Buku Ajar

PEMROGRAMAN BERORIENTASI OBJEK

Penulis : Ida
Mhd. Faisal
Samsuriah
Musdalifa
Darniati
Rosnani
Nasir
Dikwan Moeis

PUBLISHER BY
PT. Inovasi Pratama Internasional

BUKU AJAR

PEMROGRAMAN BERORIENTASI OBJEK

PT Inovasi Pratama Internasional

PEMROGRAMAN BERORIENTASI OBJEK

Penulis : Ir.Ida,S,Kom,M.T, Muhammad Faisal, Samsuriah Musdalifa, Darniati,Rosnani, Nasir,Dikwan Moeis
ISBN :
Editor : Bincar Nasution, S.Pd., M.Pd.,C.Mt
Penyunting : Ali Amran Btr, S.Pd

Desain Sampul dan Tata Letak:

InoVal

Penerbit:

PT Inovasi Pratama Internasional

Anggota IKAPI Nomor 071/SUT/2022

Redaksi:

Jl. Cempaka No. 25 Padang Sidempuan 22725

Telp. +628 5360 415005

Email: cs@ipinternasional.com

Distributor Tunggal:

PT Inovasi Pratama Internasional

Jl. Cempaka No. 25 Padang Sidempuan 22725

Telp. +628 5360 415005

Email: info@ipinternasional.com

Cetakan Pertama, 18 Mei 2024

KATA PENGANTAR

Di dunia Bahasa pemrograman computer, cukup banyak pemain yang berada di dalamnya, antara lain Pascal, Basic, C, PHP, ASP, .NET, java. Java adalah Bahasa pemrorgaman yang saat ini banyak digunakan. Java menempati urutan kedua setelah bahasa C, sedangkan urutan lima besar lainnya setelah java adalah Objective -C, C++, dan C#. Indeks tersebut didasarkan atas banyaknya programmer, pelatihan serta pihak ketiga dengan menggunakan algoritma perhitunkan dari pencarian di Google, yahoo, MSN, dan Youtube.

Java merupakan salah satu bahasa pemrograman yang banyak diminati antara lain karena java unggul ketika digunakan dalam pembuatan aplikasi berbasis mobile, juga aplikasi yang berskala enterprise. Dengan demikian, ketika Anda menguasai bahasa Java maka dapat membuat berbagai macam aplikasi yang bejalan pada multiplatform, baik pada skala kecil maupun yang berskala besar.

Pada buku ini penulis akan membahas secara mudah, sistematis, dan detail tentang apa saja yang diperlukan untuk dapat mengusai bahasa Java. Selain itu penulis akan memberikan tutorial pembuatan program yang telah tertera pada buku ini sehingga pembaca dapat mempraktikan secara langsung.

Penulis

Ir.Ida,S,Kom,M.T

DAFTAR ISI

BAB I INSTALASI KOMPONEN NETBEANS

Tujuan

Mengidentifikasi bagian dasar dari program java

- Membedakan mana yang termasuk editor pada java.
- Mengembangkan program java sederhana menggunakan konsep pembelajaran pada bab ini dengan Netbeans.
- Menganalisa program java pertama pada pemrograman berorientasi objek

Pada akhir pembahasan, Diharapkan pembaca dapat :

1. Mengetahui cara Instalasi Java.
2. Melakukan Instalasi Java dengan menggunakan Neatbeans
3. Membuat program Java pada Netbeans.

Netbeans merupakan salah satu proyek open source yang disponsori oleh SUN Microsystem. Proyek ini berdiri pada tahun 2000 dan telah menghasilkan 2 produk, yaitu netbeans IDE (Integrated Development Environment) dan Netbeans Platform. Netbeans IDE (Integrated Development Environment) merupakan produk yang berguna untuk melakukan pemrograman baik menulis kode, mengompilasi, mencari kesalahan dan mendistribusikan program. Sedangkan Netbeans platform adalah sebuah modul yang merupakan kerangka awal/pondasi dalam membangun aplikasi desktop yang besar.

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, PFP, JavaScript dan Ajax, Ruby on Rails, Groovy dan Grails, dan C/C++. Proyek Netbeans didukung oleh komunitas pengembang yang ekstensif dan menawarkan dokmnetasi dan sumber daya pelatihan serta beragam pilihan plugin pihak ketiga. 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.

I. Instalasi Java Development Kit (JDK).

JDK(Java Development Kit) adalah aplikasi untuk pengembang bahasa Java, sebelum mempelajari bahasa Java kita diharuskan untuk menginstal JDK ini.
Dalam buku ini versi jdk yang digunakan adalah jdk-6u19-windows-i586.

-  jdk-6u19-windows-i586.exe
- Klik ganda file setup JDK "jdk-6u19-windows-i586.exe" untuk melakukan instalasi .

Gambar 1.1 Tahap awal proses instalasi JDK

- Klik tombol **Decline** untuk melanjutkan , maka tampil form pilihan fitur java yang akan digunakan.

Gambar 1.2 Proses pemilihan feature JDK

- Klik tombol **Next** untuk melanjutkan, disarankan gunakan pilihan fitur secara standart.

Gambar 1.3 Proses installing JDK

- Pada bagian akhir proses instalasi klik tombol **Finish**.

Gambar 1.4 Tahap akhir instalasi JDK

Komponen JDK antara lain compiler(javac), interpreter(java) disebut juga java virtual machine atau java runtime environment, applet viewer (appletviewer), debugger(jdb), java class library (jcl), header dan stub generator (javah), dan yang penting yaitu java documentation(javadoc).

Penjelasan penggunaan komponen JDK :

1. Kompilator (javac)

   Berfungsi untuk kompilasi file source code : *.java menjadi *.class.

   Syntax umum : javac nama_file.java.

2. Interpreter (java)

   Bertugas untuk menjalankan bytecode (*.class).

Syntax umum : java nama_file.class.

3. Applet Viewer

   Digunakan untuk menjalanakan applet viewer, namun sekarang sudah digantikan browser.

   Syntax umum : appletviewer nama_file.html

4. Java Debugger

   Bertugas melakukan debugging.

   Syntax umum : jdb option.

5. Java Class File Diassembler (javap)

   Bertugas membuat daftar method dan attribute public dari suatu kelas.

   Syntax : javap namaKelas.

6. Java Header and Stub Generator

   Bertugas menerjemahkan bahasa yang ditulis dalam bahasa Java menjadibahasa pemrograman C.

   Syntax umum : javah_namaKelas

7. Java Documentation Generator

   Menampilkan pustaka kelas, interface, constructor, dan method standart yangtelah dibuat vendor.

8. Source Code Java API

   Source code ini dapat diperoleh dari file src.zip.

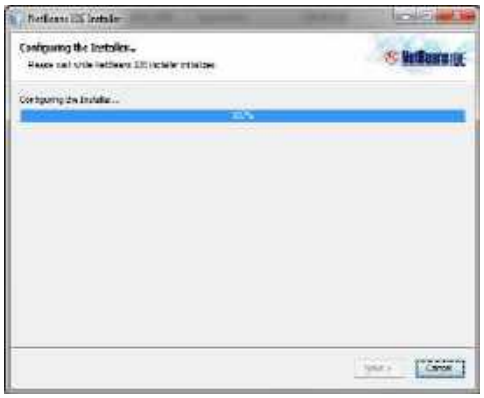
II. Instalasi IDE Netbeans 6.9.1

- 

- **Netbeans** bisa di download melalui situs http://www.netbeans.org dan **iReport** di http://ireport.sourceforge.net/.
- Klik ganda file setup **netbeans-6.9.1-ml-windows**. Sehingga akan muncul proses instalasi seperti gambar berikut.

Gambar 1.5 Proses awal instalasi netbeans.

- Tunggu proses instalasi di atas hingga selesai, hingga muncul seperti gambar di bawah ini.

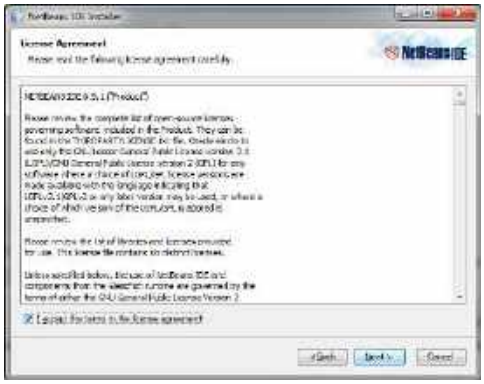


Gambar 1.6 Proses konfigurasi netbeans.

- Selanjutnya akan muncul seperti gambar dibawah ini

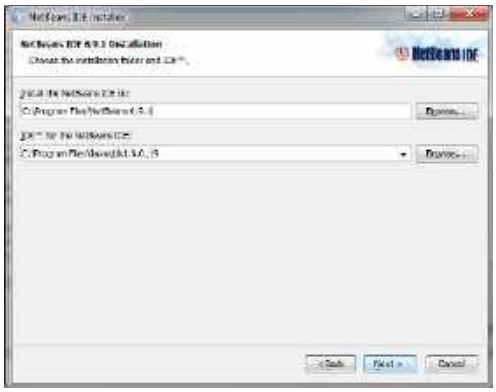
Gambar 1.7 Pilihan feature pada netbeans .

- Selanjutnya klik **Next**, hingga muncul seperti gambar di bawah ini. Beri tanda centang pada **I accept the terms…,** lalu klik **Next**.



Gambar 1.8 License Agreement.

- Tentukan directory Instalasi, lalu klik **Next**.



Gambar 1.9 Directory dan Kapasitas Instalasi.

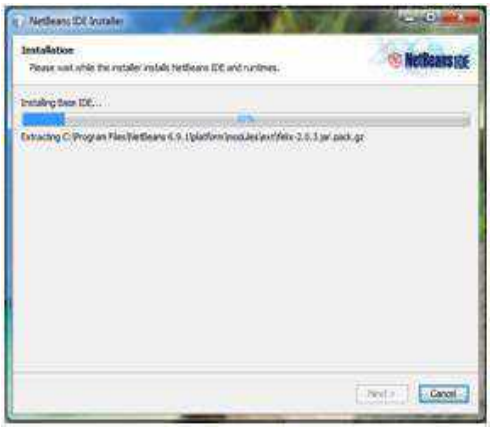
- Tentukan folder instalasi untuk **glassfish** sebagai server JavaDB, aplikasi **glassfish** berfungsi sebagai server local bagi database yang akan dikoneksikan ke netbeans.

Gambar **1.10** Directory Instalasi glassfish.

- Klik **Next**,maka muncul seperti gambar dibawah ini .

Gambar 1.11 Informasi total kapsitas install .

- Klik **Install**, tunggu proses instalasi, lamanya proses tergantung seberapa banyak paket yang di install. jika selesai klik Next.



Gambar 1.12 Proses Extract file jar.

- Hingga proses selesai maka muncul seperti gambar dibawah ini

Gambar 1.13 Tahap akhir Instalasi netbeans.

- Hapus tanda centang jika ingin mengabaikan register dan kontribusi produk Netbeans. Langkah terakhir klik **Finish**.
- Untuk menjalankan,klik ganda shortcut Netbeans 6.9.1.

- Tampilan utama jendela netbeans seperti gambar berikut.

Gambar 1.14 Tampilan utama Netbeans IDE 6.9.1

III. Komponen Netbeans IDE 6.9.1

A. Java DB

JavaDB merupakan Relation Database Manajement System (RDMS), yang ditulis dalam bahasa Java. Java DB atau yang disebut proyek derby merupakan proyek open source dibawah Apache Software Foundation (ASF). Java DB Atau Derby ditulis dan diimplementasikan secara lengkap pada pemrograman java. JavaDB menyediakan pengguna dengan mesin utama database yang kecil (Standart Database) yang banyak disertakan kedalam banyak solusi yang didasarkan pada java. JavaDB menjaga integritas data dan menyediakan dukungan terhadap transaksi yang canggih. JavaDB telah menjadi satu paket bersama JDK , sehingga pada saat Instalasi JDK JavaDB sudah disertakan dalam JDK Tersebut.

B. System Requirement Java DB

Dikarenakan javaDB dirancang dengan bahasa java, maka setiap database akan berjalan pada komputer yang telah terinstall java Virtual Machine (JVM) / Java Runtime Environment (JRE). JVM/JRE yang dibutuhkan adalah versi 1.4.2 atau yang lebih baru.

BAB II AREA KERJA NETBEANS

Tujuan

Mengidentifikasi bagian dasar dari program java

- Pokok bahasan ini menjelaskan tentang area kerja Netbeans.
- Mengetahu area kerja Netbeans pada Panel Project.
- Mengetahu area kerja Netbeans pada Panel File.
- Mengetahu area kerja Netbeans pada Panel Services.
- Mengetahu area kerja Netbeans pada Panel Navigator.
- Mengetahuai beberapa panel dan tab swing pada netbeans.

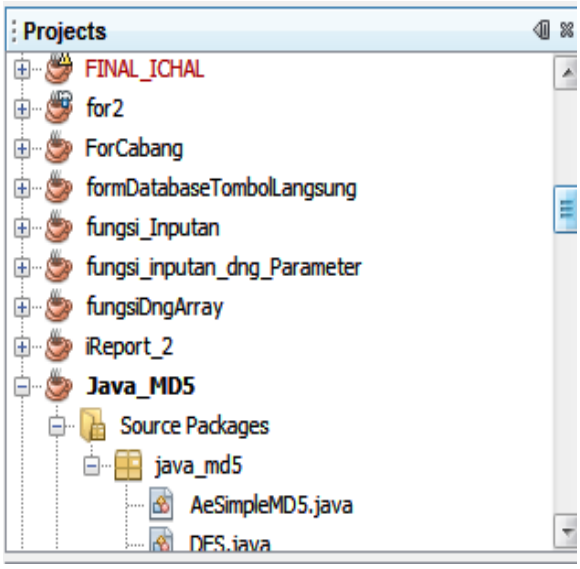
Pada akhir pembahasan, Diharapkan pembaca dapat :

1. Mengetahui area kerja Neatbeans.
2. Memahami langkah-langkah pada editor Netbeans.

Sebelum menggunakan aplikasi netbeans terlebih dahulu kita mengetahui fungsi tiap jendela / layar yang terdapat didalamnya, guna mamaksimalkan proses kerja terutama saat merancang suatu aplikasi baik berbasis desktop maupun web.

I. Panel Projects.

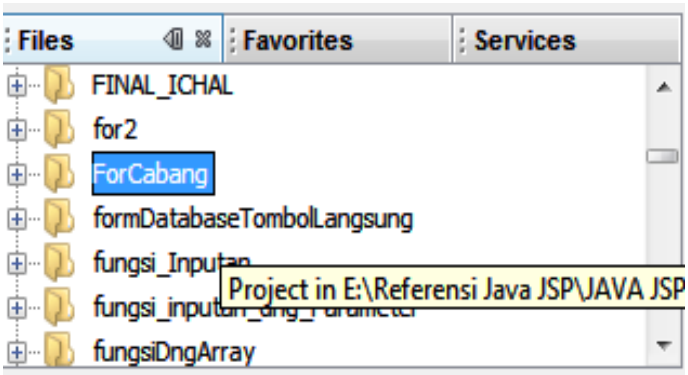
Panel projects digunakan untuk menampilkan informasi proyek yang telah dibuat. Informasi yang ditampilkan berupa source packages yang menampilkan file – file yang telah dibuat beserta packagenya. Sedangkan test packagesakan menampilkan file test Junit. Libraries menampilkan library JDK yang digunakan dalam pembuatan aplikasi.



Gambar 2.1 Panel Project

II. Panel Files.

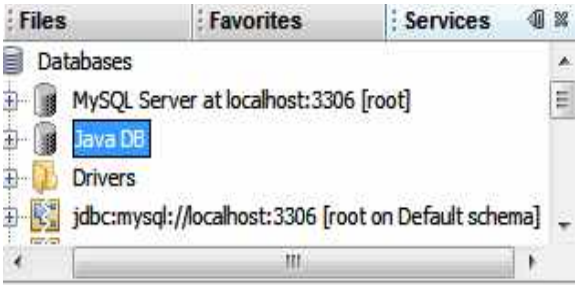
Panel files berguna untuk menampilkan file – file dalam sebuah proyek, baik file java (*.java) atau bytecode (*.class). melalui panel Project dan Files, kita dapat membuka file yang telah dibuat pada area kerja.



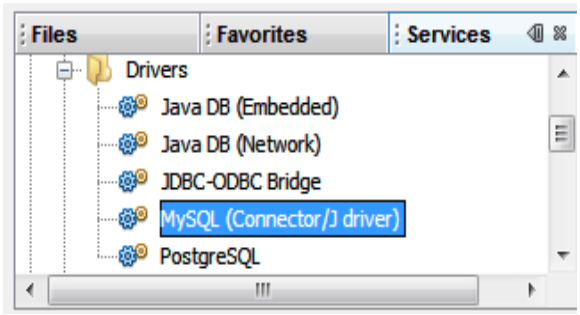
Gambar 2.2 Panel Files

III. Panel Services.

Panel service berfungsi menampilkan informasi tentang service yang disediakan atau telah diregister kedalam netbeans. Tab databases digunakan untuk menampilkan informasi aplikasi database yang digunakan. Secara default, database JavaDB telah diregister kedalam netbeans, selain itu pada tab databases juga menampilkan driver yang telah diregister, koneksi yang digunakan oleh database, nama database serta tabel yang telah dibuat sebelumnya. Melalui tab ini dapat membuat, menghapus database atau tabel.



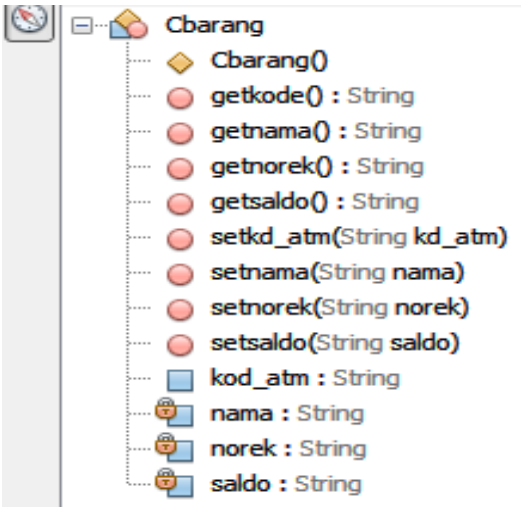
Gambar 2.3 Panel Services



Gambar 2.4 Driver IDE Netbeans

IV. Panel Navigator

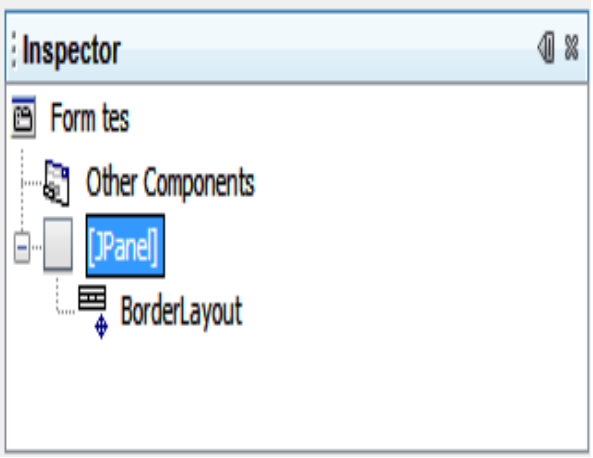
Panel navigator akan muncul apabila kita membuat proyek dan mengaktifkan salah satu document / file. Panel ini digunakan untuk menampilkan kelas dan member / method yang ada pada file. Kita dapat menuju sebuah method dengan mengklik anggota method yang tampil pada panel navigator



Gambar 2.5 Panel Navigator

V. Panel Inspector

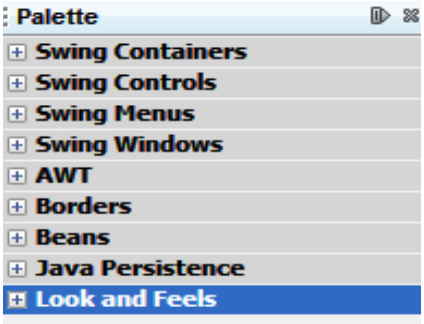
Panel Inspector akan tampil apabila kita mengaktifkan document yang mengandung container/pemrograman grafis (GUI). Panel ini menampilkan komponen yang digunakan oleh file yang bersangkutan seperti kontainer, kontrol, menu, border dll.



Gambar 2.6 Panel Inspector

VI. Panel Palette

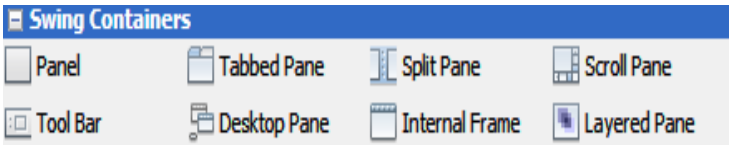
Panelpalette merupakan panel yang menyediakan tool – tool untuk membuat tampilan grafis (GUI). Panel palette dikenal dengan Gui Builder Matisse. Tool ini juga dibagi menjadi beberapa kategori, dimana setiap kategori menyediakan tool GUI Builder sesuai dengan kategorinya. Untuk menggunakannya, kita tinggal menyeret tool kedalam area design (Form).



Gambar 2.7 Panel Palette.

VII. Tab Swing Containers

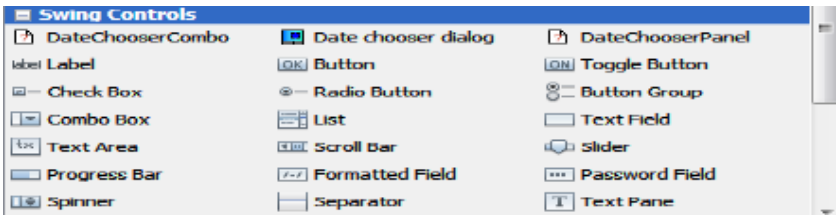
Tab Swing digunakan untuk mengategorikan container swing (komponen utama untuk menampung komponen lain) seperti **jPanel, jTabbed Pane, jSplit Pane, jScrol Pane, jTool Bar**, dll.



Gambar 2.8 Tab Swing Container.

VIII. Tab Swing Controls

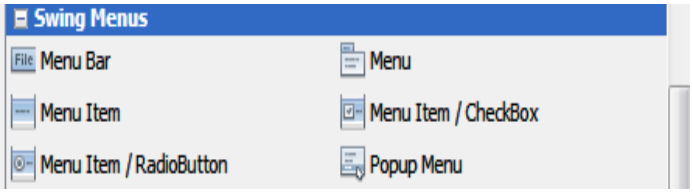
Tab swing controls menyediakan tool untuk membuat komponen operasi seperti **jLabel, jButton, jText Field,** dll



Gambar 2.9 Tab Swing Controls

IX. Tab Swing Menus

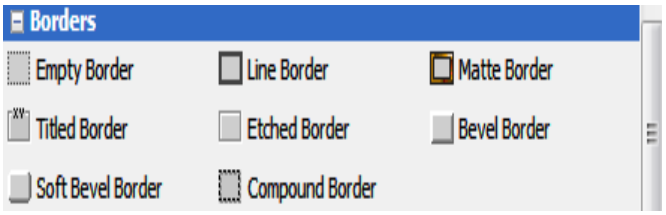
Tab swing menu menyediakan tool untuk membuat komponen menu, baik **Menu Bar**maupun **Menu Popup**



Gambar 2.10 Tab Swing Menus

X. Tab Borders

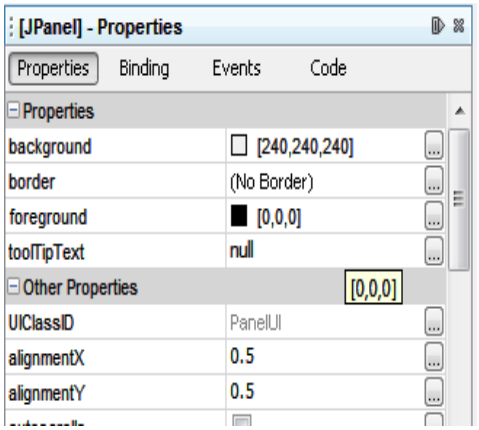
Tab border digunakan untuk membuat border / garis tepi suatu komponen atau container. Container yang paling sering digunakan adalah Container panel.



Gambar 2.11 Tab Borders

XI. Panel Properties

Panel properties berfungsi untuk menampilkan properti komponen yang aktif / terseleksi. Melalui panel ini, kita akan mengatur properti yang dimiliki oleh suatu komponen.



Gambar 2.12 Panel Properties

BAB III STRUKTUR PEMROGRAMAN JAVA

Tujuan

Mengidentifikasi bagian dasar dari program java

- Membedakan struktur yang ada pada pemrograman java.
- Mengembangkan program java sederhana menggunakan konsep pembelajaran pada bab ini dengan struktur pemrograman pada java.
- Memahami dan mempraktekkan konsep Class, Objek, Inheritansi di java

Pada akhir pembahasan, Diharapkan pembaca dapat :

1. Mengetahui struktur pemrograman Java.
2. Mengetahui elemen Class di Java
3. Membuat program pada Netbeans dengan menggunakan struktur java.

