

Modul 1 : Pemrograman Java**KOMPETENSI :**

Setelah menyelesaikan praktikum pada modul 1, praktikan diharapkan dapat mengetahui konsep yang digunakan pada pemrograman Java, mengenali bentuk deklarasi class, konstruktor, objek class dan fungsi utama pada pemrograman Java.

TUJUAN INTRUKSIONAL KHUSUS :

1. Mengetahui Konsep Pada Pemrograman Java
2. Mengetahui Tools Yang Digunakan Dalam Pemrograman Java
3. Mengenali Class, Konstruktor, Objek dan Fungsi Utama Pada Pemrograman Java serta implementasinya.

TUGAS PENDAHULUAN

1. Jelaskan apa itu pemrograman berbasis objek !
2. Jelaskan 3 konsep dalam pemrograman berbasis objek !
3. Tuliskan deklarasi class, konstruktor, fungsi utama dan objek pada pemrograman Java !
4. Tuliskan jenis-jenis tipe data pada pemrograman java.
5. Jelaskan yang dimaksud dengan atribut dan method pada pemrograman java.
6. Tuliskan deklarasi atribut dan method pada java.
7. Tuliskan contoh program yang menggunakan atribut dan method !

TEORI DASAR:**1. Pemrograman Java**

Java merupakan salah satu bahasa pemrograman yang sudah ada dari era 1990-an. Bahasa Pemrograman Java adalah bahasa pemrograman yang mengadopsi sintaks dari bahasa C dan C++. Salah satu sifat pemrograman Java yang menjadi keuntungan dalam menggunakan bahasa pemrograman ini adalah bersifat *multiplatform* yaitu Java dapat dijalankan pada banyak platform atau sistem informasi.

2. Tools Pemrograman Java

Untuk membuat program atau aplikasi menggunakan Java ada beberapa tools yang dibutuhkan. Hal yang sangat penting dan wajib adalah compiler bahasa pemrograman Java yaitu JDK (Java Development Kit). Kemudian Text Editor untuk menuliskan source code pemrograman Java. Namun, sekarang sudah ada IDE (Integrated Development Environment – Lingkungan Pengembangan Terintegrasi) untuk bahasa pemrograman Java seperti Netbeans, Eclipse, IntelliJIDEA.

3. Konsep Pemrograman Java

Konsep yang digunakan dalam pemrograman Java adalah OOP (Object Oriented Programming) atau Pemrograman Berorientasi Objek. OOP adalah sebuah metodologi atau cara berpikir dalam melakukan pemrograman dimana pendefinisian tipe data disertai dengan pendefinisian fungsi. Struktur data yang seperti ini disebut dengan istilah object. Paradigma pemrograman OOP dapat dilihat sebagai interaksi sebuah object dalam melakukan tugasnya. OOP ini terdiri dari 3 konsep yaitu :

- a. Enkapsulasi, yaitu pemberian hak akses oleh akses modifier (public, private, protected, default)
- b. Inheritance yaitu konsep pewarisan. Konsep ini akan dibahas di modul selanjutnya.
- c. Polimorfisme yang artinya banyak bentuk. Konsep ini akan dibahas di modul selanjutnya.

Untuk dapat mengetahui dan memahami konsep OOP tersebut, maka hal yang mesti dipahami adalah Class, Objek, dan Fungsi Main pada pemrograman Java.

4. Class, Objek dan Fungsi Main Pada Pemrograman Java**1) Class atau Kelas**

Class atau kelas adalah kelompok objek-objek yang memiliki karakteristik yang sama (yang sejenis). Dalam ruang lingkup pemrograman, class ini sering dianalogikan sebagai sebuah cetakan dalam kehidupan nyata, dimana dari sebuah cetakan kita bisa membuat banyak kue dan kue ini bisa kita sebut sebagai object.

Deklarasi Class Pada Java

```
Akses_Modifier    class    NamaClass {
    . . .
}
```

Contoh :

```
public class Rumah {
    . . .
}
```

Pada pemrograman java, Setiap Class akan disimpan dalam 1 file yang berekstensi “.java” dengan nama file harus sesuai dengan nama class yang sudah dibentuk. Untuk contoh diatas, untuk penyimpanannya adalah “**Rumah.java**”.

2) Konstruktor

Konstruktor adalah sebuah tipe khusus dari method yang digunakan untuk membuat dan menginisialisasi sebuah object baru. Berikut adalah property dari konstruktor :

- Konstruktor memiliki nama yang sama dengan nama class.
- Konstruktor tidak memiliki tipe data dan tidak mengembalikan nilai.
- Konstruktor tidak dapat dipanggil secara langsung, namun harus dipanggil dengan menggunakan operator *new* pada pembentukan sebuah *class*.

Deklarasi Konstruktor Pada Java

```
AksesModifier    NamaKelas (parameter) {
    statement
}
```

Contoh :

```
public class Rumah {
    . . .
}
```

Untuk Class Diatas, Untuk Pembuatan Konstrukturnya adalah :

```
public Rumah( ){
    . . .
}
```

3) Objek Class

Setiap class pasti memiliki objek, satu kelas ini dapat terbentuk banyak objek. Objek ini digunakan umumnya untuk berelasi dengan objek lainnya.

Deklarasi Objek Pada Java

```
NamaClass    NamaObjek    =    new NamaClass( );
```

Contoh :

```
public class Rumah {
    . . .
}
```

Untuk Class Diatas, Untuk Pembuatan Objeknya adalah :

```
Rumah    r = new Rumah( );
```

Pada pemrograman Java, tidak menutup kemungkinan dalam satu program dibutuhkan lebih dari 1 Class yang mana artinya dalam 1 program terdapat banyak Class dan 1 fungsi utama. Untuk memberi dan mengambil nilai dari suatu atribut atau method dari Class lain maka dibutuhkan objek dari Class tempat pendeklarasian atribut atau method tersebut.

Contoh :

```
public class Alamat {
    String alamat=" ";
}
```

```

public class Rumah {
    public static void main( String [] args){
        //Untuk menggunakan atribut/variabel alamat pada Class
        Alamat maka dibutuhkan objek dari Class Alamat.
        Alamat al = new Alamat();
        al.alamat = "Batua Raya X ";
    }
}

```

4) Fungsi Utama atau Fungsi Main.

Fungsi utama adalah fungsi yang pertama kali dijalankan. Didalam fungsi utama diperlukan objek untuk menggunakan variable atau procedure dari luar fungsi utama.

| Deklarasi Objek Pada Java |
|--|
| <pre> AksesModifier static void main (String [] args) { . . . } </pre> |
| Contoh : |
| <pre> public class Rumah { public static void main (String [] args) { System.out.println ("Hello Word"); } } </pre> |

PRAKTIKUM MANDIRI

| 1. Peralatan | | | | | | | | |
|---|-----------------|-----------------|---|--|---|--|--|--|
| a) Perangkat komputer / PC / Laptop / Notebook . b) Sistem operasi Windows / Linux (optional Mac OS) c) Aplikasi Geany / Notepad | | | | | | | | |
| 2. Prosedur | | | | | | | | |
| a) Baca dan pahami semua tahapan praktikum dengan cermat. b) Gunakan fasilitas yang disediakan dengan penuh rasa tanggung jawab. c) Rapihkan kembali setelah menggunakan komputer (mouse, keyboard, kursi, dll) d) Perhatikan sikap anda untuk tidak mengganggu rekan praktikan lain e) Pastikan diri anda tidak menyentuh sumber listrik. | | | | | | | | |
| 3. Kegiatan Praktikum | | | | | | | | |
| 1) Buka texteditor yang akan digunakan. 2) Buat file baru kemudian simpan file tersebut dengan nama “ HelloWord.java ” 3) Kerjakan percobaan yang telah disediakan dan tuliskan kesimpulan dari percobaan terebut. 4) Setelah percobaan selesai, tutup semua perangkat lunak yang telah digunakan. 5) Matikan PC dan Rapihkan meja praktikum. | | | | | | | | |
| 4. Percobaan | | | | | | | | |
| <table><tr><th>Percobaan</th><th>Hasil Percobaan</th></tr><tr><td>Buatlah sebuah Class dengan nama class “ HelloWord” dengan akses modifier class “public”.</td><td></td></tr><tr><td>Tambahkan konstruktor pada class HelloWord dengan isi konstruktor menampilkan nama dan stb anda.</td><td></td></tr><tr><td>Tambahkan fungsi utama pada class HelloWord.</td><td></td></tr></table> | Percobaan | Hasil Percobaan | Buatlah sebuah Class dengan nama class “ HelloWord ” dengan akses modifier class “ public ”. | | Tambahkan konstruktor pada class HelloWord dengan isi konstruktor menampilkan nama dan stb anda. | | Tambahkan fungsi utama pada class HelloWord . | |
| Percobaan | Hasil Percobaan | | | | | | | |
| Buatlah sebuah Class dengan nama class “ HelloWord ” dengan akses modifier class “ public ”. | | | | | | | | |
| Tambahkan konstruktor pada class HelloWord dengan isi konstruktor menampilkan nama dan stb anda. | | | | | | | | |
| Tambahkan fungsi utama pada class HelloWord . | | | | | | | | |

