Modul Praktikum

Pemrograman Berorientasi Objek

Dosen Pengampu:

Didik Kurniawan, M.Kom.

Penyusun:

Firmansyah

Feri Krisnanto

Mei Rusfandi

Ichwan Almaza

Muammar Rizki F.I.

Faiz Azmi Rekatama

Garnies Hafitma Yora

Edisi 1 (2017)

Laboratorium Komputasi Dasar

Jurusan Ilmu Komputer

FMIPA Universitas Lampung

Deskripsi Mata Kuliah

Paradigma pengembangan perangkat lunak berorientasi objek adalah salah satu metode/pendekatan dalam membangun suatu sistem, dan aplikasi perangkat lunak dengan mengikuti model-model objek seperti dalam dunia nyata. Pada kuliah ini akan dijelaskan prinsip utama secara konprehensif dan kasus-kasus nyata sederhana pengembangan suatu perangkat lunak. Seperti bahasa pemrograman OO, analisis dan peracangan berbasis OO, dan implementasi dengan menggunakan bahasa OO.

Pokok Bahasan: Paradigma bahasa pemrograman. Konsepsi pemrograman OOP, pengertian objek, class, method, pesan, pewarisan,konstruktor dan Polimorphisme. Analisis dan perancangan berbasis OO, perancangan menggunakan tool UML, seperti class diagram, use case, activity diagram, interaksi diagram. Tool pengembangan perangkat lunak dan bahasa pemrograman OO, seperti UML, java, C++.

Tujuan Perkuliahan

Mengerti perbedaan pemrograman OOP dan non OOP, pengembangan perangkat lunak berbasis OO. Mengerti dan mampu mengerti mengenai konsep OOP, objek, pesan, class, inheritance, overriding, method dapat membuat CRC untuk suatu penyelesaian kasus nyata sederhana. Mampu merancang dan membangun perangkat lunak berbasis pada modeling paradigma UML, dan mengimplementasikan dengan bahasa pemrograman OO, seperti java dan C++.

Deskripsi Isi Perkuliahan

Pada kuliah ini mahasiswa mempraktekkan konsep-konsep OOP dalam penyelesaian masalah-masalah dengan program komputer.

Daftar Isi

Pertemuan 1

Pengenalan Mata Kuliah Pemrograman Berorientasi Objek

Tujuan Instruksional **:** Pengantar
Mengidentifikasi bagian dasar dari program java

- Membedakan mana yang termasuk ke dalam java literals, tipe data dasar, tipe variabel, pengidentifikasian dan operator.
- Mengembangkan program java sederhana menggunakan konsep pembelajaran pada bab ini.
- Menganalisa program java pertama

Kompetensi yang Diharapkan **:**

1. Mahasiswa diharapkan dapat mengetahui pemrograman berorientasi objek.
2. Mahasiswa diharapkan dapat membuat program berorientasi objek.

Waktu Pertemuan : 100 menit

Latar Belakang

Pada bagian ini, kita akan mendiskusikan mengenai bagian dasar pemrograman java. Kita akan memulai dengan mencoba menjelaskan bagian dasar dari program Hello.java yang telah diperkenalkan pada bab sebelumnya. Kita juga akan mendiskusikan beberapa pedoman cara menulis script atau petunjuk penulisankode dalampenulisan program lebih efektif dan mudah dibaca.

Percobaan 1

Program 1-1

```
public class RelasiDemo {
public static void main(String[] args) {      int i = 37;
int j = 42;  int k = 42;      System.out.println("Nilai
variabel...");
   System.out.println(" i = " + i);
   System.out.println(" j = " + j);
   System.out.println(" k = " + k);
   System.out.println("Lebih besar dari...");
   System.out.println(" i > j = " + (i > j));
   System.out.println(" k > j = " + (k > j));
   System.out.println("Lebih besar dari atau sama dengan...");
   System.out.println(" i >= j = " + (i >= j));
   System.out.println(" j >= i = " + (j >= i));
   System.out.println("Lebih kecil dari...");
   System.out.println(" i < j = " + (i < j));
   System.out.println("Lebih kecil dari atau sama dengan...");
   System.out.println(" j <= i = " + (j <= i));
   System.out.println(" k <= j = " + (k <= j));
   System.out.println("Sama dengan...");
   System.out.println(" k == j = " + (k == j));
   System.out.println("Tidak sama dengan...");
System.out.println(" k != j = " + (k != j));  }
}
```

Tuliskan outputnya:

Percobaan 2

Program 1-2

```
public class OperatorLogika {
    public static void main(String []args){
        boolean a=true,b=true;
        System.out.println("A\tB\tA && B\tA OR B\tA XOR
B\tNOT A");
         for (int i=0;i<2;i++) {
for (int j=0;j<2;j++){
                System.out.println(a+"\t"+b+"\t"+ (a &&
b)+"\t"                          +(a || b)+"\t"+(a ^ b)
+"\t"+(!a));                         b=!b;
            }
a=!a;
        }
    }
}
```

Tuliskan Outputnya

Percobaan 3

Program 1-3

```
public class Operator
{
    public int a,b;
boolean c,d,e;
double f,g;
    public void testOperator()
    {
a=19;
b=2;
c=true;
d=false;
e=!c;
g=0.2;          if
(a%2==1)
b=a++;
else
b=++a;
        f=(c&&!d||e)? a/g : a/(g+1);
System.out.println("a : "+a);
        System.out.println("b : "+b);
        System.out.println("e : "+e);
System.out.println("f : "+f);
    }
    public static void main(String args[])
    {
        Operator oper= new Operator();
        oper.testOperator();
        System.out.println(oper.checkMonth(9, 2014));
    }
    public int checkMonth(int month,int year)
    {
        int day=0;
if (month<8){
if(month%2==1)
day=31;          else
if(month==2)
            day = (year%4==0)? 29: 28;
else
            day=30;
} else {
if(month%2==1)
day=30;             else
day=31;
        }
        return day;
    }
}
```

Jelaskan Logika pada method checkMonth:

Soal Latihan

Diberikan tiga angka, tuliskan program yang menghasilkan output angka dengan nilai terbesar diantara tiga angka tersebut. Gunakan operator kondisi ?: yang telah kita pelajari sebelumnya (HINT: Anda akan perlu menggunakan dua operator ternary ?: untuk memecahkan permasalahan ini). Sebagai contoh , diberikan angka 10, 23 dan 5. Program anda akan menghasilkan output:

number 1 = 10
number 2 = 23
number 3 = 5
Nilai tertingginya adalah angka = 23

Tuliskan kode program anda:

Pertemuan 2

Pengenalan Program Netbeans dan Eclipse

Tujuan Instruksional	**:**
Pokok bahasan ini menjelaskan tentang aplikasi Netbeans dan Eclipse.	
Kompetensi yang Diharapkan	**:**
Mahasiswa diharapkan dapat mengenal dan memahami aplikasi Netbeans dan Eclipse.	
Waktu Pertemuan	**:** 100 menit

NetBeans NetBeans IDE (Integrated Development Environment) adalah sebuah lingkungan pengembangan terintegrasi yang tersedia untuk Windows, Mac, Linux, dan Solaris. Proyek NetBeans terdiri dari open-source IDE dan platform aplikasi, yang memungkinkan pengembang untuk secara cepat membuat web, enterprise, desktop, dan aplikasi mobile menggunakan platform Java, serta JavaFX, PHP, JavaScript dan Ajax, Ruby dan Ruby on Rails, Groovy dan Grails, dan C/C++. Proyek NetBeans didukung oleh komunitas pengembang yang ekstensif dan menawarkan dokumentasi dan sumberdaya pelatihan serta beragam pilihan plugin pihak ketiga.

Pada pembahasan ini digunakan NetBeans versi 7.1. IDE NetBeans 7.1 memperkenalkan dukungan untuk JavaFX 2.0 dengan mengaktifkan siklus kompilasi/debug/profil pengembangan secara penuh untuk aplikasi jafaFX 2.0. Rilis ini juga menyediakan perangkat tambahan Swing GUI Builder secara signifikan, dukungan CSS3, dan perangkat untuk visual debugging untuk Swing dan antarmuka pengguna untuk JavaFX. Tambahan termasuk dukungan Git yang terintegrasi ke dalam IDE, fitur baru PHP debugging, beberapa perbaikan pada JavaEE dan Maven, dan banyak lainnya.

NetBeans 7.1 tersedia dalam bahasa Inggris, Brasil, Portugis, Jepang, Rusia dan Cina.

Instalasi NetBeans

Untuk dapat menggunakan NetBeans, kita harus menginstalasi NetBeans dan JDK. Keduanya dapat di download secara gratis di http://www.netbeans.com/ dan http://www.oracle.com/. Supaya lebih mudah dalam menginstall, install JDK terlebih dulu baru kemudian install NetBeans.

Tampilan NetBeans
Setelah instalasi selesai kita dapat menggunakan NetBeans dengan tampilan pembuka seperti pada gambar berikut ini:

Tampilan NetBeans IDE seperti terlihat pada gambar berikut ini:

Membuat Project

Untuk membuat project baru kita dapat memilih menu File>>New Project, sehingga tampil jendela seperti pada gambar berikut ini:

Karena kita akan membuat aplikasi java, maka pada Categories pilih Java dan pada Projects pilih Java Application. Kemudian klik tombol Next. Untuk latihan membuat project, pada Project Name isi dengan Latihan dan kosongkan tanda check padaCreate Main Class dan Set as Main Project, sehingga tampilan seperti pada gambar berikut ini:

Tentukan juga di mana project akan disimpan dengan mengisikannya pada Project Location. Kemudian klik tombol Finish. Selanjutnya klik tanda tambah (+) pada Source Packages, kemudian kita tambahkan package pada project yang kita buat dengan cara klik kanan pada Source Packages dan pilih New >> Java Package... seperti pada gambar berikut ini:

Pada Package Name isi dengan APLIKASIKONSOL dan klik tombol Finish.

