**LAPORAN PRAKTIKUM**

**ALGORITMA DAN PEMROGRAMAN**

**Pertemuan Ke – 10**



**DISUSUN OLEH :**

**HELDA LUDYA SAFITRI**

**175410186**

**TEKNIK INFORMATIKA**

**STMIK AKAKOM**

**YOGYAKARTA**

**2017**

**PERTEMUAN KE-10**

**CREATING CLASS**

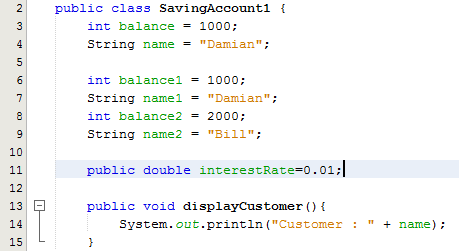
1. **TUJUAN**
2. Mahasiswa dapat membuat kelas.
3. Mahasiswa dapat membuat test/main class.
4. Mahasiswa dapat membuat objek.
5. Mahasiswa dapat Mengakses atribut dan metode.
6. **DASAR TEORI**

Class merupakan salah satu konsep fundamental pemrograman berorientasi objek atau dapat diilustrasikan sebagai prototipe yang digunakan untuk menciptakan objek.

Objek adalah sebuah komponen software yang strukturnya mirip dengan objek pada dunia nyata. Setiap objek dibuat dari satu set data (sifat) dimana variable menjabarkan esensial karakter dari objek dan juga terdiri dari satu set metode yang menjabarkan bagaimana tingkah laku dari objek.

1. **PEMBAHASAN PRAKTIK**

**PRAKTIK 1 – Membuat Kelas**

.

Class merupakan blueprint untuk menciptakan suatu object. Pada praktik 1 a,b,c dibuat dalam 1 class yaitu SavingAccount1.

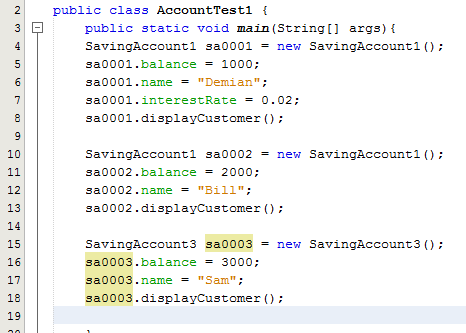
Baris 3-4 merupakan sebuah variable yang hanya bisa di akses di dalam kelas tersebut.

* Int balance : merupakan pendeklarasian variable balance dengan tipe data integer yang bernilai 1000.
* String name : merupakan pendeklarasian variable name dengan tipe data String yang bernilai “Demian”

Kemudian pada baris-baris selanjutnya juga terdiri dari variable dengan tipe data sesuai nilainya dan hanya bisa diakses dalam class tersebut.

Public void displayCustomer() merupakan method yang bisa di akses di dalam class atau di luar class tanpa harus menggunakan pengembalian nilai atau nilai return.

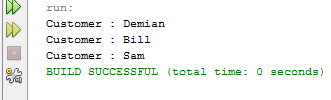
**PRAKTIK 2 – The main Method as a driver Class**



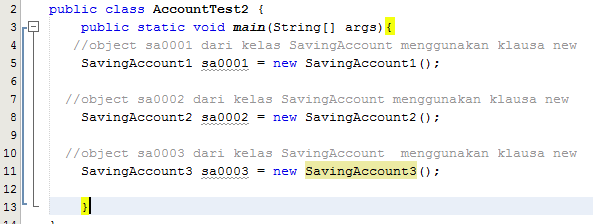
**SavingAccount1 sa0001=new SavingAccount();** digunakan untuk mendeklarasikan class lain yaitu class SavingAccount1yang dibuat pada praktik sebelumnya. **sa0001** merupakan variable atau properti yang mendeklarasikan konstruktor. Baris 5-7 digunakan untuk mengubah nilai dari property atau variable yang ada di class SavingAccount1 dengan menggunakan variable yang mendeklarasikan konstruktor tersebut. Baris 8 digunakan untuk memanggil method yang telah kita buat tadi yaitu **sa0001.displayCustomer();**.