| | | |
|---|--|--|
| | | |
| Buatlah program tersebut untuk menghasilkan output berupa text “Hello Word !!!” | | |
| Lengkapi program tersebut hingga menghasilkan output berupa text “Nur / 13020140003 “ “Ini Pemrograman Java” | | |
| Buatkan objek class HelloWorld tersebut di dalam fungsi utama. | | |
| Ubah akses modifier class dari public menjadi private . Apa yang terjadi ? | | |
| Ubah keyword println menjadi print . Apa yang terjadi ? Tuliskan perbedaan diantara keduanya. | | |

EVALUASI PRAKTIKUM

1. Bagaimana cara membuat lebih dari satu objek class dengan class yang sama ?
2. Apakah bisa membuat lebih dari satu fungsi utama ? Jelaskan alasannya.
3. Buatlah sebuah program sederhana yang menghasilkan output berupa biodata diri sendiri. Sertakan dengan gambar output program.
4. Buatlah sebuah program dengan mengimplementasikan konstruktor berparameter ! Sertakan dengan gambar output.
5. Tuliskan kesimpulan praktikum pada kolom kesimpulan.

Modul 2 : Atribut & Method

KOMPETENSI :

Setelah menyelesaikan praktikum pada modul 2, praktikan diharapkan dapat memahami tipe data, atribut dan method serta mampu menggunakan atau mengimplementasikannya dalam pemrograman Java.

TUJUAN INTRUKSIONAL KHUSUS :

1. Mengetahui Apa Atribut pada Pemrograman Java.
2. Mengetahui Deklarasi Umum dari Atribut dan Method Pada Pemrograman Java.
3. Mampu mengimplementasikan Penggunaan Atribut dan Method Pada Pemrograman Java.
4. Mengetahui Apa Itu Method Pada Pemrograman Java.
5. Mengetahui Jenis-jenis Method dan Penggunaannya.

TUGAS PENDAHULUAN

1. Tuliskan apa itu package pada java!
2. Tuliskan langkah-langkah membuat package pada netbeans!
3. Tuliskan apa itu import package dan class serta fungsinya!
4. Bagaimana cara mengimport package dan class!
5. Jelaskan apa yang dimaksud dengan enkapsulasi!
6. Tuliskan contoh program yang menggunakan method setter dan getter (konsep enkapsulasi).

TEORI DASAR:

1. Tipe Data

Tipe data adalah jenis data yang dapat ditampung dalam suatu variable. Pada pemrograman java tipe data terbagi dua yaitu tipe data primitive dan tipe data referensi. Tipe data primitive adalah tipe data yang memiliki nilai dan ukuran tertentu dan bukan referensi dari objek atau class seperti int, char, float, double, boolean, dll. Sedangkan tipe data referensi atau tipe data class adalah tipe data yang merupakan referensi dari objek atau class seperti Integer, String, Boolean, dll.

2. Atribut

Atribut atau variabel merupakan satuan dasar penyimpanan dalam program Java. Atribut/variabel ini ada 3 macam yaitu sebagai berikut.

- Variabel Global adalah variabel yang dideklarasikan diluar fungsi/prosedur sehingga dapat digunakan oleh semua fungsi dan prosedur yang ada dalam suatu class.
- Variabel Lokal adalah variabel yang dideklarasikan didalam suatu fungsi/prosedur sehingga hanya dapat digunakan oleh fungsi/prosedur tempat pendeklarasiannya.
- Variabel Static yaitu variabel yang pada saat dideklarasikan menggunakan keyword 'static', variabel static ini artinya variabel yang bersifat instan maksudnya dalam pemanggilannya tidak dibutuhkan referensi dari suatu objek classnya.

Deklarasi Atribut / Variabel Pada Java

TipeData NamaVariabel ;

Contoh :

`int nilai;` → Contoh Variabel Menggunakan Tipe Data Primitive

`String nama1, nama2;` → Contoh Variabel Menggunakan Tipe Data Referensi

3. Konversi Tipe Data

Konversi tipe data adalah merubah tipe data suatu variabel. Berikut beberapa cara konversi pada pemrograman java.

- a. Mengubah data string ke numerik integer

```
int varInteger = Integer.parseInt ( strVarInteger ) ;
```

- b. Mengubah data String ke numerik double

```
double varDouble = Double.parseDouble ( strVarDouble ) ;
```

- c. Mengubah data dari string ke short

```
short varShort = Short.parseShort ( strVarShort ) ;
```

- d. Mengubah data dari string ke long

```
long varlong = Long.parseLong ( strVarLong ) ;
```

- e. Mengubah data dari string ke long

```
float varFloat = Float.parseFloat ( strVarFloat ) ;
```

Terdapat banyak jenis konversi bilangan mulai dari konversi tipe data nonnumerik (text) ke tipe data numerik ataupun sebaliknya.

4. Method

Method adalah kumpulan kode perintah untuk melakukan sebuah proses yang diberi nama yang disebut sebagai nama method. Nama method sebagai rujukan untuk sekumpulan kode tersebut. Method pada pemrograman java adalah istilah untuk prosedur dan fungsi (*function*) pada pemrograman yang lain. Method pada pemrograman java ada banyak jenisnya yaitu sebagai berikut.

- a. Konstruktor

Konstruktor adalah sebuah fungsi yang dijalankan pertama kali pada sebuah class. Konstruktor ini berperan dalam inisialisasi objek pada suatu class. Konstruktor digunakan untuk memberi nilai awal pada suatu class. Konstruktor tidak dapat mengembalikan nilai balik. Deklarasi konstruktor harus memiliki nama sesuai dengan nama classnya.

| Deklarasi Konstruktor Pada Java |
|--|
| <pre>AksesModifier NamaClass (){ //statement }</pre> |
| Contoh : |
| <pre>public class Modul4 { public Modul4(){ //statement } }</pre> |

- b. Procedure dan Function

Procedure adalah method yang bertipe data **void** sehingga tidak mengembalikan nilai.

| Deklarasi Method Tidak Mengembalikan Nilai |
|---|
| <pre>AksesModifier void NamaMethod (){ //statement }</pre> |
| Contoh : |
| <pre>public void Biodata (){ //statement }</pre> |

Function adalah method yang bertipe data selain dari **void** seperti int, String, float, dll. Method ini mengembalikan nilai balik sesuai dengan tipe data method.

| Deklarasi Method Mengembalikan Nilai |
|--|
| <pre>AksesModifier TipeData NamaMethod (){ return Nilai; //nilai yang dikembalikan harus bertipe data sesuai tipe data method }</pre> |
| Contoh : |
| <pre>public int nilai(){ return 100; }</pre> |

- c. Method Overloading

Overloading adalah kondisi dimana terdapat dua atau lebih method yang memiliki nama method yang sama tetapi parameter yang berbeda. Konsep overloading ini juga berlaku pada konstruktor.