Setelah nama package selesai kita buat, klik kanan pada nama package tersebut dan pilih New, kemudianJava Class... seperti gambar di bawah ini:

Pada Class Name isi Hello dan pastikan nama Package adalah APLIKASIKONSOL, sehingga seperti pada gambar berikut ini :

Di dalam class Hello ketik kode program di bawah ini:

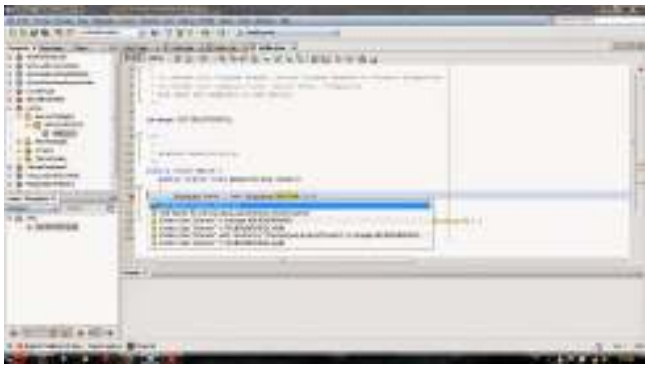
```
public static void main(String args[])
{ Scanner baca = new Scanner(System.in);
String nama; System.out.print("Nama : ");
nama = baca.nextLine();
System.out.println("Hello "+nama+", ini aplikasi console menggunakan NetBeans");
}
```

Lihat gambar:

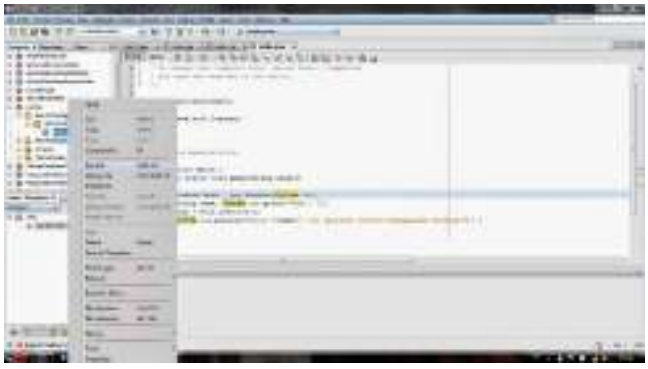
Pada kode program di atas akan tampil keterangan kesalahan, karena penggunaan class Scanner harus mengimport java.util.Scanner. Untuk mengimportnya dapat dilakukan dengan menambahkan statement berikut ini:

```
import java.util.Scanner
```

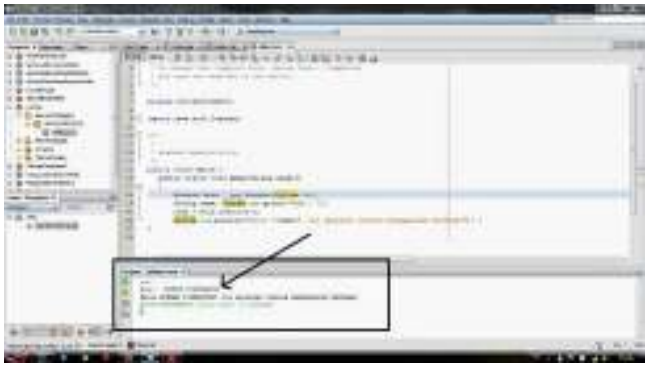
di bawah nama package, atau dengan mengklik gambar bola lampu di sebelah kirinya dan memilih Add import for java.util.Scanner seperti pada gambar berikut ini:



Untuk menjalankan program yang telah kita buat, klik kanan pada nama class kemudian pilih Run File (atau tekan tombol Shift dan F6) seperti pada gambar berikut ini:



Jika berhasil maka akan tampil pesan berikut dibawahnya :



Sebagai pengenalan Eclipse adalah sebuah IDE (Integrated Development Environment) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (platform-independent). Berikut ini adalah sifat dari Eclipse:

§ Multi-platform: Target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.

§ Mulit-language: Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya.

§ Multi-role: Selain sebagai IDE untuk pengembangan aplikasi, Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya.

Eclipse pada saat ini merupakan salah satu IDE favorit dikarenakan gratis dan open source, yang berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan plug-in (sumber : wikipedia).

Lalu bagaimana langkah-langkah untuk dapat melakukan pemrograman dengan Eclipse ini?

Pertama kita unduh terlebih dahulu perangkat lunak ini di situs resmi Eclipse : www.eclipse.org .
Kemudian lanjutkan ke halaman download maka akan dibawa ke halaman dengan tampilan berikut :

Pilih file yang dikehendaki :

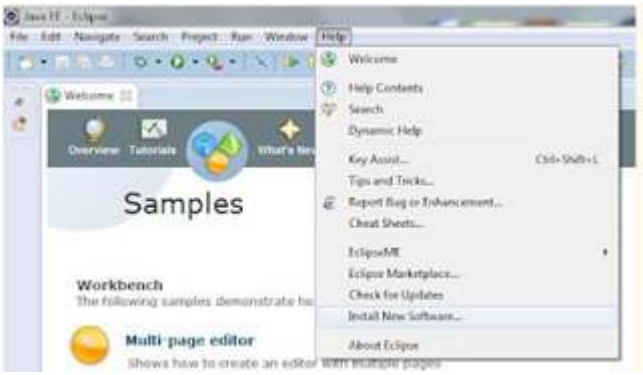
Klik link di sebelah logo download.

Eclipse yang saya pilih adalah Eclipse Juno dan file instalasinya portabel sehingga file yang diunduh berupa format .zip yang hanya tinggal di ekstrak pada direktori file tertentu pada komputer

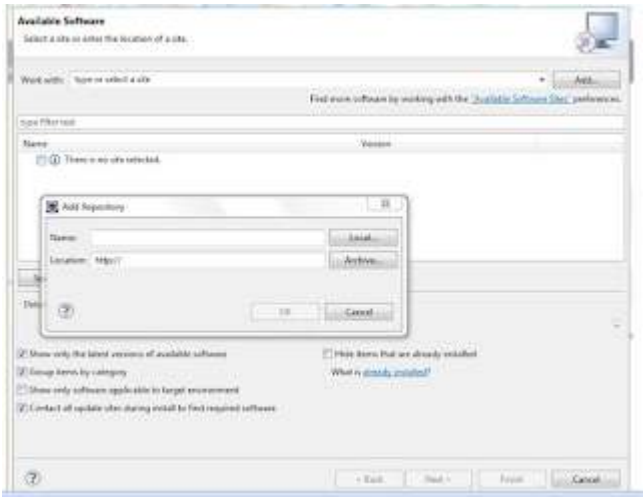
Setelah terekstrak, buka folder eclipse, lalu jalankan eclipse.exe

sebagai tambahan , diperlukan Java Development Kit (JDK) dan Eclipse ME sebagai plugin pemrograman mobile java untuk mendapatkan JDK dapat diunduh dari http://www.oracle.com/technetwork/java/javase/downloads/jdk-6u25-download-346242.html sesuaikan dengan sistem operasi pada komputer. dan untuk Eclipse ME dapat diperoleh dari : www.eclipseme.org

Install JDK pada komputer dengan mengikuti petunjuk instalasi pada installernya. Sedangkan untuk Eclipse ME diinstall dengan cara menjalankan Eclipse terlebih dahulu. Setelah Eclipse terbuka tampilannya gunakan menu help - Install New Software



Akan dibawa ke tampilan sebagai berikut



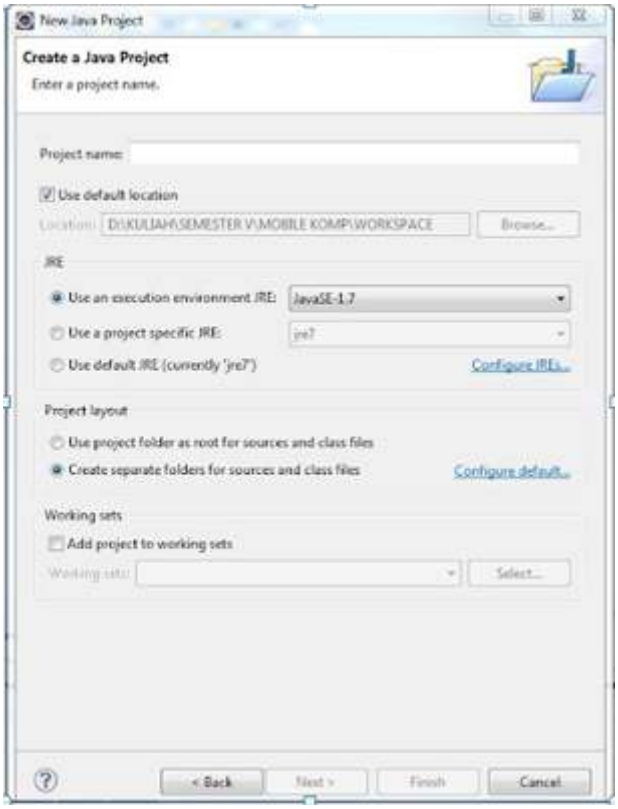
Klik tombol Add.. akan muncul window Add Repository, klik Archive, cari file EclipseME yang sudah diunduh. Kemudian OK.

Ikuti perintah instalasi dan tunggu prosesnya. Setelah selesai maka akan diminta untuk merestart program Eclipse, pilih Yes / OK.

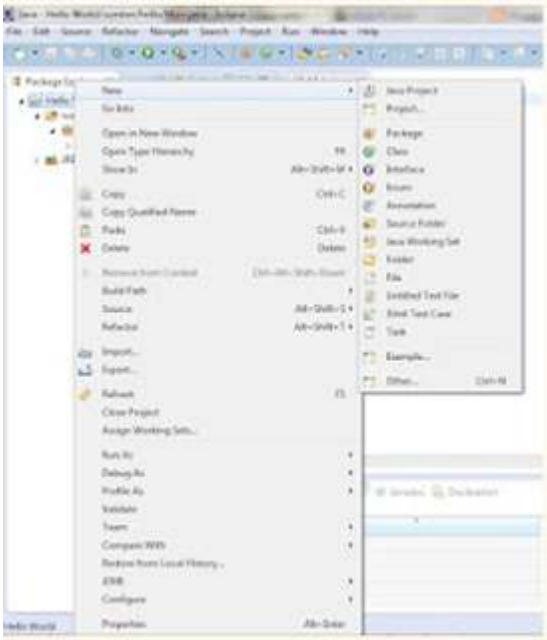
Setelah eclipse terbuka lagi coba untuk membuat program awalan "Hello World"

Buat project baru dengan melalui menu File - New - Project

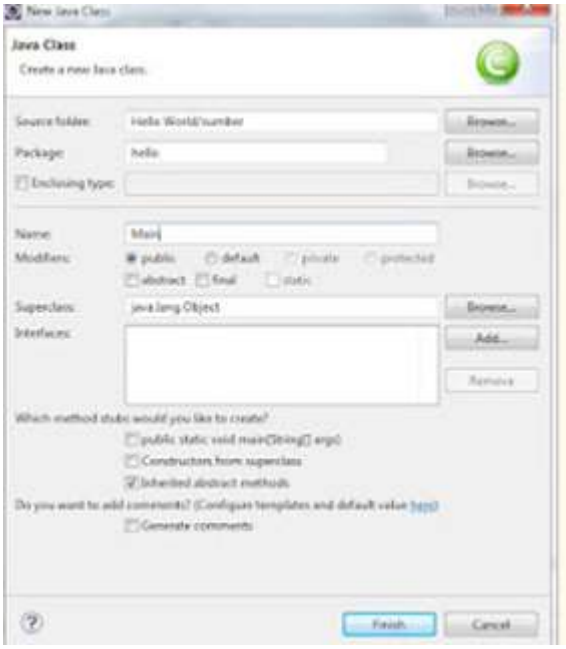
Pilih Java Project dan isi form project baru.



