OOP (**Object Oriented Programming)**adalah Pemrograman berorientasi objek. Pemrograman ini mempertinggi kualitas dan produktifitas pengembangan software. Program pengembangannya dilakukan dengan pendekatan building block.  
OOP atau singkatan dari Object Oriented Programming merupakan cara atau metode baru dalam membuat program, dan dalam teknik membuat program OOP adalah dengan mengetahui terlebih dahulu Objeknya, kemudian atribut(**field**/**data**)dan Perilaku (**Fungsi**).berbeda dengan programan Prodesural(konvensional),membuat program memakai aspek input, proses, dan output.  
oke, kita selanjutnya kita masuk ke Karekteristik **OOP**

***1.Karekteristik OOP***Karakteristik (ciri) suatu program termasuk dalam OOP, apabila terdapat Pembungkusan(Encapsulation)**, Polymorphisme**, dan Turunan (**Inheritance**)  
  
Baik, Mari Kita membahas apa yang saya sebutkan di atas satu persatu  
  
**1.2 Pembungkusan** (Encapsulation)  
  
Pembungkusan(Encapsulaton)merupakan suatu karakteristik OOP dimana program terbungkus menjadi satu) data (properti/field) dan perilaku (fungsi/method) artinya lebih memperhatikan aspek internal daripada aspek eksternal.  
  
Contoh ; Di dalam sebuah program sering kita jumpai tombil *Button* **Close** (Sebagai Data), kemudian kita sisipkan*Method* **System.exit(0)** untuk keluar dari sistem Java.  
  
**1.3 Polymorphisme**  
  
Polymorphisme merupakan suatu karakteristik OO dimana objek yang berbeda dapat memakai method Polymorphisme dengan nama yang sama pada suatu kelas (class)  
  
**1.4 Turunan**(Inheritance)  
  
Turunan Merupakan Suatu Karakteristik OOP dimana suatu kelas (Parent Class/Base Class) yang dapat diturunkan ke kelas yang lain (child class/Derived class), sehingga kelas anak dapat memiliki data atau perilaku kelas orangtuanya.

Di dalam OOP, minimal kita harus kenalan dulu dengan yang namanya.

**Java : Sekilas Tentang Object Oriented Programing(OOP/Pemrograman Berorientasi Objek)**

Java Merupakan salah satu bahasa pemrograman yang berorientasi objek. OOP Atau Object Oriented Programing Adalah merupakan model pemrograman yang berorientasikan kepada objek. Semua data dan fungsi di dalam model pemrograman ini dibungkus dalam objek. Bandingkan dengan logika pemrograman terstruktur. Setiap objek dapat menerima pesan, memproses data, dan mengirim pesan ke objek lainnya. Model pemrograman ini merupakan model pemrograman yang mudah untuk dikembangkan dan di maintenance.  
  
Ada berbagai macam konsep yang dipakai dalam model pemrograman berbasis OOP diantaranya : Kelas, Objek, Abstrak, Polimorfisme, Enkapsulasi Dalam kesempatan ini kita akan pelajari Kelas, Objek serta beberapa macam akses modifier.

**Class, Objek, Methode dan Penerapan Parameter**

**A. Class**

Class didefinisikan sebagai blue print atau prototype yang mendefinisikan variable-variabel  
dan metode-metode yang umum untuk semua objek. Classs berisi atribut dan perilaku obyek  
yang dibuatnya. Contohnya : class manusia, class tumbuhan, dll.  
Bentuk umum , Penamaan suatu class mempunyai aturan umum :

a. Tidak menggunakan spasi atau menggunakan spasi dengan diganti “\_” (misal class Sarjana\_Mipa)  
b. Tidak menggunakan kata-kata yang telah dipakai oleh compiler (reserved Word, lihat bab sebelumnya)

Modifier saat deklarasi kelas adalah :  
• Tidak ditentukan (default) Kelas tersebut dapat diakses oleh kelas lain dalam satu package.  
• Public  
¬ Main Class/ Class Induk  
Modifier1 Class Namaclass modifier2 {  
//tipedata namaatribut/variable  
Public static void main (String[ ] args){  
//character yang akan di tampilkan  
}  
}

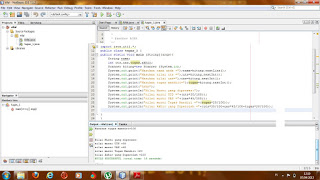
¬ Sub Class/ Class Anak  
Modifier1 Class Namaclass modifier2 {  
//tipedata namaatribut/variable  
}  
This page was created using BCL ALLPDF Converter trial software.  
To purchase, go to http://store.bcltechnologies.com/productcart/pc/instPrd.asp?idproduct=1NMR – Nusa Mandiri Lab B Sore2

Jika sebuah kelas tersebut dapat digunakan oleh kelas lain tanpa memperdulikan,apakah kelas lain yang menggunkannya itu berasal dari package yang sama atau berbeda.