d. Method Overriding

Overriding adalah kondisi dimana terdapat dua atau lebih method dengan nama yang sama yang berada dalam konsep pewarisan / inheritance. *(Akan dibahas di modul polymorphisme.)*

PRAKTIKUM MANDIRI

| |
|---|
| 1. Peralatan |
| a) Perangkat komputer / PC / Laptop / Notebook . b) Sistem operasi Windows / Linux (optional Mac OS) c) Aplikasi Geany / Notepad |
| 2. Prosedur |
| a) Baca dan pahami semua tahapan praktikum dengan cermat. b) Gunakan fasilitas yang disediakan dengan penuh rasa tanggung jawab. c) Rapihkan kembali setelah menggunakan komputer (mouse, keyboard, kursi, dll) d) Perhatikan sikap anda untuk tidak mengganggu rekan praktikan lain e) Pastikan diri anda tidak menyentuh sumber listrik. |
| 3. Kegiatan Praktikum |
| 1) Buka texteditor (geany) yang akan digunakan. 2) Buat file baru kemudian simpan file tersebut dengan nama Hitung.java 3) Lengkapi program berikut. <pre> public class Hitung { public () →KONSTRUKTOR { nama = "Ismail"; stb = "13020100101";} buatlah atribut <i>nama</i> dan <i>stb</i> bertipe <i>String</i> public float tambah (parameter){mengembalikan hasil jumlah <i>nilai1</i> dan <i>nilai2</i> } public float selisih (parameter){mengembalikan hasil selisih <i>nilai1</i> dan <i>nilai2</i> } public float kali (parameter){mengembalikan hasil kali <i>nilai1</i> dan <i>nilai2</i>} public float bagi (parameter){mengembalikan hasil bagi <i>nilai1</i> dan <i>nilai2</i>} public void tampilNama(){<i>nilai</i> dari variabel <i>stb</i> dan <i>nama</i>} public static void main(String[] args) { buatlah objek dari class "<i>Hitung</i>" buatlah variabel <i>nilai1</i> dan <i>nilai2</i> bertipe <i>int</i> nilai1 = 10; nilai2 = 5; String pilihan = "tambah" ; //pilihan : tambah, selisih, kali, bagi if (pilihan.equals("tambah")){ System.out.println(nilai1 + "+" + nilai2 + " = " + panggil method <i>tambah</i>); } else if (pilihan.equals("selisih")){ System.out.println(nilai1 + "-" + nilai2 + " = " + panggil method <i>selisih</i>); } else if (pilihan.equals("kali")){ System.out.println(nilai1 + "x" + nilai2 + " = " + panggil method <i>kali</i>); } else if (pilihan.equals("bagi")){ System.out.println(nilai1 + "/" + nilai2 + " = " + panggil method <i>bagi</i>); } else { System.out.println("Tidak Ada Pilihan !"); } System.out.println("-----"); panggil method <i>tampilNama</i>. } } </pre> |

OUTPUT :

Output jika nilai pilihan = tambah :

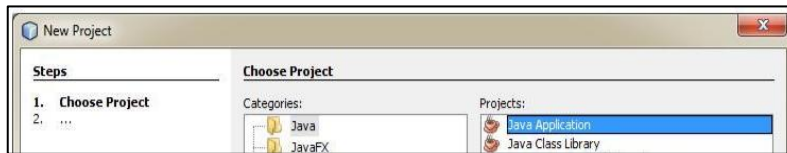
```
10 + 5 = 15.0
```

```
-----  
Nama :
```

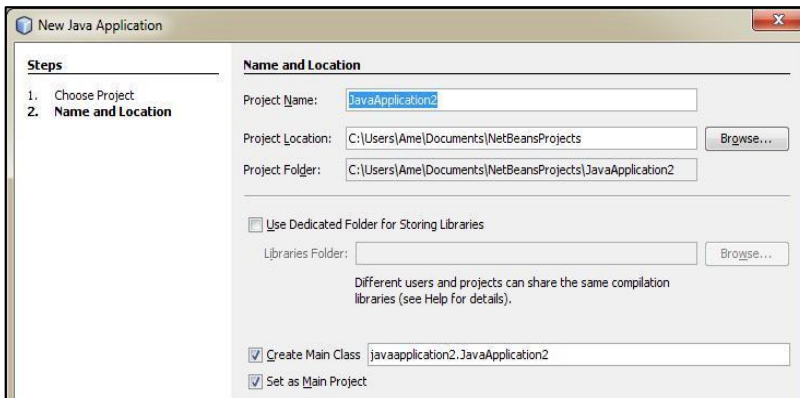
```
Stb :
```

EVALUASI PRAKTIKUM

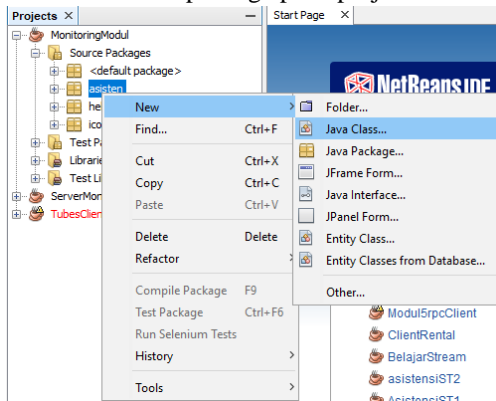
1. Buatlah contoh program yang mengimplementasikan penggunaan variabel local, global dan static !
2. Buatlah contoh program yang mengimplementasikan method procedure dan function!
3. Jelaskan kapan method berparameter dan method nonparameter digunakan ! Berikan contohnya !
4. Pada program nomor 2, ubah semua method yang dibuat menjadi method static. Tuliskan pesan error yang terjadi dan jelaskan maksud error tersebut !
5. Bagaimana cara memanggil atribut atau method yang bersifat static ! Jelaskan !
6. Tuliskan minimal 6 poin kesimpulan pada modul ini!



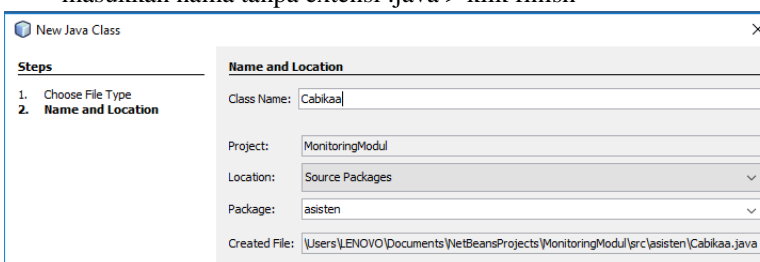
- Setelah itu, akan tampil jendela seperti berikut ini, pada bagian **project name** silahkan masukkan stambuk anda kemudian klik finish.



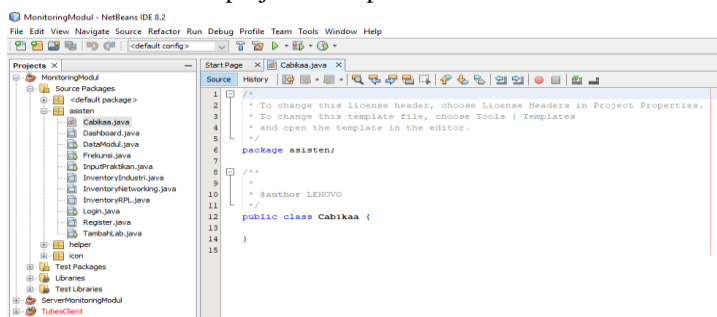
- Klik kanan package pada project anda untuk membuat class pada editor netbeans.



- Akan tampil jendela seperti berikut ini, masukkan nama class java pada isian **class name**, masukkan nama tanpa ekstensi .java > klik finish



- Maka akan tampil jendela seperti ini :



- Editor siap digunakan.

2. Package & Import

Package adalah : suatu cara untuk memanager atau mengelompokkan class-class berdasarkan kesamaan atau kemiripan fungsi.