Tambahkan folder source dan package melalui menu berikut :



Setelah proses selesai tambahkan new class (File - New - Class)



Ketikkan syntax Hello World

Kemudian jalankan program tersebut, dan muncullah sebagai berikut:

Pertemuan 3

Pengenalan Pemrograman Berorientasi Objek

Tujuan Intruksional	**:**
Pokok Bahasan ini menjelaskan tentang bagaimana cara membuat gerbang logika.	
Kompetensi Yang Diharapkan	**:**
1. Mahasiswa diharapkan telah memahami apa itu gerbang logika.	
Waktu Pertemuan	**:** 100 Menit

Jika anda telah biasa membuat program secara prosedural, yakni menulis program-program dari baris pertama sampai dengan baris terakhir secara berurutan, konsep pemrograman berbasis objek mungkin sedikit susah dipahami. Dalam tutorial pertama tentang OOP ini, kita akan membahas dulu pengertian pemrograman berbasis objek.

Pengertian Pemrograman Berorientasi Objek

Pemrograman Berorientasi Objek atau Object Oriented Programming (OOP) adalah sebuah tata cara pembuatan program (programming paradigm) dengan menggunakan konsep "objek" yang memiliki data (atribut yang menjelaskan tentang objek) dan prosedur (function) yang dikenal dengan method. (http://en.wikipedia.org/wiki/Object-oriented_programming)

Dalam pengertian sederhananya, OOP adalah konsep pembuatan program dengan memecah permasalahan program dengan menggunakan objek. Objek dapat diumpamakan dengan 'fungsi khusus' yang bisa berdiri sendiri. Untuk membuat sebuah aplikasi, berbagai objek akan saling bertukar data untuk mencapai hasil akhir.

Berbeda dengan konsep fungsi atau 'function' di dalam pemrograman, sebuah objek bisa memiliki data dan function tersendiri. Setiap objek ditujukan untuk mengerjakan sebuah tugas, dan menghasilkan nilai akhir untuk selanjutnya dapat ditampilkan atau digunakan oleh objek lain.

Fungsi Pemrograman Berorientasi Objek dalam PHP

PHP bukan bahasa pemrograman yang 'murni' berbasis objek seperti Java. Bahkan, konsep OOP dalam PHP baru hadir dalam PHP versi 4, dan disempurnakan oleh PHP versi 5. Dengan kata lain,OOP di PHP merupakan 'fitur tambahan'. Anda bisa membuat situs web dengan PHP tanpa menggunakan objek sama sekali.

Dalam studi pemrograman, pembuatan program dalam PHP tanpa menggunakan objek disebut juga dengan pemrograman prosedural atau pemrograman fungsional. Dikenal dengan pemrograman prosedural, karena kita memecah kode program menjadi bagian-bagian atau fungsi-fungsi kecil, kemudian menyatukannya untuk menghasilkan nilai akhir.

Dengan membuat program secara prosedural, aplikasi bisa dibuat dengan cepat dan mudah dipelajari jika dibandingkan dengan pemrograman berbasis objek (bagi anda yang pernah mempelajari Java, tentu telah 'melewati' hal ini). Keuntungan pemrograman berbasis objek baru terasa ketika program tersebut telah 'besar' atau kita bekerja dengan tim untuk membagi tugas. Konsep 'objek' untuk memisahkan program menjadi bagian-bagian yang berdiri sendiri akan memudahkan dalam membuat program.

Saya tidak akan panjang lebar menjelaskan tentang keuntungan atau kerugian menggunakan OOP. Sebagai programmer web, OOP adalah salah satu makanan wajib. Pembuatan website modern saat ini akan lebih mudah jika menggunakan template kode program yang dikenal dengan framework. Daripada kita membuat situs mulai dari awal, menggunakan framework akan mempercepat proses kerja. Dan framework PHP hampir semuanya dibuat menggunakan OOP.

Pemrograman berbasis objek tidak hanya berisi 'object'. Selanjutnya kita akan belajar OOP PHP, dan pertama kita akan membahas tentang pengertian class, object, property dan method. Keempat 'keyword' inilah yang menjadi pondasi dasar dari Pemrograman Berorientasi Objek. Selain pengertian, kita juga akan mempelajari cara penulisannya dengan PHP.

Untuk memudahkan pemahaman dan agar sejalan dengan istilah aslinya, saya tetap menggunakan istilah bahasa inggris untuk kata kunci PHP, seperti: class, object, property dan method.

Pengertian Class dalam Pemrograman Berbasis Objek

Class adalah 'cetak biru' atau 'blueprint' dari object. Class digunakan hanya untuk membuat kerangka dasar. Yang akan kita pakai nantinya adalah hasil cetakan dari class, yakni object.

Sebagai analogi, class bisa diibaratkan dengan laptop atau notebook. Kita tahu bahwa laptop memiliki ciri-ciri seperti merk, memiliki keyboard, memiliki processor, dan beberapa ciri khas lain yang menyatakan sebuah benda tersebut adalah laptop. Selain memiliki ciri-ciri, sebuah laptop juga bisa dikenakan tindakan, seperti: menghidupkan laptop atau mematikan laptop.

Class dalam analogi ini adalah gambaran umum tentang sebuah benda. Di dalam pemrograman nantinya, contoh class seperti: koneksi_database dan profile_user.

Di dalam PHP, penulisan class diawali dengan keyword class, kemudian diikuti dengan nama dari class. Aturan penulisan nama class sama seperti aturan

penulisan variabel dalam PHP, yakni diawali dengan huruf atau underscore untuk karakter pertama, kemudian boleh diikuti dengan huruf, underscore atau angka untuk karakter kedua dan selanjutnya. Isi dari class berada dalam tanda kurung kurawal.

Berikut adalah contoh penulisan class dalam PHP :

```php
<?php
class laptop {
   // isi dari class laptop...
}
?>
```

Pengertian Property dalam Pemrograman Berorientasi Objek

Property (atau disebut juga dengan atribut) adalah data yang terdapat dalam sebuah class. Melanjutkan analogi tentang laptop, property dari laptop bisa berupa merk, warna, jenis processor, ukuran layar, dan lain-lain.

Jika anda sudah terbiasa dengan program PHP, property ini sebenarnya hanyalah variabel yang terletak di dalam class. Seluruh aturan dan tipe data yang biasa diinput kedalam variabel, bisa juga diinput kedalam property. Aturan tata cara penamaan property sama dengan aturan penamaan variabel.

Berikut adalah contoh penulisan class dengan penambahan property :

```php
<?php
class laptop {
   var $pemilik;
   var $merk;
   var $ukuran_layar;
   // lanjutan isi dari class laptop...
}
?>
```

Dari contoh diatas, $merk, $ukuran_layar dan $jenis_processor adalah property dari class laptop. Seperti yang kita lihat, penulisan property di dalam PHP sama dengan cara penulisan variabel, yakni menggunakan tanda dollar ($). Sebuah class tidak harus memiliki property.

Pengertian Method dalam Pemrograman Berorientasi Objek

Method adalah tindakan yang bisa dilakukan didalam class. Jika menggunakan analogi class laptop kita, maka contoh method adalah: menghidupkan laptop, mematikan laptop, mengganti cover laptop, dan berbagai tindakan lain.

Method pada dasarnya adalah function yang berada di dalam class. Seluruh fungsi dan sifat function bisa diterapkan kedalam method, seperti argumen/parameter, mengembalikan nilai (dengan keyword return), dan lain-lain.

Berikut adalah contoh penulisan class dengan penambahan method :

```php
<?php
class laptop {
    function hidupkan_laptop() {
    //... isi dari method hidupkan_laptop
    }

    function matikan_laptop() {
    //... isi dari method matikan_laptop
    }

    ... //isi dari class laptop
}
?>
```

Dari contoh diatas, function hidupkan_laptop() dan function matikan_laptop() adalah method dari class laptop. Seperti yang kita lihat, bahwa penulisan method di dalam PHP sama dengan cara penulisan function. Sebuah class tidak harus memiliki method.

Pengertian Object dalam Pemrograman Berbasis Objek

Object atau Objek adalah hasil cetak dari class, atau hasil 'konkrit' dari class. Jika menggunakan analogi class laptop, maka objek dari class laptop bisa berupa: laptop_andi, laptop_anto, laptop_yukcoding, dan lain-lain. Objek dari class laptop akan memiliki seluruh ciri-ciri laptop, yaitu property dan method-nya.

Proses 'mencetak' objek dari class ini disebut dengan 'instansiasi' (atau instantiation dalam bahasa inggris). Pada PHP, proses instansiasi dilakukan dengan menggunakan keyword 'new'. Hasil cetakan class akan disimpan dalam variabel untuk selanjutnya digunakan dalam proses program.

Sebagai contoh, berikut adalah cara membuat obje laptop_andi dan laptop_anto yang dibuat dari class laptop :

```php
<?php
class laptop {
   //... isi dari class laptop
   }

$laptop_andi = new laptop();
$laptop_anto = new laptop();
?>
```

Dari contoh diatas, $laptop_andi dan $laptop_anto merupakan objek dari class laptop. Kedua objek ini akan memiliki seluruh property dan method yang telah dirancang dari class laptop.

Petemuan 4

Struktur Kontrol

Tujuan Instruksional :
Pokok bahasan ini:

- Menggunakan struktur kontrol keputusan (if, else, switch) yang digunakan untuk memilih blokkode yangakan dieksekusi
- Menggunakan struktur kontrol pengulangan (while, do-while, for) yang digunakanuntuk melakukanpengulangan padablok kode yangakan dieksekusi
- Menggunakan statement percabangan (break,continue, return) yang digunakan untuk mengatur redirection dari program

Kompetensi yang Diharapkan :
Mahasiswa diharapkan dapat memahami apa itu struktur kontrol

Waktu Pertemuan : 100 menit

Latar Belakang

Pada bab sebelumnya, kita sudah mendapatkan contah dari program sequential, dimana statement dieksekusi setelah statement sebelumnya dengan urutan tertentu. Pada bagian ini, kita mempelajari tentang struktur kontrol yang bertujuan agar kita dapat menentukan urutan statement yang akan dieksekusi. Struktur kontrol keputusan adalah statement dari Java yang mengijinkan user untuk memilih dan mengeksekusi blok kode dan mengabaikan blok kode yang lain.

