Konsep OOP di Java

Mengenal Object, Class, Method, dan Property

Burung itu ibarat objek yang punya nama, mata, kaki, sayap, warna, jenis. Burung juga dapat terbang, berkicau dan berjalan. Mata, kaki sayap dan warna dalam dunia pemrograman disebut juga atribut atau properties. Sedangkan aktifitasnya yaitu terbang, berkicau dan berjalan dalam dunia pemrograman disebut sebagai method.

Class adalah cetakan, template, prototype,tempat dari object, sedangkan object adalah isi dari kelas. Satu kelas dapat mempunyai object lebih dari satu atau banyak. Contoh sederhananya seperti berikut : satu cetakan bias menghasilkan banyak kue.

Contoh penerapannya di Java :

1. public class Burung {
3. String nama; //atribut nama
5. String jenis; //atribut jenis
7. String warna; //atribut warna
9. String sayap; //atribut sayap
11. //method terbang
13. public void terbang(){
15. this.sayap = “mengepak”;
17. System.out.println(“Burung “+this.nama+” Sedang Terbang”);
19. }
21. //method tidur
23. public void tidur(){
25. this.sayap = “diam”;
27. System.out.println(“Burung “+this.nama+” Sedang Tidur”);
29. }
31. }

Dari contoh diatas bisa diambil pengerian Attribute adalah Berbagai variable yang mengitari class, dengan nilai datanya bisa ditentukan di object. nama, warna, sayap, dan jenis adalah atribute dari class Burung. Method adalah implementasi dari bagaimana aktifitas / bekerjanya sebuah class. terbang dan tidur adalah method atau behaviour dari class Burung

Ada 3 Konsep Dasar OOP pada Java :

A. Encapsulation

Encapsulation adalah pembungkus, pembungkus disini dimaksudkan untuk menjaga suatu proses program agar tidak dapat diakses secara sembarangan atau di intervensi oleh program lain.

Dalam OOP Encapsulation di wujudkan dalam bentuk “class”. Seperti yang telah dijalaskan sebelumnya dalam sebuah class terdapat property dan method yang memiliki hak akses tertentu terhadap environment/lingkungan-nya, hak akses ini biasa di sebut Access Modifier, access modifier terdiri dari private, protected, dan public.

• private

memberikan hak akses hanya kepada anggota class tersebut untuk menggunakan dan/atau mengubah nilai dari property atau method tersebut.

• protected

memberikan hak akses kepada anggota class nya dan anggota class hasil inheritance (penurunan sifat) dari class tersebut.

• public

memberikan hak akses kepada property dan method agar dapat digunakan diluar class tersebut.

Contoh Program Encapsulation:

1. public class Siswa {
3. public String nama;
5. public String nrp;
7. public void Info() {
9. System.out.println(“Saya adalah”);
11. System.out.println(“Nama ” + nama);
13. System.out.println(“Nrp ” + nrp);
15. }
17. }
19. // Setelah itu kita buat New Class lagi dengan nama IsiData
21. public class IsiData {
23. public static void main(String[] args) {
25. Siswa IT = new Siswa();
27. IT.nama = “aji M”;
29. IT.nrp = “0910651015″;
31. IT.Info();
33. }
35. }

B. Inheritance

inheritance adalah pewarisan yang artinya sebuah class dapat mewarisi atribut dan metho dari class lain. untuk mendeklarasikan suatu class sebagai subclass dilakukan dengan cara menambahkan kata kunci extends setelah deklarasi nama class, kemudian diikuti dengan nama parent class-nya. Kata kunci extends tersebut memberitahu kompiler Java bahwa kita ingin melakukan perluasan class.

Contoh Programnya :

2. public class Mamalia () {
4. System.out.println(“Ciri umum mamalia adalah bernapas dengan paru-paru, ” +
6. “berkembang biak dengan cara beranak”);
8. }
10. class () Paus extends Mamalia () {
12. String x = “Ikan Paus”;
14. System.out.println(“Mamalia yang Anda pilih adalah “+ x );
16. System.out.println(“Mamalia ini berada di laut”);
18. System.out.println(“Paus termasuk jenis mamalia terbesar”);
20. }
22. class Sapi () extends Mamalia (){
24. String x = “Sapi”;
26. System.out.println(“Mamalia yang Anda pilih adalah “+ x );
28. System.out.println(“Mamalia ini berada di darat”);
30. System.out.println(“Sapi termasuk jenis mamalia pemamah biak”);
32. }

C. Polymorism

suatu aksi yang memungkinkan pemrogram menyampaikan pesan tertentu keluar dari hirarki obyeknya, dimana obyek yang berbeda memberikan tanggapan/respon terhadap pesan yang sama sesuai dengan sifat masing-masing obyek.

Polymorism dapat berarti banyak bentuk, maksudnya yaitu kita dapat menimpa (override), suatu method, yang berasal dari parent class (super class) dimana object tersebut diturunkan, sehingga memiliki kelakuan yang berbeda.

Contoh program java nya :

Simpan dengan nama BeautyfullMahasiswa.java

2. public class BeautyfullMahasiswa extends Mahasiswa{
4. public void printNama(){
6. System.out.println(“Hallo, saya Mahasiswa yg baik hati dan tidak sombong”);
8. Super.printNama();
10. }
12. }
14. Simpan dengan nama Main.java
15. <pre>
17. public class Main{
19. public static void main(String[] args]) {
21. BeautyfullMahasiswa mhs = new BeautyfullMahasiswa();
23. mhs.setNim(“0910651015”);
25. mhs.setNama(“aji M”);
27. mhs.printNim();
29. mhs.printNama();
31. }
33. }

Kemudian compile dan run.

C:\>javac BeautyfullMahasiswa.java

C:\>javac Main.java

C:\>java Main

Kontruksi objek mahasiswa

Nim: 0910651015

Hallo, saya Mahasiswa yg baik hati dan tidak sombong.

Nama:  aji M

C:\>

Terlihat bahwa disini kita mengoverride method printNama(), dengan menambahkan beberapa teks tambahan, sehingga method printNama(), pada class BeautyfullMahasiswa, memiliki perilaku yang berbeda dengan class Mahasiswa.

http://ti.unmuhjember.ac.id/rpl/?p=85