Dalam versi Java ada 6 package besar yaitu :

- java.io
- java.lang
- java.applet
- java.awt
- java.net
- java.util

Secara fisik, package dapat berupa folder yang berisi file/class, interface/enum lain yang kegunaannya hampir sama sehingga patut untuk dikelompokkan. Package mempengaruhi hak akses ke class lainnya.

Menggunakan Package

Ada 2 cara menggunakan suatu package yaitu:

- Apabila Class yang berhubungan dengan Class lain berada pada package sama maka tidak diperlukan **import package**.
- Apabila Class yang berhubungan dengan Class lain berada pada package yang berbeda maka diperlukan **import package**.

Mengimport Package

`Import namaPackage>NamaClass;` atau `Import namaPackage.*;`

3. Enkapsulasi

Enkapsulasi artinya pembungkusan, pembungkusan yang dimaksud adalah membungkus suatu class, atribut, atau method menggunakan access modifier untuk memberikan hak akses. Ada 4 macam modifier yaitu :

- a. Default
Modifier default merupakan akses untuk class yang hanya bisa diakses oleh class dalam package yang sama.
- b. Public
Modifier public yaitu hak akses yang dapat di akses dari manapun, baik itu dari class yang berbeda maupun dari package yang berbeda.
- c. Protected
Hak akses untuk tipe protected hampir sama dengan public akan tetapi tidak bisa diakses dari paket yang berbeda kecuali merupakan turunan.
- d. Private
Modifier private yaitu hak akses yang hanya bisa di akses oleh kelas itu sendiri. Untuk mengambil atau memberikan nilai pada atribut tersebut digunakan method setter dan getter.

| No | Modifier | Class Sama | Paket Sama | | Paket Berbeda | |
|----|-----------|------------|------------|--------|---------------|--------|
| | | | Extends | Instan | Extends | Instan |
| 1 | public | ✓ | ✓ | ✓ | ✓ | ✓ |
| 2 | protected | ✓ | ✓ | ✓ | ✓ | |
| 3 | default | ✓ | ✓ | ✓ | | |
| 4 | private | ✓ | | | | |

PRAKTIKUM MANDIRI

1. Peralatan

- a) Perangkat komputer / PC / Laptop / Notebook .
- b) Sistem operasi Windows / Linux (optional Mac OS)
- c) Aplikasi Netbeans

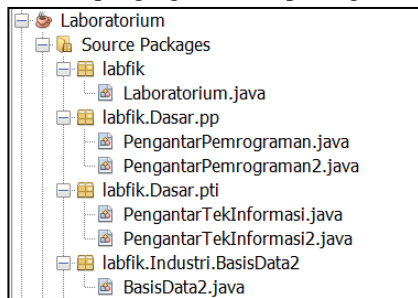
2. Prosedur

- a) Baca dan pahami semua tahapan praktikum dengan cermat.
- b) Gunakan fasilitas yang disediakan dengan penuh rasa tanggung jawab.
- c) Rapihkan kembali setelah menggunakan komputer (mouse, keyboard, kursi, dll)
- d) Perhatikan sikap anda untuk tidak mengganggu rekan praktikan lain
- e) Pastikan diri anda tidak menyentuh sumber listrik.

3. Kegiatan Praktikum

- 1) Buka aplikasi netbeans yang akan digunakan.

- 2) Buat sebuah project java dengan nama "Laboratorium".
- 3) Buatlah pengorganisasian package dan class sesuai dengan susunan berikut.



- 4) Buatlah sebuah project baru dengan nama project "**Modul3**". Kemudian buat sebuah package "**Mahasiswa**" yang didalamnya terdapat 2 class yaitu class "**Utama**" dan class "**Mahasiswa**".

```

public class Mahasiswa {
    public String stb, nama;
}

public class Utama {
    public static void main(String[] args) {
        Mahasiswa mhs = new Mahasiswa();
        mhs.nama="Udin";
        mhs.stb="13020110909";
        System.out.println("Stb : " + mhs.stb);
        System.out.println("Nama : " + mhs.nama);
    }
}
  
```

- Setelah membuat class seperti diatas, ubah akses modifier variabel nama dan stb yang ada pada class Mahasiswa dari public menjadi private. Akan muncul pesan error !
- Untuk penanganan error tersebut, dilakukan dengan menambahkan method setter dan getter pada atribut / variabel yang berakses modifier private pada class **Mahasiswa**.
- Untuk Method Setter, digunakan untuk mengeset/memberikan nilai. Berikut method setter untuk variabel **stb**.

```

public void setStb (String stb){
    this.stb = stb;
}
  
```

- Untuk Method Getter, digunakan untuk mengambil nilai. Berikut method getter untuk variabel **stb**.

```

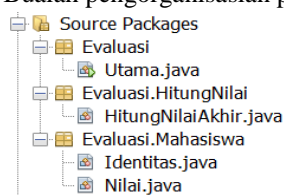
public String getStb (){
    return stb;
}
  
```

- Buatkan pula method setter dan getter untuk variabel **nama**.

- 5) Setelah percobaan selesai, tutup semua perangkat lunak yang telah digunakan.
- 6) Matikan PC dan Rapihkan meja praktikum.

EVALUASI PRAKTIKUM

1. Bualah pengorganisasian package dan class seperti berikut.



2. Setelah mengerjakan soal nomor1, kerjakan program berikut sesuai dengan instruksi.

```
package Evaluasi.Mahasiswa;
public class Identitas {
    buatlah variabel nama dan stambuk bertipe String dan akses modifier private
    buatlah method setter dan getter untuk kedua variabel tersebut
}

package Evaluasi.Mahasiswa;
public class Nilai {
    buatlah variabel tugas1, tugas2, mid dan uas bertipe int dan akses modifier private
    buatlah method setter dan getter untuk variabel-variabel tersebut
}

package Evaluasi.HitungNilai;
public class HitungNilaiAkhir {
    1. buatlah method nilaiTugas bertipe double dengan parameter tugas1, tugas2
       bertipe int. Method ini mengembalikan nilai rata-rata tugas1 dan tugas2.
    2. buatlah method nilaiAkhir bertipe double dengan parameter tugas bertipe
       data double, mid dan uas bertipe data int. Method ini mengembalikan hasil
       dari (tugas*0.4)+(mid*0.3)+(uas*0.3).
}

Package Evaluasi;
public class Utama {
    public static void main (String [] args){
        public double na = 0, tugas = 0;
        berikan nilai pada variabel nama dan stambuk pada class Mahasiswa
        berikan nilai pada variabel tugas1, tugas2, mid dan uas pada class Nilai
        tugas = rata-rata tugas1 dan tugas2 yang dihitung menggunakan method
        nilaiTugas pada class HitungNilaiAkhir dimana tugas1 untuk
        parameter1 dan tugas2 untuk parameter2.
        na = hasil perhitungan tugas, mid dan uas yang dihitung menggunakan
        method nilaiAkhir pada class HitungNilaiAkhir dimana tugas untuk
        parameter1, mid untuk parameter2, dan uas untuk parameter3.
        tampilkan nama dan stb dari class Identitas
        tampilan tugas, mid, uas, dan na.
    }
}
```

Modul 4 : Penginputan & Exception

KOMPETENSI :

Setelah menyelesaikan praktikum pada modul 4, praktikan diharapkan dapat menggunakan semua jenis penginputan dan exception handling pada pemrograman Java.

TUJUAN INTRUKSIONAL KHUSUS :

1. Mengetahui apa itu Scanner, BufferedReader dan JOptionPane.
2. Dapat mengimplementasikan Scanner, BufferedReader dan JOptionPane.
3. Mengetahui apa itu exception handling.
4. Mampu mengimplementasikan penggunaan try-catch, throw, dan throws.

TUGAS PENDAHULUAN

1. Tuliskan dan jelaskan tentang konsep Inheritance !
2. Berikan contoh konsep inheritance dalam kehidupan sehari-hari !
3. Apa itu keyword super pada konsep inheritance ? Jelaskan !
4. Apa itu method overriding ? Jelaskan !
5. Buatlah contoh program yang mengimplementasikan konsep inheritance !