Percobaan 1

Program 2-1

```
public class Grade4 {
    public static void main(String[]args ){
        String  names[]  =  {"Beah",  "Bianca",  "Lance",
"Beah"};          int  count = 0;
        for(int i=0;i<names.length; i++){
if(      !names[i].equals("Beah")      ){
continue;
            }
count++;
        }

  }
}
```

*Berapa nilai variabel **count** ?*

Percobaan 2

Program 2-2

```
public class Grade6  {
    public static void main(String[]args ){
outerLoop:
        for(int i=0;i<5;i++ ){
for(int j=0;j<5;j++ ){
                System.out.println("Inside for(j)loop");
//message1                  if(j==2 )  continue outerLoop;
            }
            System.out.println("Inside for(i)loop"); //message2
        }
    }
     }
```

Tuliskan output dan jelaskan logika pada class Grade6 :

Output:	Penjelasan:

Percobaan 3

Program 2-3

```
      s SwitchCaseExample {  static
      void main(String[] args) {

      grading('A');
      grading('C');
      grading('E');
      grading('G');
}

public static void grading(char grade) {
      int success;
      switch (grade) {
      case 'A':
            System.out.println("Excellent
grade");         success = 1;          break;
      case 'B':
            System.out.println("Very good
grade");         success = 1;          break;
      case 'C':
            System.out.println("Good
grade");         success = 1;
      break;      case 'D':        case
'E':       case 'F':
System.out.println("Low grade");
                  success =
0;                break;
            default:
                System.out.println("Invalid grade");
                  success = -1;
                  break;
            }
 passTheCourse(success); //pemanggilan method
passTheCourse
      }

      public static void passTheCourse(int success)
{           switch (success) {
                        //tambahkan kode program disini
            }
      }
}
```

Modifikasi kode program diatas sehingga memberikan output:

```
Excellent grade
Final result: Success
Good grade
Final result: Success
Low grade
Final result: Fail
Invalid grade
No result
```

Ketentuan

Nilai variabel *success*	**Keterangan yang dicetak**
-1	No result
0	Final Result: Fail
1	Final Result: Success
Selainnya	Uknknown result

Jawab:

Soal Latihan

Program 2-4

```
public                class
WhileRectangle            {
public    int    height=3;
public int width = 10;
    public                  void
displayRectangle() {        int
colCount  =  0;             int
rowCount  =  0;            while
(rowCount       <       height){
colCount = 0;
            while   (colCount   <   width){
System.out.print("@");
colCount++;
            }
            System.out.println();
rowCount++;
        }
    }
  //Tambahkan   kode   di
sini }
```

Berdasarkan kode pada percobaan 4, lakukan:

1. Buat method main lalu Panggil method displayRectangle, lalu tuliskan outputnya pada kolom berikut:

2. Ganti loop **while ... do** menjadi **do ... while**, akan tetapi masih memberikan output yang sama. Tuliskan kodenya pada kolom berikut

Pertemuan 5

Java Array

Tujuan Intruksional :
Pokok Bahasan ini menjelaskan:

- Mendeklarasikan danmembuat array
- Mengakses elemen-elemen didalam array
- Menentukanjumlahelement didalam sebuah array
- Mendeklarasikan danmembuat array multidimensi

Kompetensi Yang Diharapkan :
Mahasiswa diharapkan telah memahami apa itu Java Array

Waktu Pertemuan : 100 Menit

Latar Belakang

Pada bagian ini, kita akan mendiskusikan mengenai array dalam Java. Pertama, kita akan mendefinisikan apa yang dimaksud denganarray, kemudian Kita juga akan mendefinisikan bagaimana mendeklarasikannya dan menggunakannya dalam Java.

Dibagian sebelumnya,kita telah mendiskusikan bagaimana cara pendeklarasian berbagai macamvariabel dengan menggunakan tipe data primitif. Dalam pendeklarasian variabel, kita sering menggunakan sebuah tipe data beserta namavariabel atau identifier yang unik, dimana untuk menggunakan variabel tersebut, kitaakanmemanggil dengan nama identifiernya.

Sebagai contoh, kita memiliki tiga variabel dengan tipe data int yang memiliki identifier yang berbeda untuk tiap variabel.

```
int     number1;
int     number2;
int     number3;
number1  =  1;
number2  =  2;
number3 = 3;
```

Seperti yang dapat Anda perhatikan pada contoh diatas, hanya untuk menginisialisasi dan menggunakan variabel terutama pada saat variabel-variabel tersebut memiliki tujuan yang sama, dirasa sangat membingungkan. Di Java maupun di bahasa pemrograman yang lain, mereka memiliki kemampuan untuk menggunakan satu variabel yang dapat menyimpan sebuah data list dan kemudian memanipulasinya dengan lebih efektif. Tipe variabel inilah yang disebut sebagai array.

Sebuah array akan menyimpan beberapa item data yang memiliki tipe data sama di dalam sebuah blok memori yang berdekatan yang kemudian dibagai menjadi beberapa slot. Bayangkanlah array adalah sebuah variabel – sebuah lokasi memori tertentu yang memiliki satu nama sebagai identifier, akan tetapi ia dapat menyimpan lebih dari sebuah value.

Percobaan 1

Program 3-1

```
public class ContohArray {
    public static void main(
String[] args ){          int[] ages
= new int[100];          for( int
i=0; i<ages.length; i++ ){
            System.out.print(
ages[i] );
        }
    }
}
```

Buat penjelasan berkaitan dengan kode program di atas!

Program 3-2

```
public class ArrayMultiDimensi {

    public static void main( String[] args ){
int[][][] threeD = new int[8][16][24];
        String[][][] dogs = {
           {{ "terry", "brown","m" },{ "Kristin",
"white","f" },
           { "toby", "gray","f"},{ "fido", "black","m"}},};
       //Tambahkan kode disini
} }
```

Gunakan looping **for** dan atribut **length** untuk mencetak semua elemen data dari variabel array dogs.

Soal Latihan

Berikut ini adalah array multi dimensi yang menyatakan isi dari sebuah buku alamat:

```
String entry[][][]= {{{"Florence", "735-1234", "Manila"},
        {"Joyce", "983-3333", "Quezon City"},
        {"Becca", "456-3322", "Manila"}}};
```

Cetak buku alamat tersebut dalam format berikut ini:

```
Name : Florence
Tel. #  : 735-1234
Address : Manila
Name : Joyce
Tel. #  : 983-3333
Address  : Quezon City
Name : Becca
Tel. #  : 456-3322
Address : Manila
```

Tuliskan kode programnya di bawah ini:

Pertemuan 6

Bekerja dengan Java Class Library

Tujuan Instruksional :

- menjelaskan OOP dan beberapa konsepnya
- perbedaan antara class dan object
- pebedaan antara instance variables/method dan class (static) variable/method
- menjelaskan method apa dan bagaimana memanggil method parameter
- mengidentifikasi beberapa jangkauan dari sebuah variable
- memilih tipe data primitive dan object
- membandingkan objects dan menjabarkan class dari objects

Kompetensi yang Diharapkan :
Mahasiswa diharapkan dapat memahami java class library.

Waktu Pertemuan : 100 menit

Latar Belakang

Pada bagian ini, kita akan mengantarkan beberapa konsep dasar dari Object-Oriented Programming (OOP). Selanjutnya kita akan membahas konsep dari classes dan bagaimana menggunakan class dan anggotanya. Perubahan dan pemilihan object juga akan dibahas. Sekarang, kita akan fokus dalam menggunakan class yang telah dijabarkan dalam Java Class library, yang nantinya kita akan membahas bagaimana membuat class anda sendiri.

OOP berputar pada konsep dari object sebagai dasar element dari program anda. Object tersebut dikarakterisasi oleh sifat/attribut dan tingkah lakunya. Sebagai contoh di dunia nyata, objek sebuah mobil mempunyai sifat tipe transmisi, warna dan manufaktur. Mempunyai kelakuan berbelok, mengerem dan berakselerasi. Dengan cara yang sama pula kita dapat mendefinisikan perbedaan sifat dan tingkah laku dari objek-objek yang lainnya.

Percobaan 1 Pass By Reference

Program 4-1

```
public class TestPassByReference {
public static void main(String[] args) {
int []ages = {10,11,12};
for(int i=0;i<ages.length;i++){
System.out.println(ages[i]);
        }
test(ages);
        for(int i=0;i<ages.length;i++){
System.out.println(ages[i]);
        }
    }
    public static void test(int []arr){
for(int i=0;i<arr.length;i++){
arr[i]=i+50;
        }
    }
}
```

Percobaan 2 Perbandingan Objek

Program 4-2

```
class EqualsTest {
    public static void main(String[]
arguments) {           String str1, str2;
str1 = "Ilkom Unila";         str2 = str1;
        System.out.println("String1: " + str1);
        System.out.println("String2: " + str2);
        System.out.println("Same object? " + (str1 ==
str2));           str2 = new String(str1);
        System.out.println("String1: " + str1);
        System.out.println("String2: " + str2);
        System.out.println("Same object? " + (str1 ==
str2));           System.out.println("Same value? " +
str1.equals(str2));
    }
}
```

Tuliskan outputnya dan berikan penjelasan tentang perbedaan operator pembanding (==) dengan method ***equals()!***

Output:	Penjelasan:

Parameter

Program 4-3

```
public class Elevator {
    public boolean doorOpen=false;
public int currentFloor = 1;
public final int TOP_FLOOR = 5;
public final int BOTTOM_FLOOR = 1;
public void openDoor() {
        System.out.println("Opening door.");
doorOpen = true;
        System.out.println("Door is open.");
    }
    public void closeDoor() {
        System.out.println("Closing door.");
doorOpen = false;
        System.out.println("Door is closed.");
    }
    public void goUp() {
        System.out.println("Going up one floor.");
currentFloor++;
        System.out.println("Floor: " + currentFloor);
    }
    public void goDown() {
        System.out.println("Going down one floor.");
currentFloor--;
        System.out.println("Floor: " + currentFloor);
    }
    public void setFloor(int
desiredFloor) {          while
(currentFloor != desiredFloor){
if(currentFloor < desiredFloor) {
goUp();              } else {
goDown();
          } }     }
    public int getFloor(){
return currentFloor;
    }
    public boolean checkDoorStatus(){
return doorOpen;
    }
}
```

Program 4-4

```
public class ElevatorTest {
public static void main(String[] args) {
Elevator myElevator = new Elevator();
myElevator.openDoor();
myElevator.closeDoor();
myElevator.goUp();
myElevator.goUp();
myElevator.goUp();
myElevator.openDoor();
myElevator.closeDoor();
myElevator.goDown();
myElevator.openDoor();
myElevator.closeDoor();
myElevator.goDown();
myElevator.setFloor(myElevator.TOP_FLOOR);
myElevator.openDoor();
```

Soal Latihan

Dengan kata-kata Anda sendiri, definisikan istilah-istilah berikut ini:

a. Class

b. Object

c. Instantiate

d. Instance Variable

e. Instance Method

f. Class Variables atau static member variables

g. Constructor

Pertemuan 6

Membuat Class Sendiri

Tujuan Intruksional :
Pokok Bahasan ini menjelaskan:

- Membuat kelas mereka sendiri
- Mendeklarasikan atribut dan *method* pada *class*
- Menggunakan referensi *this* untuk mengakses *instance data*
- Membuat dan memanggil *overloaded method*
- Mengimport dan membuat *package*
- Menggunakan *access modifiers* untuk mengendalikan akses terhadap *class member*

Kompetensi Yang Diharapkan :
Mahasiswa diharapkan dapat memahami class mereka sendiri.

