

TIF1202 – Pemrograman Berorientasi Objek

HO 05 - Inheritance

Opim Salim Sitompul

Department of Information Technology
Universitas Sumatera Utara



Outline

- 1 Tujuan
- 2 Pendahuluan
- 3 Pengertian Inheritance
- 4 Keuntungan Inheritance
- 5 Menggunakan Fungsi Constructor pada Inheritance
- 6 Menurunkan lebih dari satu kelas
- 7 Multiple Inheritance



Tujuan

- Dalam kuliah ini mahasiswa diharapkan:
 - Memahami inheritance dan mengapa penggunaannya menguntungkan
 - Memahami bagaimana menggunakan fungsi constructor bersama inheritance
 - Mengetahui ilmu dan seni OOP
 - Mengetahui konsep multiple inheritance
 - Memahami anggota kelas protected



Pendahuluan

- Kelas sering sekali memiliki karakteristik yang mirip.
- Dengan memanfaatkan hubungan pada objek-objek, jumlah kode program yang harus ditulis akan sangat berkurang.
 - Waktu pemrograman menjadi lebih singkat
 - Meningkatkan keterbacaan program
 - Mengurangi potensi kesalahan
- Membangun kelas berdasarkan hubungan antar objek dilakukan dengan memanfaatkan sifat inheritance – kelas yang diturunkan dapat mewarisi sifat-sifat dari kelas dasarnya.



Pendahuluan

- Mulai dari sebuah kelas sederhana (disebut *base class*), kemudian bangun kelas yang lebih kompleks.
- Contoh:
 - *Base Kelas*: mesin
 - *Kelas Turunan*: mesin boat, mesin mobil, mesin sepeda motor.



Pengertian Inheritance

- Sering sekali dua atau lebih kelas berbagi karakteristik (anggota) yang sama.
- Pada saat mendefinisikan kelas, upayakan seumum mungkin
 - Definisikan sebuah kelas dasar yang memuat fungsi-fungsi anggota yang sama.
 - Bangun kelas berikutnya dari kelas dasar tersebut.
 - Anggota fungsi dan variabel yang ada dalam kelas level pertama (*base class*) hendaklah berlaku pada tiap-tiap kelas yang diturunkan (*derived class*).



Keuntungan Inheritance

- Jika anggota-anggota kelas yang sama ditulis di dalam setiap kelas, akan banyak sekali kode program sehingga mudah terjadi error dan sulit di-debug.
- Kelas turunan akan mewarisi karakteristik kelas dasar.
- Contoh Program 5.1:



Keuntungan Inheritance

```

1 //Contoh5_1.cpp
2 #include <iostream>
3 using namespace std;
4
5 class Hitung
6 {
7     protected:
8         int cacah;
9     public:
10         Hitung() { cacah = 0; }
11         Hitung(int c) { cacah = c; }
12         int Inkremen() { return cacah++; }
13         int Dekremen() { return cacah--; }
14         int getHitungan() { return cacah; }
15         ~Hitung() { }
16 };

```



Keuntungan Inheritance

```
17 class HitungMaju : public Hitung
18 {
19     public:
20         int Inkremen() { return cacah++; }
21 };
22
23 class HitungMundur : public Hitung
24 {
25     public:
26         int Dekremen() { return cacah--; }
27 };
```



Keuntungan Inheritance

```
28 int main()  
29 {  
30     HitungMaju c1;  
31     HitungMundur c2;  
32     Hitung c3(10);  
33  
34     cout << "c1_=_ " << c1.getHitungan() << endl;  
35     c1.Inkremen(); c1.Inkremen(); c1.Inkremen();  
36     cout << "c1_=_ " << c1.getHitungan() << endl;  
37  
38     cout << "c2_=_ " << c2.getHitungan() << endl;  
39     c2.Dekremen(); c2.Dekremen();  
40     cout << "c2_=_ " << c2.getHitungan() << endl;
```



Keuntungan Inheritance

```
41     cout << "c3_=_ " << c3.getHitungan() << endl;
42     c3.Inkremen();
43     cout << "c3_=_ " << c3.getHitungan() << endl;
44
45     return 0;
46 }
```



Keuntungan Inheritance

- Kelas dasar:
 - class Hitung
- Kelas turunan:
 - Titik dua di sebelah kanan nama kelas menyatakan nama base class.
 - class HitungMaju : public Hitung
 - class HitungMundur : public Hitung
- Agar dapat diakses oleh kelas turunan, member anggota pada kelas dasar hendaklah bersifat protected.



