
OPERATOR OVERLOAD OOP C++



Nama : Muhammad Fathi Farhat  
NRP : 5024211065  
Departemen : Teknik Komputer  
Mata Kuliah : Pemrograman Lanjutan  
Kelas : B  
Link Github : [https://github.com/Blindheroes/tugas-2\\_operator-overload](https://github.com/Blindheroes/tugas-2_operator-overload)

INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA  
2022

## DEKLARASI CLASS DAN ATRIBUTE

```
9  class PersegiPanjang{
10     public:
11
12     float y_min ;
13     float y_max ;
14
15     float x_min ;
16     float x_max ;
17 }
```

Pada line 9-16 dideklarasikan class bernama **PersegiPanjang** dengan atribute y\_min, y\_max, x\_min, dan x\_max menggunakan tipe data float agar dapat di isi oleh bilangan pecahan.

## KONSTRUKTOR

```
18 //konstruktor; panjang = vertikal(y), lebar = horizontal(x); object(TP x, TP y, kor. x, kor.y);
19 PersegiPanjang(float titikPusat_x, float titikPusat_y, float lebar, float panjang){
20     PersegiPanjang::x_min = titikPusat_x - (lebar / 2);
21     PersegiPanjang::x_max = titikPusat_x + (lebar / 2);
22
23     PersegiPanjang::y_min = titikPusat_y - (panjang / 2);
24     PersegiPanjang::y_max = titikPusat_y + (panjang / 2);
25
26 }
```

pada line 19-24 dibuat konstruktor dengan parameter (tp x, tp y, kor. x, kor. Y). yangmana variabel panjang dan lebar akan dibagi 2 lalu di tambah dengan titik pusat hingga jadilah koordinat y\_min, y\_max, x\_min, dan x\_max.

## OPERATOR ==

```
29 //operator overloading ==
30 bool operator == (PersegiPanjang object2){
31     int irisan_x = 0, irisan_y = 0;
32
33     //syarat beririsan pada sumbu Y
34 }
```

Pertama, operator dideklarasikan dengan format [ tipe data + operator + tanda operator + parameter]

```
35 // jika object2 berada disebelah atas
36 if(PersegiPanjang::y_max >= object2.y_min && PersegiPanjang::y_min <= object2.y_max){
37     irisan_y = 1;
38 }
39 // jika object2 berada disebelah bawah
40 else if ( object2.y_min <= PersegiPanjang::y_max && object2.y_max >= PersegiPanjang::y_min)
41     irisan_y = 1;
42 }
43 }
```

Mengecek apakah persegi 1 dan 2 beririsan pada sumbu y

```

44 //syarat beririsan pada sumbu X
45
46 // jika object2 berada disebelah atas
47 if(PersegiPanjang::x_max >= object2.y_min && PersegiPanjang::y_min <= object2.x_max){
48     irisan_y = 1;
49 }
50 // jika object2 berada disebelah bawah
51 else if ( object2.y_min <= PersegiPanjang::x_max && object2.x_max >= PersegiPanjang::y_min){
52     irisan_y = 1;
53 }

```

Mengecek apakah persegi 1 dan 2 beririsan pada sumbu x

```

54
55 //jika beririsan pada sumbu X atau Y
56 if (irisan_x == 1 || irisan_y == 1 ) return true;
57 return false;
58 }

```

Jika beririsan pada sumbu x atau y maka akan dikembalikan nilai true.