Java mempunyai struktur untuk menuliskan sintaks- sintaksnya dimulai dari **package, import, class,method**.Adapun fungsi tiap sintaks secara garis besar adalah sebagai berikut :

I. Package

Package adalah suatu cara pengelompokan dan pengorganisasian kelas ke dalam suatu library. Package bekerja dengan membuat directory dan folder baru sesuai dengan penamaan package, kemudian menyimpan file class pada folder tersebut. Deklarasi package dituliskan pada baris paling atas sebelum perintah import, sebagaimana terlihat pada struktur program java. package akan secara otomatis terbentuk pada saat anda membuat project.

Contoh:

```
package
uJI_FORM_PENERBANGAN;
import javax.swing.JFrame;
```

II. Import

Perintah Import digunakan untuk memberitahukan kepada program untuk mengacu pada kelas – kelas yang terdapat pada package tersebut bukan menjalankan kelas tersebut. Dalam program, kita dapat mengimport hanya kelas tertentu dan dapat pula mengimport semua kelas yang terdapat pada package.untuk mengimport semua kelas dapat menggunakan tanda asterisk.

Contoh :

```
package uJI_FORM_PENERBANGAN;
import javax.swing.JFrame;
import  sql.*;
```

III. Class

Class sebagaimana pada bahasa C/C++ merupakan bagian utama pada pemrograman java, didalam body class ini diidentifikasikan variabel, method, dan kelas inner. Deklarasi kelas otomatis terbentuk saat anda membuat file java.

Deklarasi class dapat dilakukan dengan sintaks sebagai berikut :

<modifier> class <nama_class> {

[deklarasi_atribut]

[deklarasi_konstruktor]

[deklarasi_metode]

}

Deklarasi atribut dapat dilakukan dengan sintaks sebagai berikut :

<modifier> <tipe> <nama atribut>;

Contoh:

```
public class daftartiket {
public  String Kode_Penerbang;
 private String Harga;
 public daftartiket() { }    public String
getKode_Penerbang()

{   return Kode_Penerbang;   }}
```

IV. Method

Method adalah bagian program yang menjelaskan tingkah laku dari objek yang akan diinstance. Method tidak dapat berdiri sendiri sebagaimana kelas,

dimana letak penulisan berada didalam body kelas. Method berdasarkan jenisnya dapat dibagi menjadi beberapa bagian seperti : konstruktor, fungsi/prosedure dan main.

Deklarasi metode dapat dilakukan dengan sintaks sebagai berikut :

<modifier> <return_type> <nama_metode> ([daftar argumen])

{

[<statement>]

}

Dalam java terdapat dua buah metode

1. Fungsi (Non Void), merupakan metode yang memiliki nilai balik jika metode tersebut dipanggil, cara pembuatan sebuah fungsi adalah dengan cara menentukan nilai baliknya, lalu membuat nama metodenya.

```
Class Manusia { String nama;
// fungsi
      Public String ambilNama() {
// untuk mengembalikan nilai gunakan kata kunci return
Return nama;
}
}
```

Contoh :

```
public ResultSet Method_isi_KdTiket(int x)
{
 Connection koneksi=KoneksiTiket.getConnection();
 ResultSet rs=null;
 try{
          Statement st=koneksi.createStatement();
          String sql="select * from tiket where
          kd_penerbang='"+x+"'";
          rs=st.executeQuery(sql);
}
       catch (Exception e)
     {
JOptionPane.showMessageDialog(null,e.getMessage()
,"Error",0);
       }
       return rs;
}
```

2. Prosedur (Void), merupakan metode yang tidak memiliki nilai balik, cara pembuatan prosedur sama dengan fungsi namun bedanya, nilai baliknya menggunakan kata kunci void.

Saat membuat sebuah fungsi maka untuk mengembalikan nilainya, harus menggunakan kata kuci return, diikuti nilai yang akan dikembalikannya. Untuk mengambil nilai balik dari fungsi.

V. Method Instance

Variabel dan method dalam obyek java secara formal diketahui sebagai variabel instance dan method instance. Hal ini dilakukan untuk membedakan dari variabel class dan method class. Class adalah sturktur dasar dari OOP,class terdiri dari dua tipe dari anggota dimana disebut dengan field (attribut/properti) dan method. Field merupakan tipe data yang didefinisikan oleh class, sementara method merupakan operasi. Sebuah obyek adalah sebuah instance (keturunan) dari class.

Contoh :

```
package metod_instance;

public class kelas_instance {

String nama, nim, belajar, olahRaga, makan, minum;

public void nama  (){System.out.println("Nama mahasiswa : "
+nama);}

public void nim (){System.out.println("Nim : " +nim);}

public void belajar () {System.out.println("Belajar " +belajar);}

public void olahRaga () {System.out.println("Olah raga :" +olahRaga);}

public void makan (){System.out.println("Makanan : " +makan);}

public void minum (){System.out.println("Minuman : " +minum);}

}
```

Class Kelas_instance diatas berisi variabel dan beberapa class instance yang berdiri sendiri dari kelas induk, namun pada penggunaanya pada method main dibentuk objek yang berisi varibel dan method yang berada pada class kelas_instanse. Hingga pada penggunaannya isi dari variabel dapat berubah - ubah dan dapat

dibentuk objek baru dari class kelas_instanse tersebut tanpa merubah sintaks yang ada pada kelas tersebut.

Contoh :

```
package metod_instance;
public class Main {
public static void main(String[] args) {
kelas_instance mahasiswa1= new kelas_instance();
kelas_instance mahasiswa2= new kelas_instance();
kelas_instance mahasiswa3= new kelas_instance();

mahasiswa1.nama="Arthur";
mahasiswa1.nim="01094034";
mahasiswa1.belajar="Java";
mahasiswa1.olahRaga="PS3,Catur";
mahasiswa1.makan="Ikan Bakar";
mahasiswa1.minum="Juice Wortel\n";
```

VI. Method Main

Method main adalah method utama yang paling pertama dipanggil untuk menjalankan program. Sebuah program yang tidak mempunyai method main tidak akan bisa dieksekusi/ dijalankan.

Contoh:

```
package uJI_FORM_PENERBANGAN;
public class Main {
   public static void main(String[] args) {
      MENUUTAMAFRAME eni= new MENUUTAMAFRAME();
      eni.setVisible(true);
   }
```

- **public static,** modifier public berarti method tersebut dapat dibaca oleh setiap kelas, sedangkan static berarti method main hanya dapat diakses oleh kelas itu sendiri.
- **void,** berartimethod main tidak mengembalikan sebuah nilai.
- **String[ ] args** dapat juga ditulis **String args[ ]** merupakan parameter input method main.

VII. Class Abstrac

Class abstrac merupakan hirarki tertinggi dari suatu kelas. Kelas ini berisi variabel – variabel umum dan deskripsi method tanpa disertai implementasi. Kelas ini merupakan basis dari penurunan kelas – kelas lainnya, sehingga tidak dapat diInstance secara langsung menjadi object. Kelas turunanlah yang nantinya bertugas mendefenisikan method secara detail.

Contoh:

```
public abstract class daftarPenerbangan {
public  String Kode_tiket="";
private String Kode_penerbang="";
private String Nama_penerbang="";

    public daftarPenerbangan() {
    }
 public String getKode_tiket(){
        return Kode_tiket;
    }
    public void setKode_tiket(String Kode_tiket){
        this.Kode_tiket=Kode_tiket;
    }
 public String getKode_penerbang(){
        return Kode_penerbang;
    }
    public void setKode_penerbang(String
Kode_penerbang){
        this.Kode_penerbang=Kode_penerbang;
    }
```

Setelah pembentukan kelas abstrac , penggunaan method serta pengisian variabel dilakukan pada class lainnya. Pada umumnya pengisian dilakukan pada class main atau class dalam suatu form, dengan mengimport jika berada pada package yang berbeda.

Contoh :

```
import data.daftarPenerbangan;
public class Penerbangan extends javax.swing.JFrame
{
private void
SAVEActionPerformed(java.awt.event.ActionEvent evt)
{
      Kode_tiket = (String)kd_tiket.getText();
      Kode_penerbang=(String)kd_penerbang.getText();
     Nama_penerbang=(String)nm_penerbang.getText();
}
```

VIII. Inheritansi

Konsep inheritance ini mengadopsi dunia riil dimana suatu entitas/obyek dapat mempunyai entitas/obyek turunan. Dengan konsep inherintance, sebuah class dapat mempunyai class turunan. Suatu class yang mempunyai class turunan dinamakan parent class atau base class. Sedangkan class turunan itu sendiri seringkali disebut subclass atau child class. Suatu subclass dapat mewarisi apa apa yang dipunyai oleh parent class-nya, sehingga member dari suatu subclass adalah terdiri dari apa-apa yang ia punyai dan juga apa-apa yang ia warisi dari class parent-nya. Kesimpulannya, boleh dikatakan bahwa suatu subclass adalah tidak lain hanya memperluas (extend) parent class-nya.

Di dalam Java untuk mendeklarasikan suatu class sebagai subclass dilakukan dengan cara menambahkan kata kunci extends setelah deklarasi nama class, kemudian diikuti dengan nama parent class-nya. Kata kunci extends tersebut memberitahu compiler Java bahwa kita ingin melakukan perluasan class. Berikut adalah contoh deklarasi inherintance :

Contoh :

```
Public class B extends A{
…
}
```

Contoh diatas memberitahukan compiler Java bahwa kita ingin meng-extend class A ke class B. Dengan kata lain, class B adalah subclass (class turunan) dari class A, sedangkan class A adalah parent class dari class B.

Java hanya memperkenankan adanya single inherintance. Konsep single inheritance hanya memperbolehkan suatu subclass mempunyai satu parent class. Dengan konsep single inheritance ini, masalah pewarisan akan dapat diamati dengan mudah.

Dalam konsep dasar inheritance dikatakan bahwa suatu subclass adalah tidak lain hanya memperluas (extend) parent class-nya.

Pengaksesan member yang ada di parent class dari subclass-nya tidak jauh berbeda dengan pengaksesan member subclass itu sendiri(Elenia et al., 2020).

Suatu parent class dapat tidak mewariskan sebagian member-nya kepada subclass-nya. Sejauh mana suatu member dapat diwariskan ke class lain, ataupun suatu member dapat diakses dari class lain, sangat berhubungan dengan access control (control pengaksesan).

Di dalam java, control pengaksesan dapat digambarkan dalam tabel berikut ini :

Modifier	Class yang sama	Package yang sama	Subclass	Class manapun
private	√			
default	√	√		
protected	√	√	√	
public	√	√	√	√

Kata kunci *super* dipakai untuk merujuk pada member dari parent class, sebagaimana kata kunci this yang dipakai untuk merujuk pada member dari class itu sendiri. Adapun format penulisannya adalah sebagai berikut :

Super data_member	% merujuk pada data member pada parent class
Super function_member()	% merujuk pada function member pada parent class
Super()	% merujuk pada konstruktor pada parent class

Inharitansi adalah proses pewarisan variabel dan methhod dari suatu kelas kepada kelas yang lain. Kelas super juga disebut kelas induk sedangkan subkelas biasa disebut kelas anak. Subkelas mewarisi semua metode dan variabel superkelasnya dan secara otomatis superkelas memberikan perilakunya ke subkelas pewaris.

Adapun cara untuk membuat kelas turunan dengan menggunakan kata kunci **extends.**

Contoh:

- Kelas Sistem_informasi yang merupakan subkelas Stmik_Profesional. Pada data kelas Sistem_informasi ditambah data alamat2, jurusan, jenjang dan jumlah mahasiswa.

```
public class Stmik_Profesional {
String kampus;  String alamat1;
String jurusan1;  String jurusan2;
String jurusan3;
String jenjang;
int total_D3;
public Stmik_Profesional()
   {
   kampus="";
   alamat1="";
   jurusan1="";
   jurusan2="";
   jurusan3="";
   jenjang="";
   total_D3=0;
   }
public Stmik_Profesional( String newKampus,String
newAlamat1,String newJurusan1, String newJurusan2,String
newJurusan3,String newJenjang,int Tot_D3)
{
kampus=newKampus;
alamat1=newAlamat1;
jurusan1=newJurusan1;
jurusan2=newJurusan2;
jurusan3=newJurusan3;
jenjang=newJenjang;
total_D3=Tot_D3;
}
```

Lanjutan....

```
public void setkampus(String newKampus)
{ kampus = newKampus;}

public String getKampus()
{ return kampus;}

public void setalamat1(String newAlamat1)
{ alamat1 = newAlamat1; }

public String getAlamat1()
{ return alamat1; }

public void setjurusan1(String newjurusan1)
{ jurusan1 = newjurusan1; }

public String getJurusan1()
{ return jurusan1; }

public void setjurusan2(String newjurusan2)
{ jurusan2 = newjurusan2;}

public String getJurusan2()
{ return jurusan2; }
```

Lanjutan....

```
public void setjurusan3(String newjurusan3)
{ jurusan3 = newjurusan3;}

public String getJurusan3()
{ return jurusan3; }

public void setjenjang(String newjenjang)
{ jenjang = newjenjang; }

public String getJenjang()
{ return jenjang;}

public void settotal_D3(int tot_D3)
{ total_D3=tot_D3; }

public int getTotal_D3()
{ return total_D3;}

static void isi_super(){
Stmik_Profesional Prof = new Stmik_Profesional("STMIK
Profesional Makassar","Jl A.P Pettarani No.27","Manajement
Informatika","Teknik Komputer","Komputerisasi
Akuntansi","Diploma 3",1500); }}
```

Pembentukan class Stmik_Profesional sebagai kelas super merupakan kelas yang berisi method dan variabel yang berhubungan dengan biodata kampus satu. Untuk menambahkan data mengenai kampus dua tanpa harus merubah isi dari class Stmik_Profesional, di buatlah class baru yaitu class Sistem_informasi dimana constructor yang ada pada class induk(Stmik_Profesional) tinggal ditambahkan dengan data yang diinginkan.

Pada class Sistem_informasi sebagai subkelas dilakukan deklarasi pengisian constructor secara keseluruhan hingga ditampilkan.

```
public class Sistem_Informasi  extends  Stmik_Profesional{
   String jurusan4;    String alamat2;
   String Jenjang  ;    int Jum_S1;
   public Sistem_Informasi ()
   {
super();
   jurusan4="";
   alamat2="";
   Jenjang="";
   Jum_S1=0;
   }

public Sistem_Informasi(String newKampus,String
newAlamat1,String newJurusan1, String newJurusan2,String
newJurusan3,String newJenjang,int Tot_D3, String
newjurusan4,String newAlamat2,String new_jenjang,int
jumlah_S1)
   {
super (newKampus,newAlamat1,newJurusan1,
            newJurusan2,newJurusan3,newJenjang,Tot_D3);
         jurusan4=newjurusan4;
         alamat2=newAlamat2;
         Jenjang=new_jenjang;
         Jum_S1=jumlah_S1;
   }

public void setjurusan4(String newJurusan4)
{
jurusan4=newJurusan4;
}
public String getJurusan4()
{
return jurusan4;
}
```

Lanjutan....

```
public void setalamat2(String newalamat2)
{ alamat2 = newalamat2; }
public String getAlamat2()
{ return alamat2; }

public void setJenjang(String newJenjang)
{ Jenjang = newJenjang; }
public String getJenjang2()
{ return Jenjang;}

public void setjuml_S1(int jumlah_S1)
{ Jum_S1 = jumlah_S1;}
public int getJumlah_S1()
{ return Jum_S1; }

 public static void Sistem_Informasi(String[] args) {
Sistem_Informasi Prof = new Sistem_Informasi("STMIK
Profesional Makassar","Jl A.P Pettarani No.27","Manajement
Informatika","Teknik Komputer","Komputerisasi
Akuntansi","Diploma 3",1500,"Sistem Informasi","Jl G.Latimojong
18","Strata 1",500);
System.out.println("=============KAMPUSKU==========");
System.out.println("Nama                    \t: "+Prof.getKampus());
System.out.println("Alamat 1                \t: "+Prof.getAlamat1());
System.out.println("Jenjang                 \t: "+Prof.getJenjang());
System.out.println("Jurusan1                \t: "+Prof.getJurusan1());
System.out.println("Jurusan2                \t: "+Prof.getJurusan2());
System.out.println("Jurusan3                \t: "+Prof.getJurusan3());
System.out.println("Jumlah Mahasiswa\t: "+Prof.getTotal_D3());
System.out.println("Alamat 2                \t: "+Prof.getAlamat2());
System.out.println("Jurusan                 \t: "+Prof.getJurusan4());
System.out.println("Jenjang                 \t: "+Prof.getJenjang2());
System.out.println("Jumlah Mahasiswa\t: "+Prof.getJumlah_S1());
   }
}
```

IX. Enkapsulasi

Kita dapat menyembunyikan information dari suatu class sehingga anggota-anggota class tersebut tidak dapat diakses dari luar. Adapun caranya adalah cukup dengan memberikan akses kontrol private ketika mendeklarasikan suatu atribut atau method. Encapsulation (Enkapsulasi) adalah suatu cara untuk menyembunyikan implementasi detail dari suatu class(Elenia et al., 2020).

Enkapsulasi mempunyai dua hal mendasar, yaitu :

- Information hiding
- Menyediakan suatu perantara (method) untuk pengaksesan data

Enkapsulasi merupakan cara membungkus data dengan method yang membangun suatu kelas. Dalam java enkapsulasi mempunyai dua method, yaitu method yang diawali **set** dan **get**. awalan set menunjukan bahwa method digunakan untuk memberikan nilai pada variabel, sedangkan get method merupakan get yang digunakan untuk mendapatkan nilai dari variabel.

```
package enkapsulasi_java;
public class klas_Variabel {
   private String nim="";
   private String nama="";
   private String kelas="";

   public String getnim()
   {  return nim; }

    public void setnim(String nim)
    {  this.nim=nim;  }

public String getnama()
   { return nama; }

    public void setnama(String nama)
    { this.nama=nama; }

    public void setkelas(String kelas)
    { this.kelas=kelas;  }

    public String getkelas()
    {
        return kelas;
    }
}
```

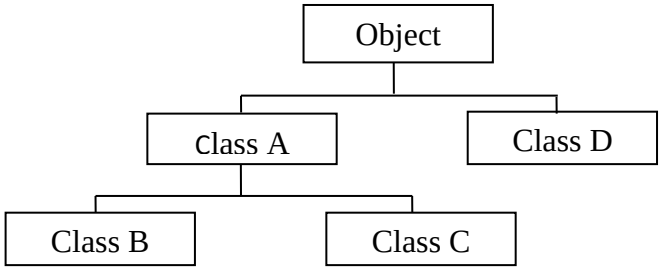
Setelah dilakukan pengkapsulan, proses pengisian nilai dan deklarasi dilakukan pada class Main.

```
package enkapsulasi_java;
public class Main
{
    public static void main(String[] args)
{
        klas_Variabel biodata = new klas_Variabel();
        biodata.setnim("01094034");
    biodata.setnama("Muhammad Faisal Palawa");
        biodata.setkelas("SI - 41");

        System.out.println("====Biodata Mahasiswa====");
        System.out.println(
" Nim :"+biodata.getnim()+"\n
Nama :"+biodata.getnama()+"\n
Kelas : "+biodata.getkelas());
   }
}
```

X. Polymorfisme

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 yang 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.

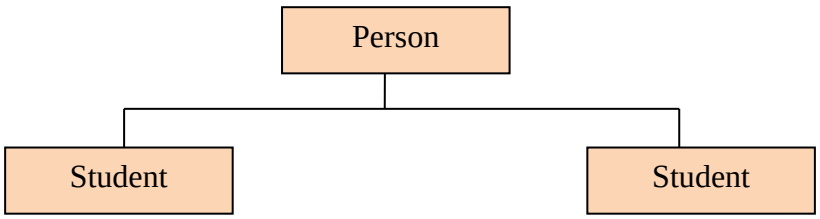
Dalam Java, semua class, termasuk class yang membangun Java API, adalah subclasses dari superclass Object. Contoh hirarki class diperlihatkan di bawah ini. Beberapa class diatas class utama dalam hirarki class dikenal sebagai superclass. Sementara beberapa class di bawah class pokok dalam hirarki class 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 subclass. 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 mendefiniskan 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(FT et al., 2018).

Contoh lain yang menunjukkan properti polimorfisme adalah ketika kita mencoba melalui referensi ke method. Misalkan kita punya method statis print Information 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.

Polymorfisme adalah istilah yang digunakan untuk menjelaskan bagaimana suatu objek dapat menerima banyak bentuk. Dalam java polymorfisme adalah sebuah method yang mempunyai nama sama namun parameternya berbeda.

Contoh:

```
package polymorfisme_ku;
public class Main{
    public String memory;
    public String Harddisk;
    public int harga;

    public Main (String newmemory,String newharddisk,int
newharga){
    memory = newmemory;
    Harddisk = newharddisk;
    harga  = newharga;
    }
    public static void main(String[] args) {
        Main Toshiba = new Main("4 GB","500 GB",5000000);
        Main Axio = new Main("2 GB","300 GB",4700000);
        System.out.println("Daftar Harga Notebook");

        System.out.println("Spesifikasi Toshiba :.....");
        System.out.println("Memory   \t: "+Toshiba.memory);
        System.out.println("Harddisk \t: "+Toshiba.Harddisk);
        System.out.println("Harga    \t: "+Toshiba.harga);

System.out.println("==============================");
        System.out.println("Spesifikasi AXIO :.....");
        System.out.println("Memory   \t: "+Axio.memory);
        System.out.println("Harddisk \t: "+Axio.Harddisk);
        System.out.println("Harga    \t: "+Axio.harga);
    }
}
```

Pada penggunaanya spesifikasi tiap notebook dapat diatur melalui method **Main** dengan membentuk objek baru sesuai dengan merknya dan antara tiap objek berisi nilai yang berbeda.

BAB IV MENGGUNAKAN NETBEANS 6.9.1

Tujuan

Mengidentifikasi bagian dasar dari program java

- Membedakan editor yang ada pada pemrograman java.
- Mengembangkan program java menggunakan konsep pembelajaran pada bab ini dengan menggunakan Netbeans.
- Menganalisa program java pada pemrograman berorientasi objek

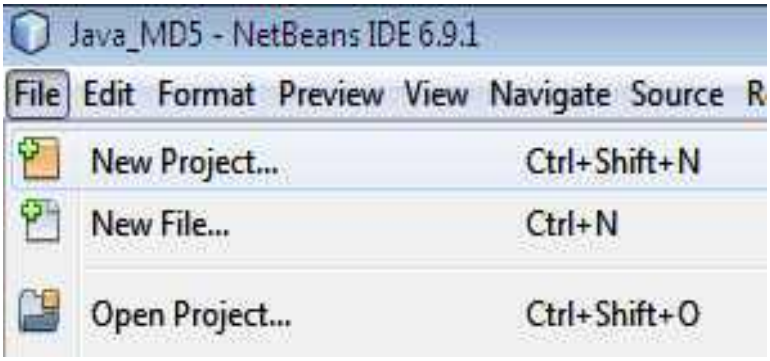
Pada akhir pembahasan, Diharapkan pembaca dapat :

1. Mengetahui tipe data di java dan cara menggunakannya.
2. Mengetahui cara mendeklarasikan variabel di java dan cara menggunakannya
3. Mengetahui dan memahami conditional statement di java juga mampu untuk menggunakannya.
4. Mengetahui cara kerja reader input di java.