Keuntungan Inheritance

- Sebuah anggota yang dideklarasikan public dapat diakses di keseluruhan program (apabila objek berada dalam ruang lingkup)
- Sebuah anggota yang dideklarasikan private hanya dapat diakses menggunakan fungsi anggota kelas yang sama.
- Sebuah anggota yang dideklarasikan protected dapat diakses secara langsung oleh kelas turunan, tetapi tidak dapat diakses oleh yang lain.



Menggunakan Fungsi Constructor pada Inheritance

- Constructor masih dapat digunakan untuk menginisialisasi class inheritance.
- Perbedaan utamanya adalah constructor kelas turunan harus memanggil constructor kelas dasar.
- Contoh Program 5.2:



A set of small navigation icons typically found in Beamer presentations, including symbols for back, forward, search, and other slide controls.

Menggunakan Fungsi Constructor pada Inheritance

```
16 class Lingkaran : public BangunGeometri
17 {
18     private:
19         float luas, keliling;
20     public:
21         Lingkaran(int r) : BangunGeometri(r)
22             {luas = 0.0; keliling = 0.0;}
23         void getJari2();
24         void hitungLuas();
25         void hitungKeliling();
26         void displayHasil();
27 };
```



Menggunakan Fungsi Constructor pada Inheritance

```
28 void Lingkaran::displayHasil()
29 {
30     cout << "Lingkaran_berjari-jari_" << besaran
31         << endl;
32     cout << "Luas_=_ " << luas << endl;
33     cout << "Keliling_=_ " << keliling << endl;
34 }
35
36 void Lingkaran::getJari2()
37 {
38     cout << "Berikan_jari-jari_lingkaran:_ ";
39     cin >> besaran;
40 }
```



Menggunakan Fungsi Constructor pada Inheritance

```
41 void Lingkaran::hitungLuas()  
42 {  
43     luas = PI * besaran * besaran;  
44 }  
45  
46 void Lingkaran::hitungKeliling()  
47 {  
48     keliling = 2 * PI * besaran;  
49 }
```



Menggunakan Fungsi Constructor pada Inheritance

```
50 int main()  
51 {  
52     Lingkaran r1(5);  
53     r1.hitungLuas();  
54     r1.hitungKeliling();  
55     r1.displayHasil();  
56  
57     return 0;  
58 }
```



Menurunkan lebih dari satu kelas

- Dari satu buah base class dapat diturunkan beberapa kelas
- Masing-masing contoh kelas turunan akan mewarisi sifat-sifat kelas dasar. Selain itu dapat memiliki sifat-sifat tersendiri yang membedakannya dengan kelas turunan yang lain.
- Contoh 5.3:



Menurunkan lebih dari satu kelas

```

1 //Contoh5_3.cpp
2 #include <iostream>
3 #include <cstring>
4 using namespace std;
5
6 class kendaraan
7 {
8     protected:
9         char nama[64];
10        char bbakar[64];
11        int roda;
12    public:
13        kendaraan(char *, char *, int);
14        void show_kendaraan();
15        ~kendaraan() { }
16 };

```



Menurunkan lebih dari satu kelas

```

17 kendaraan::kendaraan(char *nama, char *bbakar,
    int roda)
18 {
19     strcpy(kendaraan::nama, nama);
20     strcpy(kendaraan::bbakar, bbakar);
21     kendaraan::roda = roda;
22 }
23
24 void kendaraan::show_kendaraan()
25 {
26     cout << "Nama:_" << nama << endl;
27     cout << "Bahan_bakar:_" << bbakar << endl;
28     cout << "Roda:_" << roda << endl;
29 }
    
```



Menurunkan lebih dari satu kelas

```
30 class sepedamotor : public kendaraan
31 {
32     private:
33         int mesin;
34     public:
35         sepedamotor(char *nama, char *bbakar, int
            roda,
36                     int mesin);
37         void show_sepedamotor();
38 };
```