## OPERATOR +

```

//operator overloading +
PersegiPanjang operator + (PersegiPanjang object2){

    PersegiPanjang hasil(0,0,0,0);
}

```

Pertama, operator dideklarasikan dengan format [ tipe data + operator + tanda operator + parameter] dan dideklarasikan juga object hasil.

```

67     if(PersegiPanjang::x_max >= object2.x_max){
68         hasil.x_max = PersegiPanjang::x_max;
69     }else hasil.x_max = object2.x_max;
70
71     if(PersegiPanjang::x_min <= object2.x_min){
72         hasil.x_min = PersegiPanjang::x_min;
73     }else hasil.x_min = object2.x_min;

```

Mengambil nilai x yang paling besar dan kecil dari kedua persegi

```

75         if(PersegiPanjang::y_max >= object2.y_max){
76             hasil.y_max = PersegiPanjang::y_max;
77         }else hasil.y_max = object2.y_max;
78
79         if(PersegiPanjang::y_min <= object2.y_min){
80             hasil.y_min = PersegiPanjang::y_min;
81         }else hasil.y_min = object2.y_min;
82

```

Mengambil nilai y yang paling besar dan kecil dari kedua persegi

```

85
86         return hasil;
87     }
88

```

Dan nilai yg di kembalikan adalah object hasil

#### OPERATOR ++

```

//operator overloading ++
void operator ++ (){

    PersegiPanjang::x_max *= 2;
    PersegiPanjang::x_min *= 2;
    PersegiPanjang::y_max *= 2;
    PersegiPanjang::y_min *= 2;

}

```

Pertama, operator dideklarasikan dengan format [ tipe data + operator + tanda operator]. lalu deklarasikan nilai atribut y\_min, y\_max, x\_min, dan x\_max dengan dikalikan 2.

#### OPERATOR -

```

104 //operator overloading -
105 PersegiPanjang operator - (PersegiPanjang object2){
106
107     PersegiPanjang hasil(0,0,0,0);

```

Pertama, operator dideklarasikan dengan format [ tipe data + operator + tanda operator + parameter] dan dideklarasikan juga object hasil.

```

109     if(PersegiPanjang::x_max >= object2.x_max){
110         hasil.x_max = object2.x_max;
111     }else hasil.x_max = PersegiPanjang::x_max;
112
113     if(PersegiPanjang::x_min <= object2.x_min){
114         hasil.x_min = object2.x_min;
115     }else hasil.x_min = PersegiPanjang::x_min;

```

lalu ambil nilai x yg lebih kecil di antara persegi 1 & 2.

```

116
117     if(PersegiPanjang::y_max >= object2.y_max){
118         hasil.y_max = object2.y_max;
119     }else hasil.y_max = PersegiPanjang::y_max;
120
121     if(PersegiPanjang::y_min <= object2.y_min){
122         hasil.y_min = object2.y_min;
123     }else hasil.y_min = PersegiPanjang::y_min;
124
125
126     return hasil;

```

kemudian ambil nilai y yg lebih kecil di antara persegi 1 & 2 dan return berupa object hasil.

## OPERATOR –

```

130     //operator overloading --
131     void operator -- (){
132
133         PersegiPanjang::x_max /= 2;
134         PersegiPanjang::x_min /= 2;
135         PersegiPanjang::y_max /= 2;
136         PersegiPanjang::y_min /= 2;
137
138     }
139

```

Pertama, operator dideklarasikan dengan format [ tipe data + operator + tanda operator]. lalu deklarasikan nilai atribut y\_min, y\_max, x\_min, dan x\_max dengan dibagi 2.

## Tambahan :

### 1. Class Kartesius\_1\_object

Berfungsi untuk menggambar bentuk persegi pada bidang kartesius sesuai titik koordinat yg di ketahui.

### 2. Class Kartesius\_2\_object

Berfungsi untuk menggambar bentuk 2 persegi pada bidang kartesius sesuai titik koordinat yg di ketahui.

Cara memanggil:

A. Operator +

```
478 //memanggil operator overload +  
479 hasil = object1 + object2;  
480 hasil.printData(3);
```

B. Operator ++

```
494 //memanggil operator overload ++  
495 ++object1;  
496 object1.printData(1);
```

C. Operator -

```
486 //memanggil operator overload -  
487 hasil = object1 - object2;  
488 hasil.printData(3);
```

D. Operator --

```
510 //memanggil operator overload --  
511 --object1;
```

E. Operator ==

```
//jika sama maka akan di eksekusi  
while (object1 == object2)  
{
```