Penjelasan baris selanjutnya sama dengan baris sebelumnya, namun pemanggilan class harus sesuai dengan class sebelumnya.

Hasil Output :



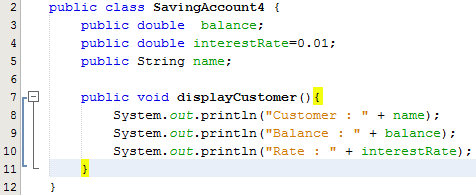
**PRAKTIK 3 – Membuat Objek**

****

Pada praktik 3 hanya membuat konstruktor menggunakan klausa new yang digunakan untuk menginisialisasi object, nama konstruktor harus sama dengan nama class.

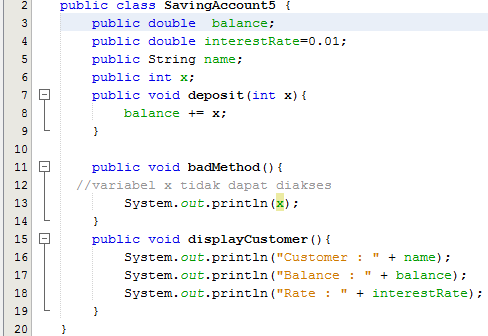
**PRAKTIK 4 – Variable Scope**

1. Field/atribute



**Public double balance;** artinya variable balance akan dideklarasikan dengan tipe data double, karena menggunakan modifier public maka variable-variable tesebut juga bisa diakses didalam kelas itu sendiri maupun di luar class atau didalam class lain. Baris 5 merupakan sebuah method yang digunakan untuk melakukan sebuah operasi yang dapat dilakukan oleh object dimana method ini tidak memiliki nilai pengembalian atau return. Dan method ini bisa di akses di dalam maupun di luar class karena menggunakan modifier public. Kemudian baris 8-10 merupakan statement dari method yang dibuat yaitu **public void displayCustemer().**

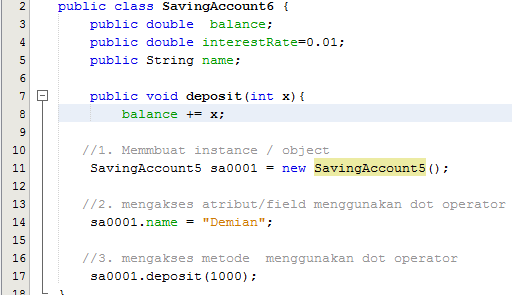
1. Variable yang dibuat dalam metode tidak dapat diakses diluar metode tersebut



Baris 7-8 merupakan sebuah method dengan 1 parameter. Public **void deposit (int x)** artinya untuk mendeklarsikan variable x dengan tipe data interger. Dan method ini bisa di akses di dalam maupun di luar class karena menggunakan modifier public. Baris 10-12 juga merupakan sebuah method yang bisa di akses di luar maupun didalam class tesebut. Statement di dalam method ini akan membuat program error, **System.out.println(x)**; variable x tidak akan bisa di akses karena tidak ada didalam parameter method tersebut yaitu **public void badMethod()**.

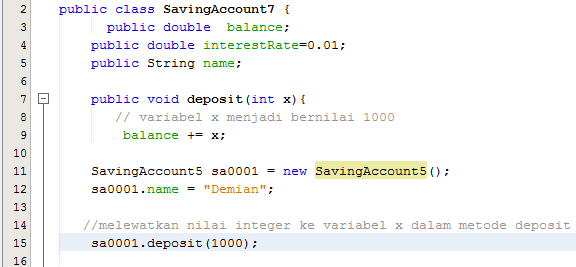
Variable yang dideklarasikan di dalam sebuah method tidak akan bisa di akses di oleh method lain selain dari method yang mendeklarasikannya.

