

BAB 10

CLASS (SETTER & GETTER)

10.1. TUJUAN PEMBELAJARAN

Tujuan dari pembelajaran pada pertemuan *setter* dan *getter*, diantaranya mahasiswa diharapkan mampu:

1. Memahami dan mampu mengimplementasi class ke dalam program.
2. Memahami konsep *setter* dan *getter*.
3. Membuat program dengan menerapkan *setter* dan *getter* pada class.

10.2. DASAR TEORI

Class merupakan cetak biru (*blueprint*) atau prototipe/kerangka yang mendefinisikan variabel – variabel (data) dan fungsi – fungsi (perilaku) umum dari sebuah objek tertentu. Prinsip kerja *class* yaitu dengan mengumpulkan data member dan fungsi member (juga dikenal sebagai metode) yang berhubungan satu sama lain untuk membentuk objek yang memiliki *comportement* dan data tertentu. Setiap objek yang dibuat dari *class* tersebut akan memiliki properti dan *comportement* yang sama seperti yang ditentukan dalam class itu sendiri.

Class memiliki tingkat akses yang merupakan cara bagaimana kita membatasi akses terhadap atribut atau method dalam sebuah class. Ada tiga tingkat akses dalam OOP (Object Oriented Programming), diantaranya:

1. Public: Semua bagian dari program dapat mengakses atribut atau method yang ditandai sebagai public.
2. Private: Atribut atau method yang ditandai sebagai private hanya bisa diakses oleh class itu sendiri.
3. Protected: Atribut atau method yang ditandai sebagai protected hanya bisa diakses oleh class itu sendiri dan turunannya.

Penggunaan tingkat akses ini sangat penting untuk memastikan bahwa logika aplikasi berjalan sesuai dengan yang diharapkan dan membatasi akses terhadap bagian yang tidak perlu untuk diakses oleh bagian lain dari aplikasi.

Pada implementasinya, yang membedakan *class* dengan tipe data sederhana adalah fungsi pendeklarasiannya. Di mana pada tipe data sederhana biasanya digunakan untuk

mendeklarasikan variabel ‘normal’, sedangkan *class* digunakan untuk mendeklarasikan sebuah variabel yang berupa objek. Sebagai contoh, jika terdapat sebuah *class* yang diberi nama “Class Animalia”, maka semua data dan perilaku yang dimiliki oleh kategori animalia akan dibungkus ke dalam kelas tersebut.

Di dalam *class* terdapat istilah *setter* dan *getter*. *Setter* dan *Getter* merupakan metode pada *class* yang digunakan untuk mengatur dan mengambil nilai dari data member (property) pada objek. Sederhananya *Setter* digunakan untuk mengatur nilai dari data member, sedangkan *Getter* digunakan untuk mengambil nilai dari data member. Biasanya *setter* dan *getter* dapat ditemui pada saat membuat program yang terdapat member function dan enkapsulasi (hak akses **public**, **private** dan **protected**). **Tujuan digunakan *setter* dan *getter* agar program bisa melakukan pemrosesan atau proses validasi sebelum mengakses data member.**

10.2.1. *Setter*

Setter merupakan member function yang dipakai untuk memberikan nilai ke dalam sebuah data member. Umumnya, dalam penulisan sebuah program, nama member function pada *setter* akan diawali dengan kata “set” yang diikuti dengan nama data member yang akan di-set. Sebagai contoh, jika kita ingin membuat *setter* untuk data member mahasiswa, maka nama member functionnya menjadi setMahasiswa(). Atau *setter* untuk data member luasLingkaran akan menjadi setLuasLingkaran().

Sebagai contoh, perhatikan program di bawah ini :

```
#include <iostream>

using namespace std;

class Jurusan {

    private:
        string mahasiswa;
        int npm;

    public:
        void setMahasiswa(string var1) {
```

```

        mahasiswa = var1;
    }

    void setNPM(int var2) {
        npm = var2;
    }
};

int main()
{
    Jurusan jurusanRina;

    jurusanRina.setMahasiswa("Rina");
    jurusanRina.setNPM(212310008);

    return 0;
}

```

Pada program di atas hak akses yang digunakan sudah diubah dari **public** menjadi **private**. Sehingga data member mahasiswa dan npm tidak bisa diakses di luar class Jurusan. Untuk kondisi seperti ini, program dapat ditambahkan *setter* `setMahasiswa()` dan `setNPM()`. Kedua *member function* ini butuh satu parameter yang akan menampung argumen inputan. Parameter `var1` akan diinput ke dalam data member Mahasiswa, sedangkan parameter `var2` akan diinput ke dalam data member NPM. Terakhir kedua *setter* tersebut akan dipanggil kembali pada fungsi utama (`main`) menggunakan perintah `jurusanRina.setMahasiswa("Rina")` dan `jurusanRina.setNPM(212310008)`. Setelah menjalankan kedua *member function* ini, maka data member Mahasiswa dan NPM akan berisi string “Rina” dan angka 212310009

10.2.2. Getter

Jika *setter* digunakan untuk mengatur nilai dari data member, maka *getter* digunakan untuk mengambil nilai dari data member. Penamaan *getter* diawali dengan “get” yang diikuti dengan nama data member yang akan di-get. Sebagai contoh, jika kita ingin membuat *getter*

untuk data member Mahasiswa, maka nama member functionnya menjadi `getMahasiswa()` atau *getter* untuk data member `luasLingkaran` akan menjadi `getLuasLingkaran()`.