TEORI DASAR:

1. Penginputan

a. Scanner

Scanner pada java merupakan sebuah class yang menyediakan fungsi-fungsi untuk membaca inputan pengguna. Class Scanner ini didapatkan dari package java.util. Scanner dapat membaca inputan pengguna dalam beberapa tipe data.

b. JOptionPane

JOptionPane merupakan sebuah class yang menyediakan fungsi-fungsi dalam menerima inputan dan menampilkan output pengguna dengan memunculkan dialog box. Class ini berada pada package javax.swing.JOptionPane.

c. BufferedReader

BufferedReader adalah salah satu class library di dalam pemrograman java (*java.io.BufferedReader*) sejak JDK Versi 1, yang fungsinya untuk membaca text dengan menggunakan Buffering dengan kemampuan membaca dari skala besar dalam satu waktu. BufferedReader akan membungkus dan membaca karakter input stream menggunakan objek dari class InputStreamReader (*java.io.InputStreamReader*) sebagai argument.

2. EXCEPTION

Exception adalah sebuah peristiwa yang menjalankan alur proses normal pada program. Peristiwa ini biasanya berupa kesalahan(error) dari beberapa bentuk. Beberapa contoh dari exception yaitu *ArrayIndexOutOfBoundsException*, yang terjadi jika kita mencoba mengakses elemen array yang tidak ada, atau *NumberFormatException*, yang terjadi ketika kita mencoba melalui parameter bukan angka dalam method *Integer.parseInt*.

3. MENANGANI EXCEPTION

Untuk menangani exception dalam Java, digunakan blok Try Catch. Try Catch ini adalah salah satu fitur dari Java untuk menangkap eksepsi atau kesalahan.

Bentuk Umum Try Catch

```
try
{<blok_programs>; }
catch(<parameter>)
{<blok_yang_dijalankan_jika_terjadi_eksepsi>;}
finally
{<blok_yang_dijalankan_terakhir>;}
```

4. MELEMPAR EXCEPTION

Disamping menangkap exception, Java juga mengizinkan seorang user untuk melempar sebuah exception. Syntax pelemparan exception cukup sederhana.

```
throw <exception object>;
```

Jika sebuah method dapat menyebabkan sebuah exception namun tidak menangkapnya, maka digunakan keyword *throws*. Aturan ini hanya berlaku pada checked exception.

Berikut ini penulisan syntax menggunakan keyword *throws* :

```
<type> <methodName> (<parameterList>) throws
<exceptionList>
{ <methodBody> }
```

Sebuah method perlu untuk menangkap ataupun mendaftarkan seluruh exceptions yang mungkin terjadi, namun hal itu dapat menghilangkan tipe Error, RuntimeException, ataupun subclass-nya.

PRAKTIKUM MANDIRI

| |
|---|
| 1. Peralatan |
| <ul style="list-style-type: none"> a) Perangkat komputer / PC / Laptop / Notebook . b) Sistem operasi Windows / Linux (optional Mac OS) c) Aplikasi Netbeans |
| 2. Prosedur |
| <ul style="list-style-type: none"> a) Baca dan pahami semua tahapan praktikum dengan cermat. b) Gunakan fasilitas yang disediakan dengan penuh rasa tanggung jawab. c) Rapihkan kembali setelah menggunakan komputer (mouse, keyboard, kursi, dll) d) Perhatikan sikap anda untuk tidak mengganggu rekan praktikan lain e) Pastikan diri anda tidak menyentuh sumber listrik. |
| 3. Kegiatan Praktikum |
| <ul style="list-style-type: none"> 1) Buka aplikasi netbeans yang akan digunakan. 2) Buat sebuah project java dengan nama “Penginputan”. 3) Buatlah pengorganisasian package dan class sesuai dengan susunan berikut. <div data-bbox="375 1032 774 1238" data-label="Diagram"> <pre> graph TD Penginputan --> SourcePackages SourcePackages --> penginputanContohBufferedReader[penginputan.ContohBufferedReader] SourcePackages --> penginputanContohJOptionPane[penginputan.ContohJOptionPane] SourcePackages --> penginputanContohScanner[penginputan.ContohScanner] penginputanContohBufferedReader --> LuasSegitiga1[LuasSegitiga.java] penginputanContohJOptionPane --> LuasSegitiga2[LuasSegitiga.java] penginputanContohScanner --> LuasSegitiga3[LuasSegitiga.java] </pre> </div> 4) Lanjutkan kegiatan praktikum dengan menyelesaikan percobaan berikut. <div data-bbox="347 1272 1420 2022" data-label="Code-Block"> <pre> package penginputan.contohscanner; import java.util.Scanner; public class LuasSegitiga { public double luas (int alas, int tinggi){ return (0.5*alas*tinggi);} public static void main (String [] args){ LuasSegitiga ls = new LuasSegitiga(); int alas, tinggi; Scanner sc = new Scanner (System.in); System.out.print("Masukkan Alas Segitiga : "); alas=sc.nextInt(); System.out.print("Masukkan Tinggi Segitiga : "); tinggi=sc.nextInt(); System.out.println("Luas Segitiga :"+ls.luas (alas, tinggi)); } } package penginputan.contohjoptionpane; import javax.swing.JOptionPane; public class LuasSegitiga { public double luas (int alas, int tinggi){ return (0.5*alas*tinggi);} public static void main (String [] args){ LuasSegitiga ls = new LuasSegitiga(); </pre> </div> |

```

String a, t;
int alas, tinggi;
a = JOptionPane.showInputDialog("Masukkan Alas Segitiga : ");
t = JOptionPane.showInputDialog ("Masukkan Tinggi Segitiga : ");
alas = Integer.parseInt(a);
tinggi = Integer.parseInt(t);
JOptionPane.showMessageDialog(null, ls.luas(alas, tinggi));
}
}

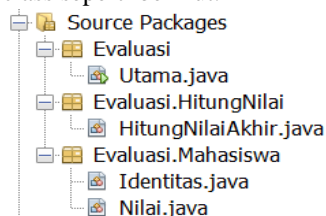
package penginputan.bufferedereader;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
public class LuasSegitiga {
    public double luas (int alas, int tinggi){ return (0.5*alas*tinggi);}
    public static void main (String [] args){
        LuasSegitiga ls = new LuasSegitiga();
        Bufferedreader bf = new BufferedReader(new
        InputStreamReader(System.in));
        String a, t;
        int alas, tinggi;
        System.out.print("Masukkan Alas Segitiga : ");
        a = bf.readLine();
        System.out.print("Masukkan Tinggi Segitiga : ");
        t = bf.readLine();
        alas = Integer.parseInt(a);
        tinggi = Integer.parseInt(t);
        System.out.println("Luas Segitiga :"+ls.luas (alas, tinggi));
    }
}

```

- 5) Tambahkan fitur program diatas yaitu validasi terdapat pesan error menggunakan Joptionpane ketika menginputkan huruf pada penginputan angka.
- 6) Setelah percobaan selesai, tutup semua perangkat lunak yang telah digunakan.
- 7) Matikan PC dan Rapihkan meja praktikum.

EVALUASI PRAKTIKUM

1. Buatlah sebuah project dengan nama project stambuk anda dan buatlah pengorganisasian package dan class seperti berikut.