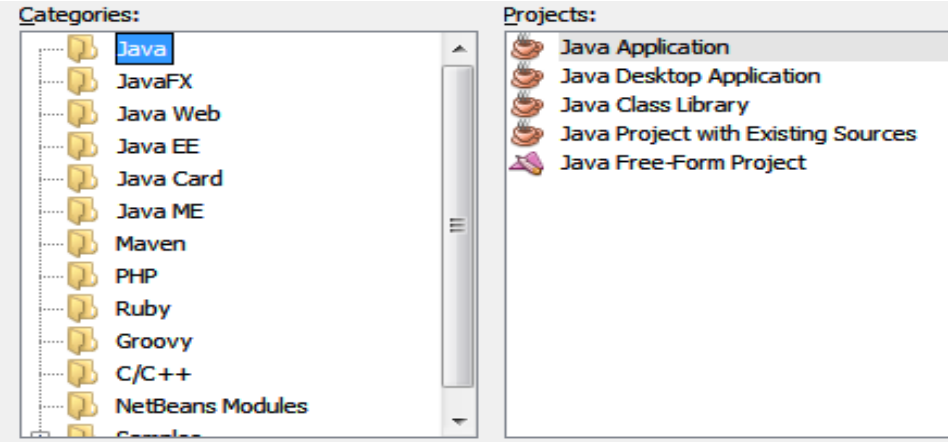
I. Pembentukan project

Prosedure pembuatan project sebagai berikut:

- Klik **New project** >Klik **Java > Java Aplication > Next**

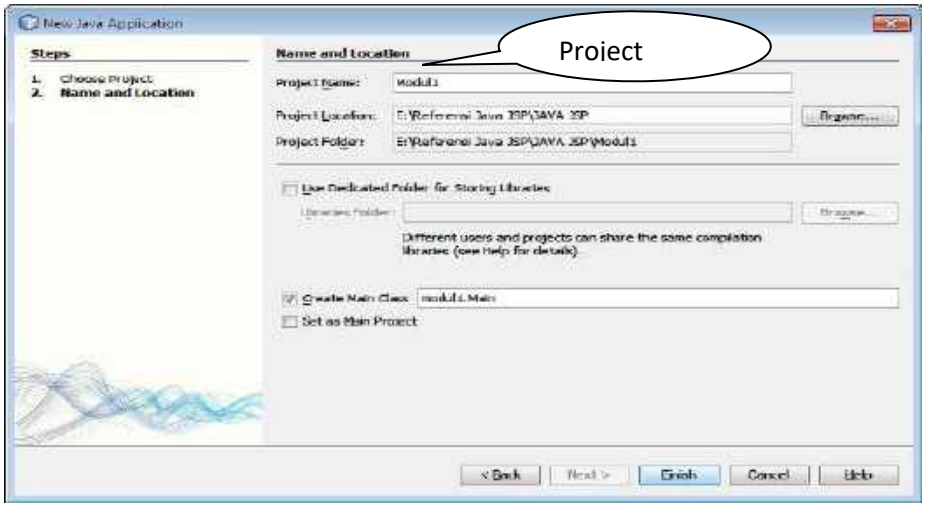


Gambar 4.1 New Project



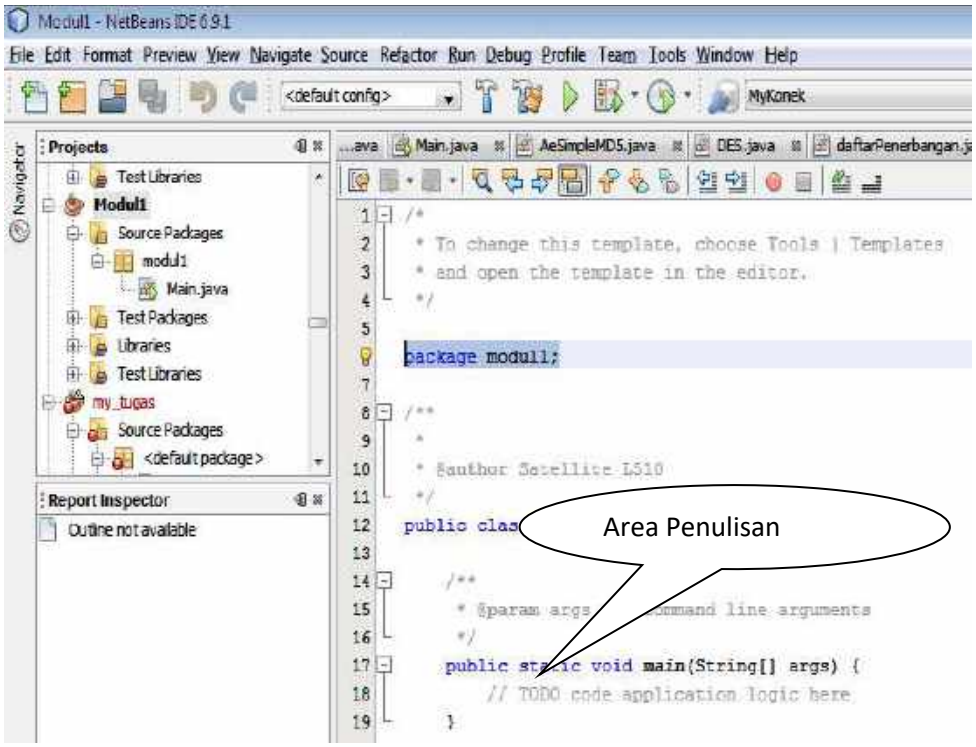
Gambar 4.2 Pilihan java aplication

- Isikan **Project Name**, misalkan Modul1, **Project Location**isikan folder dimana kita akan menyimpan data, misalnya pada D:\ Modul1> **Finish**



Gambar 4.3 Pengisian project name

- Maka akan tampil jendela seperti gambar dibawah ini:
- klik Plus tanda (+) pada **Project Modul1** >**Source Packages**> **Modul1** >double klik >**Main.java** maka kita akan aktif pada area penulisan coding java.



Gambar 4.4 Area penulisan coding / sintaks

- Pada jendela penulisan coding terdapat bagian yang dinamakan **Comment**, berikut beberapa penulisan Comment Java :
- /*...text...*/ : Compiler akan mengabaikan text diantara /* text */.
- /** documentasi */ : Compiler akan mengabaikan text diantara /**..*/.
- //..... : Compiler mengabaikan text satu baris.
- Comment diabaikan oleh compiler, tapi sangat berguna untuk memberi catatan terhadap program yang anda buat, sehingga dapat membantu anda memahami logika yang saat ini dibuat dikemudian hari.
- Penulisan sintaks java sifatnya case sensitive, jadi huruf kecil dan huruf besar dibedakan.

II. Syarat penamaan Class

Dalam pemrograman java terdapat beberapa syarat pembentukan / penamaan class antara lain:

a. Diawali huruf kapital
b. Bila lebih dari satu kata, huruf kedua diawali huruf kapital juga.
c. Tidak boleh mengandung spasi.
d. Karakter yang diizinkan adalah huruf dan angka.

e. Pada bahasa java terdapat istilah kode Escape, yaitu yang penulisannya diawali dengan simbol “\”

Tabel daftar karakter Escape.

Kode	Keterangan
\ b	Backspace
\ n	Baris Baru (Line Feed)
\ r	Carriage return
\ t	Tabulasi

Contoh penamaan dalam class :

```
public class Game {
    public static void main(String[] args){

        // membuat objek player
        Player petani = new Player();

        // mengisi atribut player
        petani.name = "Petani Kode";
        petani.speed = 78;
        petani.healthPoin = 0;

        // menjalankan method
        petani.run();

        if(petani.isDead()){
            System.out.println("Game Over!");
        }

    }
}
```

```
run:
Petani Kode is running...
Speed: 78
Game Over!
BUILD SUCCESSFUL (total time: 3 seconds)
```

III. Variabel

Variabel merupakan tempat atau wadah untuk menyimpan nilai / value pada bahasa pemrograman. Pada Pemrograman java, semua variabel harus dideklarasikan sebelum mereka dapat digunakan. Bentuk dasar dari sebuah deklarasi variabel yang ditampilkan di sini:

Type identifier [= value]

Jenis ini merupakan salah satu tipe data Java, Identifier adalah nama variabel. Menyatakan lebih dari satu variabel dari jenis tertentu, menggunakan daftar dipisahkan koma. Berikut adalah beberapa contoh deklarasi variabel dari berbagai jenis. Perhatikan bahwa beberapa mencakup inisialisasi.

```
int a, b, c;                // deklarasi 3 variabel a,b,c bertipe integer.
long d = 3, e, f = 5;    // deklarasi 3 variabel d, e, f  bertipe long
byte z  = 22;               // deklarasi dan inisialisasi variabel z.
char x = ‘ x’ ;             // variabel x yang diberikan value / nilai x.
String nama = “adi” ; // variabel nama yang diberikan value / nilai adi.
```

Variabel adalah suatu tempat penyimpanan yang bersifat temporary dimemori yang digunakan dalam suatu program, karena bersifat sementara maka apabila program selesai dijalankan maka isi dari variabel akan hilang. Variabel dapat bersifat lokal, misalkan didalam perulangan for, atau dapat pula berupa variabel instant yang dapat diakses oleh semua dalam class, berikut cara mendeklarasi dan memberikan nilai terhadap variabel :

```
public class Main {
    public Main() { }
    public static void main(String[] args) {
int nil;
   for (nil = 1; nil < 10 ; nil++)
   {if (nil%2 == 1)
     System.out.println(nil+".");
   }          }    }
```

Pada potongan program diatas yang dimaksud variabel adalah **nil** dengan tipe data Int (bilangan Integer).

IV. Tipe Data

Tipe data bisa juga dikatakan sebagai sifat dari suatu variabel, yang hanya menyatakan model data yang diproses, bukan menyatakan tempat untuk menyimpan data tersebut. Pada pemrograman java tipe data secara umum dapat dikelompokan ke dalam 2 jenis, yaitu tipe data angka/numerik dan huruf/string.

a. Numeric

Tipe data numeric adalah tipe data yang menangani penampungan data berupa bilangan bulat maupun bilangan real baik negatif maupun positif. Sebuah tipe data mempunyai range nilai yang berbeda , berikut tabel range nilai tipe data :

Operasi	Range Nilai	Ukuran Memori
Byte	-27(-128) s/d -27-1(127)	8 bit signed
Short	-215 (-32758) s/d -215-1 (	16 bit signed
Integer	-231 s/d 231-1	32 bit signed
Long	-263 s/d 263-1	64 bit signed
Float	-3.4E38 s/d 3.4E38	32 bit IEEE 754
Double	-1.7E308 s/d 1.7E308	64 bit IEEE 754

Dalam pemrograman java untuk mengubah suatu data angka bertipe string menjadi numerik dapat dilakukan konversi dengan menggunakan perintah sebagai berikut :

Integer.parseInt(nilaiString)>>mengubah String menjadi Integer

Double.parseInt(nilaiString)>>mengubah String menjadi Double

Short.parseInt(nilaiString)>>mengubah String menjadi Short.

b. String.

Tipe data string adalah tipe data yang digunakan untuk menampung data berupa karakter atau huruf. Berdasarkan banyak karakter, tipe data string dapat dibagi menjadi 2 bagian, yaitu tipe data char dan tipe data string. Tipe data char hanya dapat menampung satu karakter, sedangkan string mampu menampung banyak karaker.

Contoh : char nilai; .....String nama;

kita dapat menkonversi tipe data lain numerik/char menjadi string dengan cara berikut :

String.valueOf(Tipedatalain) ;

Integer.toString(nilaiInteger) ;

namaObject.toString() ;

V. Operator Java

Operator adalah suatu karakter khusus yang memerintahkan compiler untuk melakukan sesuatu operasi terhadap sejumlah operand.

Beberapa contoh operator pada java.

Operator Aritmatika

Operator	Hasil
+	Penjumlahan
-	Pengurangan
*	Perkalian
/	Pembagian
%	Modulus
++	Increment
--	Decrement
+ =	Persamaan Penjumlahan
-=	Persamaan Pengurangan

Operator Logika

Operator	Hasil
&&	And
||	Or
!	Not

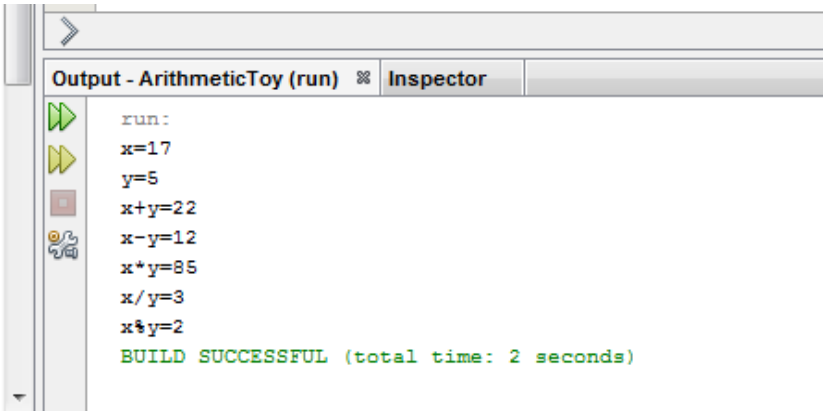
Operator Relasi

Operator	Hasil
==	Sama Dengan
!=	Tidak sama dengan
>	Lebih besar dari
<	Lebih kecil dari
>=	Lebih besar dari atau sama dengan
<=	Lebih kecil dari atau sama dengan

Contoh penggunaan operator aritmatika

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package arithmetictoy;

/**
 *
 * @author Samsuriah
 */
public class ArithmeticToy {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        int x=17, y=5;
        System.out.println("x="+x);
        System.out.println("y="+y);
        System.out.println("x+y="+(x+y));
        System.out.println("x-y="+(x-y));
        System.out.println("x*y="+(x*y));
        System.out.println("x/y="+(x/y));
        System.out.println("x%y="+(x%y));
    }

}
```

Hasil Output dari sintaks diatas adalah:



Gambar 4.5 Layar output console netbeans penggunaan operator java

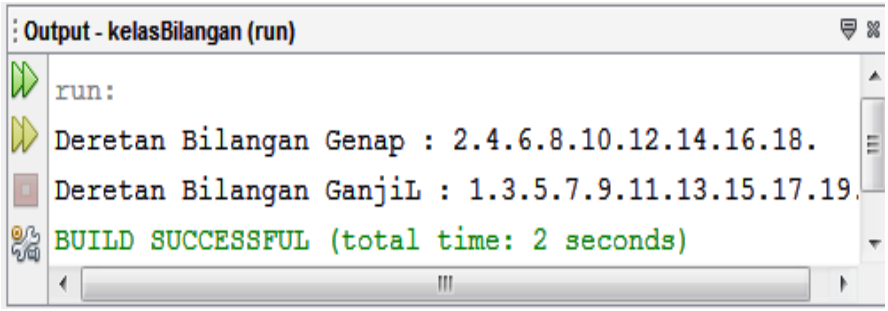
VI. Menampilkan Informasi ke layar

Untuk menampilkan hasil eksekusi sintaks ke layar dapat digunakan class system dari library java, yaitu System.Out.print atau System.Out.println untuk menggabung String digunakan tanda + sebagai penghubung.

Contoh penggunaan fungsi **System.Out.print.**

```
public class Main {
    private static void bilanganGanjil ()
    {
      for (int x = 1; x < 20 ;x++){
   if (x%2==1)
     System.out.print(x+".");
   }
    }

    private static void bilanganGenap ()
    {
    for (int i = 1; i < 20 ;i++){
   if (i%2==0)
     System.out.print(i+".");
    }
}

    public static void main(String[] args) {
System.out.print("Deretan Bilangan Genap : ");
      bilanganGenap();
      System.out.println();
System.out.print("Deretan Bilangan GanjiL : ");
 bilanganGanjil();
    }
}
```

Hasil Output dari sintaks diatas adalah:



Gambar 4.6 Layar output console netbeans

VII. Menerima inputan dari keyboard

Dalam program java untuk menangkap inputan yang diberikan keyboard dapat digunakan Class BuffredReader yang disediakan oleh java API , cara penggunaannya yaitu dengan meletakan sintaks Import java.io.* pada program. Disarankan menggabungkan fungsi **Try catch** dengan **fungsi readLine().**

Contoh penggunaan fungsi BuffredReader.

```
package perulangan_inputan;
import java.io.*;
public class Main {

    public static void main(String[] args)
 {
    try
{
    int n1,nil_awal;
    int n2,nil_akhir;

BufferedReader uji = new BufferedReader(new
InputStreamReader(System.in));

   System.out.println("Input Nilai Awal..");

   nil_awal=Integer.valueOf(uji.readLine()).intValue();

System.out.println("Input Nilai Akhir..");

   nil_akhir=Integer.valueOf(uji.readLine()).intValue();

System.out.println("Perulangan For Antara "+nil_awal+" dan "+
nil_akhir);
for (n1=nil_awal; n1<=nil_akhir;n1++)
{
System.out.println(n1);
}
}
  catch (Exception e){  System.out.print(e);}
 }
}
```

Pada diatas variabel nil_awal dan nil_akhir akan berisi inputan dari keyboard sebelum nantinya dilakukan proses increment.

BAB V KONTROL PROGRAM DALAM JAVA

Tujuan

Pokok bahasan ini :

- Menggunakan struktur kontrol keputusan (if, else, switch) yang digunakan untuk memilih blokkode yang akan dieksekusi
- Menggunakan struktur kontrol pengulangan (while, do-while, for) yang digunakan untuk melakukan pengulangan pada blok kode yang akan dieksekusi
- Menggunakan statement percabangan (break, continue, return) yang digunakan untuk mengatur redirection dari program.

Pada akhir pembahasan, Diharapkan pembaca dapat :

1. Memahami apa itu struktur control
2. Membuat kode dengan menggunakan struktur kontrol

Latar Belakang

Pada bab sebelumnya, kita sudah mendapatkan contoh dari program, 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 control keputusan adalah statement dari Java yang mengijinkan user untuk memilih dan mengeksekusi blok kode dan mengabaikan blok kode yang lain.

I. Pernyataan If

a. If Tunggal

Pernyataan if merupakan salah satu bentuk pernyataan yang berguna untuk mengambil keputusan terhadap sebuah kemungkinan. Bentuk pernyataan if berupa :

```
If (kondisi) {
        // yang akan dijalankan
}
```

Sebuah Instrukksi **If** digunakan unutk mengatasi kondisi percabangan dalam java dan pemrograman lainnya, blok instruksi yang terletak setelah **if** akan dikerjakan jika logika dari kondisi dibelakangnya bernilai **true**contohpernyataan If tunggal adalah :

```
package if_tunggal;
import java.io.DataInputStream;
public class Main {
   public static void main(String[] args)
 {
      DataInputStream Beli = new DataInputStream(System.in);
try
{
System.out.print("Inputkan Jumlah pembelian : ");
String input = Beli.readLine();
double jumlah = Double.parseDouble (input);
System.out.print ("\n Discount : ") ;

if ((jumlah >= 15000) && (jumlah <= 110000)) System.out.println
("Discount 10%");
if ((jumlah > 110000) && (jumlah <= 150000)) System.out.println
("Discount 20%");
if ((jumlah > 150000) && (jumlah <= 1100000)) System.out.println
("Discount 30%");
if ((jumlah > 1100000))System.out.println ("Discount 50%");
}
catch (Exception e)
{
System.out.println ("\n Periksa Inputan.....");
}}}
```

Ternary operator adalah operator untuk menuliskan statement if dengan lebih sederhana. Tapi operator ini bisa digunakan jika terdapat dua kondisi, jika nilai yang dievaluasi true, maka nilai pertama yang diambil dan jika salah maka nilai kedua yang diambil.

Contoh kasus

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:

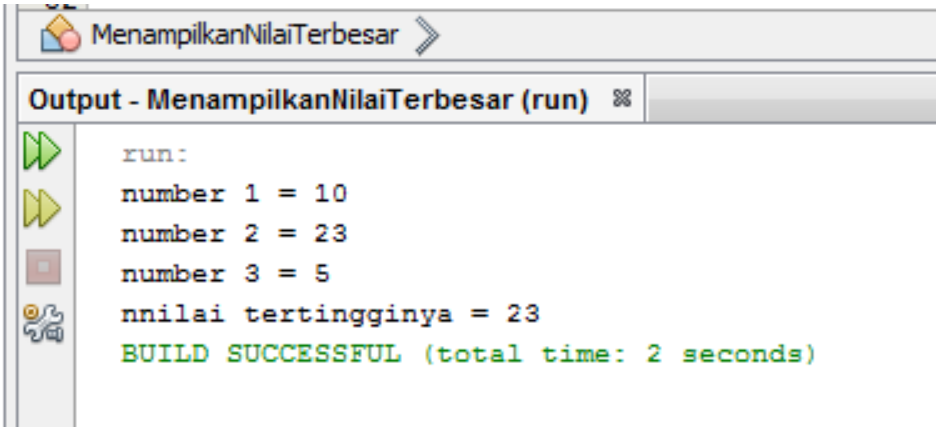
number 1 = 10 number 2 = 23 number 3 = 5

Nilai tertingginya adalah angka = 23

Penggunaan dalam menuliskan statement if dengan lebih sederhana

```
public class MenampilkanNilaiTerbesar {


   /**
    * @param args the command line arguments
    */
   public static void main(String[] args) {
      // TODO code application logic here
      int number1=10;
      int number2=23;


      int score = 0;
      score=(number2
number3)?(number2>number1)?number2:number1:number3;
      System.out.println("number 1 = "+number1);
      System.out.println("number 2 = "+number2);
      System.out.println("number 3 = "+number3);
      System.out.println("nilai tertingginya = "+score);
   }

}
```

Hasil dari sintaks diatas adalah :



b. If Majemuk

Bentuk lain percangan dengan if adalah bentuk if majemuk yang merupakan susunan perintah if sedemikian rupa sehingga jika hasil logika bernilai true maka perintah berikutnya diabaikan.

contoh dari pernyataan If majemuk adalah :

```
package if_Mejemuk;
import java.io.DataInputStream;

public class Main {
public static void main(String[] args)
 {
DataInputStream Beli = new DataInputStream(System.in);
try
{
System.out.print("Inputkan Jumlah pembelian : ");

String input = Beli.readLine();
double jumlah = Double.parseDouble (input);

System.out.print ("\nDiscount : ") ;
if ((jumlah >= 7000) && (jumlah <= 10000)) System.out.println
("Discount 10%");
else
if ((jumlah > 10000) && (jumlah <= 70000)) System.out.println
("Discount 20%");
else
if ((jumlah > 70000) && (jumlah <= 100000))
System.out.println ("Discount 30%");
else
System.out.println ("Discount 50%");
}
catch (Exception e)
{
System.out.println ("\n Periksa Inputan...");
}
}
}
```

Kata kunci **else** digunakan sebagai penghubung antar pernyataan if yang akan diseleksi dalam satu tingkat.

II. Pernyataan Switch

Pernyataan bersyarat digunakan untuk melakukan tindakan yang berbeda berdasarkan pada kondisi yang berbeda. Gunakan pernyataan switch untuk memilih salah satu dari banyak blok kode yang akan dieksekusi(Litbang, 2016).