Waktu Pertemuan : 100 Menit

Latar Belakang
Setelah kita mempelajari penggunaan *class* dari Java Class Library, kita akan mempelajari bagaimana menuliskan sebuah *class* sendiri. Pada bagian ini, untuk mempermudah pemahaman pembuatan *class*, kita akan membuat contoh *class* dimana akan ditambahkan beberapa data dan fungsi-fungsi lain.

Percobaan 1 Method Accessor dan Method Mutator
Perhatikan Kode Berikut

Program 5-1

```
public class PrivateShirt {
private int idBaju = 0;
    private String keterangan = "-Keterangan Diperlukan-";
    //Kode warna R=Merah, G=Hijau, B=Biru, U=Tidak
Ditentukan     private char kodeWarna = 'U';     private
double harga = 0.0;     private int jmlStok = 0;

    public char getKodeWarna() {
return kodeWarna;
    }
    public void setKodeWarna(char kode)
{         switch (kode) {         case
'R':         case 'G':         case 'B':
kodeWarna = kode;              break;
default:
            System.out.println("Kode Warna salah, gunakan R, G, atau
B");
        }
    }
 }
```

Program 5-2

```
      public class PrivateShirt {
      private int idBaju = 0;
          private String keterangan = "-Keterangan Diperlukan-";
      //Kode warna R=Merah, G=Hijau, B=Biru, U=Tidak Ditentukan
      private char kodeWarna = 'U';     private double harga =
      0.0;     private int jmlStok = 0;

          public char getKodeWarna() {
      return kodeWarna;
          }
          public void setKodeWarna(char kode)
      {         switch (kode) {         case
      'R':         case 'G':         case 'B':
      kodeWarna = kode;              break;
      default:
                  System.out.println("Kode Warna salah, gunakan R, G,
      atau B");         }
          }
       }
```

Program 5-3

```
public class PrivateShirtTest {
public static void main(String[] args) {         PrivateShirt
privShirt = new PrivateShirt();
char kodeWarna;
privShirt.setKodeWarna('R');
kodeWarna = privShirt.getKodeWarna();
        System.out.println("Kode Warna: " + kodeWarna);
        privShirt.setKodeWarna('Z');
        kodeWarna = privShirt.getKodeWarna();
        System.out.println("Kode Warna: " + kodeWarna);
    } }
```

Tuliskan outputnya dan berikan penjelasan mengenai perbedaan fungsi method ***getKodeWarna()*** *dan* ***setKodeWarna()****!*

Output:	Penjelasan:

Percobaan 2 Constructor

Program 5-4

```
class Kotak {
double panjang;
double lebar;
double tinggi;

    Kotak(double p, double l, double t)
{          panjang = p;          lebar =
l;          tinggi = t;
    }
    double hitungVolume() {
        return (panjang * lebar * tinggi);
    }
}
class DemoConstructor {
    public static void main(String[] args) {
Kotak k1, k2;
        k1 = new Kotak(4, 3, 2);
k2 = new Kotak(6, 5, 4);

  System.out.println("Volume kotak 1 = " + k1.hitungVolume() + " cm");
 System.out.println("Volume kotak 2 = " + k2.hitungVolume() + " cm");
    }
}
```

Tuliskan outputnya dan jelaskan kegunaan constructor ***Kotak()****!*

Output:	Penjelasan:

Percobaan 3 Constructor Overloading

Program 5-5

```
class Language {
String name;

    Language() {
        System.out.println("Constructor method called.");
}

    Language(String t) {
name = t;
    }      public static void main(String[]
args) {          Language cpp  = new
Language();          Language java = new
Language("Java");
cpp.setName("C++");          java.getName();
cpp.getName();
    }
    void setName(String t) {
name = t;
    }
    void getName() {
        System.out.println("Language name: " + name);
    }
}
```

*Bagian manakah yang dinamakan **constructor overloading**? Berikan penjelasan singkat!*

Percobaan 4 Method Overloading

Program 5-6

```
public class ShirtTwo {     public
int shirtID = 0;
    public String description = "-description required-";
    //The color codes are R=Red, B=Blue, U=Unset
public char colorCode = 'U';     public double price =
0.0;     public int quantityInStock = 0;

    public void setShirtInfo(int ID, String desc, double cost) {
shirtID = ID;          description = desc;          price = cost;
    }
    public void setShirtInfo(int ID, String desc, double cost, char
color) {          shirtID = ID;          description = desc;
price = cost;          colorCode = color;
    }      public void setShirtInfo(int ID, String desc, double cost,
char color, int quantity) {          shirtID = ID;          description
= desc;          price = cost;          colorCode = color;
        quantityInStock = quantity;
    }

     //This method displays the values for an item      public
void display() {
        System.out.println("Item ID: " + shirtID);
        System.out.println("Item description: " + description);
        System.out.println("Color Code: " + colorCode);          "     +
       System.out.println("Item price: " + price);
       System.out.println("Quantity in stock:
       quantityInStock);
           } }
```

Program 5-7

```
       class ShirtTwoTest {
           public static void main(String args[]) {
       ShirtTwo shirtOne = new ShirtTwo();
               ShirtTwo shirtTwo = new ShirtTwo();
               ShirtTwo shirtThree = new ShirtTwo();

               shirtOne.setShirtInfo(100, "Button Down", 12.99);
       shirtTwo.setShirtInfo(101, "Long Sleeve Oxford", 27.99,
       'G');
               shirtThree.setShirtInfo(102, "Shirt Sleeve T-Shirt",
       9.99, 'B', 50);
       shirtOne.display();
       shirtTwo.display();
       shirtThree.display();
           }
       }
```

*Tuliskan outputnya dan jelaskan logika dari bagian yang dinamakan **method overloading**!*

Output:	Penjelasan:

Soal Latihan
Tugas Anda adalah membuat sebuah *class* yang memuat data-data pada buku alamat. Tabel berikut mendefinisikan informasi yang dimiliki oleh buku alamat.

Attribut	Deskripsi
Nama	Nama Lengkap perseorangan
Alamat	Alamat Lengkap
Nomor Telepon	Nomor telepon personal
Alamat E-Mail	Alamat E-Mail personal

Buat implementasi dari *method* sebagai berikut :

- Menyediakan *accessor* dan *mutator method* terhadap seluruh atribut
- Dua Constructor atau lebih

Pertemuan 7

Pewarisan dan Polimorfisme

Tujuan Intruksional :
Pokok Bahasan ini menjelaskan:

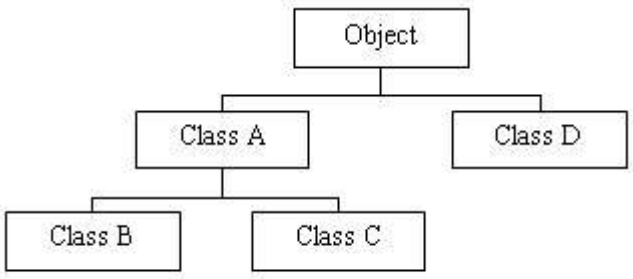
- Mendefinisikan superclasses dan subclasses
- Override method dari superclasses
- Membuat method final dan class final

Kompetensi Yang Diharapkan :
Mahasiswa diharapkan dapat memahami pewarisan dan polimorfisme

Waktu Pertemuan : 100 Menit

Latar Belakang
Dalam bagian ini, kita akan membicarakan bagaimana suatu class dapat mewariskan sifat dari class yang sudah ada. Class ini dinamakan subclass dan induk class dinamakan superclass. Kita juga akan membicarakan sifat khusus dari Java dimana kita dapat secara otomatis memakai method yand tepat untuk setiap object tanpa memperhatikan asal dari subclass object. Sifat ini dinamakan polimorfisme. Pada akhirnya, kita akan mendiskusikan tentang interface yang membantu mengurangi penulisan program.

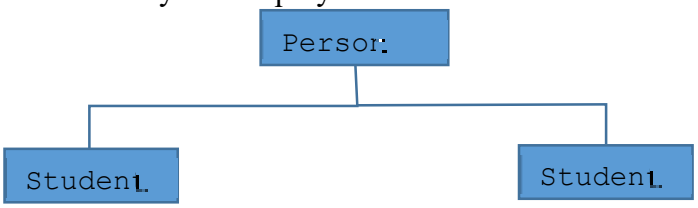
Dalam Java, semua class, termasuk class yang membangun Java API, adalah subclasses dari superclass Object. Contoh hirarki class diperlihatkan di bawah ini. Beberapa class di atas class utama dalam hirarki class dikenal sebagai superclass. Sementara beberapa class di bawah class pokok dalam hirarki class dikenal sebagai subclass dari class tersebut.



Class hierarchy in Java.

Pewarisan adalah keuntungan besar dalam pemrograman berbasis object karena suatu sifat atau method didefinisikan dalam superclass, sifat ini secara otomatis diwariskan dari semua subclasses. Jadi, Anda dapat menuliskan kode method hanya sekali dan mereka dapat digunakan oleh semua subclass. Subclass hanya butuh mengimplementasikan perbedaannya sendiri dan induknya.

Interface adalah jenis khusus dari blok yang hanya berisi method signature(atau constant). Interface mendefinisikan sebuah(signature) dari sebuah kumpulan method tanpa tubuh. Interface mendefinisikan sebuah cara standar dan umum dalam menetapkan sifat-sifat dari class-class. Mereka menyediakan class-class, tanpa memperhatikan lokasinya dalam hirarki class, untuk mengimplementasikan sifat-sifat yang umum. Dengan catatan bahwa interface-interface juga menunjukkan polimorfisme, dikarenakan program dapat memanggil method interface dan versi yang tepat dari method yang akan dieksekusi tergantung dari tipe object yang melewati pemanggil method interface. Sekarang, class induk Person dan subclass Student dari contoh sebelumnya, kita tambahkan subclass lain dari Person yaitu Employee. Di bawah ini adalah hierarkinya,



Dalam Java, kita dapat membuat referensi yang merupakan tipe dari superclass ke sebuah object dari subclass tersebut. Kemampuan dari referensi untuk mengubah sifat menurut object apa yang dijadikan acuan dinamakan polimorfisme. Polimorfisme menyediakan multiobject dari subclasses yang berbeda untuk diperlakukan sebagai object dari superclass tunggal, secara otomatis menunjuk method yang tepat untuk menggunakannya ke particular object berdasar subclass yang termasuk di dalamnya.