2. Setelah mengerjakan soal nomor1, kerjakan program berikut sesuai dengan instruksi.

```

package Evaluasi.Mahasiswa;
public class Identitas {
    buatlah variabel nama dan stambuk bertipe String dan akses modifier private
    buatlah method setter dan getter untuk kedua variabel tersebut
}

package Evaluasi.Mahasiswa;
public class Nilai {

```


buatlah variabel **tugas1**, **tugas2**, **mid** dan **uas** bertipe **int** dan akses modifier **private**
buatlah method **setter** dan **getter** untuk variabel-variabel tersebut

```
package Evaluasi.HitungNilai;  
public class HitungNilaiAkhir {  
    1. buatlah method nilaiTugas bertipe double dengan parameter tugas1, tugas2  
       bertipe int. Method ini mengembalikan nilai rata-rata tugas1 dan tugas2.  
    2. buatlah method nilaiAkhir bertipe double dengan parameter tugas bertipe  
       data double, mid dan uas bertipe data int. Method ini mengembalikan hasil  
       dari  $(tugas*0.4)+(mid*0.3)+(uas*0.3)$ .  
}
```

```
Package Evaluasi;  
public class Utama {  
    public static void main (String [] args){  
        public double na = 0, tugas = 0;  
        buatlah penginputan untuk nama, stambuk menggunakan penginputan Scanner.  
  
        buatlah penginputan untuk tugas1, tugas2, mid dan uas  
        menggunakan penginputan BufferedReader.  
  
        berikan nilai pada variabel nama dan stambuk pada class Mahasiswa dari  
        penginputan nama dan stb  
  
        berikan nilai pada variabel tugas1, tugas2, mid dan uas pada class Nilai  
        dari penginputan tugas1, tugas2, mid dan uas.  
  
        tugas = rata-rata tugas1 dan tugas2 yang dihitung menggunakan method  
             nilaiTugas pada class HitungNilaiAkhir dimana tugas1 untuk  
             parameter1 dan tugas2 untuk parameter2.  
        na = hasil perhitungan tugas, mid dan uas yang dihitung menggunakan  
             method nilaiAkhir pada class HitungNilaiAkhir dimana tugas untuk  
             parameter1, mid untuk parameter2, dan uas untuk parameter3.  
        tampilkan nama dan stb dari class Identitas menggunakan JOptionPane  
        tampilkan tugas, mid, uas, dan na menggunakan JOptionPane.  
    }  
}
```

Modul 5 : Inheritance

KOMPETENSI :

Setelah menyelesaikan praktikum pada modul 5, praktikan diharapkan mampu memahami konsep inheritance dan mampu mengimplementasikannya dalam pemrograman Java.

TUJUAN INTRUKSIONAL KHUSUS :

1. Mengetahui apa itu konsep inheritance.
2. Dapat mengimplementasikan konsep inheritance.

TUGAS PENDAHULUAN

1. Tuliskan apa itu konsep polimorfisme!
2. Bagaimana bentuk polimorfisme pada pemrograman java!
3. Apa itu method Overloading ? Jelaskan!
4. Apa itu method Overriding ? Jelaskan !
5. Buatlah program dengan mengimplementasikan konsep polimorfisme!

TEORI DASAR

1. INHERITANCE

Inheritance merupakan pewarisan atribut dan method pada sebuah class yang diperoleh dari class yang telah terdefiniskan. Class yang diwariskan disebut dengan subclass (class anak), sedangkan class yang mewariskan disebut dengan superclass (class induk). Salah satu keuntungan dari inheritance adalah Subclass dapat merubah atau memodifikasi apa yang telah diwariskan oleh Superclass. Untuk membuat class anak diperlukan deklarasi class menggunakan extends.

```
class <nama-class> extends <nama-  
superclass>{  
    //isi program  
}
```

- Super : ini digunakan untuk memanggil konstruktor dari superclass atau menjadi variabel yang mengacu pada superclass.

2. OVERRIDING

Method subclass override terhadap method superclass ketika subclass mendeklarasikan method yang signaturnya serupa ke method dalam superclass. Signature dari method hanyalah informasi yang ditemukan dalam definisi method bagian atas. Signature mengikutkan tipe return, nama dan daftar parameter method tetapi itu tidak termasuk acces modifier dan tipe yang lain dari kata kunci seperti final dan static. Inilah perbedaan dari method overloading. Method overloading secara singkat didiskusikan dalam sub bagian pada kata kunci this. Method ini hanya terjadi jika merupakan implementasi dari pewarisan.

PRAKTIKUM MANDIRI

1. Peralatan

- a) Perangkat komputer / PC / Laptop / Notebook .
- b) Sistem operasi Windows / Linux (optional Mac OS)
- c) Aplikasi Netbeans

2. Prosedur

- a) Baca dan pahami semua tahapan praktikum dengan cermat.
- b) Gunakan fasilitas yang disediakan dengan penuh rasa tanggung jawab.
- c) Rapihkan kembali setelah menggunakan komputer (mouse, keyboard, kursi, dll)
- d) Perhatikan sikap anda untuk tidak mengganggu rekan praktikan lain
- e) Pastikan diri anda tidak menyentuh sumber listrik.

3. Kegiatan Praktikum

- 1) Buka aplikasi netbeans yang akan digunakan.
- 2) Buat sebuah project java dengan nama "Inheritance".
- 3) Lanjutkan kegiatan praktikum dengan menyelesaikan program berikut.

| | | |
|--|---|--|
| | <p>SuperClass</p> <pre>public class BangunDatar { int jumlaHSisi; double luas, keliling; String bangundatar; void Luas() { System.out.println("Luas "+bangundatar+" = "+luas); } void Keliling() { System.out.println("Keliling "+bangundatar+" = "+keliling); } }</pre> | |
| | <p>SubClass</p> <pre>public class Segitiga extends BangunDatar { int alas, tinggi; } public class Persegi extends BangunDatar { int sisi; }</pre> | |
| | <p>Class Utama</p> <pre>public class Utama { public static void main (String[] args) { Segitiga s3 = new Segitiga(); s3.bangundatar="Segitiga"; s3.jumlaHSisi=3; s3.alas=6; s3.tinggi=4; s3.luas=(0.5*s3.alas*s3.tinggi); s3.Luas(); Persegi segi = new Persegi(); segi.bangundatar="Persegi"; segi.sisi=4; segi.luas=segi.sisi*segi.sisi; segi.keliling=4*segi.sisi; segi.Luas(); segi.keliling(); } }</pre> | |
| | <ol style="list-style-type: none"> 4) Tambahkan Method Info() bertipe void pada Class BangunDatar untuk menampilkan kalimat "Ini Adalah Bangun Datar. Ada Banyak Jenis-jenis Bangun Datar." 5) Tambahkan Method Info() bertipe void pada semua SubClass (Class Segitiga dan Class Persegi) untuk menampilkan kalimat "Ini Adalah Bangun Datar Berupa Segitiga/Persegi." 6) Pada Class Utama pada method utama, Tambahkan perintah untuk menampilkan method Info() dari SuperClass (BangunDatar) dan method info() pada SubClass (Segitiga dan Persegi) 7) Setelah percobaan selesai, tutup semua perangkat lunak yang telah digunakan. 8) Matikan PC dan Rapihkan meja praktikum. | |

EVALUASI PRAKTIKUM

1. Buatlah sebuah program dengan tema yang berbeda setiap praktikan. Program yang mengimplementasikan konsep enkapsulasi (Setter & Getter), Inheritance serta Overriding. Program yang dibuat menggunakan penginputan.
2. Sertakan dengan output dan alur program.

Modul 6 : Polimorfisme

KOMPETENSI :

Setelah menyelesaikan praktikum pada modul 6, praktikan diharapkan mampu memahami konsep polimorfisme dan mampu mengimplementasikannya dalam pemrograman Java.

TUJUAN INTRUKSIONAL KHUSUS :

1. Mengetahui apa itu konsep polimorfisme.
2. Dapat mengimplementasikan konsep polimorfisme.

TUGAS PENDAHULUAN

1. Tuliskan tentang abstract class pada pemrograman java !
2. Bagaimana bentuk umum abstrak class pada pemrograman java !
3. Buatlah program dengan mengimpelementasikan konsep abstrak class!
4. Jelaskan tentang interface pada pemrograman java !
5. Tuliskan bentuk umum insterface pada pemrograman java !
6. Buatlah program dengan mengimplementasikan konsep interface !

TEORI DASAR:

1. Polimorfisme

Polimorfisme berarti banyak bentuk. Ada beberapa definisi berbeda tentang polimorpisme yang berkaitan dengan pemrograman berorientasi obyek.