```
switch(n)
{
case 1;
execute code block 1
break;
case 2;
execute code block 2
break;
default;
code to be executed if in is different from case 1 and 2
}
```

Cara kerja dari pernyataan switch diumpamakan Anda memiliki ekspresi n tunggal (paling sering variabel) yang dievaluasi sekali. Nilai ekspresi tersebut dibandingkan dengan nilai-nilai untuk setiap kasus dalam struktur. Jika ada yang sama, blok kode yang terkait dengan kasus adalah yang akan dieksekusi. Gunakan break untuk mencegah kode berlari ke kasus berikutnnya secara otomatis.

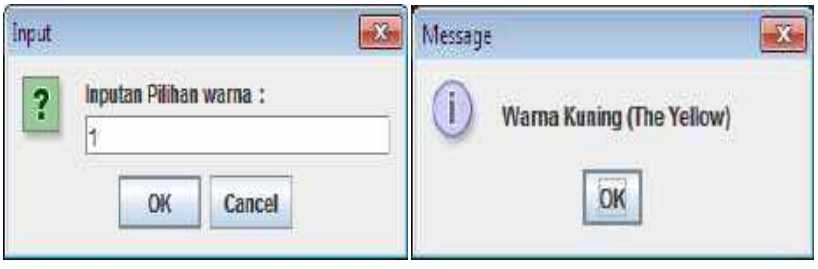
```
<script type=”text/javascript”>
//you will receive a different greeting based
//on what day it is. Note that Sunday=0,
//Monday=1, Tuesday=2, etc.
var d=new Date();
var theday=d.getDay();
switch (theDay)
{
case 5;
document.write(“Finally Friday”);
break;
case 6;
document.write(“Super Saturday”);
break;
case 0;
document.write(“Sleepy Sunday”);
break;
default;
document.write(“I’m looking forward to this weekend!”);

}
</script>
```

Perintah switch memungkinkan untuk melakukan sejumlah tindakan berbeda terhadap sejumlah kemungkinan nilai.Pada perintah switch terdapat pernyataan break, yang akan digunakan untuk mengendalikan eksekusi keakhir pernyataan switch. Pada umumnya switch tidak dapat digunakan untuk ekspresi string.

Contoh program menggunakan percabangan **switch** adalah :

```
package swicth_inputan;
import javax.swing.JOptionPane;

public class Main {
public static void main(String[] args) {
try{
int Warna;
String input;
input = JOptionPane.showInputDialog
("Inputkan Waktu :");
Warna = Integer.parseInt (input);

switch (Warna)
{
case 0: JOptionPane.showMessageDialog (null,
"Warna Merah (The Red)");

case 1: JOptionPane.showMessageDialog (null,
"Warna Kuning (The Yellow)");
break;

default: JOptionPane.showMessageDialog (null,
Warna + " Tidak Terdaftar: Input Salah...Periksa Inputan");
}
} catch (Exception e)
 {
    System.out.println("Periksa Inputan");
}
System.exit (0);
}
}
```

Hasil dari sintaks diatas adalah:



Dalam penggunaanya, pernyataan **if** dan pernyataan **switch** masing – masing memiliki kelebihan dan kekurangan, jadi penggunaannya sesuai dengan kebutuhan dan kondisi yang akan dikerjakan.

III. Pernyataan For

Perulangan for merupakan perulangan yang memiliki variabel untuk melakukan pengkondisian, berbeda dengan while dan do-while yang kita harus membuat sebuah variabel diluar untuk melakukan pengkondisian, pada perulangan for, ditempatkan sebuah blok untuk membuat variabel dan melakukan proses pengkondisian. Bentuk pernyataan for sebagai berikut :

```
for (inisialisasi; kondisi; penaikan/penurunan) {
        instruksi
}
```

Pernyataan for dikenal sebagai pernyataan untuk mengendalikan proses berulang dengan jumlah yang sudah ditetapkan sebelumnya.

Contoh penggunaan pernyataan for adalah:

```
public class Main {
    public Main() {
    }
    public static void main(String[] args) {
     int nilgenap,nilganjil;

     System.out.println("Anda Mencetak Bilangan Genap");
 for (nilgenap = 1; nilgenap <= 20 ; nilgenap++ )
 { if (nilgenap%2==0)
{ System.out.println(nilgenap);}
      }

     System.out.println("Anda Mencetak Bilangan Ganjil");
for (nilgenap = 1; nilgenap <= 20 ; nilgenap++ )
{    if (nilgenap%2==1)
{ System.out.println(nilgenap);}
     }
  }   }}
```

Keterangan :

- Bagian inisialisasi digunakan untuk memberikan nilai pada variabel yang digunakan untuk mengontrol pengulangan. (**nilgenap = 1)**
- Bagian kondisi digunakan untuk mengontrol pengulangan untuk dilanjutkan atau diakhiri. (**nilgenap <= 20**)

- Bagian increment/decrement digunakan untuk menaikan atau menurunkan nilai variabel pengontrolan pengulangan. (**nilgenap++**)
- Variabel **nilgenap = 1**dengan type Integer, digunakan untuk mendeklarasikan jumlah dan memberikan nilai 1 sebagai nilai awal.
- Pernyataan **nilgenap <= 20**digunakan untuk menetapkan batas perulangan lebih kecil dari 21.
- Pernyataan **nilgenap++**merupakan kondisi penaikan nilai variabel nilgenap.
- Pernyataan **nilgenap %2==0** digunakan untuk menyeleksi bilangan genap yang berada antara 1 – 20.

IV. Pernyataan While

Pernyataan while berguna untuk melakukan proses perulangan untuk kondisi, selama kondisi tersebut bernilai benar (true), maka perulangan akan terus berjalan, dan terhenti ketika kondisi bernilai salah (false).

Pernyataan while berguna untuk melakukan proses yang berulang

Bentuk umum pernyataan while adalah :

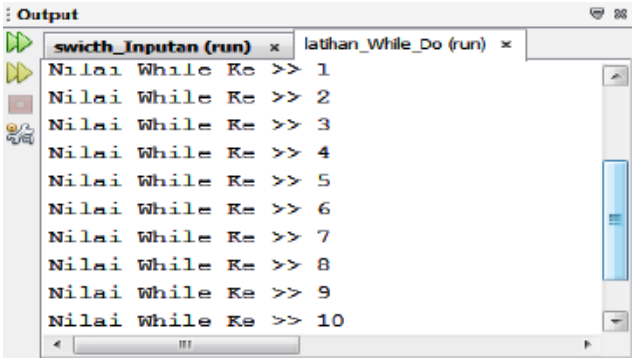
```
While (Kondisi)
{
Blok_Pernyataan;
}
```

Bagian blok pernyataan akan dijalankan berulang secara terus menerus selama kondisi bernilai **true** (benar)

Contoh program menggunakan pernyataan while adalah :

```
package latihan_while;
public class Main {
    public Main() { }
    public static void main(String[] args)
    {
  int Urutan = 0;
 while (Urutan <= 10)
    {
      Urutan ++;
if (Urutan ==10) break;
     System.out.println("Nilai While Ke >> "+ Urutan);
      }
}}
```

Pada program diatas akan tercetak “Nilai While Ke 1” sampai“Nilai While Ke 10”, sesuai dengan penyataan While(Urutan <=10) dan pernyataan break pada nilai 10 (if (Urutan ==10) break). Walaupun nilai awal dari variabel adalah 0 namun pada saat dieksekusi variabel langsung melakukan proses increment sehingga variabel bernilai 1.



V. Pernyataan Do...While

Pernyataan do..while menyerupai pernyataan while, akan tetapi pernyataan do..while melakukan pengecekan terhadap suatu kondisi setelah melakukan perintah yang ada didalamnya. Looping akan berhenti jika kondisi bernilai false.

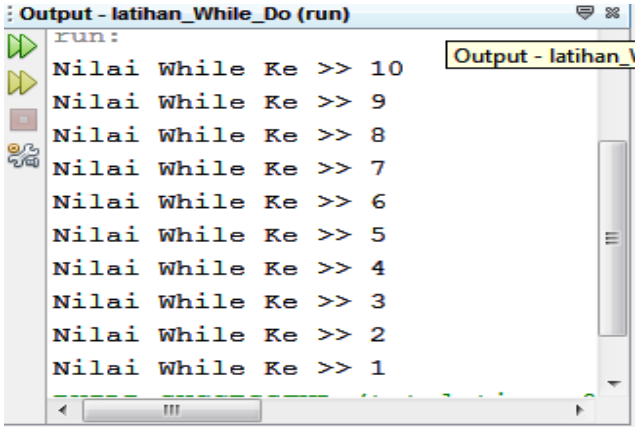
Bentuk umum pernyataan **do..while**:

```
Do {
Blok_Pernyataan;
} While (Kondisi);
```

Contoh program pernyataan **do..while** adalah :

```
package latihan_while;
public class Main {
    public Main() { }
    public static void main(String[] args)
    {
  int Urut=10;
    do{
      System.out.println("Nilai While Ke >> "+Urut);
Urut --;
     }
    while (Urut>= 1);
     }
}
```

Hasil program diatas akan menampilkan seperti gambar berikut :



sesuai dengan penyataan proses decrement dimana nilai awalnya adalah 10 dan batasan nilai = 1(**While** (i >=1)) merupakan pernyataan looping**Do..while** sebanyak 10 kali.

VI. Array

Suatu array adalah sebuah struktur data yang terdiri atas banyak variabel dengan tipe data sama, menyimpan satu jenis data (variabel).

Array merupakan sekumpulan obyek yang memiliki tipe data yang sama dan dapat di akses secara random dengan menggunakan index. Array mempunyai panjang yang tetap, artinya ketika kita mendeklarasikan suatu array dengan pangjang 10, maka array tersebut panjangnya akan tetap 10 walaupun kita hanya memakai 5 elemen(Elenia et al., 2020).

Array adalah variabel tunggal yang dipakai untuk menyimpan sekumpulan data sejenis. Untuk membedakan tempat penyimpanan satu data dengan data yang lainnya.

beberapa cara mendeklarasikan variabel array dalam java, yaitu :

- Pendeklarasikan element tak langsung, yaitu tanpa menyebutkan beberapa element yang diperlukan.

  Bentuk umum :

```
Int[ ] angka = new int [ numeric ];
```

Contoh program adalah:

```
package array1d;
public class Main {
    public static void main(String[] args) {
    int [] nilai = new int[5];
        nilai[0]=1;
        nilai[1]=2;
        nilai[2]=3;
        nilai[3]=4;
        nilai[4]=5;
       System.out.println(nilai[0]);
       System.out.println(nilai[1]);
       System.out.println(nilai[2]);
       System.out.println(nilai[3]);
       System.out.println(nilai[4]);
}  }
```

- Pendeklarasikan element secara langsung, yaitu dengan menyebutkan elemen yang diperlukan.

  Bentuk umum : ***Int[ ] angka = {"x","xx"};***

Contoh program adalah:

```
package array_deklarasi_element;
public class Main {
    public static void main(String[] args) {
        String[] bunga = {"Melati", "Mawar", "Seroja"};
System.out.println("Juamlah Array =" + bunga.length);
System.out.println("=========================");
        for (int cari = 0; cari < bunga.length; cari++) {
        System.out.println("bunga Ke : "+cari+" =
"+bunga[cari]+".");
        }    }
}
```

- Deklarasikan Array satu dimensi,

  Untuk mendeklarasikan array dapat dilakukan dengan cara berikut ini :

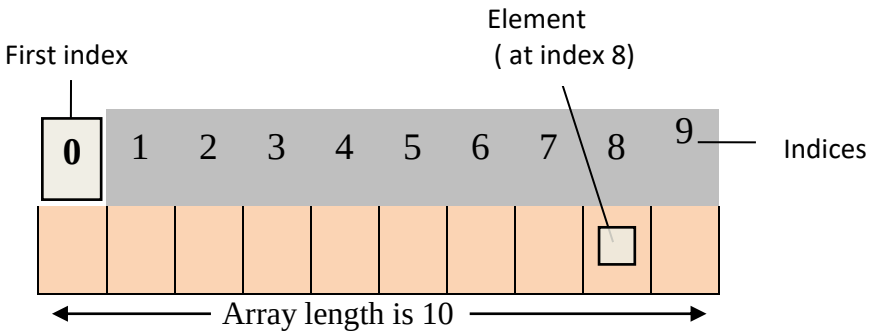
```
Int value [];
value =  new int [ 100] ; // menginisialisasikan
                          // [100] merupakan batas ruangan array
```

Atau bisa deklarasi sekaligus menginisialisasikan

```
Int value [] = new int [100] ;
```

Cara pendeklarasian tanpa batas

```
Int value [] = {1, 2, 3} ;    // {1, 2, 3} adalah isi dari array
```



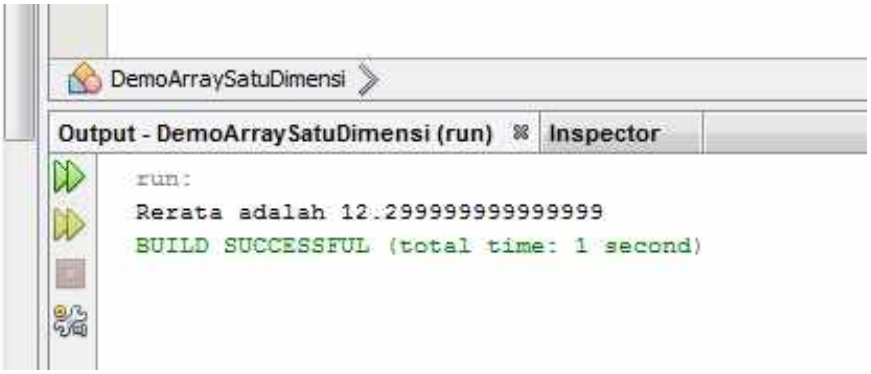
Visual suatu array

Untuk mengetahui panjang array yaitu gunakan syntax nama array.length berdasarkan contoh diatas nama array = value maka untuk mengetahui panjang value kita gunakan syntax value.length. cara pemanggilan yaitu value[0] akan menampilkan angka 1.

Contoh program pernyataan Array 1 dimensi adalah :

```
//Demonstrasi array satu-dimensi
public class DemoArraySatuDimensi {
   public static void main(String args[]) {
      double arrayAngka[] = {10.1, 11.2, 12.3, 13.4, 14.5};
      double hasil = 0;
      int i;

      for(i=0; i<5; i++)
         hasil = hasil + arrayAngka[i];
      System.out.println("Rerata adalah " + hasil / 5);
   }
}
```

Hasil program diatas akan menampilkan seperti gambar berikut :



- Deklarasi Array Multidimensi bisa

  Array tidak hanya terdiri dari 1 dimensi, array juga bisa terdiri dari 2 dimensi, 3 dimensi dan n-dimensi. Array diatas 3 dimensi sangat jarang digunakan karena sangat sulit untuk di gambarkan. Array multidimensi diimplementasikan sebagai array dalam array. Untuk lebih jelas mengenai cara pendeklarasian array multidimensi dan penggunaannya, silahkan ketikkan program dibawah ini :

Array 2 dimensi:

Char multiChar[ ][ ]=new char[10][5];

Char multiChar2[ ][ ]= {{‘a’,’b’,’c’},{‘d’,’e’,’f’}}; inisialisasi array 2 dimensi

Array 3 dimensi:

int tigaDimensi[ ][ ]=new int[10][10][10];

int tigaDimensi[ ][ ]={{{1,2,3},{4,5,6}},{{1,2,3},{4,5,6}}}; inisialisasi array 3 dimensi

Tambahan:

Array lebih dari 2-dimensi sangat jarang dipakai.

BAB VI BEKERJA DENGAN GUI NETBEANS 6.9.1

Tujuan :

Pokok Bahasan ini menjelaskan :

- Membuat form sederhana dengan memanfaatkan komponen GUI
- Mendapatkan input dari command-line
- Mengetahui cara untuk memanipulasi properties dari sistem
- Membaca standart menulis file
- Membaca dan menulis file

Pada akhir pembahasan, Diharapkan pembaca dapat :

1. Memahami aplikasi berbasis teks
2. Membuat kode dengan menggunakan GUI

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 member 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. Keismpulannya, komponen AWT dapat digunakan bersama dengan komponen Swing.

Java swing merupakan toolkit GUI pada Java yang sering dipaksa untuk membuat aplikasi dengan Interface berbasis grafis.

Beberapa komponen dari Java swing yaitu:

- Button : Tombol.
- Label : Teks untuk memberikan suatu keterangan.
- Text Field : Media input text sepanjang 1 baris.
- Text Area : Media input text dengan ukuran bisa lebih dari 1 baris.
- Menu Bar : Bar yang biasanya menu utama suatu aplikasi.
- Menu : Menu-menu pada aplikasi.
- Table : Untuk menampilkan data dalam bentuk tabel.
- Combo box : Media input untuk memilih beberapa opsi dari opsi yang tersedia.
- Tool bar : Bar-bar untuk memilih tool-tool yang disediakan aplikasi dan b\iasanya ditampilkan dalam bentuk ikon.

Setiap komponen memiliki metode setter maupun getter untuk mengakses atributnya. Semisal untuk Jlabel terdapat metode setText() untuk mengubah tulisan pada label dan pada JTextField terdapat method getText() untuk mengambil data yang diinputkan ke dalam teks.

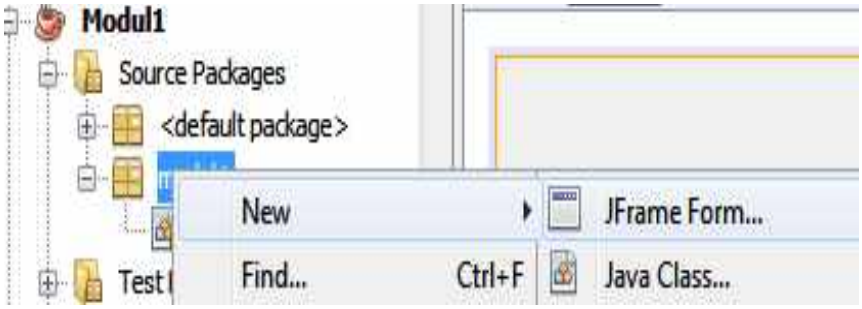
Setiap komponen juga memiliki yang disebut event listener (atau kadang disebut event handler) yaitu suatu aksi yang dilakukan ketika terjadi suatu event tertentu. Misal, ketika tombol ditekan, ketika teks ditulis dalam text field, dan sebagainya.

I. Membuat form.

Form adalah objek atau wadah yang digunakan untuk meletakan berbagai komponen yang dibutuhkan dalam mendesain suatu aplikasi.

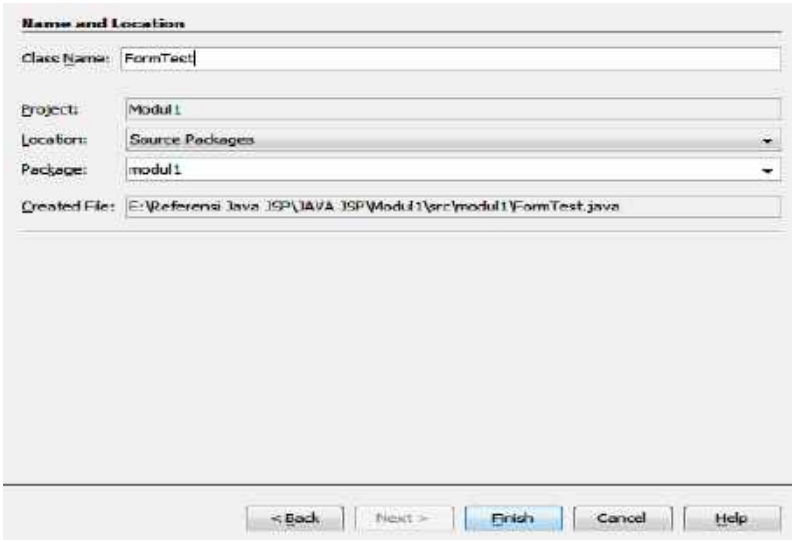
Prosedure pembuatan form pada netbeans adalah :

- Aktif pada package yang akan digunakan
- Klik kanan nama package lalu pilih **New > Jframe Form**



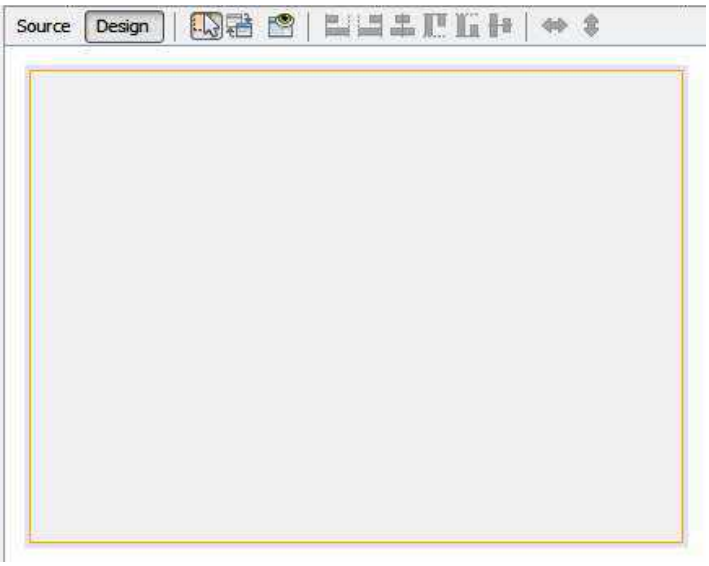
Gambar 6.1 Pembuatan form baru

- Maka akan tampil jendela new Jframe Form seperti berikut



Gambar 6.2 Pengisian Class name pada form

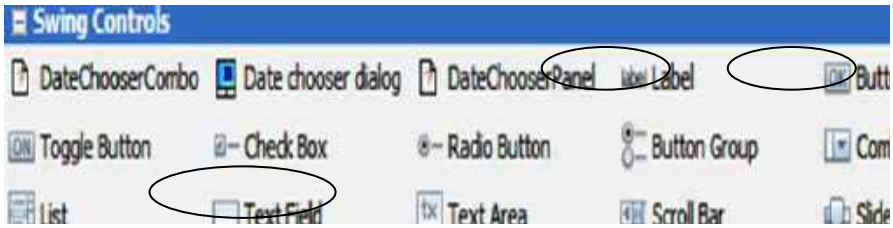
- Pada bagian **Class Name** isi dengan nama form yang diinginkan misalkan formTest , Lalu klik **Finish**
- maka akan tampil form yang siap untuk digunakan



Gambar 6.3 Form pada netbeans

II. Menggunakan objek jLabel, jTextFielddan jButton

Penggunaan komponen pada netbeans lebih sering terfokus pada palette **Swing Control** seperti gambar berikut :



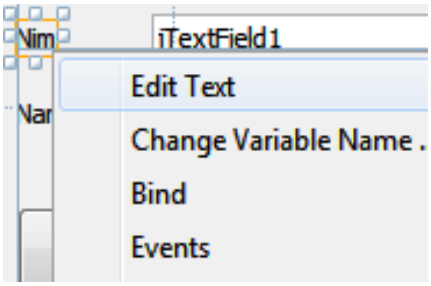
Gambar 6.4 Text field,label button pada palette Swing controls

Contoh penggunaaan:

Membuat form input data mahasiswa.

- Setelah pembuatan form selesai pada langkah sebelumnya, selanjutnya pilih objek yang diinginkan untuk mendesain form input data mahasiswa.
- Klik objek label pada palette swing control lalu seret kedalam form
- Lakukan langkah yang sama untuk objek lain yang diinginkan.
- Selanjutnya atur properties tiap komponen

- Untuk merubah teks pada label, klik kanan objek label lalu pilih **Edit Teks** atau tekan F2 pada Keyboard



- Untuk mengisi **name** suatu komponen klik kanan komponen lalu pilih **Change Variable Name**
- Lalu isi nama komponen yang diinginkan
- Atur semua komponen seperti tabel properties berikut

Komponen	Properties	Isi
Jlabel	Text	Nim
Jlabel	Text	Nama
JTextField	Name	Enim
	Text	Kosongkan
JTextField	Name	Enama
	Text	Kosongkan
JButton	Name	eClear
	Text	Clear
JButton	Name	eTutup
	Text	Exit

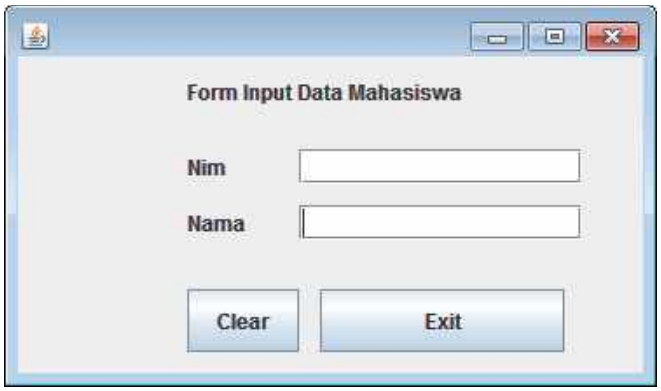
- Setelah semua pengaturan komponen selesai, selanjutnya klik ganda pada **button clear** , maka kita akan masuk ke jendela source, lalu ketik sintaks berikut;

```
enim.setText("");
    enama.setText("");
    enim.requestFocus();
```

- Selanjutnya klik ganda pada **button Exit** ,lalu ketik sintaks berikut;

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent
evt) {  System.exit(0);
      setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);    }
```

- Running Program maka hasil pembuatan form seperti gambar berikut :



Gambar 6.5 Form Input data mahasiswa

III. Menggunakan objek jRadioButton, jComboBox dan jTextArea

Untuk menggunakan komponen **Radio Button**, **Combobox,**dan **TextArea**, dapat kita temukan pada **Palette Swing Controls**

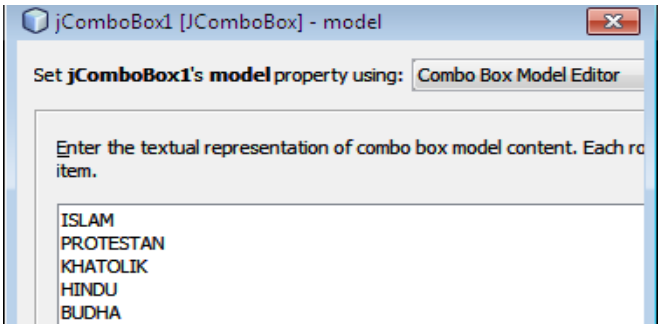
Gambar 6.6 Komponen Radio Button,Combobox,Text Area.

Contoh program :

- Tambahkan komponen **jRadioButton** sebanyak 2 pada program data mahasiswa sebelumnya.
- Tambahkan komponen **jComboBox**, ganti namanya menjadi cdAgama. Lalu klik **model** pada tab properties.

Gambar 6.7 Item Model pada Combobox

- Ganti item1,item2...dengan ISLAM,KHATOLIK, PROTESTAN, HINDU dan BUDHA.