Contoh lain yang menunjukkan properti polimorfisme adalah ketika kita mencoba melalui referensi ke method. Misalkan kita punya method statis printInformation yang mengakibatkan object Person sebagai referensi, kita dapat me-referensi dari tipe Employee dan tipe Student ke method ini selama itu masih subclass dari class Person.

Percobaan 1: Super class dan Sub class

Program 6-1

```
public class Person {
protected String name;
protected String address;
/**
* Default constructor
*/
    public Person(){
        System.out.println("Inside
Person:Constructor");         name = "";
address = "";
    }
/**
* Constructor dengan dua parameter
*/
    public Person( String name, String address)
{         this.name = name;         this.address
= address;
    }
/**
* Method accessor
*/
    public String getName() {
return name;
    }
    public String getAddress() {
return address;
}
    public void setName(String name) {
this.name = name;
    }
    public void setAddress(String add)
{          this.address = add;
    }
}
```

Program 6-2

```
public class Student extends Person{
public Student(){
        //super( "SomeName", "SomeAddress");
        //super();
        //super.name = "name";
        System.out.println("Inside Student:Constructor");
    }
}
```

Program 6-3

```
public class Employee extends Person {
public Employee(){
        //super( "SomeName", "SomeAddress");
        //super();
        //super.name = "name";
        System.out.println("Inside Employee:Constructor");
    }
}
```

```
public class Utama {
    public static void main(
String[] args) {          Student
anna = new Student();
        Employee riska = new
Employee();
    }
}
```

Percobaan 2: Override Method

Override methode adalah mengimplementasikan kembali method super class oleh sub class. Anda dapat mengoverride semua non-final method superclass.

Tambahkan kode berikut ke dalam class Student.

Program 6-4

```
public String getName(){
        System.out.println("Student
Name : "+ this.name);
return "";
    }
```

Lalu panggil method setName dan getName dalam method main class Utama, seperti di bawah ini:

Program 6-5

```
        Student anna = new Student();
Employee riska = new Employee();
anna.setName("Anna Kornicova");
riska.setName("Riskandar Irwanto");
String stdName= anna.getName();
        String empName= riska.getName();
```

Percobaan 2: Final Class dan Final Method

Untuk mengetahui apa yang dimaksud dengan final class dan final method, coba anda ubah class Person menjadi final, lalu perhatikan class Student dan Employee pada netbeans atau anda compile kembali ke dua class tersebut, apakah terjadi kesalah? Tuliskan komentar kesalahan yang ditunjukan oleh netbeans atau ketika anda compile dengan command.

Sekarang anda kembalikan kembali class Person seperti semula, lalu anda ubah method getName() pada class person menjadi final, tuliskan kembali komentar kesalahannya.

Bagaimana kesimpulan Anda tentang final class dan final method?

Percobaan 3: Polimorfisme

Kembalikan kode program Anda seperti pada **Percobaan 1** lalu buat program pada class Utama menjadi:

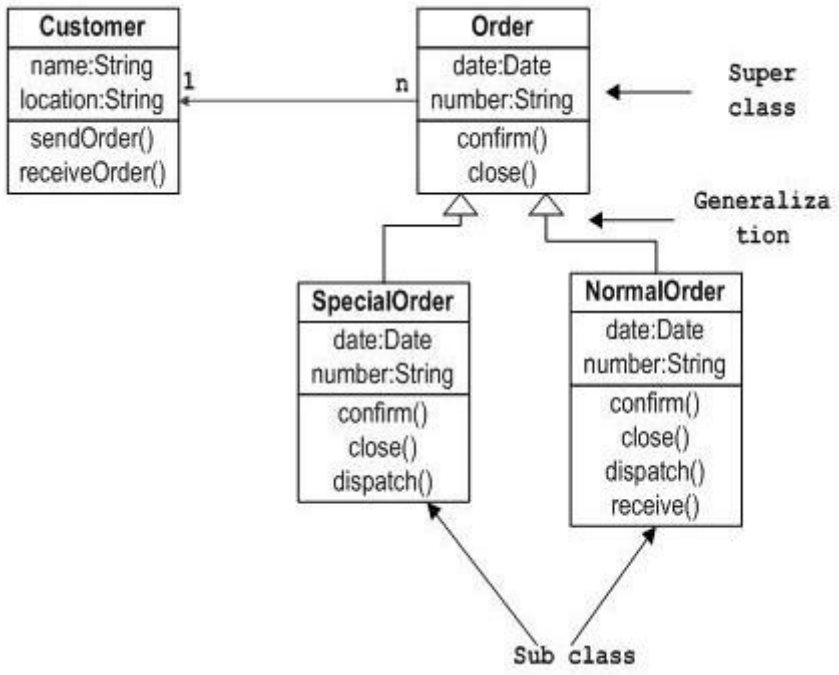
Program 6-6

```
public class Utama {
    public static void main( String[] args) {
        Person p;
        Student std=new Student();
        Employee emp=new Employee();
p = std;
        p.setName("Yahya");
        p.getName();
    }
}
```

Tuliskan output dan beri penjelasan

Output	Penjelasan

Soal Latihan
Implementasikan Class Diagram berikut



Pertemuan 8

Class Abstrak dan Interface

Tujuan Intruksional :
Pokok Bahasan ini menjelaskan:

- Mendefiniskan class abstract
- Membuat interface
- Membuat sub class dari class abstract
- Mengimplementasikan interface dan
- Mengoverride Method

Kompetensi Yang Diharapkan :
Mahasiswa diharapkan dapat memahami class abstrak dan Interface.

Waktu Pertemuan : 100 Menit

Latar Belakang

Sebuah class abstract adalah class yang tidak dapat di-instantiate. Seringkali muncul di atas hirarki class pemrograman berbasis object, dan mendefinisikan keseluruhan aksi yang mungkin pada object dari seluruh subclasses dalam class.

Method ini dalam class abstract yang tidak mempunyai implementasi dinamakan method abstract. Untuk membuat method abstract, tinggal menulis deklarasi method tanpa tubuh class dan digunakan menggunakan kata kunci abstract. Contohnya,

```
public abstract void someMethod();
```

Interface adalah jenis khusus dari blok yang hanya berisi method signature(atau constant). Interface mendefinisikan sebuah(signature) dari sebuah kumpulan method tanpa tubuh.

Interface mendefinisikan sebuah cara standar dan umum dalam menetapkan sifatsifat dari class-class. Mereka menyediakan class-class, tanpa memperhatikan lokasinya dalam hirarki class, untuk mengimplementasikan sifat-sifat yang umum. Dengan catatan bahwa interface-interface juga menunjukkan polimorfisme, dikarenakan program dapat memanggil method interface dan versi yang tepat dari method yang akan dieksekusi tergantung dari tipe object yang melewati pemanggil method interface.

Percobaan 1 Abstract Class

Buat class Person menjadi sbb:

Program 7-1

```
public abstract class Person {
protected String name;
protected String address;
public Person(){
        System.out.println("Inside Person:Constructor");
name = "";
address = "";
}
public Person( String name, String address) {
this.name = name;
this.address = address;
}
    public String getName() {
return name;
    }
    public String getAddress() {
return address;
    }
    public void setName(String name) {
this.name = name;
    }
    public void setAddress(String add) {
this.address = add;
    }     public abstract void
walk();
}
```

Perhatikan class Student dan Employee, apakah ada kesalahan seteah class Person diubah menjadi abstract dan terdapat method abstract walk? Tuliskan komentar kesalahan yang ditampilkan!

Apa yang harus dilakukan agar sub classs Person tidak terjadi kesalahan?

Percobaan 2 Interface

Buat sebuah interface sbb:

Program 7-2

```
public interface Relasi {
public boolean isGreater( Object a, Object b);
public boolean isLess( Object a, Object b);
public boolean isEqual( Object a, Object b); }
```

Lalu buat sebuah class Line yang mengimplementasikan interface tersebut!

Program 7-3

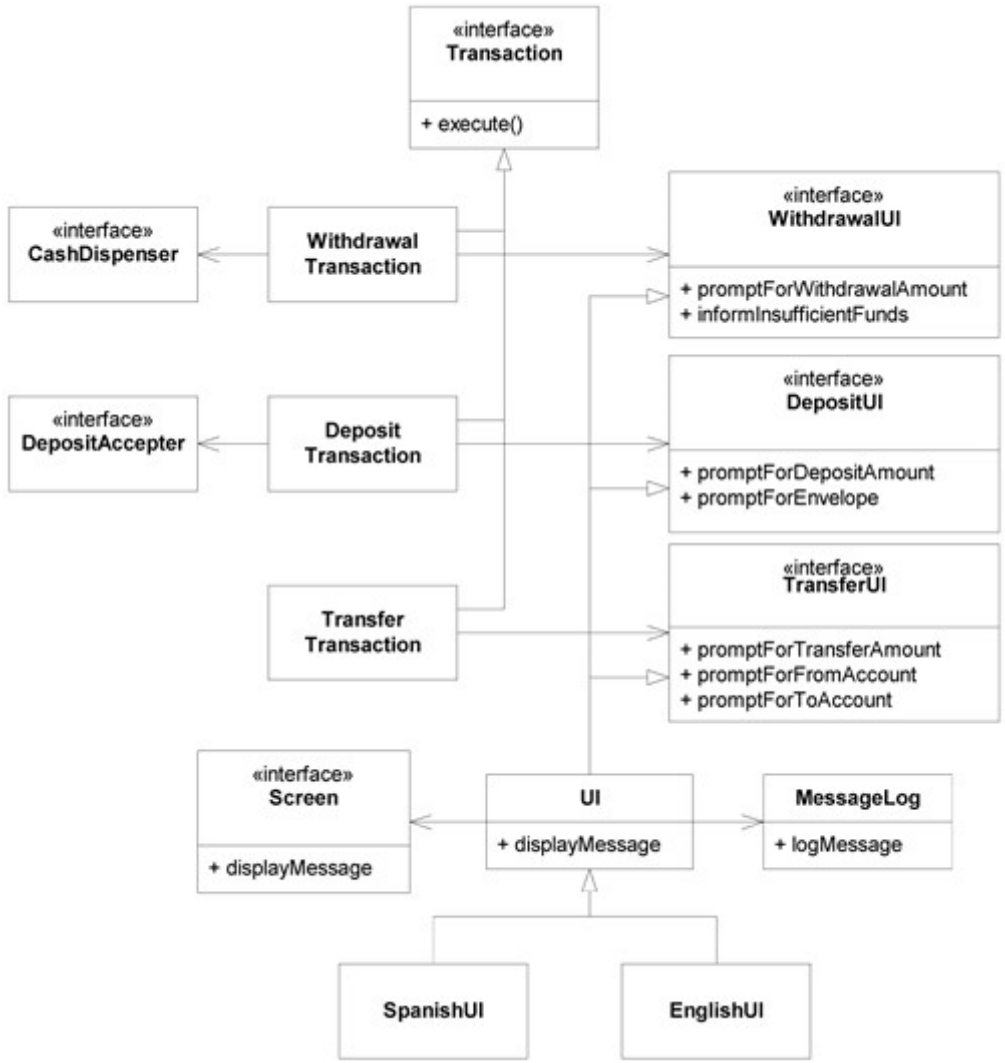
```
public class Line implements Relasi {

}
```

Lengkapi kode di atas agar tidak terjadi kesalahan. Apa yang harus Anda lakukan?

