1. 4 acces modifiers

* Public

public acces modifier berarti suatu data bisa dibaca oleh class mana saja dalam suatu computer. Dengan mempublic suatu data, class mana saja juga bias merubah data tersebut.

* Private

Private acces modifier berarti data tersebut hanya bias dibaca oleh class itu sendiri. Tentu saja data tersebut hanya bisa di rubah oleh class itu sendiri. Sehingga agar class lain bisa mengakses data tersebut dibutuhkan suatu method setter dan getter. Yang method itu sendiri berguna untuk mengubah dan membaca data yang tadi di private kan itu.

* Protected

Protected modifier pada suatu data mengartikan bahwa data tersebut bisa di akses hanya oleh subclass dari class tersebut. Yang dimaksud dari subclass disini yaitu class yang merupakan inheritance atau turunan dari class tersebut.

* Default(not given any acces modifier)

Jika suatu data tidak diberikan acces modifier, maka secara default data tersebut akan hanya dapat diakses oleh class yang berasal dari package yang sama.

1. Overloading & overriding

Overloading dan overriding ini sendiri berasal dari Teknik pemrogaman oop yaitu polymorphism. Dimana dengan adanya overloading dan overriding ini user diperbolehkan memiliki method dengan nama yang sama tetapi bentuknya banyak.

* Overloading

Overloading dilakukan dengan cara memberikan nama yang sama tetapi mempassing variable yang berbeda.

Contoh:

**public** Salesman() {

**super**();

}

**public** Salesman(String nama, String alamat, String jabatan) {

**super**(nama, alamat, jabatan);

}

* Overriding

Overriding terjadi ketika class memiliki suatu method dan ketika class tersebut memiliki subclass dengan nama method yang sama tetapi isi atau hasil dari method tersebut berbeda . ketika hal ini terjadi lah overriding terjadi

Contoh:

**public** **class** Employee {

**public** **int** hitung\_gaji(){

**return** Company.*basic\_salarry*;

}

}

**public** **class** Salesman **extends** Employee{

@Override

**public** **int** hitung\_gaji() {

**int** sum = 0;

**for**(**int** i = 0 ; i < **this**.list\_transaksi.size() ; i++){

sum += **this**.list\_transaksi.get(i).getTotal\_profit();

}

**return** sum;

}

}

3 method arraylist

* + Add

Add berarti menambahkan suatu objek pada arraylist yang sudah dibuat.

* + Size

Mengembalikan berapa jumlah data yang ada pada arraylist tersebut

* + Get

Mengembalikan objek yang sebelumnya sudah ditambahkan pada arraylist tersebut.