

DELIVERABLE LATIHAN KENDARAAN: C++

IF2210 Object Oriented Programming



Dipersiapkan oleh:

Kelompok 8

Muhammad Alif Putra Yasa	13520135
Ghazian Tsabit Alkamil	13520165

**PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG**

2022

Daftar Isi

Daftar Isi	1
Deskripsi Program	2
class Kendaraan	2
class KoleksiKendaraan	4
class Bus	6
class Minibus	8
class Mobil	9
Link Source Code	10

Deskripsi Program

Program terdiri dari beberapa kelas, yaitu:

1. Kendaraan
2. KoleksiKendaraan
3. Bus
4. Minibus
5. Mobil

Masing-masing kelas terdiri dari dua file, yaitu file *header* bertipe *hpp* dan file *source code* bertipe *cpp*.

class Kendaraan

Kelas ini merepresentasikan sebuah kendaraan. Kelas ini merupakan kelas abstrak, dapat dilihat dari metode virtual abstrak *biayaSewa*. Berikut *source code* dari program tersebut:

```
/*Kendaraan.hpp*/
#ifndef __KENDARAAN_H__
#define __KENDARAAN_H__

#include <string>

class Kendaraan {
    private:
        int nk;           // Nomor Kendaraan
        std::string kategori;
        std::string merk;
        int tk;           // Tahun Keluar
    public:
        Kendaraan();
        Kendaraan(int nomorK, std::string cat, std::string merk, int
tk );
        Kendaraan(const Kendaraan& K);
        ~Kendaraan();
        virtual void printInfo();
        virtual int biayaSewa(int lamaSewa) = 0;

};

#endif
```

```

/*Kendaraan.cpp*/
#include "Kendaraan.hpp"
#include <iostream>
#include <ostream>

Kendaraan::Kendaraan(){
    this->nk = 0;
    this->tk = 0;
    this->merk = "XXX";
    this->kategori = "mobil";
}

Kendaraan::Kendaraan(int nomorK, std::string cat, std::string merk,
int tk ){
    this->nk = nomorK;
    this->tk = tk;
    this->merk = merk;
    this->kategori = cat;
}

Kendaraan::Kendaraan(const Kendaraan& K){
    this->nk = K.nk;
    this->tk = K.tk;
    this->kategori = K.kategori;
    this->merk = K.merk;
}

Kendaraan::~Kendaraan(){

}

void Kendaraan::printInfo(){
    std::cout <<
        "Nomor Kendaraan : " << this->nk << std::endl <<
        "Kategori          : " << this->kategori << std::endl <<
        "Merk              : " << this->merk << std::endl <<
        "Tahun Keluar      : " << this->tk << std::endl;
}

```

```

int Kendaraan::biayaSewa(int lamaSewa){
    if(this->kategori == "Bus"){
        return 1000000 * lamaSewa;
    } else if (this->kategori == "Minibus"){
        if(lamaSewa <= 5){
            return 5000000;
        } else {
            return 5000000 + 500000 * (lamaSewa - 5);
        }
    } else if(this->kategori == "Mobil"){
        return 500000 * lamaSewa;
    } else {
        return 0;
    }
}

```

class KoleksiKendaraan

Kelas ini merepresentasikan kumpulan kendaraan. Karena kelas Kendaraan merupakan kelas abstrak, kelas ini menyimpan pointer ke sebuah kendaraan di dalam *array*-nya. Berikut merupakan *source code*-nya:

```

/*KoleksiKendaraan.hpp*/
#ifndef __KOLEKSI_KENDARAAN_H__
#define __KOLEKSI_KENDARAAN_H__
#define DEFAULT_SIZE 100

#include "Kendaraan.hpp"

class KoleksiKendaraan {
    private:
        Kendaraan** AoK;
        int nEff;
        int size;
    public:
        KoleksiKendaraan();
        KoleksiKendaraan(int sz);
        KoleksiKendaraan(const KoleksiKendaraan& obj);
        ~KoleksiKendaraan();
}

```

```

    void printAll();
    void operator<<(Kendaraan& K);
    void operator<<(KoleksiKendaraan& KK);
};

```

```

#endif

```

```

/*KoleksiKendaraan.cpp*/
#include "KoleksiKendaraan.hpp"
#include <iostream>
#include <ostream>

KoleksiKendaraan::KoleksiKendaraan(){
    this->AoK = new Kendaraan*[DEFAULT_SIZE];
    this->nEff = 0;
    this->size = DEFAULT_SIZE;
}

KoleksiKendaraan::KoleksiKendaraan(int sz){
    this->AoK = new Kendaraan*[sz];
    this->nEff = 0;
    this->size = sz;
}

KoleksiKendaraan::KoleksiKendaraan(const KoleksiKendaraan& KK){
    this->AoK = new Kendaraan*[KK.size];
    for(int i = 0; i < KK.nEff; i++){
        this->AoK[i] = KK.AoK[i];
    }
    this->size = KK.size;
    this->nEff = KK.nEff;
}

KoleksiKendaraan::~KoleksiKendaraan(){
    delete [] this->AoK;
}

void KoleksiKendaraan::printAll(){