1. Mengakses Field dan metode dari kelas lain



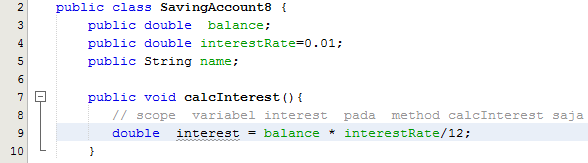
Baris 10 merupakan bentuk dari instance object yang akan digunakan untuk mengakses atribut/field dan method dari class lain.**SavingAccount sa0001=new** **SavingAccount5()** merupakan bentuk konstruktor yang digunakan untuk membuat sebuah instance object. **Sa0001.name=”Demian”;** artinya kita akan mengkases field/atribut dari class yang sudah dibuat sebelumnya dengan cara memberikan nilai pada field/variable tersebut. Namun program ini akan error karena variable/field tersebut tidak pernah dideklarasikan di dalam class yang kita panggil maka program pun secara otomatis tidak bisa membaca atau mengeksekusi variable tersebut karena tidak pernah dideklarasikan.

1. Melewatkan nilai ke metode



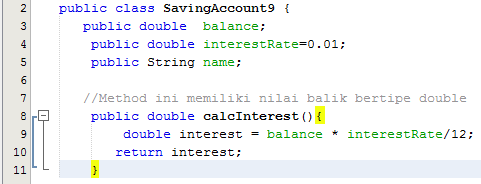
Variable x dengan tipe data integer akan menjadi bernilai 1000 dalam metode deposit.

1. Bagaimana jika dibutuhkan nilai dari metode



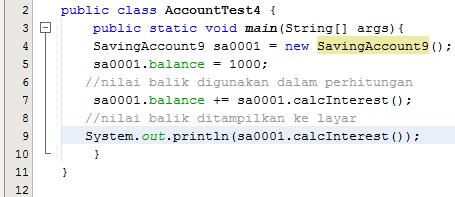
Method yang digunakan yaitu **public void calcInterest()** dimana didalam method ini tedapat sebuah variable atau field sebagai nilai dari method tersebut. **double interest=balance\*interestRate/12;** artinya variable ini akan melakukan proses operasi aritmatika. Dan variable ini hanya bisa di diakses di dalam method calcInterest saja dan tidak bisa bisa diakses di luar method tersebut atau lebih dikenal dengan nama scope variable blok kode.

1. Nilai balik/Return Value dari Metode



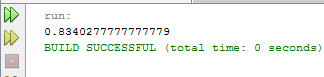
Setiap method yang di deklarasikan dengan nonVoid maka method tersebut membutuhkan nilai balik/return. **Public double calcInterest()** merupakan return tipe yang akan di gunakan ketika method ingin mengembalikan nilai. **Return interest;** digunakan untuk mengembalikan nilai dari type data yang kita gunakan sebelumnya.

1. Contoh Nilai Balik



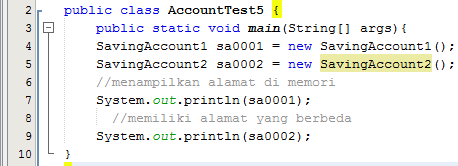
Baris 4 digunakan untuk mengakses atribut dari class yang kita panggil dengan menggunakan instance object yang kita buat**. Sa0001.calcInterest()** merupakan bentuk pemanggilan method dengan nilai balik. Dan baris 9 digunakan untuk manampilkan nilai balik di output.

Hasil Output :



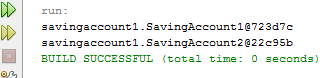
**PRAKTIK 5 - Objects in Memory**

1. Keyword new

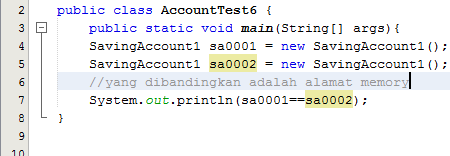


Baris 4-5 merupakan bentuk instance object atau proses pemanggilan class lain ke dalam class tersebut. sa0001 dan sa0002 merupakan sebuah variable yang mendeklarasikannya sebagai instance object didalam class baru. **new SavingAccount();** merupakan bentuk dari konstruktor yang merupakan sebuah method khusus yang digunakan untuk menginisialisasikan object, dan penulisan nama konstruktor harus sama dengan nama classnya. Karena variable sa0001 dan sa0002 merupakan variable yang mendeklarsikan sebuah instance object maka jika ditampilkan di output maka yang akan muncul adalah alamat memory tempat menyimpan variable tersebut.

