MODUL PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK

PENGENALAN PEMROGRAMAN BERORIENTASI OBJEK

Deskripsi Singkat

Praktikum pemrograman berorientasi objek adalah praktikum yang menggunakan bahasa Java sebagai bantuan dalam memahami konsep pemrograman berorientasi objek. Materi praktikum berisi teori, latihan dan soal pemrograman.

Tujuan

1. Memahami konsep objek dengan menggunakan software BlueJ.
2. Menambah fungsi atau method lain pada contoh projek BlueJ.
3. Membuat program Java sederhana dengan BlueJ.
4. Mendeklarasikan suatu class dengan atribut dan methodnya.
5. Membuat method constructor, mutator dan accessor.
6. Mencipta objek dan menggunakan objek.

Prasyarat

Siswa telah melakukan praktikum 1-3.

Materi 1 : BlueJ

BlueJ merupakan suatu *Integrated Development Environment* (IDE) yang dirancang di Universitas Monash, Australia. BlueJ ditujukan sebagai alat bantu untuk mengajarkan pemrograman berorientasi objek. BlueJ menyediakan pemvisualan dari struktur-struktur kelas dengan menggunakan diagram UML. BlueJ memberikan kemudahan kepada pengguna untuk langsung berinteraksi dengan objek yang dicipta.

Materi 2 : Konsep Penting

Objek merepresentasikan ‘sesuatu’ dari dunia nyata atau dari suatu domain masalah. Objek adalah sesuatu yang memiliki keadaan, perilaku dan identitas.

Kelas merepresentasikan objek yang sama. Kelas merupakan satu set objek yang mempunyai atribut dan perilaku yang sama.

Objek dari suatu kelas merupakan anggota (**instance**) dari kelas tersebut. Oleh karena itu jika ingin mencipta suatu objek, maka kita perlu melakukan proses **instantiation** dari kelasnya terlebih dahulu.

Objek memiliki **keadaan (state)**. Keadaan objek merupakan ciri-ciri objek. Keadaan objek diwakili dengan atribut pada pemrograman dimana nilai dapat disimpan didalamnya. Nilai disimpan di dalam variable yang dapat memiliki **tipe data** primitive (seperti int, float, Boolean, long,…) ataupun berupa objek dari kelas tertentu (seperti String, Integer, Bentuk, Lingkaran).

Objek memiliki **perilaku**. Perilaku objek akan muncul jika kita berkomunikasi dengan objek dengan memanggil **method**nya. Setiap objek dapat saling berkomunikasi dengan memanggil methodnya.

Setiap method dapat memiliki **parameter** yang menyediakan metode pengiriman nilai untuk tugas tertentu di dalam method. Setiap parameter akan memiliki **tipe data** tertentu.

Method dapat memiliki ***return value*** (nilai kembali) yang akan mengembalikan informasi tertentu mengenai objek. Return value juga memiliki tipe data tertentu.

Method terdiri dari method ***constructor***, ***accessor*** dan ***mutator***. **Method constructor** adalah method yang digunakan untuk mencipta objek. Secara default sudah ada, namun jika ingin membuat method constructor yang sesuai keinginan juga dapat dilakukan. **Method accessor** adalah method yang mengembalikan nilai dari keadaan suatu objek, secara tidak langsung memberi akses kepada suatu keadaan objek. **Method mutator** adalah method yang mengubah keadaan dari suatu objek.

Secara ringkasnya method accessor digunakan untuk meminta informasi, sedangkan method mutator digunakan untuk meminta objek mengubah keadaan (state) nya.

Materi 3 : Class, Atribut dan Method

Class/ kelas merupakan kumpulan dari objek. Di dalam pemrograman berorientasi objek (PBO), class-lah yang dibuat kode programnya. Sedangkan objek akan diciptakan saat aplikasi dijalankan. Di dalam Unified Modelling Language (UML), class direpresentasikan sebagai kotak persegi dengan 3 ruang, seperti ilustrasi gambar di bawah. Atribut/ instance variable merupakan data yang dimiliki oleh setiap objek dan ada nilainya. Jika tidak diberi nilai awal, maka nilai dari atribut tersebut menggunakan nilai default dari tipe data. Method merupakan perilaku dari objek.