- a. Method Overloading
Overloading adalah mendefinisikan dua atau lebih method di dalam kelas yang sama, dengan nama yang sama, namun dengan deklarasi parameter yang berbeda.
- b. Konstruktor Overloading
Konstruktor overloading merupakan situasi dimana terdapat lebih dari satu konstruktor dengan tipe data parameter yang berbeda.
- c. Method Overriding
Method subclass override terhadap method superclass ketika subclass mendeklarasikan method yang signaturenya serupa ke method dalam superclass. Method ini hanya terjadi jika merupakan implementasi dari **pewarisan**.

PRAKTIKUM MANDIRI

1. Peralatan

- a) Perangkat komputer / PC / Laptop / Notebook .
- b) Sistem operasi Windows / Linux (optional Mac OS)
- c) Aplikasi Netbeans

2. Prosedur

- a) Baca dan pahami semua tahapan praktikum dengan cermat.
- b) Gunakan fasilitas yang disediakan dengan penuh rasa tanggung jawab.
- c) Rapihkan kembali setelah menggunakan komputer (mouse, keyboard, kursi, dll)
- d) Perhatikan sikap anda untuk tidak mengganggu rekan praktikan lain
- e) Pastikan diri anda tidak menyentuh sumber listrik.

3. Kegiatan Praktikum

- 1) Buka aplikasi netbeans yang akan digunakan.
- 2) Buat sebuah project java dengan nama "Polimerfisme".
- 3) Lanjutkan kegiatan praktikum dengan menyelesaikan program berikut.

```
SuperClass
public class BangunDatar {
    int jumlahisi;
    double luas, keliling;
    String bangundatar;

    void info() {
        System.out.println("Ini bangun datar");
    }
}
```

| | | |
|---|--|--|
| | <pre> void Luas() { System.out.println("Luas "+bangundatar+" = "+luas); } void Keliling() { System.out.println("Luas "+bangundatar+" = "+keliling); } void Luas(int sisi) { luas = sisi*sisi; System.out.println("Luas persegi : "+luas); } void Luas(int a, nt t) { luas = a*t*0.5; System.out.println("Luas Segitiga : "+luas); } </pre> | |
| | <pre> public class Prsegi extends BangunDatar { int sisi; void info() { Sytem.out.println("Ini bangun persegi"); } } </pre> | |
| | <p>Class Utama</p> <pre> public clas Utama { public static void main(String[] args) { Segitiga s3 = new Seitisa(); S3.bangundatar="Segitiga"; S3.jumlahsisi=3; S3.alas=6; S3.tinggi=4; S3.luas= (0.5*s3.alas*s3.tinggi); S3.Luas(); Persegi segi = new Persegi(); segi.bangundatar="Persegi"; segi.sisi=5; segi.luas=segi.sisi*segi.sisi; segi.keliling=4*segi.sisi; segi.Luas(); segi.Keliling(); } } </pre> | |
| <p>4) Pada SuperClass (BangunDatar) untuk semua atribut berikan akses modifier private. Jika terjadi error silahkan perbaiki program tersebut tanpa merubah akses modifier.</p> <p>5) Kemudian ubah semua akses modifier pada method di Class BangunDatar menjadi protected.</p> <p>6) Setelah percobaan selesai, tutup semua perangkat lunak yang telah digunakan.</p> <p>7) Matikan PC dan Rapihkan meja praktikum.</p> | | |

EVALUASI PRAKTIKUM

1. Buatlah sebuah program dengan tema yang berbeda setiap praktikan. Program yang mengimplementasikan konsep polimerfisme.

Modul 7 : Abstract Class dan Interface

KOMPETENSI :

Setelah menyelesaikan praktikum pada modul 7, praktikan diharapkan mampu memahami abstract class & interface serta mampu mengimplementasikannya dalam pemrograman Java.

TUJUAN INTRUKSIONAL KHUSUS :

1. Mengetahui apa itu abstract class dan interface.
2. Dapat mengimplementasikan abstract class dan interface.

TUGAS PENDAHULUAN

1. Tuliskan apa itu ArrayList dan HashMap!
2. Bagaimana bentuk umum ArrayList dan HashMap pada pemrograman java!
3. Buatlah program dengan mengimpelementasikan ArrayList!
4. Buatlah program dengan mengimpelementasikan HashMap!

TEORI DASAR:

1. Abstract Class

Abstract class adalah class yang terletak pada posisi tertinggi pada hirarki class. Class ini digunakan sebagai basis bagi penurunan kelas lainnya, sehingga abstract class tidak dapat dinstansiasi secara langsung menjadi object. Suatu abstract class dapat mengandung method konkrit dan/atau method abstract. Abstract method adalah suatu method yang tidak mempunyai badan method, hanya berupa nama method dan parameter inputan method. Ketika suatu class abstract diturunkan, maka subclass harus mengimplementasikan semua method abstract pada superclass. Jika tidak mengimplementasikan semua method abstract maka subclass harus dideklarasikan sebagai abstract class.

Deklarasi Abstract Class

```
AksesModifier abstract class NamaClass {
    //statement
}
```

Contoh :

```
public abstract class Hitung{
    //statement
}
```

Deklarasi Abstract Method

```
AksesModifier abstract TipeData NamaMethod(parameter)
```

Contoh :

```
public abstract void HasilHitung ();
```

2. Interface

Interface adalah kumpulan method yang hanya memuat deklarasi dan struktur method tanpa detail implementasinya. Sedangkan detail dari method berada pada class yang mengimplementasikan interface tersebut. Interface digunakan bila anda ingin mengaplikasikan suatu method yang spesifik, yang tidak diperoleh dari proses inheritance. Method yang dapat dibuat pada interface adalah method abstract.

PRAKTIKUM MANDIRI

1. Peralatan

- a) Perangkat komputer / PC / Laptop / Notebook .
- b) Sistem operasi Windows / Linux (optional Mac OS)
- c) Aplikasi Netbeans

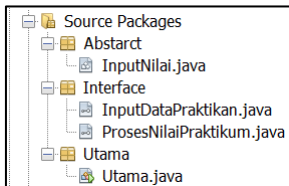
2. Prosedur

- a) Baca dan pahami semua tahapan praktikum dengan cermat.
- b) Gunakan fasilitas yang disediakan dengan penuh rasa tanggung jawab.
- c) Rapihkan kembali setelah menggunakan komputer (mouse, keyboard, kursi, dll)

- d) Perhatikan sikap anda untuk tidak mengganggu rekan praktikan lain
- e) Pastikan diri anda tidak menyentuh sumber listrik.

3. Kegiatan Praktikum

- 1) Buka aplikasi netbeans yang akan digunakan.
- 2) Buat sebuah project java dengan nama "Modul7" dengan struktur project sebagai berikut.



- 3) Lanjutkan kegiatan praktikum dengan menyelesaikan program berikut.

Abstract Class

```

package Abstarct;
public abstract class InputNilai {
    public int acc1, acc2, acc3, acc4, mid, uas;
    public abstract void inputNilaiAcc();
    public abstract void inputNilaiUjian();
    public void tampilNilai() {
        System.out.println("Nilai Acc1 : " + acc1);
        System.out.println("Nilai Acc2 : " + acc2);
        System.out.println("Nilai Acc3 : " + acc3);
        System.out.println("Nilai Acc4 : " + acc4);
        System.out.println("Nilai MID : " + mid);
        System.out.println("Nilai Final : " + uas);
    }
}
  
```

Interface

```

package Interface;
public interface ProsesNilaiPraktikum {
    public float nilaiTugas1();
    public float nilaiTugas2();
    public float nilaiAkhir(float tugas1, float tugas2, float
mid, float uas);
}

package Interface;
public interface DataPraktikan {
    public void InputNamaStb();
    public void InputPraktikum();
}
  
```

Class Utama

```

package Utama;
public class Utama extends InputNilai implements DataPraktikan,
ProsesNilaiPraktikum {
    String nama, stb, praktikum;
    Scanner sc;
    BUAT KONSTRUKTOR CLASS UTAMA YANG BERISI PEMBUATAN OBJEK
DARI Scanner sc.
    @Override
    public void inputNilaiAcc(){
  
