**PENGERTIAN ENKAPSULASI**  
  
Enkapsulasi/ Pembungkusan (Encapsulation)  
Enkapsulasi merupakan suatu cara pembungkusan data dan method yang menyusun suatu kelas sehingga kelas dapat dipandang sebagai suatu modul dan cara bagaimana menyembunyikan informasi detail dari suatu class (information hiding). Dalam OOP, enkapsulasi sangat penting untuk keamanan serta menghindari kesalahan permrograman, pembungkus disini dimaksudkan untuk menjaga suatu proses program agar tidak dapat diakses secara sembarangan atau di intervensi oleh program lain. Konsep enkapsulasi sangat penting dilakukan untuk menjaga kebutuhan program agar dapat diakses sewaktu-waktu, sekaligus menjaga program tersebut.  
Dua hal yang mendasar dalam enkapsulasi yakni :  
• Information hiding.  
• Interface to access data.  
  
Information hiding  
Sebelumnya kita dapat mengakses anggota class baik berupa atribut maupun method secara langsung dengan menggunakan objek yang telah kita buat. Hal ini dikarenakan akses kontrol yang diberikan kepada atribut maupun method yang ada di dalam class tersebut adalah 'public'. Kita dapat menyembunyikan informasi dari suatu class sehingga anggota class tersebut tidak dapat diakses dari luar, caranya adalah hanya dengan memberikan akses kontrol 'private' ketika mendeklarasikan atribut atau method. Proses ini disebut dengan information hiding.  
  
Interface to access data   
Jika kita telah melakukan information hiding terhadap suatu atribut pada suatu class, lalu bagaimana cara melakukan perubahan terhadap atribut yang kita sembunyikan tersebut, caranya adalah dengan membuat suatu interface berupa method untuk menginisialisasi atau merubah nilai dari suatu atribut tersebut.  
Manfaat utama teknik encapsulation adalah kita mampu memodifikasi kode tanpa merusak kode yang telah digunakan pada class lain.  
Enkapsulasi memiliki manfaat sebagai berikut:  
> Modularitas  
Source code dari sebuah class dapat dikelola secara independen dari source code class yang lain. Perubahan internal pada sebuah class tidak akan berpengaruh bagi class yang menggunakannya.  
> Information Hiding  
Penyembunyian informasi yang tidak perlu diketahui objek lain.

Contoh Enkapsulasi :

import java.io.\*;  
import java.util.Scanner;  
class makanan  
{  
public int porsi;  
public double harga;  
}  
class nasi extends makanan  
{  
public void setharga()  
{  
harga = 1000;  
}  
}  
class lauk extends makanan  
{  
public void setharga()  
{  
harga = 2000;  
}  
}  
class sayur extends makanan  
{  
public void setharga(double h) //set harga lewat input user  
{  
harga = h;  
}  
public double getharga() //mengambil info harga dr class  
{  
return(harga);  
}  
}  
public class program //utk menjalankan program utama  
{  
public static void main(String[] args)  
{  
double h;  
sayur s = new sayur(); //create object utk masing2 class  
lauk l = new lauk();  
nasi n = new nasi();  
Scanner sc = new Scanner(System.in);  
System.out.print(“Harga : “);  
h = sc.nextDouble();  
s.setharga(h);  
System.out.println(“Harga sayur :”+s.getharga());  
n.harga = 1000; //akses langsung ke atribut  
n.porsi = 2;  
biaya = n.harga \* n.porsi;  
System.out.println(“Biaya :”+biaya);  
//menu program dan manipulasi object yg lain  
//bisa diletakkan di sini  
}  
}  
  
  
Sumber:  
  
<http://bluejundi.wordpress.com/2010/05/11/contoh-program-oop-dengan-inheritance-sederhana-dalam-bahasa-java/>

<http://rennrenn21.blogspot.com/2013/12/enkapsulasi-pada-oop.html>