Soal Latihan

Implemtasikan class diagram pada gambar berikut:



Pertemuan 9

Text Based Application

Tujuan Intruksional :
Pokok Bahasan ini menjelaskan:

- Mendapatkan input dari command-line
- Mengetahui cara untuk memanipulasi properties dari sistem
- Membaca standart input
- Membaca dan menulis file

Kompetensi Yang Diharapkan :
Mahasiswa diharapkan dapat memahami aplikasi berbasis teks

Waktu Pertemuan : 100 Menit

Percobaan 1 Membaca keseluruhan

Program 8-1

```
import java.io.*;
public class GreetUser
{

public GreetUser() {
}
public static void main(String args[]) throws IOException {
System.out.println("Hi, what's your name?");
String name;
BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));
name = br.readLine();
System.out.println("Nice to meet you, " + name + "! :)");
}
}
```

Percobaan 2 Membaca file

Program 8-2

```
import java.io.*;
public class ReadFile {
public ReadFile() {
    }
    public static void main(String args[]) throws IOException {
System.out.println("What is the name of the file to read from?");
        String filename;
        BufferedReader     br     =     new      BufferedReader(new
InputStreamReader(System.in));
filename = br.readLine();
        System.out.println("Now reading from " + filename +
"...");
        FileInputStream fis = null;
try {
            fis = new FileInputStream(filename);
} catch (FileNotFoundException ex) {
System.out.println("File not found.");}
try {             char data;             int temp;
do {
                 temp = fis.read();
data = (char) temp;                  if
(temp != -1) {
                      System.out.print(data);
                 }
             } while (temp != -1);
         } catch (IOException ex) {
             System.out.println("Problem in reading from
file.");          }
    }
}         System.out.println("Inside Employee:Constructor");
    }
}
```

Percobaan 3 Menulis file

Program 8-3

```
import java.io.*;
public class WriteFile
{     public
WriteFile() {
    }
    public static void main(String args[]) throws IOException {
System.out.println("What is the name of the file to be written
to?");
        String filename;
        BufferedReader     br     =     new      BufferedReader(new
InputStreamReader(System.in));
filename = br.readLine();
        System.out.println("Enter data to write to " + filename +
"...");
        System.out.println("Type q$ to end.");
        FileOutputStream fos = null;
try {
            fos = new FileOutputStream(filename);
        } catch (FileNotFoundException ex) {
            System.out.println("File cannot be opened for
writing.");         }         try {
            boolean done =
false;             int data;
do {
                data = br.read();
if ((char)data == 'q') {
data = br.read();                    if
((char)data == '$') {
done = true;                    } else
{fos.write('q');
fos.write(data);
                    }
                } else {fos.write(data);
}
} while (!done);
       } catch (IOException ex) {
System.out.println("Problem in reading
from the file.");        }
}
}
```

Latihan

3.1 Spasi menjadi Underscore (_)

Buatlah sebuah program yang memuat dua String sebagai argument, sumber dan nama file tujuan. Kemudian, baca file sumber dan tuliskan isi dari file tersebut terhadap file tujuan, seluruh spasi yang ada (‘ ‘) diubah menjadi underscore (‘_’).

Pertemuan 11

Abstract Windowing Toolkit dan Swing

Tujuan Intruksional :
Pokok Bahasan ini menjelaskan:

- Mendapatkan input dari command-line
- Mengetahui cara untuk memanipulasi properties dari sistem
- Membaca standart input
- Membaca dan menulis file

Kompetensi Yang Diharapkan :
Mahasiswa diharapkan dapat memahami aplikasi berbasis teks

Waktu Pertemuan : 100 Menit

Latar Belakang
Tanpa mempelajari tentang grapichal user interface (GUI) API, Anda masih tetap bisa membuat suatu program. Tetapi, program Anda akan kelihatan tidak menarik dan tidak nyaman digunakan bagi para user. Memiliki GUI yang baik dapat memberi efek pada penggunaan aplikasi. Java menyediakan banyak tool seperti Abstract Windowing Toolkit dan Swing untuk mengembangkan aplikasi GUI yang interaktif.

The Java Foundation Class (JFC), merupakan bagian penting dari Java SDK, yang termasuk dalam koleksi dari API dimana dapat mempermudah pengembangan aplikasi JAVA GUI. JFC termasuk diantara 5 bagian utama dari API yaitu AWT dan Swing. Tiga bagian yang lainnya dari API adalah Java2D, Accessibility, dan Drag and Drop. Semua itu membantu developer dalam mendesain dan mengimplementasikan aplikasi dengan visualisasi yang lebih baik.

AWT dan Swing menyediakan komponen GUI yang dapat digunakan dalam membuat aplikasi Java dan applet. Anda akan mempelajari applet pada bab berikutnya. Tidak seperti beberapa komponen AWT yang menggunakan native code, keseluruhan Swing ditulis menggunakan bahasa pemrograman Java. Swing menyediakan implementasi platform-independent dimana aplikasi yang dikembangkan dengan platform yang berbeda dapat memiliki tampilan yang sama. Begitu juga dengan AWT menjamin tampilan look and feel pada aplikasi yang dijalankan pada dua mesin yang berbeda menjadi terlihat sama. Swing API dibangun dari beberapa API yang mengimplementasikan beberapa jenis bagian dari AWT. Kesimpulannya, komponen AWT dapat digunakan bersama dengan komponen Swing.

Percobaan 1 Menampilkan Panel

Program 9-1

```
import java.awt.Frame;
import java.awt.*;
 public class GraphicPanel extends Panel
{     public GraphicPanel() {
        setBackground(Color.black);
}
public void paint(Graphics g) {
         g.setColor(new Color(0,255,0));
         g.setFont(new Font("Helvetica",Font.PLAIN,16));
         g.drawString("Hello GUI World!", 30, 100);
         g.setColor(new Color(1.0f,0,0));
         g.fillRect(30, 100, 150, 10);
     }
     public static void main(String args[]) {
         Frame f = new Frame("Testing Graphics Panel");
GraphicPanel gp = new GraphicPanel();
         f.add(gp);
         f.setSize(600, 300);
         f.setVisible(true);
     } }
```

Percobaan 2 Menampilkan Complex Layout

Program 9-2

```
import java.awt.*;
public class ComplexLayout extends Frame{
public ComplexLayout() {
    }
    public static void main(String args[]) {
ComplexLayout cl = new ComplexLayout();
        Panel panelNorth = new Panel();
        Panel panelCenter = new Panel();
        Panel panelSouth = new Panel();
panelNorth.add(new Button("ONE"));
panelNorth.add(new Button("TWO"));
panelNorth.add(new Button("THREE"));
panelCenter.setLayout(new GridLayout(4,4));
panelCenter.add(new TextField("1st"));
panelCenter.add(new TextField("2nd"));
panelCenter.add(new TextField("3rd"));
panelCenter.add(new TextField("4th"));
panelSouth.setLayout(new BorderLayout());
panelSouth.add(new Checkbox("Choose me!"),
BorderLayout.CENTER);
        panelSouth.add(new Checkbox("I'm here!"),
BorderLayout.EAST);
        panelSouth.add(new Checkbox("Pick me!"),
BorderLayout.WEST);
        cl.add(panelNorth, BorderLayout.NORTH);
cl.add(panelCenter, BorderLayout.CENTER);
cl.add(panelSouth, BorderLayout.SOUTH);
cl.setSize(300,300);
cl.setVisible(true);
    }
}
```

Percobaan 3 Menampilkan JOption Demo

Program 9-3

```
import javax.swing.*; import
java.awt.*;
 public class JOptionPaneDemo {
    JOptionPane optionPane;     public
JOptionPaneDemo() {
    }
    void launchFrame() {
        optionPane = new JOptionPane();
        String name = optionPane.showInputDialog("Hi, what's
yourname?");
optionPane.showMessageDialog(null,"Nice to meet you, " + name + ".",
                "Greeting...",optionPane.PLAIN_MESSAGE);
System.exit(0); }
         public static void main(String args[]) {
new JOptionPaneDemo().launchFrame();
    }
}
```

Soal Latihan

Buatlah tampilan GUI untuk program tic-tac-toe. Papannya terdiri dari enam kotak. Ingatlah bahwa Anda akan menambahkan kode ini pada tahap akhir untuk mengatasi interaksi antar pengguna. Jadi, desainlah papan Anda dengan benar. Pastikanlah Anda memilih komponen yang pantas untuk papan tersebut. Keluarkan semua sisi artistik Anda. Anda dapat menggunakan AWT atau Swing untuk latihan ini.

Pertemuan 12

GUI Event Handling

Tujuan Intruksional :
Pokok Bahasan ini menjelaskan:

- Menerangkan komponen-komponen delegation event model
- Mengerti bagaimana delegation event model bekerja
- Menciptakan aplikasi GUI yang berinteraksi dengan user
- Mendiskusikan manfaat dari class-class adapter
- Mendiskusikan keuntungan-keuntungan dari menggunakan anonymous class

Kompetensi Yang Diharapkan :
Mahasiswa diharapkan dapat memahami GUI Event Handling

Waktu Pertemuan : 100 Menit

Latar Belakang
Pada modul ini, Anda akan belajar bagaimana mengendalikan events triggered ketika user berinteraksi dengan aplikasi GUI Anda. Setelah menyelesaikan modul ini, Anda akan dapat mengembangkan aplikasi GUI yang dapat merespon interaksi user.

Delegasi event model menguraikan bagaimana program Anda dapat merespon interaksi dari user. Untuk memahami model, mari kita pelajari pertama-tama melalui tiga komponen utamanya.