Melanjutkan poin B.1 tentang *setter*. Untuk pengambilan nilai dari data member mahasiswa dan npm, tidak diperlukan lagi parameter dan pada program tersebut hanya perlu ditambahkan `cout << jurusanRina.getMahasiswa()` dan `cout << jurusanRina.getNPM()` yang dituliskan di dalam *function main()* tepatnya setelah pemanggilan *setter*.

```
cout << jurusanRina.getMahasiswa() << endl; // Rina
cout << jurusanRina.getNPM() << endl; // 212310008
```

10.3. PERCOBAAN

1. Buatlah workspace menggunakan Replit.
2. Buatlah project baru `Class_1` yang berisi file C++ source untuk *ex_class setter_getter*.
3. Cobalah untuk masing-masing percobaan di bawah.

Percobaan 1 : Implementasi *class*

```
#include <iostream>

using namespace std;

class Silsilah {
    protected:
        string orangtua = "Albert Wijaya dan Jenny Lim";
        string anak = "Agatha Wijaya dan Jonathan Wijaya";

    public:
        string cekSilsilah() {
            return "Ayah bernama Albert Wijaya dan ibu bernama Jenny Lim";
        }
}
```

```
};

class Cucu: public Silsilah {
    protected:
        string cucu = "Wilbert Tan dan Rayanza Wijaya";

    public:
        string cekAnak(){
            return "Anak pertama keluarga Wijaya adalah
Agatha Wijaya dan anak kedua dari keluarga Wijaya
adalah Jonathan Wijaya";
        }
};

class Anak: public Cucu {
    public:
        string lihatSilsilahnya() {
            return "Silsilah Keluarga Wijaya" ;
        }
        string cekCucu() {
            return "Cucu pertama bernama Wilbert Tan dan Cucu
Kedua bernama Rayanza Wijaya";
        }
};

int main()
```

```

{

    Anak silsilahWijaya;

    cout << silsilahWijaya.lihatSilsilahnya() << endl;
    cout << silsilahWijaya.cekSilsilah() << endl;
    cout << silsilahWijaya.cekAnak() << endl;
    cout << silsilahWijaya.cekCucu() << endl;

    return 0;
}

```

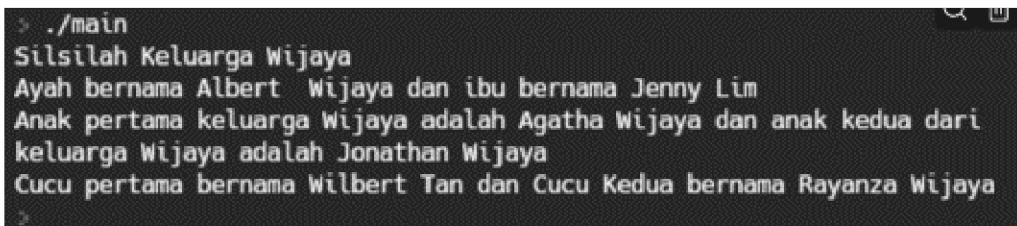
OUTPUT

Silsilah Keluarga Wijaya

Ayah bernama Albert Wijaya dan ibu bernama Jenny Lim

Anak pertama keluarga Wijaya adalah Agatha Wijaya dan anak kedua dari keluarga Wijaya adalah Jonathan Wijaya

Cucu pertama bernama Wilbert Tan dan Cucu Kedua bernama Rayanza Wijaya



```

> ./main
Silsilah Keluarga Wijaya
Ayah bernama Albert Wijaya dan ibu bernama Jenny Lim
Anak pertama keluarga Wijaya adalah Agatha Wijaya dan anak kedua dari
keluarga Wijaya adalah Jonathan Wijaya
Cucu pertama bernama Wilbert Tan dan Cucu Kedua bernama Rayanza Wijaya
>

```

Percobaan 2 : Implementasi *setter* dan *getter* pada class

```

#include <iostream>

#include <string>

using namespace std;

```

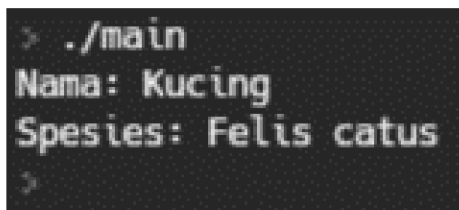
```
class Hewan {  
    private:  
        string nama;  
        string spesies;  
  
    public:  
        void setNama(string n) {  
            nama = n;  
        }  
  
        void setSpesies(string s) {  
            spesies = s;  
        }  
  
        string getNama() {  
            return nama;  
        }  
  
        string getSpesies() {  
            return spesies;  
        }  
};  
  
int main() {  
    Hewan hewan;
```

```
hewan.setNama("Kucing");  
  
hewan.setSpesies("Felis catus");  
  
cout << "Nama: " << hewan.getNama() << endl;  
  
cout << "Spesies: " << hewan.getSpesies() << endl;  
  
return 0;  
  
}
```

OUTPUT

Nama: Kucing

Spesies: Felis catus



```
> ./main  
Nama: Kucing  
Spesies: Felis catus  
>
```

10.4. TUGAS DAN LATIHAN

Contoh kasus setter dan getter di kehidupan nyata adalah perpustakaan dan pelajar. Dimana pelajar akan meminjam buku dari perpustakaan. Pelajar membutuhkan penjaga perpustakaan untuk mendapatkan buku yang ingin dipinjam maupun mengembalikan buku yang sudah dipinjam. Buatlah program berdasarkan studi kasus tersebut menggunakan setter dan getter pada c++!