

## BAB 11

### CLASS (CONSTRUCTOR DAN DESTRUCTOR)

#### 11.1. TUJUAN PEMBELAJARAN

Tujuan dari pembelajaran pada pertemuan *constructor* dan *destructor*, diantaranya mahasiswa diharapkan mampu:

1. Memahami konsep *constructor* dan *destructor*.
2. Membuat program dengan menerapkan *constructor* dan *destructor*.

#### 11.2. DASAR TEORI

##### 11.2.1. Constructor

Konstruktor (*constructor*) sendiri merupakan sebuah fungsi yang otomatis akan dipanggil setiap kali melakukan instansiasi terhadap kelas. Dapat diartikan juga suatu fungsi dari anggota suatu kelas yang memiliki nama yang sama dengan nama kelas fungsi itu berada. Konstruktor ini digunakan untuk mengalokasikan ruang untuk suatu objek dan untuk memberikan nilai awal (Stroustrup, 2013). Sama seperti fungsi biasa, *constructor* dapat ditambahkan parameter maupun dilakukan overload. Ketika membuat *constructor* pastikan nama dari fungsi *constructor* harus sama dengan nama kelasnya dan tidak memiliki tipe kembalian (tidak juga *void*).

##### 11.2.2. Destructor

*Destructor* merupakan kebalikan dari *constructor*. Destructor adalah method khusus yang dijalankan saat objek dari sebuah class dihapus atau tidak lagi digunakan. Biasanya fungsi ini dijalankan secara otomatis ketika program telah selesai dijalankan Destructor memiliki nama yang sama dengan nama class dimulai dengan tanda tilde ('~') dan tidak memiliki tipe data atau parameter. Destructor digunakan untuk membersihkan memori dan menjalankan tugas-tugas lain seperti menutup file atau membersihkan data.

#### 11.3. PERCOBAAN

1. Buatlah workspace menggunakan Replit.
2. Buatlah project baru Class\_2 yang berisi file C++ source untuk *destructor* dan *constructor*.
3. Cobalah untuk masing-masing percobaan di bawah.

##### Percobaan 1 : Implementasi *constructor* dengan String

```

#include <iostream>
#include <string>

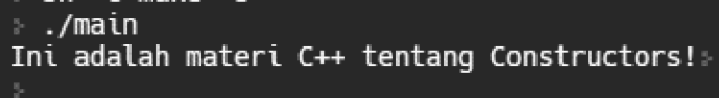
using namespace std;
class pelajaran {      // Class Name
    public:             // Access specifier
        pelajaran() {   // Constructor
            cout << "Ini adalah materi C++ tentang
Constructors!";
        }
};

int main() {
    pelajaran obj;      // membuat object dari sebuah
class
    return 0;
}

```

## OUTPUT

Ini adalah materi C++ tentang Constructors!



```

% ./main
Ini adalah materi C++ tentang Constructors!%

```

## Percobaan 2 : Implementasi *constructor* sebagai tempat inisialisasi data member

```

#include <iostream>

using namespace std;

class Laptop {

    private:
        string pemilik;

```

```

        string merk;

    public:
        Laptop(string var1, string var2) {
            pemilik = var1;
            merk = var2;

            cout << "Paket Laptop " << merk << " milik " <<
pemilik
                << " sudah dikirim" << endl;
        }

};

int main()
{
    Laptop laptopFadlan("Fadlan", "Asus");
    Laptop laptopRisma("Risma", "Acer");
    Laptop laptopCarlos("Carlos", "Lenovo");

    return 0;
}

```

## OUTPUT

Paket Laptop Asus milik Fadlan sudah dikirim

Paket Laptop Acer milik Risma sudah dikirim

Paket Laptop Lenovo milik Carlos sudah dikirim

```

Paket Laptop Asus milik Fadlan sudah dikirim
Paket Laptop Acer milik Risma sudah dikirim
Paket Laptop Lenovo milik Carlos sudah dikirim

```

## Percobaan 3 : Implementasi *destructor*

```

#include <iostream>

#include <string>

```

```

using namespace std;

class Mahasiswa {
public:
    Mahasiswa(string nama, int umur)
        : nama_(nama), umur_(umur) {
        cout << "Constructor Mahasiswa" << endl;
    }
    ~Mahasiswa() {
        cout << "==Contoh Destructur Mahasiswa==" <<
endl;
    }
    void TampilkanBiodata() {
        cout << "Nama: " << nama_ << endl;
        cout << "Umur: " << umur_ << endl;
    }

private:
    string nama_;
    int umur_;
};

int main() {
    {
        Mahasiswa mhs("Katarina", 20);
        mhs.TampilkanBiodata();
    }

    return 0;
}

```

## OUTPUT

Constructor Mahasiswa

Nama: Katarina

Umur: 20

==Contoh Destructor Mahasiswa==

```
➤ ./destructor
Constructor Mahasiswa
Nama: Katarina
Umur: 20
==Contoh Destructor Mahasiswa==
➤
```

#### 11.4. TUGAS DAN LATIHAN

Ketik ulang program di bawah ini dan berikan keterangan/komentar penjelasan terkait program yang dibuat, serta tentukan program dibawah merupakan penerapan dari *constructor* atau *destructor*!

```
#include <iostream>
using namespace std;

class contoh {
    private:
        int nilai;
    public:
        contoh (int n) {
            nilai = n;
        }

        int getNum() {
            return nilai;
        }
};

int main() {
    contoh obj(10);
    cout << "nilai yang diinput: " << obj.getNum() <<
endl;
    return 0;
}
```