IPB University
Bogor Indonesia

KOM120C -- BAHASA PEMROGRAMAN

Inheritance

- Inheritanca (Pewarisan)
- Overidding

Tim Pengajar Bahasa Pemrograman IPB University

Class Person

Implementasi dari objek orang.
Class Person ingin digunakan untuk implementasi objek Mahasiswa

```
class Person {
   private:
      string name;            // nama
      string address;         // alamat
      int age;                // usia
   public:
      Person() { name=address=""; age=0; }
      Person(string nm, String ad, int th) { name=nm; address=ad; age=th; }
      void setName(string nm) { name=nm; }
      void setAddress(string ad) { address=ad; }
      void setAge(int th) { age=th; }
      string getName() { return name; }
      string getAddress() { return address; }
      int getAge() { return age; }
      void print() { cout << name << ", " << address << age << endl; }
};
```

Kebutuhan Baru

Class Person ingin digunakan untuk implementasi objek Mahasiswa

Beberapa problem:

- Mahasiswa merupakan “person” yang memiliki atribut tambahan berupa (contoh) nim dan daftar MK yang telah diambil
- Mahasiswa dapat meng-update MK yang diambil.

Class Person harus di-EXTEND → buat class Student yang mewarisi sifat-sifat class Person → pewarisan/inheritance.

Class Student sebagai TURUNAN dari class Person → DERIVED CLASS.

Class Person sebagai INDUK dari class Student → BASE CLASS.

Class Person

Beberapa sifat harus dapat DIWARISKAN → protected and public member class

```
class Person {
   protected:          // dapat diakses oleh class ini beserta turunannya
      string name;
      string address;
      int age;
   public:
      Person() { name=address=""; age=0; }
      Person(string nm, String ad, int th) { name=nm; address=ad; age=th; }
      void setName(string nm) { name=nm; }
      void setAddress(string ad) { address=ad; }
      void setAge(int th) { age=th; }
      string getName() { return name; }
      string getAddress() { return address; }
      int getAge() { return age; }
      void print() { cout << name << ", " << address << age << endl; }
};
```

Class Student

Beberapa atribut dan prosedur mengambil dari sifat induknya (Person)

```
class MK {                          // implementasi dari struct MK
   public:
      string kodemk;
      int sks;
};
```

```
class Student : public Person {
   private:
      string nim;
      vector<MK> krs;
   public:
      Mahasiswa() { nim=""; }
      setMahasiswa(string nm, string ad, int th, string n)
         { name=nm; address=ad; age=th; nim=n; }
      void setKRS() { .... }
      .... dst.
};
```

Kaidah Pewarisan

Pada prinsipnya, setiap anggota base class diturunkan ke derived class. Hanya tingkatan aksesnya yang berbeda, tergantung jenis penurunannya.

Perkecualian: anggota private (tidak diturunkan), default constructor (otomatis).

Jenis Penurunan

Kontrol akses Untuk anggota	private	protected	public
private	-	-	-
protected	private	protected	protected
public	private	protected	public

Mengapa Pewarisan ?

Pewarisan adalah mekanisme untuk:

- Mengembangkan class baru dari class yang sudah ada
- Mendefinisikan class baru sebagai spesialisasi dari class yang sudah ada.

Harus memahami istilah base class dan derived class !

Berikan contoh ...

Mendefinisikan Pewarisan

Sintaks:

```
class <DerivedClass> : <access-level> <BaseClass>
```

<access-level> menunjukkan tipe pewarisan

- private (by default)
- public

Setiap class dapat berfungsi sebagai Base Class, sehingga Derived Class dapat dibuat sebagai Base Class. Memungkinkan turunan dari class turunan.