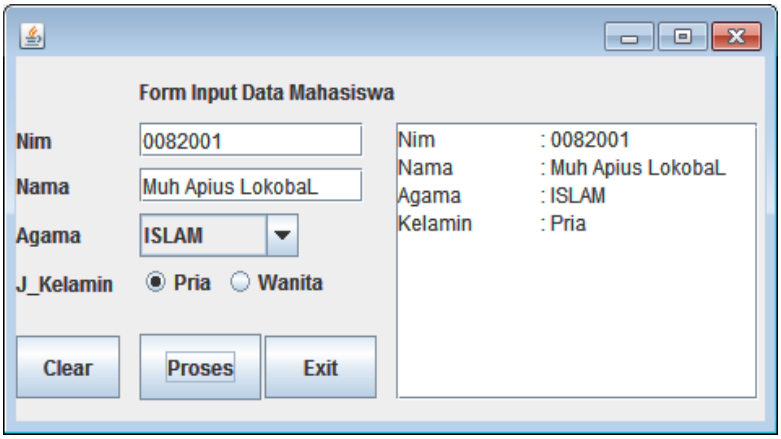


Gambar 6.8 Item pada Combobox

- Tambahkan komponen **jText Area**, isi name menjadi cetak.
- Tambahkan komponen **jButton**, lalu ubah **name** menjadi tProses lalu ketik kode berikut :

```
String nim,nama,agama,jk;
if (jRadioButton1.isSelected())
jk="Pria"; else jk="Wanita";
nim=enim.getText();
nama=enama.getText();
agama=(String)eAgama.getSelectedItem();
cetak.setText(
"Nim\t: "+nim+"\n"+
"Nama\t: "+nama+"\n"+
"Agama\t: "+agama+"\n"+
"Kelamin\t: "+jk+"\n");
```

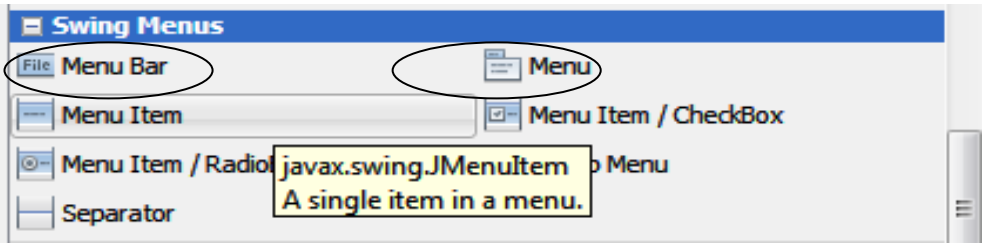
- Hasil program seperti gambar dibawah ini :



Gambar 6.9 Running program combobox dan Text Area

IV. Menggunakan jMenuBar,jMenu,jSeparator dan jMenuItem.

Komponen menu digunakan untuk mengontrol form lain, sehingga memudahkan kita pada saat memanggil form yang diinginkan. Komponen **jMenuBar** dapat ditemukan pada palette **Swing Menus** seperti gambar berikut :

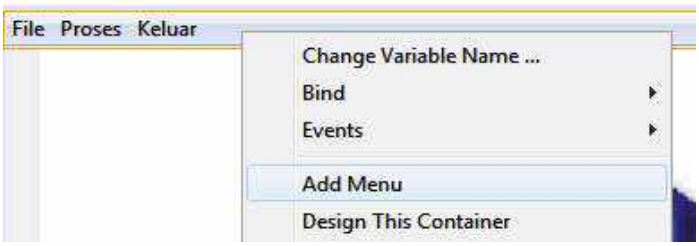


Gambar 6.10 Komponen Menu Bar

Contoh penggunaaan:

Membuat Form Menu Utama.

- Klik **Menu Bar** pada palette **Swing Menus lalu** seret kedalam form
- Prosedure untuk menambah atau membuat menu pada menu bar :
- Klik kanan pada **Menu Bar** pilih **Add Menu**

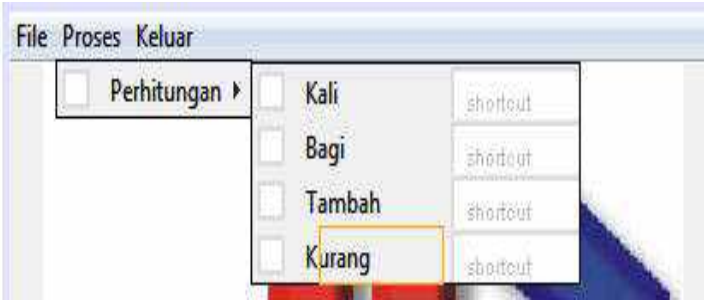


Gambar 6.11 Proses Add Menu

- Kemudian ubah **Name** dan **Text**pada properties sesuai keinginan.
- Prosedure menambah atau membuat Menu Item pada menu adalah :
- Klik kanan pada **Menu** pilih **Add form Palette**=>**menu Item**

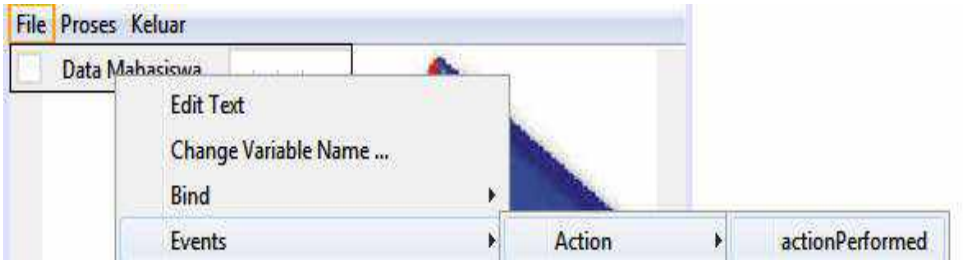
Gambar 6.12 Proses Add Menu Item

- Kemudian ubah **Name** dan **Text** nya pada properties.
- Melalui menu utama nantinya akan dipanggil form input data mahasiswa dan form perhitungan. Seperti gambar berikut :



Gambar 6.13 Menu Utama

- Untuk menampilkan form Input data mahasiswa, klik menu File=> Data Mahasiswa > Klik kanan >**Event** >**Action**>**actionPerformed**



Gambar 6.14 Pilihan event action menu item

- Ketik Kode Berikut :

```
FormInputDataMhs dataMhs = new FormInputDataMhs ();
    dataMhs.setVisible(true);
```

- Lakukan pemanggilan **class FormInputDataMhs**, lalu bentuk objek baru dari class tersebut dengan nama **dataMhs,** selanjutnya tampilkan form dengan menggunakan **setVisible.**
- Pada menu Keluar pada Event **MouseClicked** ketik kode berikut :

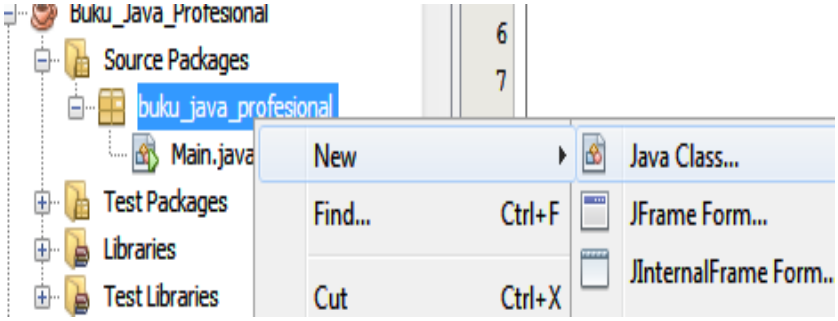
```
int answer = JOptionPane.showConfirmDialog(null,"Yakin Ingin
Keluar ??","Informasi", JOptionPane.YES_NO_OPTION);       if
(answer==JOptionPane.YES_OPTION)
{ System.exit(0);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
  } else  {  return;   }
  }
```

V. Membentuk class.

Selain membentuk form dalam project, dapat juga dilakukan pembentukan class, dimana melalui class tersebut dapat dilakukan proses enkapsulasi, pewarisan, polymorfisme, abstrac, inheritansi serta proses lainnya yang dapat dieksekusi melalui kejadian yang berulang, hingga tidak terjadi pemborosan sintaks.

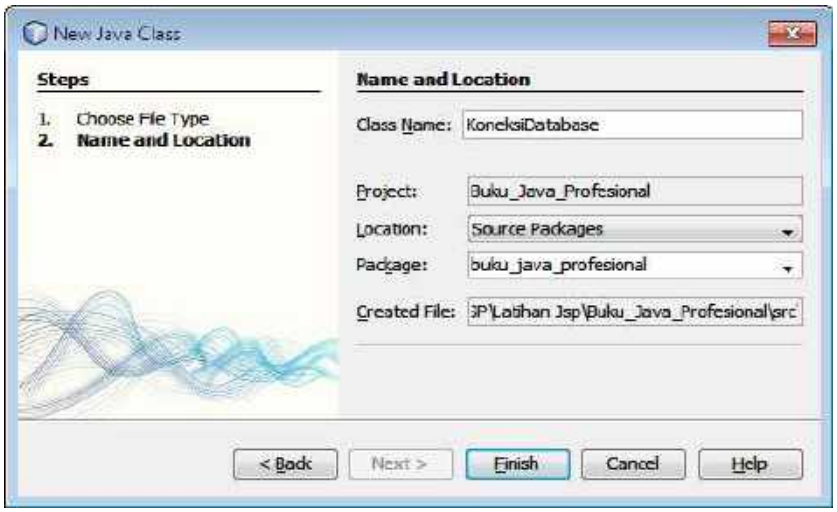
Langkah pembentukan class adalah :

- Aktif pada package yang akan digunakan
- Klik kanan nama package lalu pilih **New > Java Class**



Gambar 6.15 Pembuatan class baru

- Maka akan tampil jendela new java class seperti berikut :



Gambar 6.16 Pengisian Class name pada form

- Pada bagian **Class Name** isi dengan nama class yang diinginkan misalkan KoneksiDatabase , Lalu klik **Finish**

BAB VII IMPLEMENTASI DATABASE PADA NETBEANS

Tujuan :

Pokok Bahasan ini menjelaskan :

- Membuat form sederhana dengan memanfaatkan komponen GUI
- Mengetahui cara untuk koneksi ke database Mysql dengan memanfaatkan JDBC Connector
- Form yang telah terkoneksi dapat melakukan insert, update, delete
- Mempunyai rancangan dan menghasilkan report

Pada akhir pembahasan, Diharapkan pembaca dapat :

1. Memahami aplikasi JDBC
2. Membuat kode dengan menggunakan GUI

JDBC merupakan singkatan dari Java Database Connectivity yaitu API java yang membantu aplikasi java untuk mengeksekusi SQL statement. JDBC merupakan interface pemrograman aplikasi yang mendefinisikan bagaimana pemrograman java dapat mengakses database dalam format tabulator dari kode-kode java menggunakan sekumpulan interface standard an class-class yang tertulis dalam bahasa java.

Interface pemrograman aplikasi java menyediakan mekanisme untuk memuat package java beserta driver-driver dan memasang pada JDBC Driver Manager yang digunakan sebagai pembuat koneksi JDBC yang mendukung eksekusi syntax SQL seperti INSERT, UPDATE dan DELETE. Driver Manager adalah bagian utama dari JDBC.

Secara umum semua RDBMS (Relational Database Managemenet System) dan Java merupakan platform yang independen, jadi JDBC memungkinkan untuk membuat sebuah aplikasi database yang dapat dijalankan pada platform-platform berbeda dan berinteraksi dengan DBMS.

Singkatnya, JDBC mambantu programmer untuk membuat aplikasi java yang mengatur 3 aktivitas pemrograman ini:

1. Membantu menghubungkan dengan sumber data, seperti database.
2. Membantu mengirim query dan perintah update ke database.
3. Menerima dan memproses hasil yang diterima dari database sebagai respon dari query yang dikirim.

Seperti aplikasi lain pada umumnya, Netbeans dirancang juga untuk dapat menciptakan suatu aplikasi database desktop dan web, kemampuan netbeans untuk dapat menggunakan database tanpa menggunakan tool tambahan menjadikan netbeans tidak tergantung pada komponen pendukung saat merancang aplikasi database.

Contoh kasus yang akan digunakan adalah : "Merancang Aplikasi Pemesanan Tiket ".

A. Perancangan Database.

➢ Buatlah database melalui JavaDB atau melalui MySQL pada Xampp.

Ketentuan :

- Nama Database : **Tiket_Profesional**.
- Nama Tabel 1 : **TbPenerbangan.**
- Struktur tabel seperti gambar berikut:

Field	Type
kd_terbang	varchar(12)
Pesawat	varchar(35)
asal	varchar(35)
tujuan	varchar(35)
tgl_berangkat	date
wkt	time

Gambar 7.1 Struktur tabel tb_penerbangan

- Nama Tabel 2 : **TbTiket.**
- Struktur tabel seperti gambar berikut:

Field	Type
kd_tiket	varchar(12)
jns_tiket	varchar(12)
harga	double

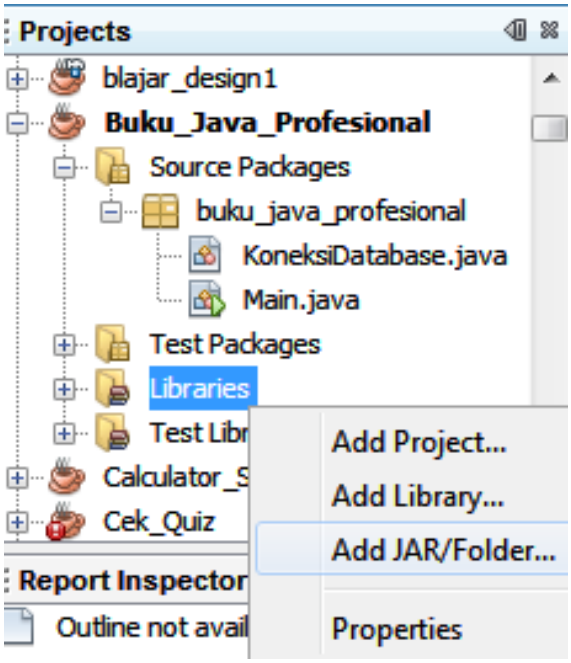
Gambar 7.2 Struktur tabel Tb_Tiket

- Nama Tabel 2 : **TbPenumpang.**
- Struktur tabel seperti gambar berikut:

Field	Type
kd_tiket	varchar(12)
kd_terbang	varchar(12)
nama	varchar(35)
jk	varchar(2)
umur	varchar(2)
telp	varchar(14)
tgl_pesan	date

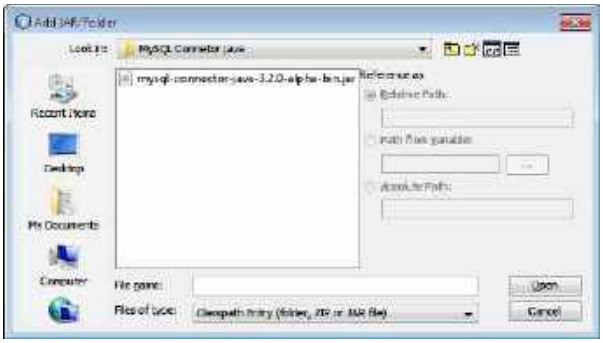
Gambar 7.3 Struktur tabel tb_penumpang

- Masukkan komponen pendukung, pada umumnya dalam bentuk file jar. Dengan cara sebagai berikut:
  - Pada jendela projects lakukan klik kanan **Libraries**> lalu klik **Add Jar/Folder**.seperti gambar dibawah ini



Gambar 7.4 Penambahan komponen jar

- Seleksi file komponen yang dibutuhkan pada direktory, lalu klik open.



Gambar 7.5 Seleksi komponen Jar pada directory

- Pada umumnya dalam pembuatan apliaksi database komponen / file jar yang dubutuhkan adalah:
  - **common-beautinul.jar**
  - **common-collection.jar**
  - **common-digester.jar**
  - **common-logging.jar**
  - **groovy-all.jar**
  - **jasperreport.jar**
  - **Connector-SQL.jar**
- Buatlah project baru dan beri nama sesuai keinginan anda.
- Rancang class **KoneksiDatabase.java**, untuk menyimpan sintaks proses koneksi kedatabase. Lalu Ketik sintaks dibawah ini :

```
import java.sql.PreparedStatement;import java.sql.Statement;
import java.sql.SQLException;import java.sql.ResultSet;
import java.sql.Connection;import java.sql.DriverManager;
public class KoneksiDatabase {
    public static Connection koneksi=null;
    public Connection getConnection()throws SQLException
    {   boolean ada_kesalahan = false;
try {Class.forName("com.mysql.jdbc.Driver");
            } catch (Exception ex)
{ System.out.println("ada kesalahan saat koneksi database
pertama : "+ ex);ada_kesalahan=true;   }
            if (!ada_kesalahan) {    try {
koneksi=DriverManager.getConnection("jdbc:mysql://localhost
/tiket_profesional","root","");
```

Lanjutan.....

```
}catch (Exception ex)
{
System.out.println("ada kesalahan saat koneksi mencari
database : "+ ex);
ada_kesalahan=true;
   }          }
      return koneksi;
   }
}
```

Lakukan pengujian dari sintaks diatas untuk menguji koneksi ke database, melalui class Main dengan sintaks berikut.

```
Connection uji_konek=null;
      try{
      KoneksiDatabase coba = new KoneksiDatabase();
uji_konek = coba.getConnection();
      } catch (Exception e)
{ System.out.print(" Koneksi GagaL....");
 }
 }
```

- Lakukan enkapsulasi melalui class **daftar_penerbangan.java,** pada class tersebut ketik sintaks berikut

```
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.Statement;
import javax.swing.JOptionPane;

public class daftar_penerbangan {

public String Kode_penerbang="";
private String pesawat="";
private String Asal="";
private String Tujuan="";
private String Tanggal="";
private String Waktu="";
private Connection konek_penerbangan = null;
```

Lanjutan...

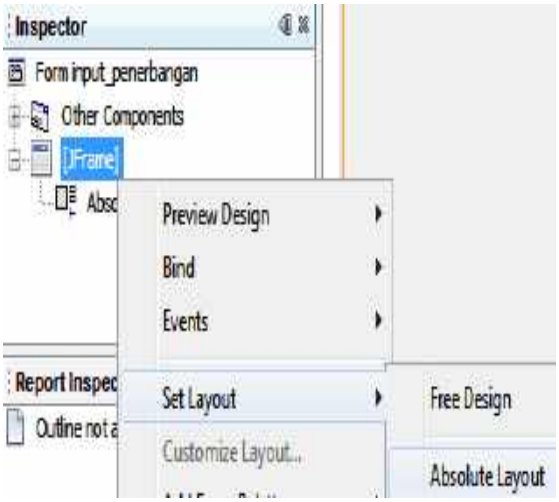
```
public String getKode_penerbang(){
        return Kode_penerbang;
    }
public void setKode_penerbang(String Kode_penerbang)
{       this.Kode_penerbang=Kode_penerbang;    }
 public String getWaktu()
{   return Waktu;  }
public void setWaktu(String Waktu)
{  this.Waktu=Waktu;  }
 public String getNama_penerbang()
{   return pesawat;    }
 public void setNama_penerbang(String Nama_pesawat)
   {   this.pesawat=Nama_pesawat;    }
 public String getAsal()
   {  return Asal;    }
 public void setAsal(String Asal)
   {this.Asal=Asal;    }
```

Lanjutan...

```
public String getTujuan(){ return Tujuan;
    }
    public void setTujuan(String Tujuan)
{       this.Tujuan=Tujuan;    }

 public String getTanggal()
{  return Tanggal;    }

    public void setTanggal(String Tanggal)
{       this.Tanggal=Tanggal;    }

    public ResultSet Penerbangan(int kode){
        KoneksiDatabase cari = new KoneksiDatabase();
        ResultSet rs=null;
        try{
            konek_penerbangan= cari.getConnection();
            Statement st=cari.createStatement();
            String sql="select * from penerbang where
kd_tiket='"+kode+"'";
            rs=st.executeQuery(sql);
        }
        catch (Exception e){

JOptionPane.showMessageDialog(null,e.getMessage(),"Error"
,0);  }   return rs;}
}
```

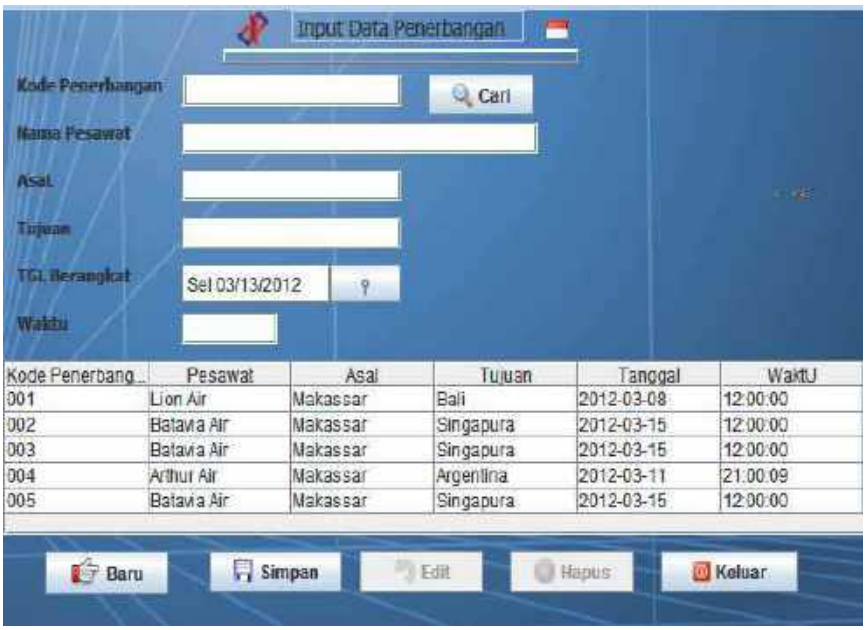
B. Perancangan Form Input.

- Rancang Form Input penerbangan dengan ketententuan berikut :
  - ✓ Nama Frame : **Finput_penerbangan**
  - ✓ Sebelum memasukan komponen ke dalam form, pada jendela **Inspector** atur **Layout** form pada **Abslute layout**. seperti gambar dibawa ini:



Gambar 7.6 Pembentukan form input data penerbangan

- Rancang form input data penerbangan seperti gambar berikut :



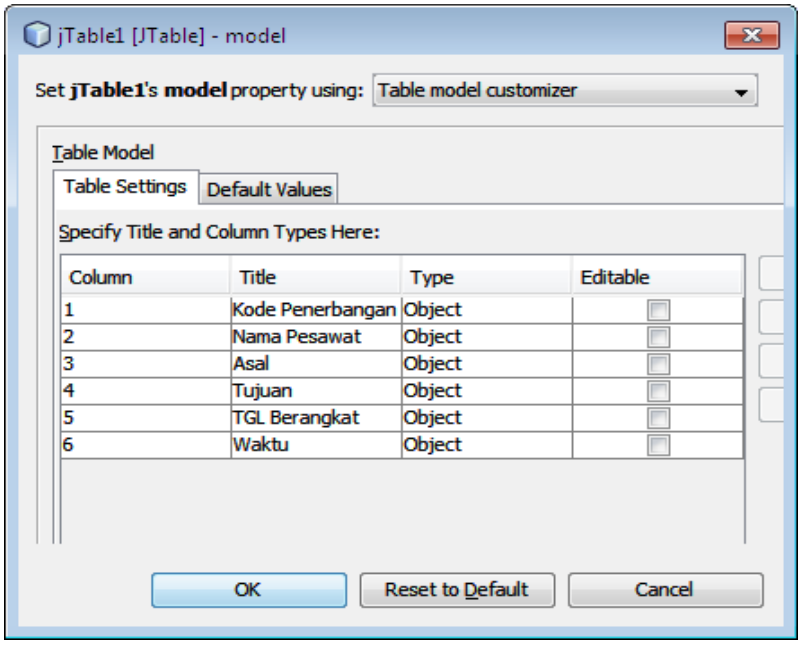
Gambar 7.7 Perancangan form input data penerbangan

Dengan ketentuan:

Palette : Swing Controls				
Komponen	**Text**	**Icon**	**Variabel name**	**Border**
jLabel4	Asal	-	Jlabel4	Ecthed Border
jLabel9	-	Vista2.jpg	Jlabel9	Soft Bevel
jButton1	Baru	New.Png	jButton1	Bevel Border
jDatePicker1	-	-	jDatePicker1	-
jTable	-	-	Tabel_terbang	Mate Border

Tabel komponen form input penerbangan

- Untuk pengaturan gambar, bentuk pakckage yang nantinya diisi file gambar. Lokasi directory berada dalam folder **namaeProject/ src/nama_package**.
- Untuk tabel pengaturan judul kolom dapat dilakukan melalui jendela properties pada **model.** Seperti gambar berikut



Gambar 7.8 Jendela model tabel

- Untuk meload data dan menampilkan dalam komponen jTabel ketik sintaks dibawah ini kedalam kelas **public class input_penerbangan extends javax.swing.JFrame { }**.

```
import javax.swing.JOptionPane;
import java.sql.Connection;
import java.sql.SQLClientInfoException;
import java.sql.Statement;
import java.sql.ResultSet;
import java.sql.*;
import javax.swing.*;
import javax.swing.table.*;
import java.util.List;
import javax.swing.table.AbstractTableModel;
import java.util.*;
```

```
Nama kelas....{

Vector row = new Vector();
Vector dataEnt = new Vector();
DefaultTableModel mod= new
DefaultTableModel(null, row);
private String Kode_tiket="";
private String Kode_penerbang="";
private String Nama_penerbang="";
private String Asal="";
private String Tujuan="";
private String Tanggal="";
private String Waktu="";
```

```
Lanjutan..

public input_penerbangan() {
        initComponents();
        KoneksiDatabase k_terbang = new KoneksiDatabase();
        try {
            konek_penerbangan =  k_terbang.getConnection();
             String sql="select * from penerbang";
             Statement stm=
konek_penerbangan.createStatement();
             ResultSet rs= stm.executeQuery(sql);
             while (rs.next())      {   }
             stm.close();
        }catch(Exception e)
        {
System.out.println(e.getMessage());
        }
```

Lanjutan..

```
row.add("Kode Penerbangan");
        row.add("Pesawat");
        row.add("Asal");
        row.add("Tujuan");
        row.add("Tanggal");
        row.add("WaktU");
         try {
            String sql="select * from tb_penerbangan order by
kd_terbang asc";
            Statement stm=
konek_penerbangan.createStatement();
            ResultSet rs= stm.executeQuery(sql);
            dataEnt=new Vector();
            Vector dataTbl=new Vector();
            while (rs.next())
            {
               Vector dataRow=new Vector();
daftar_penerbangan enty= new daftar_penerbangan();
               dataRow.add(rs.getString("kd_terbang"));
               dataRow.add(rs.getString("Pesawat"));
               dataRow.add(rs.getString("asal"));
               dataRow.add(rs.getString("tujuan"));
               dataRow.add(rs.getString("tgl_berangkat"));
               dataRow.add(rs.getString("wkt"));
```

Lanjutan..

```
enty.setKode_penerbang(rs.getString("kd_terbang"));
enty.setNama_penerbang(rs.getString("Pesawat"));
              enty.setAsal(rs.getString("asal"));
               enty.setTujuan(rs.getString("tujuan"));
               enty.setTanggal(rs.getString("tgl_berangkat"));
               enty.setWaktu(rs.getString("wkt"));
               dataEnt.add(enty);
               dataTbl.add(dataRow);
        }    mod.setDataVector(dataTbl, row);
            tabel_terbang.setModel(mod);
        }catch(Exception e)
         {   e.printStackTrace();
            System.out.println("view tables "+e.getMessage());
}
```

Sintaks diatas menghubungkan class **Finput_penerbangan** dengan class **daftar_penerbangan**, dimana pengisian variabel dalam class daftar_penerbangan dilakukan dalam class ini, sekaligus menset data kedalam tabel untuk ditampilkan.