```

```

        INPUT NILAI UNTUK ATRIBUT acc1, acc2, acc3, acc4 PADA
ABSTARCT CLASS InputNilai
    menggunakan objek sc dari Scanner.
}
@Override
public void inputNilaiUjian() {
    INPUT NILAI UNTUK ATRIBUT mid DAN uas PADA ABSTARCT CLASS
InputNilai -
    menggunakan objek sc dari Scanner.
}
@Override
public void InputNamaStb() {
    INPUT NILAI UNTUK ATRIBUT nama DAN stb menggunakan objek
sc dari Scanner.
}
@Override
public void InputPraktikum() {
    INPUT NILAI UNTUK ATRIBUT praktikum menggunakan objek sc
dari Scanner.
}
@Override
public float nilaiTugas1() {
    mengembalikan hasil dari : (acc1 + acc2) / 2;}
@Override
public float nilaiTugas2() {
    mengembalikan hasil dari : (acc3 + acc4) / 2;}
@Override
public float nilaiAkhir(float tugas1, float tugas2, float
mid, float uas) {
    mengembalikn hasil dari :
    (((tugas1+tugas2)/2)*0.3)+(mid*0.3)+(uas* 0.4);}
public static void main(String[] args) {
    InputNamaStb();
    InputPraktikum();
    inputNilaiAcc();
    inputNilaiUjian();
    float tugas1 = nilaiTugas1();
    float tugas2 = nilaiTugas2();
    float na = nilaiAkhir(tugas1, tugas2, mid, uas);
    System.out.println("Nilai Akhir : " + na);
}
}

```

- 4) Setelah percobaan selesai, tutup semua perangkat lunak yang telah digunakan.
- 5) Matikan PC dan Rapihkan meja praktikum.

EVALUASI PRAKTIKUM

1. Buatlah contoh program yang berbeda setiap praktikan. Program yang mengimplementasikan 2 interface dan 1 abstract class atau lebih, dilengkapi dengan penginputan user dan proses aritmatika, tambahkan validasi dalam penginputan angka.

Modul 8 : ArrayList & HashMap

KOMPETENSI :

Setelah menyelesaikan praktikum pada modul 8, praktikan diharapkan mampu memahami konsep ArrayList dan HashMap serta mampu mengimplementasikannya dalam pemrograman Java.

TUJUAN INTRUKSIONAL KHUSUS :

1. Mengetahui apa itu konsep ArrayList dan HashMap.
2. Dapat mengimplementasikan konsep ArrayList dan HashMap.

TEORI DASAR

1. ArrayList

ArrayList merupakan collection yang menjadi bagian dari **Java.Util.ArrayList**. Berbeda dengan array biasa, ArrayList memiliki sejumlah operasi yang lebih lengkap dan mudah digunakan dibandingkan dengan array biasa. ArrayList dibentuk untuk menutupi kekurangan yang ada pada Array, contohnya dalam Array sudah diberikan batasan tampungan data sedangkan ArrayList tidak memiliki batasan tampungan dimana dapat menambah data baru secara dinamis tanpa harus menentukan ukurannya di awal. Berikut ini merupakan bentuk dasar listing dari ArrayList :

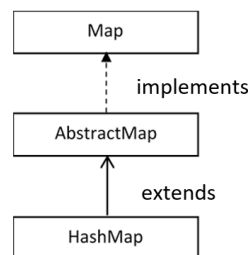
```
ArrayList< Object > nama ArrayList = new ArrayList<>();
```

Contoh :

```
ArrayList< Integer > data = new ArrayList<>();
```

2. HashMap

Class HashMap merupakan class turunan dari class AbstractMap dan implementasi dari interface Map.



HashMap adalah sebuah class yang berisi sekumpulan pasangan nilai (value) dan kunci (key). Pada penjelasan HashMap, dapat disimpulkan bahwa HashMap memiliki 2 atribut yaitu **value** dan **key**. value bisa dalam bentuk string, integer, boolean, float, double, dan objek. Sedangkan untuk key biasanya dalam bentuk string dan integer. Operasi yang dapat dilakukan terhadap HashMap antara lain :

- **Put()**, untuk memberikan nilai
- **Get()**, untuk mengambil nilai
- **Remove()**, untuk menghapus salah satu nilai
- **Clear()**, untuk menghapus semua nilai
- **Replace()**, untuk mengubah nilai

Library yang digunakan untuk HashMap berada pada **Java.Util.HashMap**. Ketika ingin menggunakan HashMap, maka kita harus mengimportnya terlebih dahulu. Berikut ini merupakan bentuk dasar kodingan HashMap :

```
HashMap< TD _key , TD_value > Nama_HashMap = new HashMap< TD_key ,
```

Contoh :

```
HashMap< Integer , Strnig > Hari = new HashMap< Integer , String
```

PRAKTIKUM MANDIRI**1. Peralatan**

- a) Perangkat komputer / PC / Laptop / Notebook .
- b) Sistem operasi Windows / Linux (optional Mac OS)
- c) Aplikasi Netbeans

2. Prosedur

- a) Baca dan pahami semua tahapan praktikum dengan cermat.
- b) Gunakan fasilitas yang disediakan dengan penuh rasa tanggung jawab.
- c) Rapihkan kembali setelah menggunakan komputer (mouse, keyboard, kursi, dll)
- d) Perhatikan sikap anda untuk tidak mengganggu rekan praktikan lain
- e) Pastikan diri anda tidak menyentuh sumber listrik.

3. Kegiatan Praktikum

- 1) Buka aplikasi netbeans yang akan digunakan.
- 2) Buat sebuah project java dengan nama "Modul8".
- 3) Lanjutkan kegiatan praktikum dengan menyelesaikan program berikut.
- 4) Buatlah sebuah class dengan nama "**Mahasiswa**" dan isi dengan listing di bawah :

```
import java.util.HashMap;
public class Mahasiswa {
    public static void main(String[] args) {
        // Membuat objek hashmap
        HashMap<Integer, String> name = new
        HashMap<Integer,String>();
        // mengisi nilai ke objek days
        name.put(1, " Petra");
        name.put(2, " Aarim");
        System.out.println("Isi objek name : " + name);
        System.out.println("Asistenurut ke-2 : " +
        name.get(2));
    }
}
```

- 5) Buat file baru kemudian simpan dengan nama "Gudang" dan isi dengan listing di bawah :

```
import java.util.ArrayList;
public class Gudang {
    public static void main(String[] args) {
        ArrayList barang = new ArrayList();
        barang.add("Monitor");
        barang.add("Keyboard");
        barang.add("Meja");
        barang.remove("Meja");
        System.out.println(barang);
        System.out.println("isi Gudang Sebanyak "+
        barang.size() +" item");
    }
}
```

- 6) Setelah percobaan selesai, tutup semua perangkat lunak yang telah digunakan.
- 7) Matikan PC dan Rapihkan meja praktikum.

EVALUASI PRAKTIKUM

1. Buatlah contoh program yang mengimplementasikan HashMap dengan menggunakan penginputan.
2. Buatlah contoh program array list bertipe data class yang merupakan class model. Class model adalah class yang didalamnya terdapat pendeklarasian variable bertipe private dilengkapi dengan setter dan getter.

LEMBAR EVALUASI PRAKTIKUM

EVALUASI PRAKTIKUM 1

| Stamp/Date | Result/Point | Signature/Name |
|------------|--------------|----------------|
| | | |

EVALUASI PRAKTIKUM 2

| Stamp/Date | Result/Point | Signature/Name |
|------------|--------------|----------------|
| | | |

EVALUASI PRAKTIKUM 3

| Stamp/Date | Result/Point | Signature/Name |
|------------|--------------|----------------|
| | | |

EVALUASI PRAKTIKUM 4

| Stamp/Date | Result/Point | Signature/Name |
|------------|--------------|----------------|
| | | |

Dosen :

Asisten 1 :

Asisten 2 :

RANGE NILAI

| | |
|----|------------|
| A | = 85 – 100 |
| A- | = 80 – 84 |
| B+ | = 75 – 79 |
| B | = 70 – 74 |
| B- | = 65 – 69 |
| C+ | = 60 – 64 |
| C | = 50 – 59 |