Hasil Output :

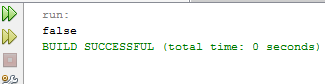


1. Comparing Object

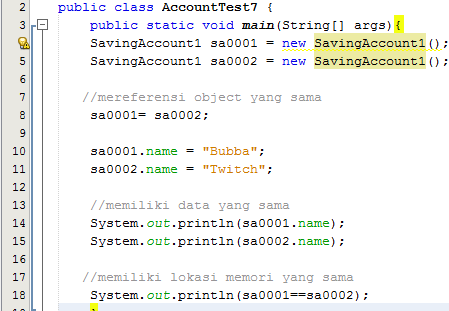


Pada praktik ini akan dibandingkan apakah lokasi atau alamat memory dari variable sa0001 dan sa0002 sama atau tidak.Karena tidak ada perintah yang menginisialisaikannya sama maka lokasi penyimpanan atau alamat penyimpanannya pada memory otomatis berbeda.Maka outputnya adalah false.

Hasil Output :

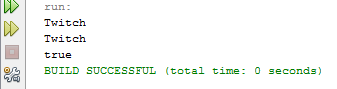
****

1. Two Refereces, One Object



Baris 4-5 untuk menginisialisasikan instance object variable sa0001 sama dengan variable sa0002. Baris 7-8 memasukkan nilai dari atribut/variable yang ada di class lain yang telah di panggil. Karena di awal sudah di inisialiasikan sama maka walaupun nilai atau data yang di masukkan berbeda nilainya akan tetep sama.Baris 10-11 untuk menampilkan nilai atau data dari **sa0001.name** dan **sa0002.name**, karena sudah diinisialisasikan sama maka nilai yang akan tampil di outputkan akan sama. Baris 14-15 untuk menampilkan hasil perbandingan apakah variable instance object sa0001 memiliki alamat memory yang sama dengan sa0002. Karena memiliki data yang sama maka secara otomatis alamat memory yang dipakai juga sama. Maka nilainya adalah true.

Hasil Output :



1. **LATIHAN**

public class CheckingAccunt{

public static void main(String[]arg){

CheckingAccount test=new CheckingAccount();

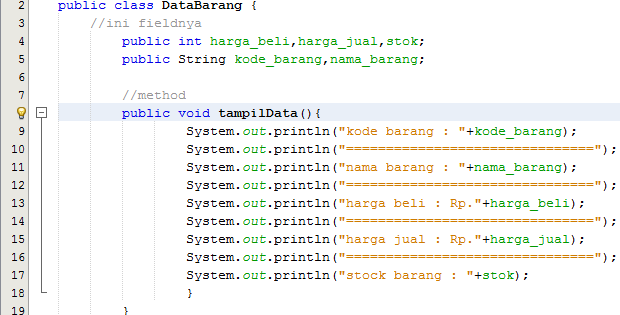
test.balance=175410186;

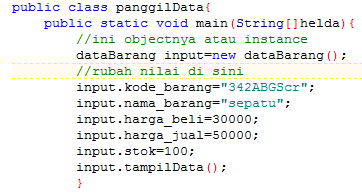
test.name="Helda Ludya Safitri";

test.tampilkanIsi();

Baris 4-5 digunakan untuk memberikan setiap nilai dari field yang ada di dalam class yang kita panggil yaitu class CheckingAccount. Baris 6 digunakan untuk memanggil fungsi method yang ada di dalam class yang kita panggil dimana method tersebut tidak memiliki nilai balik.

1. **TUGAS**

****

****

1. **KESIMPULAN**

Dari pembahasan praktikum diatas dapat disimpulkan bahwa Class merupakan salah satu konsep fundamental pemrograman berorientasi objek atau dapat diilustrasikan sebagai prototipe yang digunakan untuk menciptakan objek.

1. **LISTING**

Terlampir.