- Untuk mengatur posisi frame agar selalu berada ditengah saat pertama tampil, pada jendela inspector klik kanan frame pilih **Events-> window-> windowactivated** laluketik sintaks berikut:

```
Lakukan import pada 2 library dibawah ini:
import java.awt.Dimension;
import java.awt.Toolkit;

Masukan sintaks kedalam method "formWindowActivated"
Dimension posisi = Toolkit.getDefaultToolkit().getScreenSize();
    int x = (posisi.width - this.getWidth())/2;
    int y = (posisi.height - this.getHeight())/2;
    this.setLocation(x,y);
```

- Pada tombol **Baru** ketik sintaks berikut :

```
jButton3.setVisible(false);
jButton4.setVisible(false);
jTextField1.setText=" ";
jTextField2.setText=" ";
jTextField3.setText=" ";
jTextField4.setText=" ";
jTextField5.setText=" ";
```

Keterangan : Sintaks diatas menset tombol Edit dan Hapus pada kondisi tidak aktif. Serta area text field menjadi kosong.

- Selanjutnya ketik sintaks dibawah ini pada tombol **Simpan.**

```
KoneksiDatabase sambung = new KoneksiDatabase();
        Kode_penerbang  =(String)jTextField1.getText();
Nama_penerbang =(String)jTextField2.getText();
        Asal            =(String)jTextField3.getText();
        Tujuan          =(String)jTextField4.getText();
        Waktu           =(String)jTextField5.getText();
SimpleDateFormat t5 = new SimpleDateFormat("yyyy-MM-dd");
Tanggal         =(String) t5.format(jXDatePicker1.getDate());
if ((jTextField1.getText().equals("")) ||
    (jTextField2.getText().equals("")) ||
    (jTextField3.getText().equals("")) ||
    (jTextField4.getText().equals("")) ||
    (jTextField5.getText().equals("")))
{
JOptionPane.showMessageDialog(null,"Maaf, data anda belum
lengkap","Warning",JOptionPane.WARNING_MESSAGE);
jTextField1.requestFocus();
```

```
} else{try konek_penerbangan = sambung.getConnection();
String sql="insert into tb_penerbangan
(kd_terbang,Pesawat,asal,tujuan,tgl_berangkat,wkt)
values(?,?,?,?,?,?)" ;
PreparedStatement pStmn=
konek_penerbangan.prepareStatement(sql);
            pStmn.setString(1,Kode_penerbang);
            pStmn.setString(2,Nama_penerbang);
            pStmn.setString(3,Asal);
            pStmn.setString(4,Tujuan);
            pStmn.setString(5,Tanggal);
            pStmn.setString(6,Waktu);
          pStmn.executeUpdate();
            pStmn.close();
          }catch(Exception e)
          {  System.out.println(e.getMessage());   }
        this.dispose();
 FInput_penerbangan refresh = new FInput_penerbangan();
          refresh.setVisible(true);    }
```

Pada sintaks tombol simpan, untuk mengantisipasi terjadinya penyimpanan data yang kosong dilakukan metode dengan menolak jika komponen **jTextField** inputan dalam kondisi kosong karena nilai inputan yang berada pada komponen tersebut akan dimasukan kedalam variabel yang berhubungan dengan field pada tabel.

Untuk memasukan data kedalam tabel, hal yang harus diperhatikan adalah urutan indeks field saat merancang tabel yang disesuaikan dengan variabel yang akan mengisinya.

- Selanjutnya pada tombol **Edit** ketik sintaks berikut :

```
KoneksiDatabase sambung = new KoneksiDatabase();
Kode_penerbang  =(String)jTextField1.getText();
Nama_penerbang  =(String)jTextField2.getText();
 Asal           =(String)jTextField3.getText();
 Tujuan          =(String)jTextField4.getText();
  Waktu                =(String)jTextField5.getText();
SimpleDateFormat t5 = new SimpleDateFormat("yyyy-MM-dd");
Tanggal         =(String) t5.format(jXDatePicker1.getDate());
if ((jTextField1.getText().equals("")) ||
    (jTextField2.getText().equals("")) ||
    (jTextField3.getText().equals("")) ||
    (jTextField4.getText().equals("")) ||
    (jTextField5.getText().equals("")))
```

Lanjutan...

```
{JOptionPane.showMessageDialog(null,"Maaf, data anda belum
lengkap","Informasi",JOptionPane.WARNING_MESSAGE);
jTextField1.requestFocus();
} else
{try {konek_penerbangan = sambung.getConnection();
        String sql = "update tb_penerbangan set
Pesawat='"+jTextField2.getText()+"',"
                    + " asal='"+jTextField3.getText()+"', "
                    + "tujuan='"+jTextField4.getText()+"', "
                    + "tgl_berangkat='"+Tanggal+"',"
                    + "wkt='"+jTextField5.getText()+
                    "' where kd_terbang='"+jTextField1.getText()+"'";
          PreparedStatement pStmn=
konek_penerbangan.prepareStatement(sql);
          Statement Stmn=
konek_penerbangan.createStatement();
          Stmn.execute(sql);
          Stmn.close(); }catch(Exception e)
          {   System.out.println(e.getMessage());  }
          this.dispose();
          FInput_penerbangan refresh = new
FInput_penerbangan();
          refresh.setVisible(true);   }
```

- Selanjutnya pada tombol **Hapus**, ketik sintaks dibawah ini:

```
KoneksiDatabase sambung = new KoneksiDatabase();
Kode_penerbang  =(String)jTextField1.getText();
if ((jTextField1.getText().equals("")) )
{JOptionPane.showMessageDialog(null,"Maaf, Input anda belum
lengkap","Informasi",JOptionPane.WARNING_MESSAGE);
jTextField1.requestFocus();
} else
{try {   konek_penerbangan = sambung.getConnection();
     String sql="delete from tb_penerbangan where
kd_terbang='"+jTextField1.getText()+"' ";
          PreparedStatement pStmn=
konek_penerbangan.prepareStatement(sql);
          Statement Stmn=
konek_penerbangan.createStatement();
          Stmn.execute(sql);
          Stmn.close();
          }catch(Exception e)
          { System.out.println(e.getMessage()); }
          this.dispose();
          FInput_penerbangan refresh = new
FInput_penerbangan();
          refresh.setVisible(true);       }
```

- Buatlah class dengan nama **validasi_penerbangan.java,** Selanjutnya ketik sintaks berikut untuk mencegah terjadinya redudansi (data yang sama dalam satu tabel).

```
private Connection konek_penerbangan = null;
   public String kunci_penerbangan="";
   public String xkode="";
 public void cari(String kode_terbang)
{ try    {   KoneksiDatabase sambung = new KoneksiDatabase();
           konek_penerbangan = sambung.getConnection();
           String cari="Select * from tb_penerbangan where
kd_terbang='"+kode_terbang+"'";
Statement statement=konek_penerbangan.createStatement();
       ResultSet rs = statement.executeQuery(cari);
       if (rs.next()) {  xkode = rs.getString(1);
          xpesawat = rs.getString(2);
          xasal = rs.getString(3);
          xtujuan = rs.getString(4);
          xtgl = (String)rs.getString(5);
          xwaktu = rs.getString(6);  }    }
       catch (SQLException ex) {
          System.out.println("SQLException: " + ex.getMessage());
          System.out.println("SQLState: " + ex.getSQLState());
          System.out.println("VendorError: " + ex.getErrorCode());
       }          }
```

Sintaks tersebut nantinya akan dieksekusi melalui tombol **Simpan** pada form Input Data Penerbangan, dengan memasukan sintak berikut dan menempatkannya pada bagian sebelum sintaks untuk menyimpan dieksekusi (baris paling atas).

```
Lanjutan...
validasi_penerbangan validasi = new
validasi_penerbangan();
      validasi.cari(jTextField1.getText());
      if (validasi.xkode=="")
{ sintak menyimpan data..... }
    else
       {     int  jwb =JOptionPane.showConfirmDialog(null,"Input
Kode Penerbangan Sudah terdaftar...., Akan Melanjutkan
Proses??", "Informasi",JOptionPane.YES_NO_OPTION);
      if (jwb==JOptionPane.YES_OPTION) {
         jTextField1.setText("");
         jTextField1.requestFocus();
      } else    {  this.setDefaultCloseOperation(EXIT_ON_CLOSE);
      dispose();      }
```

- Selanjutnya untuk membantu memudahkan proses pada tombol **Edit** dan **Hapus**, pada komponen **jTable** lakukan klik kanan **Events-> Mouse-> mouseClicked** dan ketik sintak berikut :

```
int xrow = tabel_tiket.getSelectedRow();
String Kode_terbang =(String) tabel_terbang.getValueAt(xrow,
0);
String nm_pesawat   =(String) tabel_terbang.getValueAt(xrow,
1);
        String asl =(String) tabel_terbang.getValueAt(xrow, 2);
        String tjuan =(String)tabel_terbang.getValueAt(xrow, 3);
        String tgL_Br =(String) tabel_terbang.getValueAt(xrow, 4);
        String wakt=(String) tabel_terbang.getValueAt(xrow, 5);
        jTextField1.setText(Kode_terbang);
jTextField2.setText(nm_pesawat);  jTextField3.setText(asl);
```

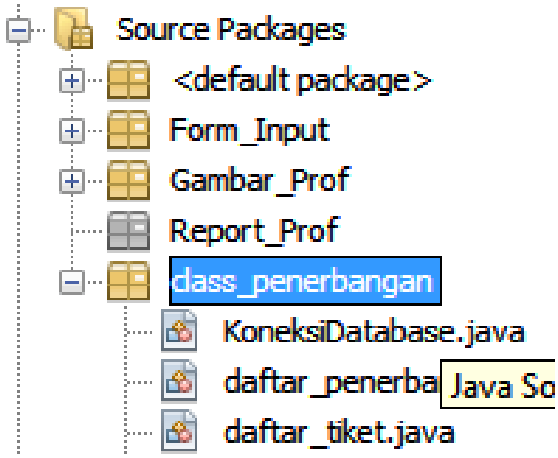
- Yang terakhir pada tombol **Cari** ketik sintaks berikut:

```
validasi_penerbangan caridata = new validasi_penerbangan();
       caridata.cari(jTextField1.getText());
       if (caridata.xkode==""){
          JOptionPane.showConfirmDialog(null,"Data Tidak
Ditemukan..","Informasi",JOptionPane.OK_OPTION);
       jTextField1.requestFocus(); }
 else { jButton3.setEnabled(true);
 jButton4.setEnabled(true);
          jTextField2.setText(caridata.xpesawat);
          jTextField3.setText(caridata.xasal);
          jTextField4.setText(caridata.xtujuan);
          jTextField5.setText(caridata.xwaktu);
jXDatePicker1.setDate(java.sql.Date.valueOf(caridata.xtgl));      }
```

- Hasil dari perancangan form Input data penerbangan adalah:

Gambar 7.9 Hasil form input data penerbangan

- Jika dilihat hasil perancangan form input data penerbangan, status tombol **Edit**dan **Hapus** yaitu tidak aktif. Hal itu dikondisikan karena penggunaan kedua tombol tersebut berdasarkan ketentuan input kode penerbangan. Maka dari itu, kedua tombol tersebut akan aktif melalui tombol **Cari**pada saat menemukan kode yang terdaftar.
- Untuk persiapan dalam merancang form input data tiket, Lakukan enkapsulasi data tiket melalui class **daftar_tiket.java,** tempatkan dalam package **class_penerbangan.** Seperti gambar berikut :



Gambar 7.10 Perancangan package class_penerbangan

Untuk melakukan pemindahan class / form ke package yang berbeda dapat menggunakan fasilitas **Repactor->Copy.**

- Pada class **daftar_tiket.java** ketik sintaks berikut:

```
public int daf_harga=0;
    public int diskon=0;
    public int bayar=0;
    public String Kode_tiket="";
    private String jenis_tiket="";
    private int harga=0;

public String getKode_tiket(){return Kode_tiket; }
    public void setKode_tiket(String Kode_tiketx){
        this.Kode_tiket=Kode_tiketx;}

 public String getjenis_tiket(){    return jenis_tiket;   }
   public void setjenis_tiket(String Jenis_tikett){
 this.jenis_tiket=Jenis_tikett;  }
```

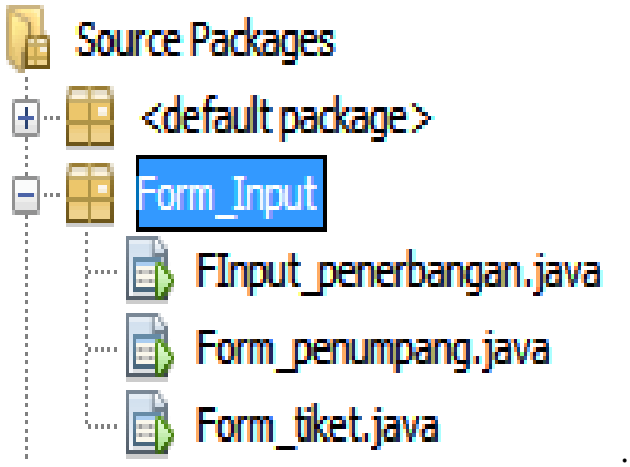
Lanjutan..

```
public void setharga(int Harga){    this.harga = Harga;   }
 public int getharga(){     return harga;    }

 public void daftar_harga (int potongan,int harga){
    daf_harga=harga;
diskon = potongan;
 if (potongan==25)
{
bayar = daf_harga-(daf_harga * 25/100);
}
 else if (potongan==50)
 {
bayar = daf_harga-(daf_harga * 50/100);
}
 else  if (potongan==70)
 {
bayar = daf_harga-(daf_harga * 70/100);}
 else {bayar = daf_harga * 1;} }
```

Pada sintaks diatas terdapat method **daftar_harga** untuk memproses perhitungan harga tiket, yang nantinya dideklarasikan melalui form input data tiket pada komponen combobox.

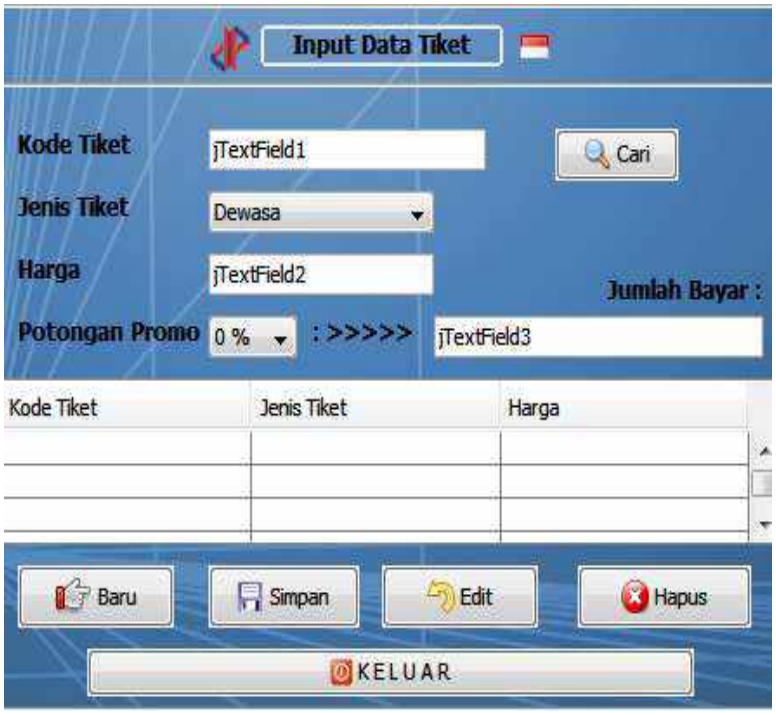
- Selanjutnya kita akan merancang form Input data tiket.

  Untuk memudahkan pengaturan tiap form, buatlah terlebih dahulu package dengan nama **Form_Input**, yang nantinya berisi daftar form yang akan digunakan dalam program seperti gambar berikut :



Gambar 7.11 Perancangan package Form_Input

- Perancangan form Input data tiket seperti gambar berikut :



Gambar 7.12 Perancangan form input data tiket

Dengan ketentuan :

Palette : Swing Controls				
Komponen	**Text**	**icon**	**Variabel name**	**Border**
jLabel4	Asal	-	Jlabel4	Ecthed Border
jLabel9	-	Vista2.jpg	Jlabel9	Soft Bevel
jButton1	Baru	New.Png	jButton1	Bevel Border
jComboBox1	-	-	jComboBox1	-
jTable	-	-	Tabel_tiket	Mate Border

- Pengaturan dan perlakuan pada form Input data tiket hampir sama dengan form sebelumnya, yaitu diawali dengan mengetikan sintaks untuk meload data dan menampilkan kedalam komponen jTabel pada kelas **public class Form_Tiket extends javax.swing.JFrame { }**.

```
Vector row = new Vector();
Vector dataEnt = new Vector();
DefaultTableModel mod= new DefaultTableModel(null, row);
private String Kode_tiket="";
private String jenis_tiket="";
private int harga=0;

private Connection konek_tiket=null;
public Form_tiket() {
KoneksiDatabase k_terbang = new KoneksiDatabase();
try { konek_tiket =  k_terbang.getConnection();
String sql="select * from tb_tiket";
Statement stm= konek_tiket.createStatement();
ResultSet rs= stm.executeQuery(sql);
while (rs.next()){   }
stm.close();
}catch(Exception e){System.out.println(e.getMessage());}
```

Lanjutan...

```
row.add("Kode Tiket");
row.add("Jenis Tiket");
row.add("Harga");
try {
String sql="select * from tb_tiket order by kd_tiket asc";
Statement stm= konek_tiket.createStatement();
ResultSet rs= stm.executeQuery(sql);
dataEnt=new Vector();
Vector dataTbl=new Vector();
while (rs.next())
{   Vector dataRow=new Vector();
daftar_penerbangan enty= new daftar_penerbangan();
dataRow.add(rs.getString("kd_tiket"));
dataRow.add(rs.getString("jns_tiket"));
dataRow.add(rs.getString("harga"));
enty.setKode_penerbang(rs.getString("kd_tiket"));
enty.setNama_penerbang(rs.getString("jns_tiket"));
enty.setAsal(rs.getString("harga"));
dataEnt.add(enty);
dataTbl.add(dataRow);}
mod.setDataVector(dataTbl, row);
tabel_tiket.setModel(mod);
}catch(Exception e)
{e.printStackTrace();
System.out.println("view tables "+e.getMessage());}
```

- Untuk mengatur posisi frame agar selalu berada ditengah saat pertama tampil, pada jendela inspector klik kanan frame pilih **Events-> window-> windowactivated lalu** ketik sintak berikut:

Lakukan import pada 2 library dibawah ini:

```
import java.awt.Dimension;
import java.awt.Toolkit;
```

Masukan sintaks kedalam method "formWindowActivated"

```
Dimension posisi = Toolkit.getDefaultToolkit().getScreenSize();
    int x = (posisi.width - this.getWidth())/2;
    int y = (posisi.height - this.getHeight())/2;
    this.setLocation(x,y);
```

- Pada tombol **Baru** ketik sintaks berikut :

```
jButton3.setEnabled(false);
        jButton4.setEnabled(false);
        jTextField1.setText("");
        jTextField2.setText("0");
        jTextField3.setText("0");
        jTextField1.requestFocus();
```

- Sebelum memasukan sintaks kedalam tombol simpan sebaiknya dibentuk dahulu class **validasi_tiket.java**. Sintaknya seperti berikut:

```
public class validasi_tiket
{
    private Connection konek_tiket = null;
    public String kunci_penerbangan="";
    public String xkode="";
    public String xjenis="";
    public int xharga=0;

 public void cari(String kode_tiket)
    { try
        {
 KoneksiDatabase sambung = new KoneksiDatabase();
           konek_tiket = sambung.getConnection();
           String cari=
"Select * from tb_tiket where kd_tiket='"+kode_tiket+"'";
        Statement statement=konek_tiket.createStatement();
        ResultSet rs = statement.executeQuery(cari);
        if (rs.next())
```

Lanjutan...

```
{ xkode = rs.getString(1);
          xjenis = rs.getString(2);
          xharga = rs.getInt(3);
         }
  }
       catch (SQLException ex) {
          System.out.println("SQLException: " +
ex.getMessage());
          System.out.println("SQLState: " + ex.getSQLState());
          System.out.println("VendorError: " + ex.getErrorCode());
       }  }}
```

- Selanjutnya ketik sintaks dibawah ini pada tombol **Simpan.**

```
KoneksiDatabase sambung = new KoneksiDatabase();
      validasi_tiket validasi = new validasi_tiket();
      validasi.cari(jTextField1.getText());  if (validasi.xkode.equals(""))
{    Kode_tiket   =(String)jTextField1.getText();
jenis_tiket  =(String)jTextField2.getText();
     harga        = new Integer(jTextField3.getText()).intValue();
 if ((jTextField1.getText().equals("")) ||
    (jTextField2.getText().equals("")) ||
    (jTextField3.getText().equals(""))){
JOptionPane.showMessageDialog(null,"Maaf, data anda belum
lengkap","Informasi",JOptionPane.WARNING_MESSAGE);
jTextField1.requestFocus();
} else
{ try {        konek_tiket = sambung.getConnection();
 String sql="insert into tb_tiket (kd_tiket,jns_tiket,harga)
values(?,?,?)" ;
PreparedStatement pStmn= konek_tiket.prepareStatement(sql);
pStmn.setString(1,Kode_tiket);
pStmn.setString(2,jenis_tiket);
 pStmn.setInt(3,harga);
 pStmn.executeUpdate();
 pStmn.close();
           }catch(Exception e) {
System.out.println(e.getMessage());}
         this.dispose();
Form_tiket refresh = new Form_tiket();
         refresh.setVisible(true);          }
       } else  {       int jwb;
       jwb =JOptionPane.showConfirmDialog(null,"Input Kode Tiket
Sudah terdaftar...., Akan Melanjutkan Proses??",
           "peringatan",JOptionPane.YES_NO_OPTION);
```

Lanjutan...

```
if (jwb==JOptionPane.YES_OPTION) {
        this.dispose();
        Form_tiket refresh = new Form_tiket();
        refresh.setVisible(true);    jTextField1.setText("");
        jTextField1.requestFocus();
} else
 {   this.setDefaultCloseOperation(EXIT_ON_CLOSE);
      dispose();    } }
```

- Selanjutnya pada tombol **Edit** ketik sintaks berikut :

```
KoneksiDatabase sambung = new KoneksiDatabase();
       Kode_tiket   =(String)jTextField1.getText();
jenis_tiket  =(String)jComboBox1.getSelectedItem();
       harga        = new Integer(jTextField3.getText()).intValue();
if ((jTextField1.getText().equals("")) ||
    (jTextField3.getText().equals("")))
{JOptionPane.showMessageDialog(null,"Maaf, data anda belum
lengkap","Informasi",JOptionPane.WARNING_MESSAGE);
jTextField1.requestFocus();
} else{try {
           konek_tiket = sambung.getConnection();
           String sql = "update tb_tiket set jns_tiket='"+
jComboBox1.getSelectedItem()+"',"+
 " harga='"+jTextField3.getText()+
    "' where kd_tiket='"+jTextField1.getText()+"'";
 PreparedStatement pStmn= konek_tiket.prepareStatement(sql);
           Statement Stmn= konek_tiket.createStatement();
           Stmn.execute(sql);
           Stmn.close();
           }catch(Exception e)
           {      System.out.println(e.getMessage());   }
           this.dispose();
           Form_tiket refresh = new Form_tiket();
           refresh.setVisible(true);
           jButton3.setEnabled(true);
          jButton4.setEnabled(true);    }
```

- Selanjutnya pada tombol **Hapus**, ketik sintaks dibawah ini:

```
KoneksiDatabase sambung = new KoneksiDatabase();
       Kode_tiket  =(String)jTextField1.getText();
if ((jTextField1.getText().equals("")) )
{JOptionPane.showMessageDialog(null,"Maaf, data anda belum
lengkap","Informasi",JOptionPane.WARNING_MESSAGE);
jTextField1.requestFocus();
} else
```