Nama Class
Atribut/ instances variable
Perilaku/ method

Dalam pemrograman Java, syntax deklarasi class:

```
public class NamaClass {
      [modifier] [tipe-data] atribut-instance variable;
      [constructor]
      [method]
}
```

Contohnya ada diagram class Siswa seperti di bawah:

Siswa
- nrp: int
+ Siswa (nrpx : int) + setNrp (nrpx : int) + getNrp () : int

Maka kode program Java seperti di bawah:

```
public class Siswa {
      private int nrp;

      public Siswa(int nrpx)
      {
            nrp = nrpx;
      }

      public void setNrp(int nrpx)
      {
            nrp = nrpx;
      }

      public int getNrp()
      {
            return nrp;
      }
}
```

Nama class

Atribut/ instance variable

Method constructor

Method mutator

Method accessor

Materi 4 : Method Constructor, Mutator dan Accessor

Method constructor merupakan method untuk memberi nilai awal pada semua atribut/ instance variable. Pada pemrograman Java, method constructor secara default sudah ada, nilai awal yang digunakan sesuai dengan tipe data. Namun kita dapat saja membuat method constructor sendiri, dan memberikan nilai awal sesuai dengan yang diinginkan. Method constructor **harus** menggunakan nama yang sama dengan nama class. Sebab method constructor inilah yang dipanggil saat objek diciptakan. Method constructor tidak memiliki tipe data kembalian, namun method constructor boleh saja memiliki parameter.

Method mutator merupakan method yang digunakan untuk mengubah nilai dari atribut/ instance variable. Method mutator biasanya bertipe data kembalian sebagai void, namun memiliki parameter. Contoh method mutator pada class Siswa di atas adalah void setNRP(int nrpx).

Method accessor merupakan method yang digunakan untuk mengakses/mengembalikan nilai dari atribut tertentu. Method accessor memiliki tipe data kembalian sehingga memiliki keyword **return** di

dalam methodnya. Namun method ini biasanya tidak memiliki parameter. Contoh method accessor pada class Siswa di atas adalah int getNrp();

Materi 5 : Mencipta dan Menggunakan Objek

Dalam aplikasi berbasis PBO, yang bekerja adalah objek di dalam memori. Objek diciptakan dengan menggunakan keyword new. Pada BlueJ, objek dapat lebih mudah lagi diciptakan tanpa memerlukan kode tambahan. Hanya dengan klik-kanan pada class lalu memlih method constructornya.

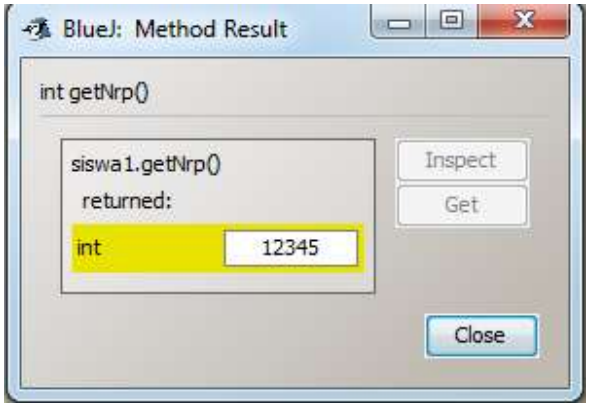
Gambar berikut ini mengilustrasikan proses menciptakan objek dari class Siswa. Kita tinggal memilih new Siswa (int nrpx) atau memilih method constructornya.

Berikutnya muncul tampilan untuk memberi nama objek dan memberi nilai bagi parameter dari method constructornya.

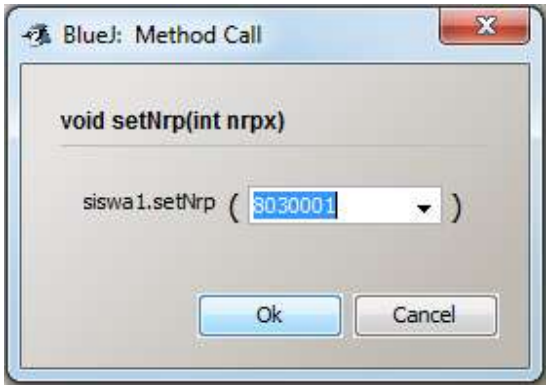
Maka muncullah objek siswa di bagian bawah.

Untuk menggunakan objek siswa1 tersebut, kita tinggal klik-kanan lalu memilih method yang diinginkan.

Contoh jika memilih method getNrp, maka akan mengembalikan nilai nrp.



Contoh jika memilih method setNrp, maka kita perlu memberi nilai pada parameter nrpx.



