

Praktikum 10

Pemrograman Berbasis Obyek (PBO)

(Abstraction)

A. Tujuan

1. Mahasiswa menerapkan konsep Abstraction (Abstraksi)

B. Dasar Teori

Konsep Abstraksi Yaitu menyembunyikan kerumitan dari suatu proses. Biasa berbentuk kelas murni yang tidak boleh memiliki objek, dan satu/lebih method-methodnya yang abstract harus diimplementasikan (override) oleh kelas turunannya. Kelas abstrak ini tidak dapat diinstansiasi.

Seperti namanya, abstract class adalah class-class yang memiliki informasi abstrak dan metode-metode dari sekumpulan data. Abstract Class tidak bisa diubah dan berlaku juga sebagai kerangka dalam penciptaan subclass-subclassnya (berperan seperti Superclass yang dibahas pada konsep Inheritance). Sebuah Abstract Classs memiliki informasi dan metode yang dapat diturunkan ke subclassnya, dan seluruh subclass akan mengikuti apa saja metode yang akan diturunkan oleh Abstract Class.

Dalam bahasa pemrograman, Abstraction merupakan salah satu konsep penting yang digunakan oleh programmer untuk mempercantik dan mempermudah penggunaan kode di masa mendatang. Membuat sebuah ringkasan dari keseluruhan kode, yang berisi atau hanya menyajikan informasi penting dan menyembunyikan detailnya, yang dimaksud menyembunyikan detail di sini adalah menyembunyikan latar belakang, implementasi atau mekanisme kerja dari informasi yang disajikan. Mirip seperti encapsulation tapi berbeda dalam tujuannya, dimana encapsulation lebih berfokus kepada pembungkusan kode, sedangkan abstraction berfokus pada penyajian kode.

Contoh Abstraksi di dunia nyata :

Sebagai contoh, mobil, sepeda motor, becak, merupakan kendaraan. Kendaraan (Abstract Class) memiliki syarat-syarat di mana suatu objek dapat dikatakan kendaraan (method and information). Bentuk kendaraan seperti mobil, motor, dsb. adalah hasil penyempitan dari kendaraan dan bersifat lebih spesifik (Subclass).

Seseorang menggunakan smartphone dimana disajikan tombol-tombol untuk mengendalikan smartphone tersebut, seperti tombol untuk “menyalakan”, “mematikan”, “menaikkan dan menurunkan suara”, yang diketahui oleh pengguna tsb hanya tau sebatas pengertian itu. Orang tersebut tidak mengetahui mekanisme kerja pada tombol-tombol tersebut hingga smartphone tersebut benar-benar melakukan tugasnya.

```
#include <iostream>
using namespace std;

int main(){
    cout<<"Halo semua";

    return 0;
}
```

Kita menggunakan cout untuk mencetak “Halo semua”. Jika kita bisa membongkar bahasa pemrograman c++ dan mencari tau latar belakang cout, kita akan tau bahwa sebenarnya cout memiliki mekanisme kerja yang cukup kompleks untuk membuatnya berfungsi mencetak sebuah kalimat pada layer (hasil compile).

Jenis- Jenis Abstraksi

1. Abstraksi menggunakan Class

```
1 #include <iostream>
2 using namespace std ;
3 class abstractionClass // deklarasi kelas
4 {
5     private :
6     int x , y ; // variabel x & y pakai private
7     public :
8     // method yang mendeklarasikan seluruh member private
9     void set_nilai ( int a , int b ) // deklarasi fungsi publik
10 {
11     x = a ;
12     y = b ;
13 }
14 void display () // deklarasi untuk display
15 {
16     cout << " x = " << x << endl ;
17     cout << " y = " << y << endl ;
18 }
19 } ;
20 int main ()
21 {
22     abstractionClass fungsi ; // membuat suatu objek dari deklarasi klas
23     fungsi.set_nilai ( 110 , 250 ) ;
24     fungsi.display () ;
25     return 0 ;
26 }
```

Hasil :

```
E:\GDRIVE\UNAIR\Mata Kuliah\PBO Praktikum\abstraksi\contoh.exe
x  =  110
y  =  250

-----
Process exited after 0.01616 seconds with return value 0
Press any key to continue . . .
```

2. Abstraksi menggunakan Header Files

```
1 #include <iostream>
2 #include <math.h> //memasukkan header math
3 using namespace std ;
4 int main ()
5 {
6     int x = 7 ;
7     int power = 2 ;
8     int result = pow ( x , power ) ; // pow(nilai,pangkat)
9     std :: cout << " The square of x is : " << result << std :: endl ;
10    return 0 ;
11 }
```

```
E:\GDRIVE\UNAIR\Mata Kuliah\PBO Praktikum\abstraksi\abstraksi_header.exe
The square of x is : 49

-----
Process exited after 0.01935 seconds with return value 0
Press any key to continue . . .
```

3. Abstraksi menggunakan kombinasi specifier

abstraksi_specifier.cpp

```
1  #include <iostream>
2  using namespace std ;
3  class Addition { // deklarasi klas untuk program
4  public :
5  // konstruktor
6  Addition ( int i = 0 ) { // konstruktor untuk kelas dengan nama yg sama
7  total = i ;
8  }
9
10 // interface to outside world
11 void numAdd ( int number ) { // fungsi tambah nilai
12 total += number ;
13 }
14
15 // interface to outside world
16 int getTotal () {
17 return total ;
18 } ;
19
20 private :
21 // hidden data from outside world
22 int total ;
23
24 } ;
25
26 int main () {
27 Addition ad ;
28 ad.numAdd ( 10 ) ;
29 ad.numAdd ( 10 ) ;
30 ad.numAdd ( 10 ) ;
31 cout << " maka jumlah seluruh nilai :| " << ad.getTotal() << endl ;
32 return 0 ;
33 }
```

Hasil :

```
E:\GDRIVE\UNAIR\Mata Kuliah\PBO Praktikum\abstraksi\abstraksi_specifier.exe
maka jumlah seluruh nilai : 30

-----
Process exited after 0.03944 seconds with return value 0
Press any key to continue . . .
```

C. Percobaan

Buatlah program menggunakan prinsip abstraksi

1. buatlah suatu program menggunakan prinsip abstraksi menggunakan header file dan class biasa untuk menghitung periode bandul dengan rumus berikut

$$T = 2\pi \left[\frac{Ke^2 + a^2}{g + a} \right]^{1/2}$$

T = periode ayunan
 Ke = jari-jari girasi terhadap pusat massa (C)
 a = jarak pusat massa
 g = percepatan gravitasi bumi

2. Buatlah suatu program menggunakan prinsip abstraksi menggunakan specifier untuk sistem parkir motor dengan biaya 500 rupiah per jam. Buatlah susunan program agar bisa digunakan untuk menghitung biaya parkir untuk tiap motor yang masuk beserta plat nomor, dan sediakan program untuk menghitung jumlah pemasukkan dari total pendapatan parkir.