
Object Oriented Programming

CONSTRUCTOR DAN DESTRUCTOR

CONSTRUCTOR

CONSTRUCTOR

- **Constructor merupakan method istimewa pada sebuah class yang secara otomatis akan dijalankan ketika sebuah obyek didefinisikan**
 - **constructor memiliki nama yang sama dengan nama class**
 - **constructor tidak memiliki return value (bersifat void)**
 - **jika constructor tidak di definisikan maka sebenarnya compiler akan secara otomatis membuat sebuah constructor tanpa parameter dan tidak memiliki statement didalamnya.**
-

CONTOH DALAM BAHASA C

```
class Sepeda{
public:
    /* ATTRIBUTES */
    int panjang, tinggi, jumlahRoda;
    string warna;
    /* END OF ATTRIBUTES */

    /* METHODS */
    Sepeda(){ // ** CONSTRUCTOR DECLARED
        this->panjang = 10;
        this->tinggi = 5;
        this->jumlahRoda = 2;
        this->warna = "silver";
    }

    void tampilkanDimensi(){
        cout<<"Dimensi sepeda adalah "<<this->panjang<<" x "<<this->tinggi<<endl;
    }
    void tampilkanWarna(){
        cout<<"warna sepeda adalah : "<<this->warna<<endl;
    }
    /* END OF OMETHODS */
};
```

CONTOH DALAM BAHASA C

```
int main(int argc, const char * argv[]) {  
    Sepeda sepedaGunung; // ** CONSTRUCTOR EXECUTE  
    // sepedaGunung adalah obyek dari class Sepeda  
  
    sepedaGunung.tampilkanDimensi();  
}
```

OUTPUT

```
Dimensi sepeda adalah 10 x 5  
Program ended with exit code: 0
```

**CONSTRUCTOR AKAN SECARA
OTOMATIS DIEKSEKUSI/DIJALANKAN
KETIKA OBYEK DARI SEBUAH CLASS
DIDEFINISIKAN**

MULTIPLE/OVERLOADING CONSTRUCTOR

- **Sebuah class dapat memiliki lebih dari satu constructor**
- **Setiap constructor dapat dibedakan berdasarkan**
 - ☒ **jumlah parameter**
 - ☒ **tipe data parameter**

CONTOH DALAM BAHASA C

```
class Sepeda{
public:
    /* ATTRIBUTES */
    int panjang, tinggi, jumlahRoda;
    string warna;
    /* END OF ATTRIBUTES */

    /* METHODS */
    Sepeda(){ // ** CONSTRUCTOR 1 DECLARED
        this->panjang = 10;
        this->tinggi = 5;
        this->jumlahRoda = 2;
        this->warna = "silver";
    }

    Sepeda(int panjang_, int tinggi_, string warna){ // ** CONSTRUCTOR 2 DECLARED
        this->panjang = panjang_;
        this->tinggi = tinggi_;
        this->jumlahRoda = 2;
        this->warna = warna;
    }

    void tampilkanDimensi(){
        cout<<"Dimensi sepeda adalah "<<this->panjang<<" x "<<this->tinggi<<endl;
    }
    void tampilkanWarna(){
        cout<<"warna sepeda adalah : "<<this->warna<<endl;
    }
    /* END OF OMETHODS */
};
```

DESTRUCTOR

DESTRUCTOR

- **Constructor merupakan method istimewa pada sebuah class yang secara otomatis akan dijalankan ketika**
 - ☒ **Akhir dari program**
 - ☒ **Akhir dari fungsi**
 - ☒ **Akhir dari block program**
 - **destructor memiliki nama yang sama dengan nama class dengan prefix '~' (tilde)**
 - **destructor tidak memiliki parameter dan tidak memiliki return value (bersifat void)**
-

CONTOH DALAM BAHASA C

```
class Sepeda{
public:
    /* ATTRIBUTES */
    int panjang, tinggi, jumlahRoda;
    string warna;
    /* END OF ATTRIBUTES */

    /* METHODS */
    Sepeda(){ // ** CONSTRUCTOR 1 DECLARED
        this->panjang = 10;
        this->tinggi = 5;
        this->jumlahRoda = 2;
        this->warna = "silver";
    }

    Sepeda(int panjang_, int tinggi_, string warna){ // ** CONSTRUCTOR 2 DECLARED
        this->panjang = panjang_;
        this->tinggi = tinggi_;
        this->jumlahRoda = 2;
        this->warna = warna;
    }

    ~Sepeda(){ // ** DESTRUCTOR DECLARED
        cout<<"destructor called"<<endl;
    }

    void tampilkanDimensi(){
        cout<<"Dimensi sepeda adalah "<<this->panjang<<" x "<<this->tinggi<<endl;
    }
    void tampilkanWarna(){
        cout<<"warna sepeda adalah : "<<this->warna<<endl;
    }
    /* END OF OMETHODS */
};
```

CONTOH DALAM BAHASA C

```
int main(int argc, const char * argv[]) {
    Sepeda sepedaGunung; // ** CONSTRUCTOR EXECUTE
    // sepedaGunung adalah obyek dari class Sepeda

    {
        Sepeda sepedaBalap; // ** constructor of sepedaBalap execute
    } //destructor of sepedaBalap execute

    sepedaGunung.tampilkanDimensi();
} //destructor of sepedaGunung execute
```

TUGAS MANDIRI

TUGAS MANDIRI

➤ **Pada program tugas sebelumnya**

- ☐ **Definisikan constructor dan destructor pada class yang telah dibuat**
 - ☐ **Berikan statement `print("constructor <nomor> called")` di dalam body constructor pada baris pertama**
 - ☐ **Berikan statement `print("destructor called")` di dalam body destructor**
 - ☐ **Pada `main()` , definisikan 1 block statement dengan dua block statement didalamnya (nested block statement)**
-

TERIMA KASIH