Lanjutan...

```
{try {   konek_tiket = sambung.getConnection();
          String sql="delete from tb_tiket where
kd_tiket='"+jTextField1.getText()+"' ";
          PreparedStatement pStmn=
konek_tiket.prepareStatement(sql);
          Statement Stmn= konek_tiket.createStatement();
          Stmn.execute(sql);
          Stmn.close();
          }catch(Exception e)
          { System.out.println(e.getMessage()); }
          this.dispose();
          Form_tiket refresh = new Form_tiket();
          refresh.setVisible(true);
          jButton3.setEnabled(true);
          jButton4.setEnabled(true);
       }{try {   konek_penerbangan = sambung.getConnection();
     String sql="delete from tb_penerbangan where
kd_terbang='"+jTextField1.getText()+"' ";
          PreparedStatement pStmn=
konek_penerbangan.prepareStatement(sql);
Statement Stmn= konek_penerbangan.createStatement();
          Stmn.execute(sql);
          Stmn.close();    }catch(Exception e)
          { System.out.println(e.getMessage()); }
          this.dispose();
 Form_tiket refresh = new Form_tiket();
          refresh.setVisible(true);       }
```

- Selanjutnya pada tombol **Cari** ketik sintaks berikut :

```
validasi_tiket caridata = new validasi_tiket();
      caridata.cari(jTextField1.getText());
      if (caridata.xkode.equals(""))
{
          JOptionPane.showConfirmDialog(null,"Data Tidak
Ditemukan..","Informasi",JOptionPane.OK_OPTION);
      jTextField1.requestFocus();
      }
else {
          jButton3.setEnabled(true);
          jButton4.setEnabled(true);
          jComboBox1.getModel().setSelectedItem(caridata.xjenis);
         jTextField3.setText(Integer.toString(caridata.xharga));
      }
```

- Untuk membantu memudahkan proses pada tombol **Edit** dan **Hapus**, pada komponen **jTable** lakukan klik kanan **Events-> Mouse-> mouseClicked** dan ketik sintak berikut :

```
int row2 = tabel_tiket.getSelectedRow();
        String Kde_tiket =(String) tabel_tiket.getValueAt(row2, 0);
        String jnis   =(String) tabel_tiket.getValueAt(row2, 1);
        String hrg =(String) tabel_tiket.getValueAt(row2, 2);

        jTextField1.setText(Kde_tiket);
        jComboBox1.getModel().setSelectedItem(jnis);
        jTextField3.setText(hrg);
```

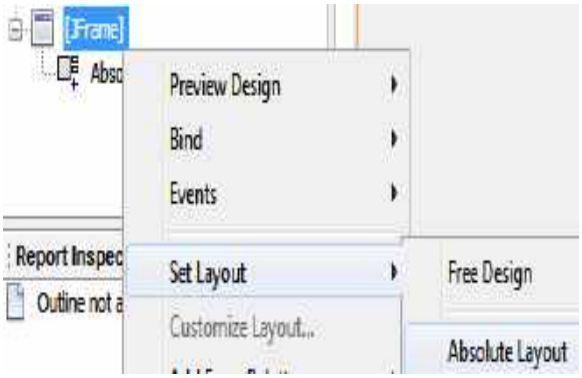
- Hasil dari perancangan form Input data tiket adalah:

Gambar 7.13 Hasil form input data tiket

- Form inputan terakhir yang akan dirancang adalah form input data penerbangan. Persiapan awal dalam merancang form input data penumpang adalah dengan melakukan enkapsulasi data penumpang melalui **daftar_penumpang.java** danmenempatkannya dalam package **class_penerbangan.** Lalu pada class **daftar_tiket.java** ketik sintaks berikut:

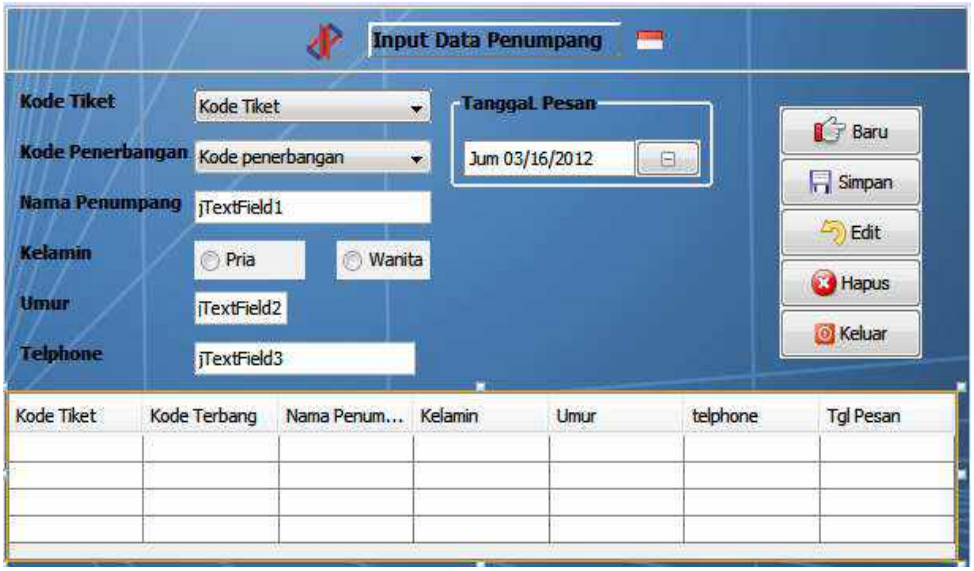
```
public String Kode_Tket="";
private String Kode_terbg="";
private String nama="";
private String kelamin="";
private String umur="";
private String telp="";
private String tgl_pesn="";
public void setKode_Tket(String kd_tket)
   {  this.Kode_Tket=kd_tket;  }
public String getKode_Tket()
   {    return Kode_Tket;  }
public void setKode_Terbg(String kd_trbg)
   {   this.Kode_terbg=kd_trbg; }
public String getKodeterbg()
   {    return Kode_terbg;  }
public void setnama (String nm)
   {   this.kelamin=nm;  }
public String getnama()
   {    return nama;    }
public void setKelamin(String jk)
   {    this.kelamin=jk;  }
public String getKelamin()
   {    return kelamin;  }
public void setumur(String usia)
   {    this.umur=usia;    }
public String getumur()
   {    return umur;  }
public void settgl_pesan(String tglPesn)
   {    this.tgl_pesn=tglPesn; }
public String gettgl_pesan()
  {    return tgl_pesn;  }
```

- Rancang Form Input penumpang dengan ketentuan sebagai berikut :
  - ✓ Nama Frame : Form_penumpang
  - ✓ pada jendela **Inspector** atur **Layout** form pada **Abslute layout**.



Gambar 7.14 Setting layout form

- Rancang Form Input data penumpang seperti gambar berikut :



Gambar 7.15 Perancangan form input data penumpang

Dengan ketentuan:

Palette : Swing Controls				
Komponen	**Text**	**icon**	**Variabel name**	**Border**
jLabel4	Umur	-	Jlabel4	Ecthed Border
jLabel9	-	Vista2.jpg	Jlabel9	Soft Bevel
jButton1	Baru	New.Png	jButton1	Bevel Border
jDatePicker1	-	-	jDatePicker1	-
jTable	-	-	tabel_pnp	Mate Border
jComboBox1	-	-	jComboBox1	-
jComboBox2	-	-	jComboBox1	-

- Untuk meload data dan menampilkan dalam komponen jTabel ketik sintaks dibawah ini kedalam kelas **public class input_penumpang extends javax.swing.JFrame { }**.

```
import javax.swing.JOptionPane;
import java.sql.Connection;
import java.sql.SQLClientInfoException;
import java.sql.Statement;
import java.sql.ResultSet;
import java.sql.*;
import javax.swing.*;
import javax.swing.table.*;
import java.util.List;
import javax.swing.table.AbstractTableModel;
import java.util.*;
```

```
Nama Kelas.....{
Vector row = new Vector();
Vector dataEnt = new Vector();
DefaultTableModel mod= new
DefaultTableModel(null, row);
private String Kode_tket="";
private String kode_terbg="";
private String nama_penumpang="";
private String kel="";
private String tlp="";
private String T_pesan="";
private Connection konek_penumpang=null;
private Connection konek_penerbangan=null;
```

Lanjutan..

```
public Form_penumpang() {
initComponents();
     KoneksiDatabase k_penumpang = new
KoneksiDatabase();
       try { konek_penumpang =
k_penumpang.getConnection();
            String sql="select * from tb_penumpang";
            Statement stm=
konek_penumpang.createStatement();
            ResultSet rs= stm.executeQuery(sql);
            while (rs.next())
 {  }
            stm.close();
       }catch(Exception e)
       {          System.out.println(e.getMessage());       }
```

Lanjutan..

```
row.add("Kode Tiket");
        row.add("Kode Penerbangan");
        row.add("Nama Penumpang");
        row.add("Kelamin");
        row.add("Umur");
        row.add("Phone");
        row.add("TGL PESAN");
         try {
        String sql="select * from tb_penumpang order by nama
asc";
            Statement stm=
konek_penumpang.createStatement();
            ResultSet rs= stm.executeQuery(sql);
            dataEnt=new Vector();
            Vector dataTbl=new Vector();
            while (rs.next())
            {   Vector dataRow=new Vector();
                daftar_penerbangan enty= new
daftar_penerbangan();
                dataRow.add(rs.getString("kd_tiket"));
                dataRow.add(rs.getString("kd_terbang"));
                dataRow.add(rs.getString("nama"));
                dataRow.add(rs.getString("jk"));
                dataRow.add(rs.getString("umur"));
                dataRow.add(rs.getString("telp"));
                dataRow.add(rs.getString("tgl_pesan"));
```

Lanjutan..

```
enty.setKode_penerbang(rs.getString("kd_tiket"));
enty.setNama_penerbang(rs.getString("kd_terbang"));
                enty.setAsal(rs.getString("nama"));
                enty.setTujuan(rs.getString("jk"));
                enty.setTanggal(rs.getString("umur"));
                enty.setWaktu(rs.getString("telp"));
                enty.setWaktu(rs.getString("tgl_pesan"));
                dataEnt.add(enty);
                dataTbl.add(dataRow);
}      mod.setDataVector(dataTbl, row);
            tabel_pnp.setModel(mod);
        }catch(Exception e)        {
System.out.println("view tables "+e.getMessage());      }
```

Sintaks diatas menghubungkan class **Finput_penumpang** dengan class **daftar_penumpang**, dimana pengisian variabel dalam class daftar_penumpang dilakukan dalam class ini, sekaligus menset data kedalam tabel untuk ditampilkan.

- Untuk mengatur posisi frame agar selalu berada ditengah saat pertama tampil, pada jendela inspector klik kanan frame pilih **Events-> window-> windowactivated lalu** ketik sintak berikut:

```
Lakukan import pada 2 library dibawah ini:
import java.awt.Dimension;
import java.awt.Toolkit;

Masukan sintaks kedalam method "formWindowActivated"
Dimension posisi = Toolkit.getDefaultToolkit().getScreenSize();
    int x = (posisi.width - this.getWidth())/2;
    int y = (posisi.height - this.getHeight())/2;
    this.setLocation(x,y);
```

- Selanjutnya didalam form terdapat komponen jCombobox1, dimana item dalam komponen ini akan diisi dengan kode tiket dari tabel tb_tiket, dan komponen jComboBox2 diisi dengan kode penerbangan dari tabel tb_penerbangan. Untuk meload kedua data tersebut, , pada jendela inspector klik kanan frame pilih **Events-> window-> WindowOpened** laluketik sintak berikut :

```
KoneksiDatabase isicombo= new KoneksiDatabase();
 try
 {konek_penumpang  = isicombo.getConnection();
 String sqlku ="select kd_tiket from tb_tiket";
 Statement stmku = konek_penumpang.createStatement();
 ResultSet panggil_sql = stmku.executeQuery(sqlku);

 String sqlku2 ="select kd_terbang from tb_penerbangan";
 Statement stmku2 = konek_penumpang.createStatement();
 ResultSet panggil_sql2 = stmku2.executeQuery(sqlku2);

while (panggil_sql.next())
 { jComboBox1.addItem(panggil_sql.getString("kd_tiket")); }

 while (panggil_sql2.next())
 {jComboBox2.addItem(panggil_sql2.getString("kd_terbang"));}
          panggil_sql.close();       panggil_sql2.close();
 }
catch (Exception e)
 {
JOptionPane.showConfirmDialog(null,"Proses koneksi
GAGAL....","Informasi",JOptionPane.YES_OPTION);
 }
```

- Pada tombol **Baru** ketik sintaks berikut :

```
jButton3.setVisible(false);
jButton4.setVisible(false);
jTextField1.setText=" ";
jTextField2.setText=" ";
jTextField3.setText=" ";
```

- Pada tombol **Simpan** ketik sintaks berikut :

```
KoneksiDatabase sambung = new KoneksiDatabase();
validasi_penumpang validasi = new validasi_penumpang();
validasi.validasi_penumpang(jComboBox1.getSelectedItem().toString()
, jComboBox2.getSelectedItem().toString());
if (validasi.kunci_xterbang.equals("") &&
validasi.kunci_xtiket.equals("")){
      Kode_tket  =(String)jComboBox1.getSelectedItem();
kode_terbg  =(String)jComboBox2.getSelectedItem();
      nama_penumpang  =(String)jTextField1.getText();
      if (jRadioButton1.isSelected()){kel = "Pria"; }
      if (jRadioButton2.isSelected()){kel = "Wanita"; }
      tlp         =(String)jTextField3.getText();
      umr         =(String)jTextField2.getText();
      SimpleDateFormat t5 = new SimpleDateFormat("yyyy-MM-dd");
      T_pesan        =(String) t5.format(jXDatePicker1.getDate());
```

```
if ((jTextField1.getText().equals("")) ||  (jTextField2.getText().equals(""))
|| (jTextField3.getText().equals("")) ||
(jComboBox1.getSelectedItem().equals("Pilih Kode Tiket")) ||
   (jComboBox1.getSelectedItem().equals("Pilih Kode Penerbangan")))
{JOptionPane.showMessageDialog(null,"Maaf, data anda belum
lengkap","Informasi",JOptionPane.WARNING_MESSAGE);
jTextField1.requestFocus();
} else{ try {
konek_penumpang = sambung.getConnection();
      String sql="insert into tb_penumpang
(kd_tiket,kd_terbang,nama,jk,umur,telp,tgl_pesan)
values(?,?,?,?,?,?,?)" ;
            PreparedStatement pStmn=
konek_penumpang.prepareStatement(sql);
            pStmn.setString(1,Kode_tket);
            pStmn.setString(2,kode_terbg);
            pStmn.setString(3,nama_penumpang);
            pStmn.setString(4,kel);
            pStmn.setString(5,umr);
            pStmn.setString(6,tlp);
            pStmn.setString(7,T_pesan);
pStmn.executeUpdate();
            pStmn.close(); }catch(Exception e)
         { System.out.println(e.getMessage()); }
```

Lanjutan....

```
this.dispose();
          Form_penumpang refresh = new Form_penumpang();
          refresh.setVisible(true);       }
      } else  {
      int jwb;
      jwb =JOptionPane.showConfirmDialog(null,"Input Kode
Penerbangan Sudah terdaftar...., Akan Melanjutkan Proses??",
          "peringatan",JOptionPane.YES_NO_OPTION);
      if (jwb==JOptionPane.YES_OPTION) {
        this.dispose();
          Form_penumpang refresh = new Form_penumpang();
          refresh.setVisible(true);
        jTextField1.setText("");
        jTextField1.requestFocus();
      } else  {
      this.setDefaultCloseOperation(EXIT_ON_CLOSE);
      dispose();       } }
```

Pada sintaks tombol simpan, untuk mengantisipasi terjadinya penyimpanan data yang kosong dilakukan metode dengan menolak proses simpan jika tidak dilakukan pemilihan item pada komponen jComboBox1 dan jComboBox2.Karena nilai inputan yang berada pada komponen tersebut akan dimasukan kedalam variabel yang berhubungan dengan field pada tabel sebagai kunci validasi.

- Pada tombol **Edit** ketik sintaks berikut

```
KoneksiDatabase sambung = new KoneksiDatabase();
 validasi_penumpang validasi = new validasi_penumpang();
validasi.validasi_penumpang(jComboBox1.getSelectedItem().toString(),
jComboBox2.getSelectedItem().toString());
if (validasi.kunci_xterbang.equals("") && validasi.kunci_xtiket.equals(""))
{ JOptionPane.showConfirmDialog(null,"Data kode
tiket"+validasi.kunci_xtiket+"dan kode terbang validasi.kunci_xterbang+
"Tidak Ditemukan","Informasi",JOptionPane.OK_OPTION);
jComboBox1.requestFocus(); } else { Kode_tket
=(String)jComboBox1.getSelectedItem();
          kode_terbg      =(String)jComboBox2.getSelectedItem();
nama_penumpang  =(String)jTextField1.getText();
if (jRadioButton1.isSelected()) kel = "Pria";  else kel = "Wanita";
tlp=(String)jTextField3.getText();umr  =(String)jTextField2.getText();
       SimpleDateFormat tfrmt = new SimpleDateFormat("yyyy-MM-dd");
       T_pesan        =(String) tfrmt.format(jXDatePicker1.getDate());
```

Lanjutan...

```
if ((jTextField1.getText().equals("")) ||    (jTextField2.getText().equals(""))
|| (jTextField3.getText().equals("")) ||
(jComboBox1.getSelectedItem().equals("Pilih Kode Tiket")) ||
(jComboBox2.getSelectedItem().equals("Pilih Kode Penerbangan")))
{JOptionPane.showMessageDialog(null,"Maaf, data anda belum
lengkap", "Informasi",JOptionPane.WARNING_MESSAGE);
jTextField1.requestFocus();
} else{try {          konek_penumpang = sambung.getConnection();
String sql = "update tb_penumpang set
nama='"+nama_penumpang+"',"
                    + "jk='"+kel+"',"
                    + "umur='"+umr+"',"
                    + "telp='"+tlp+"',tgl_pesan='"+T_pesan+"'"
                    + "where kd_tiket='"+Kode_tket+"'and kd_terbang="
                    + "'"+ kode_terbg+"'";
          Statement Stmn= konek_penumpang.createStatement();
          Stmn.executeUpdate(sql);
          Stmn.close();
          }catch(Exception e)       {
             System.out.println(e.getMessage());      }
          this.dispose();
          Form_penumpang refresh = new Form_penumpang();
          refresh.setVisible(true);
          jButton3.setEnabled(true);   jButton4.setEnabled(true);
```

Pada sintaks diatas sebelum melakukan proses edit data terlebih dahulu dilakukan proses identifikasi terhadap variabel kode tiket dan kode penerbangan, jika keduanya tidak ditemukan bersamaan dalam satu record maka data dianggap tidak ditemukan sebagaimana telah diatur sebelumnya pada class **validasi_penumpang.**

- Pada tombol **Hapus** ketik sintaks dibawah ini :

```
KoneksiDatabase sambung = new KoneksiDatabase();
 validasi_penumpang validasi = new validasi_penumpang();
validasi.validasi_penumpang(jComboBox1.getSelectedItem().toString(),
jComboBox2.getSelectedItem().toString());
      if (validasi.kunci_xterbang.equals("") &&
validasi.kunci_xtiket.equals("")){
          JOptionPane.showConfirmDialog(null,"Data kode
tiket"+validasi.kunci_xtiket+"dan kode terbang
"+validasi.kunci_xterbang+"Tidak
Ditemukan","Informasi",JOptionPane.OK_OPTION);
     jComboBox1.requestFocus();  }
```

Lanjutan...

```
else {     Kode_tket       =(String)jComboBox1.getSelectedItem();
           kode_terbg      =(String)jComboBox2.getSelectedItem();
if ((jComboBox1.getSelectedItem().equals("Pilih Kode Tiket")) ||
    (jComboBox2.getSelectedItem().equals("Pilih Kode Penerbangan")))
{  JOptionPane.showMessageDialog(null,"Maaf, data anda belum
lengkap",
                        "Informasi",JOptionPane.WARNING_MESSAGE);
jComboBox1.requestFocus();
} else   {  try {          konek_penumpang = sambung.getConnection();
      String sql = "delete from tb_penumpang where
                         kd_tiket='"+Kode_tket+"'and kd_terbang="
                      + "'"+ kode_terbg+"'";
           Statement Stmn= konek_penumpang.createStatement();
           Stmn.executeUpdate(sql);
           Stmn.close();
           }catch(Exception e)
           {  System.out.println(e.getMessage());    }
           this.dispose();
           Form_penumpang refresh = new Form_penumpang();
           refresh.setVisible(true);
           jButton3.setEnabled(true);
          jButton4.setEnabled(true); }}
```

- Pada komponen **jTable** lakukan klik kanan **Events**-> **Mouse**-> **mouseClicked** dan ketik sintak berikut :

```
int pilih = tabel_pnp.getSelectedRow();
        String Kde_tkt =(String) tabel_pnp.getValueAt(pilih, 0);
        String Kde_terbg =(String) tabel_pnp.getValueAt(pilih, 1);
        String nm_pnp   =(String) tabel_pnp.getValueAt(pilih, 2);
        String jkl =(String) tabel_pnp.getValueAt(pilih, 3);
        String tlpn =(String)tabel_pnp.getValueAt(pilih, 4);
        String umuR =(String) tabel_pnp.getValueAt(pilih, 5);
        String Tgl=(String) tabel_pnp.getValueAt(pilih, 6);

        jTextField1.setText(nm_pnp);
        jTextField2.setText(umuR);
        jTextField3.setText(tlpn);
        if (jkl.equals("Pria")){jRadioButton1.setSelected(true);}
        if (jkl.equals("Wanita")){jRadioButton2.setSelected(true);}
        jComboBox1.getModel().setSelectedItem(Kde_tkt);
        jComboBox2.getModel().setSelectedItem(Kde_terbg);
        jXDatePicker1.setDate(java.sql.Date.valueOf(Tgl));
```

- Pada tombol **Cari** ketik sintaks berikut :

```
validasi_penumpang caridata = new validasi_penumpang();
caridata.validasi_penumpang(jComboBox1.getSelectedItem().
toString(),jComboBox2.getSelectedItem().toString());

 if (caridata.kunci_xtiket.equals("") &&
caridata.kunci_xterbang.equals(""))
{
 JOptionPane.showConfirmDialog(null,"Data Tidak
Ditemukan..","Informasi",JOptionPane.OK_OPTION);
jComboBox1.requestFocus();
 }
else
{    jButton3.setEnabled(true);
jButton4.setEnabled(true);
jTextField1.setText(caridata.xnama);
jTextField2.setText(caridata.xumur);
jTextField3.setText(caridata.xtlp);

if (caridata.xjkel.equals("Pria"))
          jRadioButton1.setSelected(true);
          else
jRadioButton2.setSelected(true);

jComboBox1.getModel().setSelectedItem(caridata.kunci_xtiket);
jComboBox2.getModel().setSelectedItem(caridata.kunci_xterbang);
jXDatePicker1.setDate(java.sql.Date.valueOf(caridata.xtgl_pesan));
      }
```

- Hasil Perancangan form input data penumpang seperti berikut :

Gambar 7.16 Hasil form input data penumpang

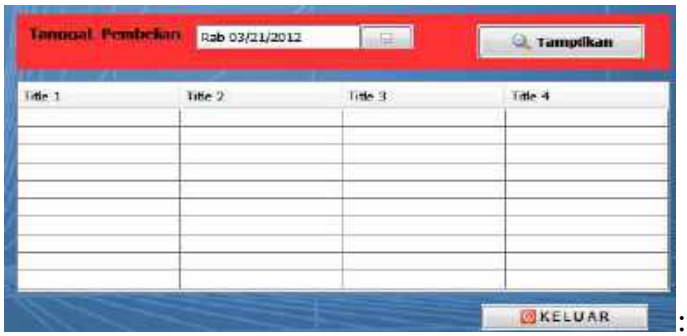
C. Perancangan Form Cari Data.

- Rancang class **daftar_caridata.java** dalam packageclass_penerbangan, yang nantinya akan digunakan pada form pencarian data penumpang. Sintaks sebagai berikut :

```
public class daftar_caridata {
public String K_tiket="";
private String kode_penerbng="";
private String namaP="";
private String tglBeli="";
private int hargaB=0;
public void setcari_Ktiket (String kt){
    this.K_tiket = kt;}
public String getCari_Ktiket(){
    return K_tiket;  }
public void setcari_Kterbang (String kdter){
    this.kode_penerbng = kdter;  }
public String getCari_Kterbg(){
    return kode_penerbng;     }
public void setcari_Npen (String NmP){
    this.namaP = NmP;}
public String getCari_Npen(){
    return namaP;}
public void setcari_TBeli (String TgBl){
    this.tglBeli = TgBl;}
public String getCari_TgBli(){
    return tglBeli;}
public void setharga(int Harga){
   this.hargaB = Harga;    }
 public int getharga(){
        return hargaB;    } }
```

- Selanjutnya rancang form pencarian data penumpang dengan nama **Form_cari.java** dan tempatkan dalam package **Form_Input**.