Objek yang telah diciptakan pada BlueJ dapat dengan mudah diketahui nilai atributnya dengan klik-kanan pada objek lalu memilih inspect.

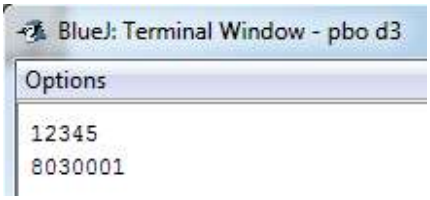
Jika kita ingin menggunakan class Siswa di dalam class X dan class X tersebutlah yang akan dijalankan. Maka class X tersebut perlu memiliki method main. Berikut contoh aplikasi yang menciptakan dan menggunakan objek dari class Siswa.

```
public class TesSiswa
{
    public static void main(String[] ar)
    {
        Siswa s = new Siswa(12345);

        System.out.println(s.getNrp());

        s.setNrp(8030001);
        System.out.println(s.getNrp());
    }
}
```

Coba simpan, kompilasi dan jalankan class TesSiswa. Contoh tampilan hasilnya.



LATIHAN 1

Buka software BlueJ. Download contoh-contoh projects BlueJ di:

http://informatika.unsyiah.ac.id/viska/pbo/projects.zip

Kita akan menggunakan contoh projek naïve-ticket-machine yang terletak dalam folder chapter02.

Projek naïve-ticket-machine ini menggambarkan mesin tiket pada stasiun kereta api yang akan mencetak tiket setelah pelanggan memasukkan sejumlah uang yang tepat sesuai dengan harga karcis. Mesin akan terus bekerja menjumlahkan total uang yang berhasil dikumpulkan.

Langkah-langkah:

- Open projek naïve-ticket-machine
- New object, buat harga tiket misalnya 1000.
- Lihat isi semua fields/state dengan klik kanan pada objek tiket dan pilih inspect.
- Anggap pelanggan datang dan memasukkan uang. Klik kanan pada objek dan pilih insertMoney dengan jumlah uang 1000.
- Lihat isi semua fields dengan perintah inspect. Amati perubahan nilainya.
- Kemudian pelanggan mencetak tiket. Klik kanan pada objek dan pilih printTicket.
- Lihat isi semua fields kembali dengan perintah inspect. Amati perubahan nilainya.
- **Apa kesimpulan anda?** Jika belum paham mengenai cara kerja mesin tiket, silakan ulang kembali mulai dari pelanggan datang.
- **Apa kesimpulan anda?** Apakah masih ada kekurangan pada mesin tiket sederhana ini?

Sekarang kita akan membuat mesin tiket sederhana yang lebih baik. Beberapa hal yang perlu anda tambahkan:

- Pastikan uang yang dimasukkan bukan negatif.
- Pastikan uang yang ada cukup untuk mencetak tiket, jika tidak pelanggan harus memasukkan uang lagi.
- Pastikan jika uang pelanggan lebih, uangnya dapat dikembalikan.

Pada bagian mana anda ingin memastikan bahwa uang yang dimasukkan bukan negatif? Tentulah pada bagian insertMoney. Jadi klik kanan pada class TicketMachine dan pilih OpenEditor. Maka window yang berisi source code dari program akan terbuka.

Untuk memastikan uang bukan negatif, tentulah kita perlu melakukan pengujian dengan if-else, seperti code di bawah:

```
public void insertMoney(int amount)
{
    if(amount > 0) {
        balance = balance + amount;
    }
    else {
        System.out.println("Masukkan uang bernilai positif: " +
                           amount);
    }
}
```

Selanjutnya pada bagian mana anda ingin memastikan uang pelanggan cukup untuk mencetak tiket? Tentu pada bagian printTicket.

```
public void printTicket()
{
    if(balance >= price) {
        // Simulate the printing of a ticket.
        System.out.println("##################");
        System.out.println("# The BlueJ Line");
        System.out.println("# Ticket");
        System.out.println("# " + price + " cents.");
        System.out.println("##################");
        System.out.println();

        // Update the total collected with the price.
        total = total + price;
        // Reduce the balance by the prince.
        balance = balance - price;

        //Jika ada uang kembalian
        int sisa = this.refundBalance();
        if(sisa > 0)
            System.out.println("Uang kembalian: " +sisa);
    }
    else {
        System.out.println("Uang anda kurang, silakan masukkan : " +
                           (price - balance) + " cents lagi.");

    }
}
```

```
/**
 * Return the money in the balance.
 * The balance is cleared.
 */
public int refundBalance()
{
    int amountToRefund;
    amountToRefund = balance;
    balance = 0;
    return amountToRefund;
}
```