Constructor untuk Turunan

Default constructor dan destructor dari base class selalu dipanggil ketika suatu objek baru dari class turunannya dibuat atau di-destroy.

```
class A {
  public:
  A() {
   cout<<"A:default\n"; }
  A(int a) {
   cout<<"A:parameter\n"; }
};
```

```
class B : public A {
  public:
  B(int a) {
    cout<<"B\n"; }
};
```

B x(1);

output:

```
A:default
B
```

Constructor untuk Turunan

Kita dapat menentukan constructor dari base class yang digunakan untuk class turunannya.

```
class A {
  public:
  A() {
   cout<<"A:default\n"; }
  A(int a) {
   cout<<"A:parameter\n"; }
};
```

```
class C : public A {
  public:
  C (int a) : A(a) {
    cout<<"C"<<endl; }
};
```

C x(1);

output:

```
A:parameter
C
```

Overriding

Class turunan dapat meng-override fungsi yang telah didefinisikan oleh base class.

Dengan overriding,

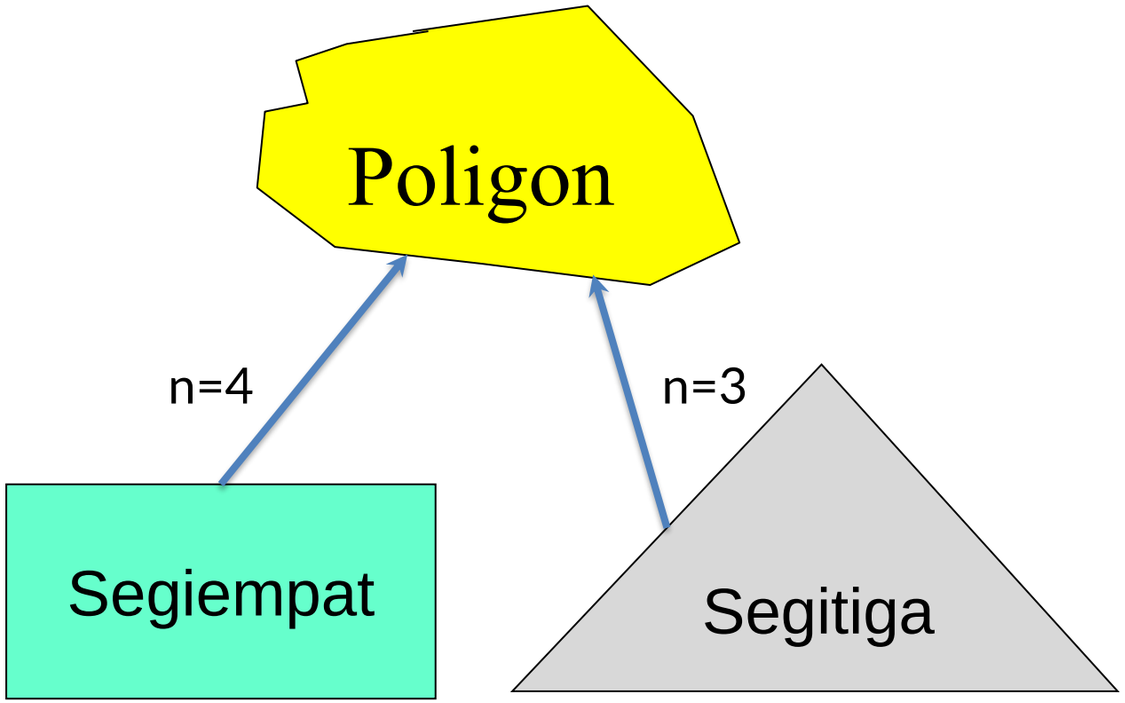
- Fungsi atau method dalam class turunan memiliki signature yang identik dengan method pada base class.
- Derive class mengimplementasikan method versinya sendiri.

```
class A {
  protected:
    int x, y;
  public:
    void print() {
      cout<<"A"<<endl; }
};
```

```
class B : public A {
  public:
    void print() {
      cout<<"B"<<endl; }
};
```