```

```

        std::cout << "### INFO KENDARAAN ###" << std::endl;
        for(int i = 0; i < this->nEff; i++){
            std::cout << std::endl;
            this->AoK[i]->printInfo();
        }
    }

    void KoleksiKendaraan::operator<<(Kendaraan& K){
        if(this->nEff < this->size){
            this->AoK[this->nEff] = &K;
            this->nEff++;
        }
    }

    void KoleksiKendaraan::operator<<(KoleksiKendaraan& KK){
        for(int i = 0; this->nEff < this->size; i++){
            *this << *KK.AoK[i];
        }
    }
}

```

class Bus

Kelas ini merupakan kelas turunan dari kelas Kendaraan. Kelas ini memiliki atribut tambahan, yaitu atribut kapasitas yang bertipe *integer*. Berikut *source code* dari kelas berikut:

```

/*Bus.hpp*/
#ifndef __BUS_H__
#define __BUS_H__

#include "Kendaraan.hpp"

class Bus : public Kendaraan {
    private:
        int kapasitas;
    public:
        Bus();
        Bus(int nomorK, std::string merk, int tk, int cap);
        Bus(const Bus& K);
        ~Bus();
        int biayaSewa(int lamaSewa) override;
        void printInfo() override;
}

```

```
};
```

```
#endif
```

```
/*Bus.cpp*/
```

```
#include "Bus.hpp"
```

```
#include <iostream>
```

```
#include <ostream>
```

```
Bus::Bus() : Kendaraan() {  
    this->kapasitas = 0;  
}
```

```
Bus::Bus(int nomorK, std::string merk, int tk, int cap) :  
Kendaraan(nomorK, "Bus", merk, tk){  
    this->kapasitas = cap;  
}
```

```
Bus::Bus(const Bus& K) : Kendaraan(K){  
    this->kapasitas = K.kapasitas;  
}
```

```
Bus::~~Bus(){  
  
}
```

```
int Bus::biayaSewa(int lamaSewa){  
    return 1000000 * lamaSewa;  
}
```

```
void Bus::printInfo(){  
    this->Kendaraan::printInfo();  
    std::cout << "Kapasitas : " << this->kapasitas << std::endl;  
}
```


class Minibus

Kelas ini merupakan kelas turunan dari kelas Kendaraan. Kelas ini memiliki satu metode tambahan, yaitu metode `diskon` yang *me-return value* bertipe *float*. Berikut merupakan *source code* dari kelas Minibus:

```
/*Minibus.hpp*/
#ifndef __MINIBUS_H__
#define __MINIBUS_H__

#include "Kendaraan.hpp"

class Minibus : public Kendaraan {
public:
    Minibus();
    Minibus(int nomorK, std::string merk, int tk );
    Minibus(const Minibus& K);
    ~Minibus();
    int biayaSewa(int lamaSewa) override;
    float diskon(int lamaSewa);
};

#endif
```

```
/*Minibus.cpp*/
#include "Minibus.hpp"
#include <iostream>

Minibus::Minibus() : Kendaraan() {}

Minibus::Minibus(int nomorK, std::string merk, int tk)
    : Kendaraan(nomorK, "Minibus", merk, tk) {}

Minibus::Minibus(const Minibus &K) : Kendaraan(K) {}
Minibus::~Minibus() {}

int Minibus::biayaSewa(int lamaSewa) {
    if (lamaSewa <= 5) {
        return 5000000;
    }
}
```

```

    } else {
        return (5000000 + 500000 * (lamaSewa - 5)) * (1 -
this->diskon(lamaSewa));
    }
}

float Minibus::diskon(int lamaSewa) {
    if (lamaSewa > 10) {
        return 0.1;
    } else {
        return 0.0;
    }
}

```

class Mobil

Kelas ini merupakan kelas turunan dari kelas Kendaraan. Kelas ini memiliki satu atribut tambahan, yaitu atribut supir yang bertipe `std::string`. Berikut merupakan source code dari kelas Mobil:

```

/*Mobil.hpp*/
#ifndef __MOBIL_H__
#define __MOBIL_H__

#include "Kendaraan.hpp"
#include <string>

class Mobil : public Kendaraan {
private:
    std::string supir;
public:
    Mobil();
    Mobil(int nomorK, std::string merk, int tk, std::string supir);
    Mobil(const Mobil& K);
    ~Mobil();
    int biayaSewa(int lamaSewa) override;
    void printInfo() override;
};

```

```
#endif
```

```
/*Mobil.cpp*/
#include "Mobil.hpp"
#include <iostream>
#include <ostream>

Mobil::Mobil() : Kendaraan() { this->supir = "XXXX"; }

Mobil::Mobil(int nomorK, std::string merk, int tk,
             std::string supir)
    : Kendaraan(nomorK, "Mobil", merk, tk) {
    this->supir = supir;
}

Mobil::Mobil(const Mobil& K) : Kendaraan(K){
    this->supir = K.supir;
}

Mobil::~Mobil(){

}

int Mobil::biayaSewa(int lamaSewa){
    return 500000 * lamaSewa;
}

void Mobil::printInfo(){
    this->Kendaraan::printInfo();
    std::cout << "Supir          : " << this->supir << std::endl;
}
```

Link Source Code

Source code dari program dapat di-download dari link GitHub:

<https://github.com/malifpy/OOP-Kendaraan>