Menurunkan lebih dari satu kelas

```

39  sepedamotor::sepedamotor(char *nama, char *
    bbakar, int roda, int mesin):kenderaan(nama,
    bbakar, roda)
40  {
41      sepedamotor::mesin = mesin;
42  }
43
44  void sepedamotor::show_sepedamotor()
45  {
46      cout << "Nama:_" << nama << endl;
47      cout << "Mesin:_" << mesin << "_Takt" <<
        endl;
48  }
    
```



Menurunkan lebih dari satu kelas

```
49 class mobil : public kendaraan
50 {
51     private:
52         int pintu;
53     public:
54         mobil(char *nama, char *bbakar, int roda, int
                    pintu);
55         void show_mobil();
56 };
57
58 mobil::mobil(char *nama, char *bbakar, int roda,
                int pintu:kendaraan(nama,bbakar,roda)
59 {
60     mobil::pintu = pintu;
61 }
```



Menurunkan lebih dari satu kelas

```

62 void mobil::show_mobil()
63 {
64     cout << "Nama:_" << nama << endl;
65     cout << "Bahan_bakar:_" << bbakar << endl;
66     cout << "Pintu:_" << pintu << endl;
67 }
68
69 int main()
70 {
71     sepedaomotor honda("CBR250RR","bensin",4,2),
        yamaha("NMax", "bensin", 4, 2);
72     mobil toyota("All_New_Rush","bensin", 4, 4);
73     mobil mitsubishi("Pajero_Sport", "solar", 4,
        4);

```



Menurunkan lebih dari satu kelas

```
74     honda.show_sepedamotor();  
75     yamaha.show_sepedamotor();  
76  
77     toyota.show_mobil();  
78     mitsubishi.show_mobil();  
79  
80     return 0;  
81 }
```



Multiple Inheritance

- Menggunakan karakteristik dari dua atau lebih kelas dasar untuk menurunkan kelas baru.
- Contoh:
 - class bundel : public buku, public disk
 - Kelas bundel mewarisi karakteristik dari dua buah kelas dasar: class buku dan class disk
- Salin dan jalankanlah Contoh Program 5.4 berikut:



Multiple Inheritance

```
1 //Contoh5_4.cpp
2 #include <iostream>
3 #include <string>
4 #include <iomanip>
5 using namespace std;
6
7 class buku
8 {
9     private:
10         char judul[64];
11         char pengarang[64];
12         int hlm;
```



Multiple Inheritance

```
13     public:
14         buku(char *judul, char *pengarang, int hlm);
15         void show_buku();
16         ~buku() {}
17 };
18
19 class disk
20 {
21     private:
22         float kapasitas;
23     public:
24         disk(float kapasitas);
25         void show_disk();
26         ~disk() {}
27 };
```



Multiple Inheritance

```

28  class bundel : public buku, public disk
29  {
30      private:
31          float harga;
32      public:
33          bundel(char *judul, char *pengarang, int hlm
              , float kapasitas, float harga);
34          void show_bundel();
35  };
36
37  buku::buku(char *judul, char *pengarang, int
              hlm)
38  {
39      strcpy(this->judul, judul);
40      strcpy(this->pengarang, pengarang);
41      this->hlm = hlm;
42  }

```



Multiple Inheritance

```
43 void buku::show_buku()
44 {
45     cout << "Judul:_ " << judul << endl;
46     cout << "Pengarang:_ " << pengarang << endl;
47     cout << "Halaman:_ " << hlm << endl;
48 }
49
50 disk::disk(float kapasitas)
51 {
52     this->kapasitas = kapasitas;
53 }
```



Multiple Inheritance

```

65 void bundel::show_bundel()
66 {
67     show_buku();
68     show_disk();
69     cout << setiosflags(ios::fixed)
70         << setprecision(2);
71     cout << "Harga:_Rp" << harga << endl;
72 }
73
74 int main()
75 {
76     bundel bukuku("Pemrograman_C_di_Lingkungan_
        Linux", "Opim_Salim_Sitompul", 320, 700,
        60000);
77     bukuku.show_bundel();
78     return 0;
79 }

```