Contoh Pewarisan #1 :: Spesialisasi

Implementasi dari objek Poligon, Segiempat, dan Segitiga



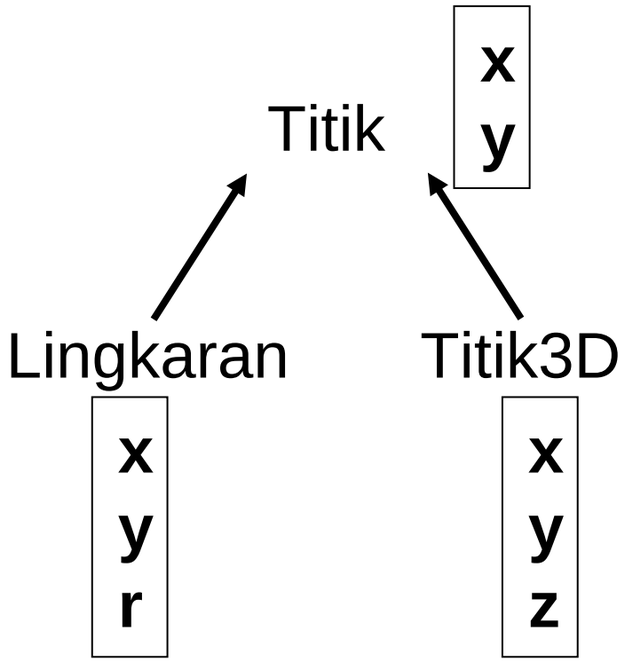

```
class Poligon {
   protected:
      int n;
      vector <double> x, y;
   public:
      void set(...);
};
```

```
class Segiempat:public Poligon {
   public:
     double luas();
};
```

```
class Segitiga:public Poligon {
   public:
     double luas();
};
```

Contoh Pewarisan #2 :: Extensification

Implementasi dari objek Titik 2D, Titik 3D, dan Lingkaran



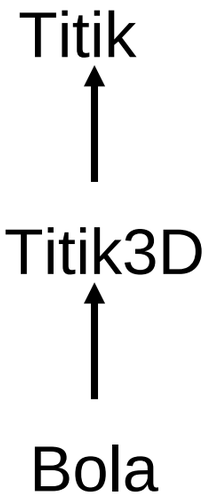

```
class Titik {
protected:
   int x, y;
public:
   void set (int a, int b);
};
```

```
class Lingkaran :
public Titik {
    double r;
};
```

```
class Titik3D :
public Titik {
    int z;
};
```

Contoh Pewarisan #3 :: Turunan dari Turunan

Titik adalah baseclass dari Titik3D, sedangkan Titik3D adalah baseclass dari Bola




```
class Titik {
  protected:
    int x, y;
  public:
    void set (int a, int b);
};
```

```
class Titik3D : public Titik {
   private:
      double z;
   ...
};
```

```
class Bola : public Titik3D {
  private:
     double r;
  ...
};
```

Latihan

Gunakan konsep OOP

Deskripsi

Praktikum-06-A-PewarisanPegawai.pdf

Format Masukan

[N : banyaknya pegawai, 1<N<100]
[N baris data pegawai : id, usia, tipe. Jika pegawai tetap, ada nilai gaji pokok.]
[T baris data gaji : id dan upah (pegawai harian) | uang lembur (pegawai tetap).]
END

Format Keluaran

Beberapa baris data id, tipe, dan penghasilan sebulan. Kelompok pegawai tetap di bagian atas, disambung dengan kelompok pegawai harian. Urutan data sesuai dengan urutan data masuk.

Contoh Input

```
5
123456 19 1 5000
989212 21 1 6000
876523 20 2
092831 20 2
187632 19 1 5000
123456 2000
876523 1000
092831 5000
187632 4000
END
```

Contoh Output

```
123456 1 7000
989212 1 6000
187632 1 9000
876523 2 1000
092831 2 5000
```