• Abstract  
Jika kelas tersebut memiliki abstract methode dan tidak dapat diinstansiasi menjadi sebuah objek.

• Final  
Kelas tidak akan dapat diturunkan lagi menjadi kelas turunan.

                                                **Penggunaan Input Output Stream**  
  
Perintah Masukan  
  
Untuk melakukan perintah masukan, kita akan menggunakan 3 (tiga) cara, yaitu InputStream, BufferedInputStream dan Scanner.  
1) InputStream  
InputStream adalah subclass Object, yang menjadi landasan untuk class-class yang biasa  
digunakan untuk membaca data dan menampilkan kelayar. Untuk penggunaan InputStream ini,  
harus menyertakan package java.io.  
  
2) InputStreamReader dan BufferedReader  
InputStreamReader digunakan membaca arus byte stream dan mengkonversi byte-byte ke  
dalam nilai-nilai bilangan bulat yang merepresentasikan karakter-karakter Unicode.  
Kelas BufferedReader membaca masukan Stream karakter dan penyangga tersebut untuk  
efisiensi. Pada penggunaannya harus mempunyai Reader Object untuk membuat versi buffered.  
Berikut konstrutor yang digunakan untuk membuat BufferedReader.  
¬ BufferedReader(Reader) digunakan untuk membuat penyangga karakter stream yang  
berhubungan dengan Reader Objek yang ditetapkan.  
¬ BufferedReader (Reader, int) digunakan untuk membuat penyangga karakter stream yang  
berhubungan dengan Reader Objek yang ditetapkan dan dengan penyangga ukuran  
integer.  
Penyangga Karakter Stream dapat dibaca menggunakan metoda read() dan read(char[], int, int)  
untuk menguraikan FileReader, serta dapat juga membaca baris dari teks dengan menggunakan  
readLine ().  
metoda readLine (), digunakan untuk kembalikan String objek yang berisi next line dari teks  
pada stream, tidak termasuk karakter atau karakter-karakter yang merepresentasikan end-of-line.  
Jika akhir stream dicapai, maka nilai string yang ingin dikembalikan bernilai dengan null.  
end-of-line ditandai dengan beberapa pernyataan berikut :  
¬ newline karakter (‘\n')  
¬ carriage return character (‘\r')  
¬ carriage return character yang diikuti oleh satu newline (“\n\r”)  
  
3) Scanner  
Class Scanner digunakan secara ekstensif untuk memasukan data dari keyboard. Tidak  
seperti perintah masukan lainnya, data-data yang dimasukan misalnya berupa angka, tidak perlu  
dilakukan konversi dari string ke integer atau tipe data lainnya. Untuk menggunakan Class  
Scanner, harus menyertakan package java.util.  
Untuk membaca baris dari text yang diinputkan dengan menggunakan metoda next(). Supaya  
string yang dibaca utuh termasuk spasi, dengan menggunakan nextLine(). Untuk membaca data  
berupa nilai integer atau tipe data angka lainnya, sebagai contoh bisa menggunakan metoda,  
seperti dibawah ini :  
¬ nextInt() untuk membaca nilai integer  
¬ nextDouble() untuk membaca nilai double.  
¬ nextFloat() untuk membaca nilai float.  
  
Perintah Keluaran  
Pada saat melakukan aktivitas dasar pemrograman, pasti tidaklah terlepas dari menampilkan  
data atau hasil kelayar. Dalam hal ini kita butuh sebuah standard output yaitu stream yang  
digunakan untuk mengirimkan keluaran kelayar. Stream tersebut yaitu System.out. Pada saat kita  
akan menampilkan kelayar, kita butuh method print atau println. Perbedaan antara print dan  
println yaitu :  
• System.out.print(Statement); , maka informasi yang ditampilkan dilayar tidak pindah  
baris.  
• System.out.println(Statement); , maka informasi yang ditampilkan dilayar pindah baris  
(line new). Statement bisa berupa pesan dan argument atau variabel. Statement biasanya  
diapit dengan tanda kutip ganda ( " " ), untuk memisahkan antara statement dan argumen  
atau variabel dipisah dengan tanda plus ( + ).  
berikut adalah tampilan printscreen system nilai yang dapat anda lihat :

[](http://3.bp.blogspot.com/-HQFC-7tjPRE/UWELhPoEUcI/AAAAAAAAAh0/PdnROMVCkj8/s1600/Untitled.jpg)

Prototype => model kerja dasar dari system produk/informasi, biasanya dibangun untuk tujuan demonstrasi/sbgai bagian dari proses pembangunan.

Tujuan => untuk mengembangkan model menjadi system final.