

LAPORAN OPERATOR OVERLOADING PEMROGRAMAN LANJUT



Disusun Oleh

Ikhwanul Abiyu Dhiyya'ul Haq

5024211048

Dosen Pengampu

Reza Fuad Rachmadi

19850403201221000

INSTITUT TEKNOLOGI SEPULUH NOPEMBER SURABAYA

TEKNIK KOMPUTER

2022

Penjelasan Logika Program

Program dapat dilihat pada link [Soellassy24/rectangle-operator-overloading: C++ program to calculate 2 rectangle by +, -, ++, --, ==, and \[\] operators \(github.com\)](https://soellassy24.github.io/rectangle-operator-overloading-C++-program-to-calculate-2-rectangle-by-+-,-,++,--,==,and[]-operators/) . Program dibuat menggunakan bahasa C++. Terdapat 3 file yang berbeda, masing-masing ada di dalam folder include dan src. Folder include diisi oleh file persegiPanjang.hpp yang berisikan deklarasi class dari persegiPanjang, sedangkan folder src diisi oleh file persegiPanjang.cpp yang berisikan definisi dari class persegiPanjang, dan program main.cpp berisikan keluaran dari seluruh operator yang diminta. File-file ini akan disatukan oleh Makefile, sehingga akan menjadi 1 file *executable*, yaitu pp.exe.

Class persegiPanjang memiliki 4 atribut private bertipe data float, agar memungkinkan operasi dalam angka desimal, yaitu xMaks, xMin, yMaks, yMin. xMaks yaitu koordinat sisi kanan persegi panjang, xMin yaitu koordinat sisi kiri persegi panjang, yMaks merupakan koordinat sisi atas persegi panjang, dan yMin merupakan koordinat sisi bawah persegi panjang. Class persegiPanjang tadi juga memiliki fungsi public, yaitu constructor dan operator overloading untuk mendapatkan operasi dengan operator + (Menambah luasan 2 persegi panjang yang berisikan), operator – (Mengambil irisan antara 2 persegi panjang yang berisikan), operator ++ (Menjadikan suatu persegi panjang memiliki luasan 2 kali sebelumnya), operator – (Menjadikan suatu persegi panjang memiliki luasan $\frac{1}{2}$ kali sebelumnya), operator [] (Mengambil masing-masing atribut dari persegi panjang), dan operator == (menentukan apakah persegi panjang saling beririsan).

Constructor class persegiPanjang memiliki input titikTengahX, titikTengahY, panjang (sumbu x), dan lebar (sumbu y). Maka, untuk mendapatkan xMin, caranya adalah titikTengahX - (panjang/2), lalu xMaks dicari dengan cara titikTengahX + (panjang/2), yMin dicari dengan cara titikTengahY - (lebar/2), dan yMaks dicari dengan cara titikTengahY + (lebar/2).

Operator overloading penambahan (+) akan menghasilkan persegi panjang dengan luasan gabungan dari dua persegi panjang, dan dengan syarat kedua persegi panjang harus beririsan. Operator + memiliki return berupa class PersegiPanjang dan dengan parameter berupa const PersegiPanjang &baru. Digunakan reference (&) agar parameter function ini akan langsung menjadi alias dari parameter dan tidak membuat variable baru, sedangkan const digunakan untuk memproteksi agar parameter rhs tidak berubah saat function dijalankan. Operator + juga dibuat menjadi const agar tidak mengubah variable yang melakukan operator, yaitu pada sisi kiri. Sebagai

contoh $pp1 + pp2$, maka dari itu $pp1$ akan melakukan operator $+$ dengan $pp2$ sebagai parameternya (baru). Dalam definisinya, akan dilakukan pengecekan terlebih dahulu apakah kedua persegi panjang tersebut beririsan atau tidak dengan menggunakan operator $==$, yaitu `"if (*this == baru)"`. Apabila kedua persegi panjang beririsan, maka komparasi akan mengeluarkan nilai `true` (1). Dan bila `true`, maka cara untuk mendapatkan $xMin$ adalah nilai $xMin$ terkecil dari kedua persegi panjang, yaitu `"min(this->xMin, baru.xMin)"`. Sedangkan untuk mendapatkan $xMaks$ adalah nilai $xMaks$ terbesar dari kedua persegi panjang, yaitu dengan `"max(this->xMaks, baru.xMaks)"`. Begitu juga untuk mencari $yMin$ dan $yMaks$. Kemudian akan dibuat variable persegiPanjang baru dan mengubah semua atributnya, lalu variable tersebut akan di-return.