1. Event Source
   Event source mengacu pada komponen GUI yang meng-generate event. Sebagai contoh, jika user menekan tombol, event source dalam hal ini adalah tombol.
2. Event Listener/Handler

   The event listener menerima berita dari event-event dan proses-proses interaksi user. Ketika tombol ditekan, listener akan mengendalikan dengan menampilkan sebuah informasi yang berguna untuk user.

3. Event Object

   Ketika sebuah event terjadi (misal, ketika user berinteraksi dengan komponen GUI), sebuah object event diciptakan. Object berisi semua informasi yang perlu tentang event yang telah terjadi. Informasi meliputi tipe dari event yang telah

terjadi, seperti ketika mouse telah di-klik. Ada beberapa class event untuk kategori yang berbeda dari user action. Sebuah event object mempunyai tipe data mengenai salah satu class ini.

Percobaan 1 Mouse Event Demo :

Program 10-1

```
import java.awt.*;
import java.awt.event.*;
public class MouseEventsDemo extends Frame implements
    MouseListener, MouseMotionListener {
    TextField tf;
    public MouseEventsDemo(String title)    {
super(title);
tf = new TextField(60);
addMouseListener(this);    }
    public void launchFrame() {
        /* Menambah komponen pada frame */
add(tf, BorderLayout.SOUTH);
setSize(300,300);
setVisible(true);  }
    public void mouseClicked(MouseEvent me) {
String msg = "Mouse clicked.";
tf.setText(msg);  }
    public void mouseEntered(MouseEvent me) {
String msg = "Mouse entered component.";
tf.setText(msg);  }
    public void mouseExited(MouseEvent me) {
String msg = "Mouse exited component.";
tf.setText(msg);  }
    public void mousePressed(MouseEvent me) {
String msg = "Mouse pressed.";
tf.setText(msg);  }
    public void mouseReleased(MouseEvent me) {
String msg = "Mouse released.";
tf.setText(msg); }
    public void mouseDragged(MouseEvent me) {
String msg = "Mouse dragged at " + me.getX() + ","
+me.getY();
tf.setText(msg);  }
    public void mouseMoved(MouseEvent me) {
String msg = "Mouse moved at " + me.getX() + "," +
me.getY();
tf.setText(msg); }
    public static void main(String args[]) {
MouseEventsDemo med = new MouseEventsDemo("Mouse
EventsDemo");
        med.launchFrame();
    } )
```

Percobaan 2 Close Frame :

Program 10-2

```
import java.awt.*;
import java.awt.event.*;
    class CloseFrame extends Frame implements WindowListener
{
        Label label;

        CloseFrame(String title){
super(title);
        label = new Label("Close the frame.");
this.addWindowListener(this);
        }
        void launchFrame(){
setSize(300,300);
setVisible(true);
        }
        public void windowActivated(WindowEvent e){
        }
        public void windowClosed(WindowEvent e){
        }
        public void windowClosing(WindowEvent e){
            setVisible(false);
            System.exit(0);
        }
        public void windowDeactivated(WindowEvent e){
        }
        public void windowDeiconified(WindowEvent e){
        }
        public void windowIconified(WindowEvent e){
        }
        public void windowOpened(WindowEvent e){
        }
        public static void main(String args[]){
            CloseFrame cf = new CloseFrame("Close Window
Example");
            cf.launchFrame();
        }
    }
```

Tuliskan Outputnya

Percobaan 3 Close Frame dengan Inner Class:

Program 10-3

```
import java.awt.*;
import java.awt.event.*;

class CloseFrame extends Frame
{
    Label label;
    CloseFrame(String title){
super(title);
        label = new Label("Close the frame.");
this.addWindowListener(new CFListener());
    }
    void launchFrame()
    {
        setSize(300,300);
setVisible(true);
    }
    class CFListener extends WindowAdapter
    {
        public void windowClosing(WindowEvent e)
        {
            dispose();
            System.exit(1);
        }
    }
    public static void main(String args[])
    {
        CloseFrame cf = new CloseFrame("Close Window Example");
cf.launchFrame();
    }
}
```

Percobaan 4 Anonymous Inner class:

Program 10-4

```
      import java.awt.*;
      import java.awt.event.*;
      class CloseFrame extends Frame
      {
          Label label;
          CloseFrame(String title)
          {
              super(title);
              label = new Label("Close the frame.");
      this.addWindowListener(new WindowAdapter()
              {
                  public void windowClosing(WindowEvent e)
```

```
                {
                    dispose();
                    System.exit(1);
                }
            });
    }
        void launchFrame()
        {
            setSize(300,300);
    setVisible(true);
        }
        public static void main(String args[])
        {
            CloseFrame cf = new CloseFrame("Close Window
    Example");
    cf.launchFrame();
        }
    }
```

Tugas Tic-Tac-Toe

Extend program papan Tic-Tac-Toe yang telah Anda kembangkan sebelumnya dan tambahkan event handlers ke kode tersebut untuk membuat program berfungsi penuh.

Permainan Tic-Tac-Toe dimainkan dengan dua pemain. Pemain mengambil giliran mengubah. Setiap giliran, pemain dapat memilih kotak pada papan. Ketika kotak dipilih, kotak ditandai oleh simbol pemain (O dan X biasanya digunakan sebagai simbol). Pemain yang sukses menaklukkan 3 kotak membentuk garis horisontal, vertikal, atau diagonal, memenangkan permainan. Permainan akan berakhir ketika pemain menang atau ketika semua kotak telah terisi.

Pertemuan 13

Thread

Tujuan Intruksional :
Pokok Bahasan ini menjelaskan:

- Mengerti perbedaan state dalam threads
- Mengerti konsep prioritas dalam threads
- Mengetahui bagaimana menggunakan method didalam class Thread
- Membuat sendiri sebuah thread
- Menggunakan sinkronisasi pada thread yang bekerja bersama-sama dan saling bergantung satu dengan yang lainya
- Memungkinkan thread untuk dapat berkomunikasi dengan thread lain yang sedang berjalan
- Mengerti dan menggunakan kemampuan concurency

Kompetensi Yang Diharapkan :
Mahasiswa diharapkan dapat memahami Threads pada source code.

Waktu Pertemuan : 100 Menit

Latar Belakang

Sebelumnya Anda terbiasa untuk membuat program yang berurutan/sekuensial. Sebuah program sekuensial berarti sebuah program yang hanya memiliki satu aliran eksekusi. Setiap eksekusi, ia memiliki sebuah titik awal eksekusi, kemudian sebuah sekuen eksekusi, dan kemudian berakhir. Selama runtime, pasti hanya satu proses yang telah dieksekusi. Bagaimanapun juga, di dunia nyata, pasti dibutuhkan sesuatu yang dapat mengatur proses yang terjadi dan berjalan bersama-sama. Oleh karena itu, thread hadir untuk menjadi solusi dalam mengatasi permasalahan tersebut.Sebuah thread merupakan sebuah pengontrol aliran program. Untuk lebih mudahnya, bayangkanlah thread sebagai sebuah proses yang akan dieksekusi didalam sebuah program tertentu. Penggunaan sistem operasi modern saat ini telah mendukung kemampuan untuk menjalankan beberapa program. Misalnya, pada saat Anda mengetik sebuah dokumen di komputer Anda dengan menggunakan text editor, dalam waktu yang bersamaan Anda juga dapat mendengarkan musik, dan surfing lewat internet di PC Anda. Sistem operasi yang telah terinstal dalam computer Anda itulah yang memperbolehkan Anda untuk menjalankan multitaskting. Seperti itu juga sebuah program (ibaratkan di PC Anda), ia juga dapat mengeksekusi beberapa proses secara bersamasama(ibaratkan beberapa aplikasi berbeda yang bekerja pada PC Anda). Sebuah contoh aplikasi

adalah HotJava browser yang memperbolehkan Anda untuk browsing terhadap suatu page, bersamaan dengan mendownload object yang lain, misalnya gambar, memainkan animasi, dan juga file audio pada saat yang bersamaan.

Percobaan 1 Menulis Object Thread sebanyak 100x

Program 11-1

```
      class PrintNameThread extends Thread {
PrintNameThread(String name) {
super(name); // menjalankan thread dengan satu kali
instantiate start(); }
public void run() {
                String name = getName();
for (int i = 0; i < 100; i++) {
                System.out.print(name);
                    }
                }
}
     class TestThread {
           public static void main(String args[]) {
                 PrintNameThread pnt1 = new PrintNameThread("A");
                 PrintNameThread pnt2 = new PrintNameThread("B");
                 PrintNameThread pnt3 = new PrintNameThread("C");
                 PrintNameThread pnt4 = new PrintNameThread("D");
                       }
           }
```

Tuliskan Outputnya :	Jelaskan Logika dari coding diatas :

Percobaan 2 Implementasi Interface Runnable

Program 11-2

```
class PrintNameThread implements Runnable {
    Thread thread;
    PrintNameThread(String name) {
thread = new Thread(this, name);
thread.start();
            }
public void run() {
       String name = thread.getName();
for (int i = 0; i < 100; i++) {
       System.out.print(name);
       }
     }
}
    class TestThread {
        public static void main(String args[]){
new PrintNameThread("A");
new PrintNameThread("B");
new PrintNameThread("C");
new PrintNameThread("D");
            }
         }
```

Outputnya :	***Jelaskan Logika coding diatas :***

Percobaan 3 Mencetak String Tanpa Sinkronisasi

Program 11-3

```
          class TwoStrings {
               static void print(String str1, String str2) {
                   System.out.print(str1);
      try {
                       Thread.sleep(500);
                       }
                   catch (InterruptedException ie) {}
                   System.out.println(str2);
                   }    }
              class PrintStringsThread implements Runnable {
                  Thread thread;
                  String str1, str2;
                  PrintStringsThread(String str1, String str2){
      this.str1 = str1;
      this.str2 = str2;
      thread = new Thread(this);
      thread.start();
                  }
                  public void run() {
                      TwoStrings.print(str1, str2);
                      }
      }
          class TestThread {
              public static void main(String args[]) {
      new PrintStringsThread("Hello ", "there.");
      new PrintStringsThread("How are ", "you?");
                      new PrintStringsThread("Thank you ", "very much!");
                  }   }
```

Soal Tugas

Dengan menggunakan AWT atau Swing, buatlah sebuah banner sederhana yang akan mencetak string yang dituliskan oleh user. String ini akan ditampilkan secara terus menerus dan program Anda harus memberikan ilustrasi bahwa string tersebut bergerak dari kiri ke kanan. Untuk memastikan bahwa proses perpindahannya tidak terlalu cepat, Anda sebaiknya menggunakan method sleep dari class Thread.

Berikut ini adalah sebuah contoh dimana Anda menuliskan ”Your name here!”.