

## PRAKTIKKUM IV INHERITANCE (PEWARISAN)

- **Inheritance ( Pewarisan / Penurunan)**

Merupakan salah satu konsep dalam OOP dimana kemampuan suatu objek untuk mewarisi cirri-ciri objek lainnya.

1. Kelas yang mewarisi sifat-sifat dari kelas lainya disebut dengan *kelas turunan (sub class)*.
2. Kelas yang mewariskan sifat-sifat kepada kelas lainya disebut dengan *kelas induk ( super class)*.

Penentu Hak Akses	Kelas Induk	Kelas Turunan
Private	Private	Tidak diwariskan
	Protected	Private
	Public	Private
Protected	Private	Tidak diwariskan
	Protected	Protected
	Public	Protected
Public	Private	Tidak diwariskan
	Protected	Protected
	Public	Public

Catatan : Konstruktor dan destruktur tidak ikut diwariskan.

### 1. Listing Program

```
#include <iostream.h>
#include <conio.h>

class Basis
{
    int alpha;
    int bravo;
public:
    void info_basis()
    { cout<<"Fungsi info_basis() dijalankan." << endl;
    }
};
```

```

class Turunan : public Basis
{
    public:
    void info_Turunan()
    { cout << " Fungsi info_Turunan() dijalankan."<< endl;
      }
};

void main()
{
    Turunan a;
    a.info_Basis();
    a.info_Turunan();
}

```

## 2. Listing Program

```

#include <iostream.h>
#include <conio.h>

class Basis
{
    protected:
    int alpha;
    int bravo;
    public:
    void info_basis()
    { cout<<"Fungsi info_basis() dijalankan." << endl;
      }
};

class Turunan : private Basis
{
    public:
    void inisialisasi( int a, int b)
    {
        alpha=a;
        bravo=b;
    }
    void info_Turunan()
    { cout << " Fungsi info_Turunan() dijalankan."<< endl;

```

```

        cout << " alpha= " << alpha << endl;
        cout << " bravo=" << bravo << endl;
    }
};

void main()
{
    Turunan a;
    a.inisialisasi(2, 10);
    a.info_Turunan();
}

```

- **Konstruktor dan destruktur pada pewarisan.**

**Proses saat obyek diciptakan :**

1. Konstruktor pada kelas induk dijalankan.
2. Konstruktor pada kelas turunan dijalankan.

**Proses pada saat obyek dihapuskan :**

1. Destruktor pada kelas turunan dijalankan.
2. Destruktor pada kelas induk dijalankan

### 3. Listing Program

```

#include <iostream.h>
#include <conio.h>
#include <string.h>
class Kendaraan
{
private:
    char nama[15];
public:
    Kendaraan(char *nama_kendaraan="XXX")
    {
        strcpy(nama, nama_kendaraan);
        cout<<"Hidupkan mesin kendaraan anda..."<<endl;
    }
    ~Kendaraan()
    {
        cout<<"Matikan mesin kendaraan anda..."<< endl;
    }
}

```

```

        void info_kendaraan()
        {
            cout << nama << "Sedang berjalan" << endl;
        }
};
class Truk : public Kendaraan
{
    public:
    Truk(char *nama_truk) : Kendaraan(nama_truk)
    {
        cout << "Hidupkan mesin truk ...." <<endl;
    }
    ~Truk()
    {
        cout << "Matikan mesin truk anda.."<<endl;
    }
};
void main()
{
    clrscr();
    Truk Fuso("TRUK FUSO");

    fuso.info_kendaraan();
    cout << "Akhir main().." << endl;
}

```

#### **4. Listing Program**

```

#include <iostream.h>
#include <conio.h>
#include <string.h>
class Kendaraan
{
protected:
    char nama[15];
public:
    Kendaraan(char *nama_kendaraan="XXX")
    {
        strcpy(nama, nama_kendaraan);
        cout<<"Hidupkan mesin kendaraan anda..."<<endl;
    }
    ~Kendaraan()
    {
        cout<<"Matikan mesin kendaraan anda..."<< endl;
    }
}

```

```

    }
    void info_kendaraan()
    {
        cout << nama << "Sedang berjalan" << endl;
    }
};
class Truk : public Kendaraan
{
    public:
    Truk(char *nama_truk)
    {
        strcpy(nama,nama_truk);
        cout << "Hidupkan mesin truk ...." <<endl;
    }
    ~Truk()
    {
        cout << "Matikan mesin truk anda.."<<endl;
    }
};
void main()
{
    clrscr();
    Truk Fuso("TRUK FUSO");

    fuso.info_kendaraan();
    cout << "Akhir main().." << endl;
}

```

## **Tugas**

1. Jelaskan istilah berikut ini :
  - a. Protected
  - b. Public
  - c. Private
  - d. Super class
  - e. Sub class