Adapun untuk operator $-$ mirip dengan operator $+$, namun kebalikannya, dimana caranya sama, namun untuk pengambilan min dan max yang berbeda. Cara untuk mendapatkan $xMin$ adalah nilai $xMin$ terbesar dari kedua persegi panjang, yaitu `"max(this-> xMin, baru.xMin)"` dan mendapatkan $xMaks$ terkecil dengan cara `"min(this-> xMaks, baru.xMaks)"`. Metode yang sama digunakan untuk mencari $yMaks$ dan $yMin$. Setelah nilai-nilai tadi didapatkan, baru di-return.

Operator $++$ memiliki jenis fungsi void, sehingga tidak memiliki nilai kembalian. Operator $++$ berfungsi untuk menambah luasan persegi panjang menjadi 2 kali luas awalnya, caranya adalah menambah panjang dan lebarnya menjadi 2 kalinya, namun titik tengahnya tetap sama. Panjang didapatkan dari `"this->xMaks - this->xMin;"` begitu pula lebar, dari `"this->yMaks - this->yMin;"`. Kedua nilai ini dimasukkan kedalam fungsi `abs()` atau `absolute`, untuk menghindari nilai negatif. $xMidpoint$ didapat dari `"this->xMin + panjang/2"` atau `"this-> xMaks - panjang/2"` juga bisa. Lalu $yMidpoint$ didapat dari `"this->yMin + lebar/2"` atau `"this-> yMaks - lebar/2"`. Nantinya variabel panjang dan lebar akan dikali 2. Setelah itu baru $xMin$, $yMin$, $xMaks$, dan $yMaks$ dirubah lagi. Untuk X maupun Y min, $x/yMidpoint$ dikurang oleh $panjang/2$ (untuk $xMin$) atau $lebar/2$ (untuk $yMin$). Untuk X maupun Y maks, $x/yMidpoint$ ditambah oleh $panjang/2$ (untuk $xMaks$) dan $lebar/2$ (untuk $yMaks$).

Operator $-$ memiliki jenis fungsi void, sehingga tidak memiliki nilai kembalian. Operator $-$ berfungsi untuk mengurangi luasan persegi panjang menjadi 2 kalinya, atau $\frac{1}{2}$ kali luasan awalnya. Operator $--$ memiliki cara yang persis seperti operator $++$ untuk logikanya, hanya saja nanti panjang dan lebar nya menjadi $panjang/2$ dan $lebar/2$.

Operator `==` berfungsi untuk menyatakan benar tidaknya 2 persegi panjang beririsan atau tidak, operator ini berjenis `bool`, yang akan mengembalikan value `true` jika `xMaks` lebih besar dari `baru.xMin` (persegi panjang satunya), `xMin` lebih kecil dari `baru.xMaks`, `yMaks` lebih besar dari `baru.yMin`, `yMin` lebih kecil dari `baru.yMaks`. Jika 4 syarat tersebut terpenuhi, maka akan mereturn nilai `true`. Jika ada 1 syarat yang tidak terpenuhi, maka akan mereturn nilai `false`.

Operator `[]` berfungsi untuk menampilkan `xMin/xMaks/yMin/yMaks` dari suatu persegi panjang, di sini saya menggunakan `switch case`, sehingga dalam operator `[]` memiliki constructor pilihan. Jika memilih 1 akan *me-return* `xMin` dari persegi panjang. 2 akan *me-return* `xMaks`, 3 akan *me-return* `yMin`, 4 akan *me-return* `yMaks`.