LATIHAN 2

Buka projects-chapter02, pilih book-exercise. Kita akan mencoba melengkapi kelas buku dengan menambahkan hal-hal berikut:

1. Tambahkan 2 method accessor yaitu **getAuthor** dan **getTitle**, yg akan mengembalikan nama pengarang dan judul.
2. Tambahkan 2 method, **printAuthor** dan **printTitle**, yg akan mencetak nama pengarang dan judul ke terminal window.
3. Tambahkan field **pages** yg bertipe int ke kelas Book, yg menyimpan jumlah halaman dlm buku. Nilai awalnya akan diset pd method constructor, jd tambahkan parameternya pd constructor.
4. Tambahkan method **getPages** yg akan mengembalikan jumlah halaman.
5. Tambahkan method **printDetails** yg akan mencetak nama pengarang, judul, jumlah halaman pd terminal window.
6. Tambahkan field **refNumber** yg bertipe String ke kelas Book, yg akan menyimpan nomor referensi perpustakaan. Pd constructor, initialize dgn string kosong (“”).
7. Tambahkan method mutator
   public void setRefNumber(String ref)
   yg akan memberi nilai pada refNumber.
8. Ubah method printDetails dengan menambahkan cetakan utk refNumber. Jika refNumber masih string kosong, maka cetak “zzz”.

LATIHAN 3

Ubah class Siswa dengan menambahkan atribut nama, method getNama dan setNama. Ilustrasi diagram classnya seperti di bawah.

Siswa
- nrp: int - nama : String
+ Siswa (nrpx : int, namax : String) + setNrp (nrpx : int) + getNrp () : int + setNama (namax : String) + getNama() : String

LATIHAN 4

```
/*
Class untuk mengecek nilai bawaan/default dari tipe data
*/
public class TesNilaiDefault {
      int nilaiInt;
      char nilaiChar;
      double nilaiDouble;
      boolean nilaiBool;
      String nilaiString;
      Siswa nilaiSiswa;
}
```

Simpan pada BlueJ lalu kompilasikan. Kemudian pilih method constructornya new TesNilaiDefault(). Sesudah muncul objek pada bagian bawah, klik-kanan lalu klik inspect. Berikut adalah tampilan yang muncul.

Terbukti disini, walaupun kita tidak membuat method constructor, tapi Java secara default sudah memiliki method constructor yang akan memberi nilai bawaan bagi semua atribut/ instance variable. Nilai bawaan tersebut dapat dilihat di atas.

Bagi tipe data int dan sejenisnya (byte, short, long), nilai bawaan adalah 0. Bagi tipe data char, nilai bawaan adalah 0 di dalam Unicode character yaitu '\u0000'. Bagi tipe data double dan float, nilai bawaan adalah 0.0. Bagi tipe data boolean, nilai bawaan adalah false.

Tipe data berupa objek dari suatu class adalah tipe data reference. Yang akan merujuk ke tempat lain (alamat) dimana objek tersebut disimpan di dalam memory. Jika objek tersebut belum memiliki data, maka nilai bawaan dari tipe data reference adalah null.

LATIHAN 5

Buatlah class Lingkaran berdasarkan diagram class berikut.

Lingkaran
- jari_jari : double
+ Lingkaran (jari : double) + setJariJari (jari : double) + getJariJari () : double

SOAL-SOAL

1. Dengan menggunakan sambungan projek yang anda kerjakan pada latihan 2, tambahkan hal-hal berikut pada projek book-exercise.

a. Ubah method **setRefNumber** sehingga field refNumber hanya akan diubah nilainya jika parameter berupa string dgn panjang minimal 3 karakter. Jika kurang dari 3 karakter maka tampilkan pesan error dan nilai field refNumber tidak berubah.
b. Tambahkan field **borrowed** bertipe int yg akan menghitung berapa kali buku telah dipinjam. Field ini bernilai awal 0.
c. Tambahkan method mutator **borrow** yg akan mengupdate nilai borrowed sehingga bertambah 1.
d. Tambahkan method accessor **getBorrowed** yg akan mengembalikan nilai dari field borrowed.
e. Ubah method printDetails dengan menambah cetakan terhadap nilai borrowed. Anda dpt menambahkan teks tambahan yg sesuai.

2. Tambahkan method cariLuas dan cariKeliling pada class Lingkaran di latihan 5.