  Gambar perancangan form sebagai berikut :



Gambar 7.17 Hasil form input data penumpang

Dengan ketentuan:

Palette : Swing Controls				
Komponen	**Text**	**icon**	**Variabel name**	**Border**
jLabel1	Tgl Beli	-	Jlabel1	Ecthed Border
jLabel2	-	Vista2.jpg	Jlabel2	Soft Bevel
jButton1	Tampil	New.Png	jButton1	Bevel Border
jDatePicker1	-	-	jDatePicker1	-
jTable	-	-	tabel_cari	Mate Border

- Untuk meload data dan menampilkan dalam komponen jTabel ketik sintaks dibawah ini kedalam kelas **public class input_penerbangan extends javax.swing.JFrame { }**.

Lanjutan..

```
private Vector row = new Vector();
private Vector dataEnt = new Vector();
private Vector dataTbl=new Vector();
private DefaultTableModel model= new
DefaultTableModel(dataTbl, row);

Vector xrow = new Vector();
Vector xdataEnt = new Vector();
Vector xdataTbl=new Vector();
DefaultTableModel xmodel= new
DefaultTableModel(xdataTbl, xrow);

private String Kode_tket="";
private String kode_terbg="";
private String nama_penumpang="";
private String kel="";
private String umr="";
private String tlp="";
private String T_pesan="";
private Connection konek_cari=null;
```

Lanjutan..

```
public Form_Cari() {
        initComponents();
         KoneksiDatabase k_cari = new KoneksiDatabase();
        try {    konek_penumpang =
k_penumpang.getConnection();
            String sqlCari="select
tb_tiket.kd_tiket,tb_penumpang.kd_terbang,nama,tgl_pesan,
harga from tb_tiket, tb_penumpang where tb_tiket.kd_tiket =
tb_penumpang.kd_tiket";
            Statement stm=
konek_penumpang.createStatement();
            ResultSet rs= stm.executeQuery(sqlCari);
            while (rs.next())            {   }
            stm.close();
        }catch(Exception e)
```

Lanjutan..

```
{  System.out.println(e.getMessage());   }
        row.add("Kode Tiket");
        row.add("Kode Penerbangan");
        row.add("Nama Penumpang");
        row.add("TanggaL Pesan");
        row.add("Harga");
         try {    String sqlCari="select
tb_tiket.kd_tiket,tb_penumpang.kd_terbang,nama,tgl_pesan,
harga from tb_tiket, tb_penumpang where tb_tiket.kd_tiket =
tb_penumpang.kd_tiket";
            Statement stm= konek_cari.createStatement();
            ResultSet rs= stm.executeQuery(sqlCari);
            dataEnt=new Vector();
            while (rs.next())
            {   Vector dataRow=new Vector();
                daftar_caridata enty= new daftar_caridata();
                dataRow.add(rs.getString("kd_tiket"));
                dataRow.add(rs.getString("kd_terbang"));
                dataRow.add(rs.getString("nama"));
                dataRow.add(rs.getString("tgl_pesan"));
                dataRow.add(rs.getString("harga"));
                enty.setcari_Ktiket(rs.getString("kd_tiket"));
                enty.setcari_Kterbang(rs.getString("kd_terbang"));
                enty.setcari_Npen(rs.getString("nama"));
                enty.setcari_TBeli(rs.getString("tgl_pesan"));
                enty.setharga(rs.getInt("harga"));
```

Lanjutan..

```
dataEnt.add(enty);
dataTbl.add(dataRow);      }
            model.setDataVector(dataTbl, row);
            tabel_cari.setModel(model);
       }catch(Exception e)
{ System.out.println("view tables "+e.getMessage());    }
        model = new DefaultTableModel(dataTbl,row){
           Class[] types = new Class [] {
              java.lang.String.class,
              java.lang.String.class,
              java.lang.String.class,
              java.lang.String.class,
              java.lang.Integer.class};
         public Class getColumnClass(int columnIndex)
         { return types [columnIndex]; }
public boolean isCellEditable  (int row, int col) {
              int cola = xmodel.getColumnCount();
              return (col < cola) ? false : true;
     } };         tabel_cari.setModel(model);    }
```

Sintaks diatas menghubungkan class **Form_cari** dengan class **daftar_caridata**, dimana pengisian variabel dalam class daftar_caridata dilakukan dalam class ini, sekaligus menset data kedalam tabel untuk ditampilkan.

Didalam class diatas terdapat sintaks yang digunakan untuk menset tabel menjadi readonly melalui method **isCellEditable**. Pengaturan readonly tabel dapat dilakukan pada saat menggunakan komponen jTable kedalam form.

- Untuk mengatur posisi frame agar selalu berada ditengah saat pertama tampil, pada jendela inspector klik kanan frame pilih **Events-> window-> windowactivated lalu** ketik sintak berikut:

Lakukan import pada 2 library dibawah ini:

```
import java.awt.Dimension;
import java.awt.Toolkit;
```

Masukan sintaks kedalam method "formWindowActivated"

```
Dimension posisi = Toolkit.getDefaultToolkit().getScreenSize();
   int x = (posisi.width - this.getWidth())/2;
   int y = (posisi.height - this.getHeight())/2;
   this.setLocation(x,y);
```

- Klik ganda pada tombol **Tampilkan** lalu ketik sintaks berikut :

```
SimpleDateFormat tglF = new SimpleDateFormat("yyyy-MM-dd");
        String tgl_pbeli = tglF.format(jXDatePicker1.getDate());
        int barisku = xmodel.getRowCount();
        for (int i = 0;i < barisku;i++)
xmodel.removeRow(0);
        KoneksiDatabase k_cari = new KoneksiDatabase();
        try {       konek_cari =  k_cari.getConnection();
String sqlCari="select
tb_tiket.kd_tiket,tb_penumpang.kd_terbang,nama,tgl_pesan,harga
from tb_tiket, tb_penumpang where tb_tiket.kd_tiket =
tb_penumpang.kd_tiket";
            Statement stm= konek_cari.createStatement();
            ResultSet rs= stm.executeQuery(sqlCari);
            while (rs.next())
            {   }  stm.close();
        }catch(Exception e)       {
           System.out.println("tidak Ada Judul");       }
        xrow.add("Kode Tiket");
        xrow.add("Kode Penerbangan");
        xrow.add("Nama Penumpang");
        xrow.add("TanggaL Pesan");
        xrow.add("Harga");
         try {          String sqlCari="select
tb_tiket.kd_tiket,tb_penumpang.kd_terbang,nama,tgl_pesan,harga
from tb_tiket,"+ " tb_penumpang where tb_tiket.kd_tiket =
tb_penumpang.kd_tiket "+ " and tb_penumpang.tgl_pesan =
'"+tgl_pbeli+"'";
```

Lanjutan.....

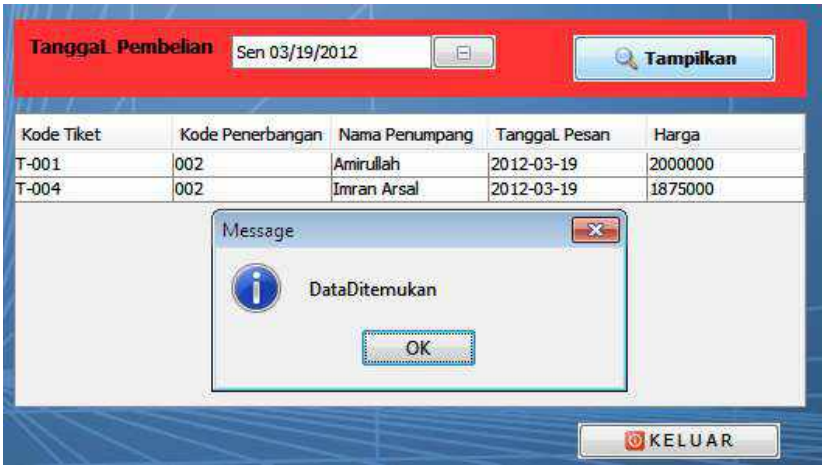
```
Statement stm= konek_cari.createStatement();
            ResultSet rs= stm.executeQuery(sqlCari);
            xdataEnt=new Vector();
            while (rs.next())
            {   Vector xdataRow=new Vector();
                daftar_caridata enty= new daftar_caridata();
                xdataRow.add(rs.getString("kd_tiket"));
                xdataRow.add(rs.getString("kd_terbang"));
                xdataRow.add(rs.getString("nama"));
                xdataRow.add(rs.getString("tgl_pesan"));
                xdataRow.add(rs.getString("harga"));
                enty.setcari_Ktiket(rs.getString("kd_tiket"));
                enty.setcari_Kterbang(rs.getString("kd_terbang"));
                enty.setcari_Npen(rs.getString("nama"));
                enty.setcari_TBeli(rs.getString("tgl_pesan"));
                enty.setharga(rs.getInt("harga"));
                xdataEnt.add(enty);
                xdataTbl.add(xdataRow);
```

Lanjutan.....

```
            } xmodel.setDataVector(xdataTbl, xrow);
            tabel_cari.setModel(xmodel);
            if (xdataTbl.isEmpty()){
            JOptionPane.showMessageDialog(this," Data Tidak
Ditemukan");
            jXDatePicker1.requestFocus();
            dispose();
            Form_Cari refresh = new Form_Cari();
            refresh.setVisible(true); }
           else {
           JOptionPane.showMessageDialog(this," DataDitemukan");
            jXDatePicker1.requestFocus();
            dispose();
            Form_Cari refresh = new Form_Cari();
            refresh.setVisible(true);
           }
       } catch(Exception e)
        {  System.out.println("view tables "+e.getMessage());   } }
```

Pada sintaks diatas terdapat object **xmodel** dari class DefaultmodelTable yang digunakan untuk menampung hasil query, dimana setiap akan dilakukan eksekusi proses query, tabel akan dikosongkan terlebih dahulu.

- Hasil perancangan form pencarian seperti gambar berikut :



Gambar 7.18 Form pencarian data penumpang

D. Perancangan Report.

Terdapat banyak tools untuk reporting dalam java. Diantaranya yang dapat digunakan adalah :

1. Jasper Report

   Merupakan software open source untuk reporting.

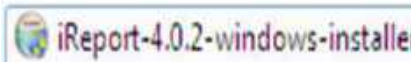
2. iReport

   Merupakan visual Designer untuk membuat laporan yang komplek menggunakan JasperReport library tanpa harus memiliki pengetahuan tentang XML.

Beberapa fitur iReport :

- 98% mendukung JasperReports tags
- Visual designer wysiwyg untuk menggambar rectangles, lines, ellipses, text fields fields, charts, sub reports…
- Built-in editor dengan syntax highlighting
- Mendukung Unicode dan bahasa non Latin (Russian, Chinese, Korean,…)
- Document structure browser
- Menggabungkan compiler dan exporter
- Mendukung semua JDBC compliant databases
- Memiliki Wizard untuk membuat report secara otomatis
- Mendukung sub reports
- Save backup
- Support for templates

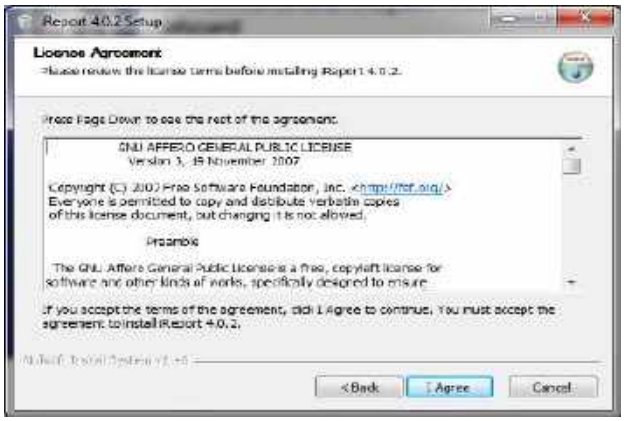
➢ Sebelum melakukan perancangan report, terlebih dahulu lakukan instalasi Ireport 4.0.2. Prosedur instalasi sebagai berikut :



➢ Klik ganda pada file **Ireport 4.0.2**

Gambar 7.19 jendela welcome instalasi Ireport 4.02

- Klik **Next**untuk melanjutkan proses instalasi.

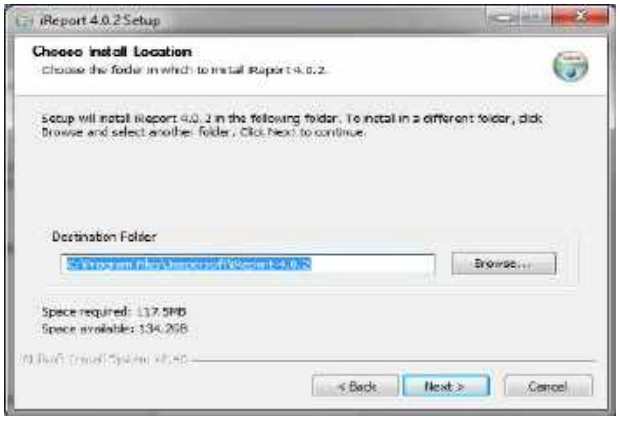


Gambar 7.20 Jendela License Agreement

- Klik tombol **I Agree** pada jendela license agreement

Gambar 7.21 Jendela pilihan komponen feature

- Klik **Next**untuk melanjutkan proses instalasi.

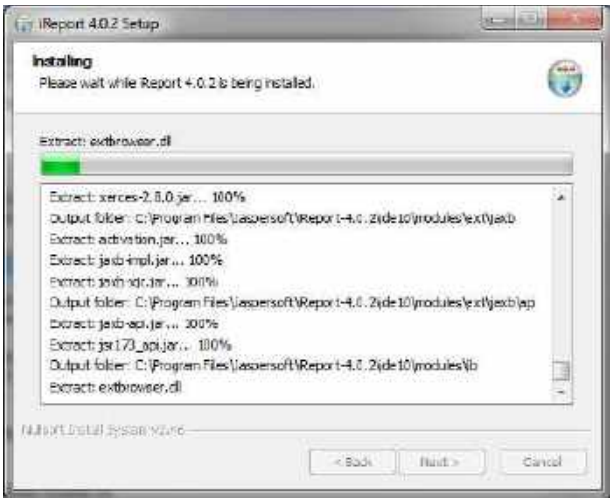


Gambar 7.22 Jendela directory instalasi IReport 4.0.2

- Tentukan directory instalasi Ireport 4.0.2 lalu klik **Next**.

Gambar 7.23 Jendela pengaturan shorcut Ireport 4.0.2

- Atur letak shortcut lalu klik **Install** untuk melanjutkan proses.



Gambar 7.24 Proses extrac file jar

- Proses extract file jar berlansung dan membutuhkan waktu beberapa menit, setelah selesai akan tampil gambar seperti berikut

Gambar 7.25 Jendela tahap akhir instalasi Ireport 4.0.2

- Tampilan IReport 4.0.2 saat loading module seperti gambar berikut .

Gambar 7.26 Tampilan Loading Ireport 4.0.2

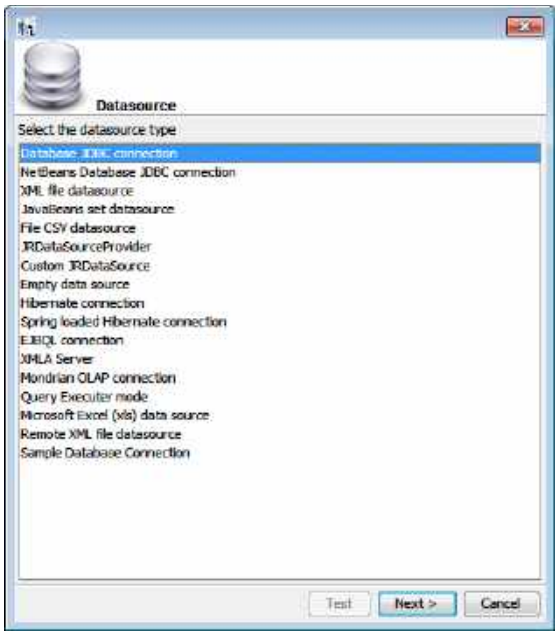
- Tampilan utama jendela IReport 4.0.2 seperti gambar berikut.

Gambar 7.27 Tampilan utama Ireport 4.0.2

- Rancang form cetak dengan kan menggunakan **Ireport 4.0.2**
- Pada project buat terlebih dahulu package yang akan menampung report dengan nama **Report_Prof**.
- Sebelum melakukan design report , bentuk terlebih dahulu datasource yang berisi koneksi database untuk report yang akan didesign.
- Langkah awal prosedure pembuatan yaitu
- Setelah aplikasi Ireport 4.0.2 aktif akan tampil seperti gambar berikut

Gambar 7.28 Jendela pembuatan datasource

- Klik gambar database atau tool datasource. Maka akan tampil seperti gambar berikut :



Gambar 7.29 Jendela pilihan tipe datasource

- Untuk tipe datasource tentukan pada bagian Database JDBC Connection. Lalu klik tombol **Next.**

Gambar 7.30 Jendela pengisian parameter datasource

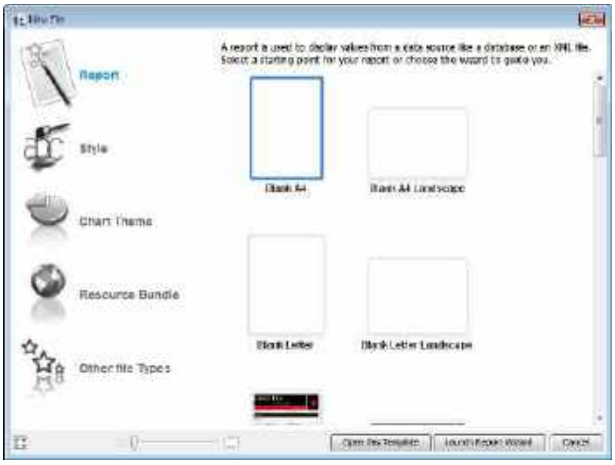
- Ketentuan pengisian parameter datasource adalah:
  - Name : DS_Tiket_profesional
  - JDBC Driver : MySQL (com.mysql.jdbc.Driver)
  - JDBC URL : jdbc:mysql://localhost/tiket_profesional

- Server Address : Localhost
- Database : tiket_profesional
- User : root
- Password : -
- Centang save password

➢ Bila semua sudah diisi dengan benar klik tombol **Test** untuk menguji koneksi jika berhasil akan tampil pesan seperti pada gambar.

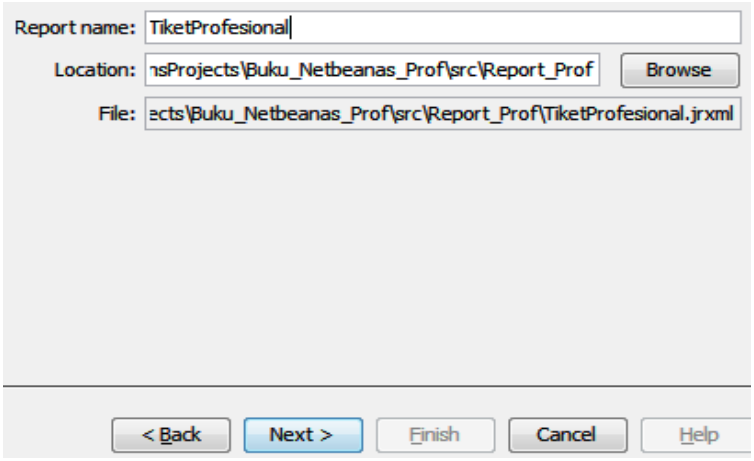
➢ Klik tombol **Save** untuk menyimpan pengaturan.

➢ Untuk membuat report baru klik menu **File => New**, maka tampil jendela model report yang ingin digunakan. klik model A4 Potrait, lalu klik launch Report Wizard Seperti gambar berikut :



Gambar 7.31 Jendela pilihan model report

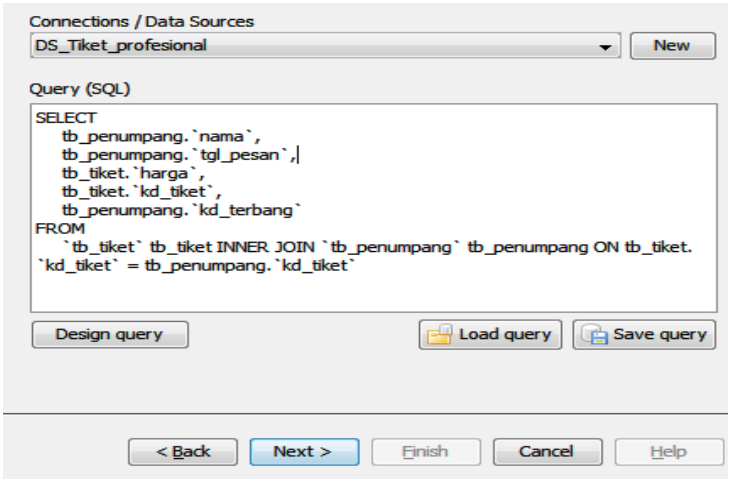
➢ Maka akan tampil jendela pengaturan letak file jrxml yang nanti setelah dicompile akan berubah menjadi file jasper.



Gambar 7.32 Jendela pengisian nama report

- Report name isi dengan nama report.
- Location : tentukan alamat package **Report_Prof**, agar pada saat dieksekusi report sudah berada dalam project.
- Contoh : C:\Users\Satellite L510\Documents\NetBeansProjects\Buku_Netbeanas_Prof\src\Report_Prof.
- Alamat package berada dalam folder **src** pada project.

➢ Klik **Next** untuk melanjutkan proses.

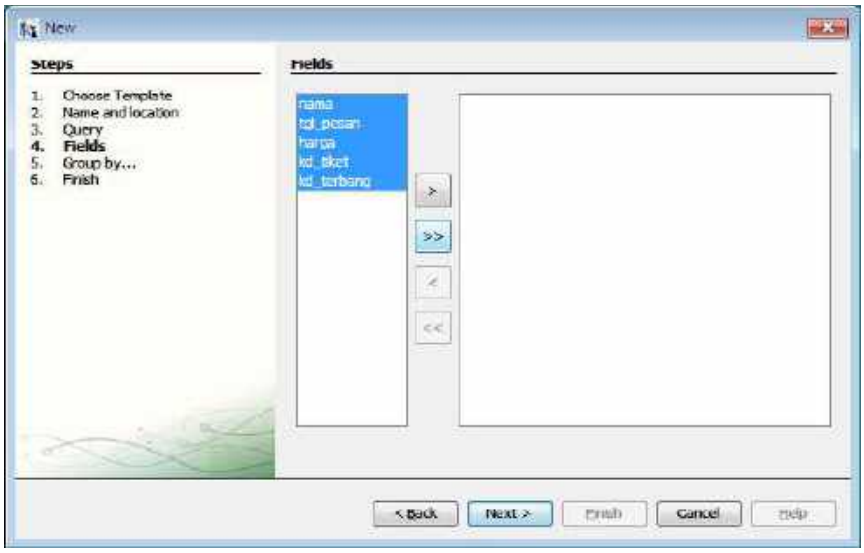


Gambar 7.33 Jendela design query report

➢ Setelah tampil seperti gambar diatas, pada bagian Query ketik sintaks berikut :

```
SELECT
    tb_penumpang.nama,
    tb_penumpang.tgl_pesan,
    tb_tiket.harga,
    tb_tiket.kd_tiket,
    tb_penumpang.kd_terbang
FROM
tb_tiket tb_tiket INNER JOIN tb_penumpang
tb_penumpang ON tb_tiket.kd_tiket =
tb_penumpang.kd_tiket
```

➢ Klik **Next** untuk melanjutkan, maka tampil gambar berikut



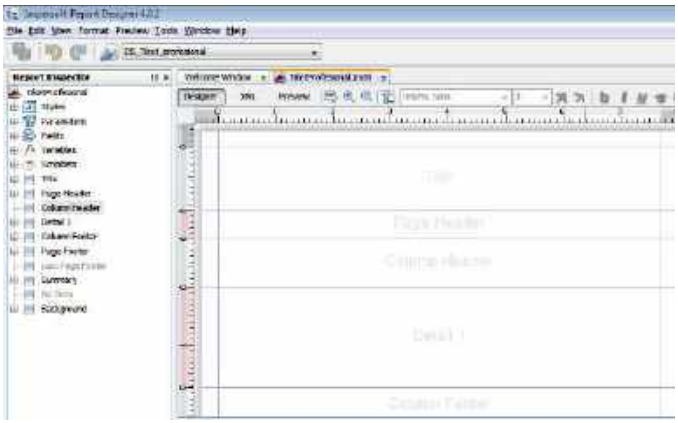
Gambar 7.34 Jendela seleksi fields

- Seleksi field yang akan digunakan dalam report. Dengan menekan tombol

- Klik tombol **Next** maka tampil seperti gambar berikut :
- Jika berhasil maka akan tempil jendela seperti berikut :

Gambar 7.35 Jendela tahap akhir pembuatan datasource

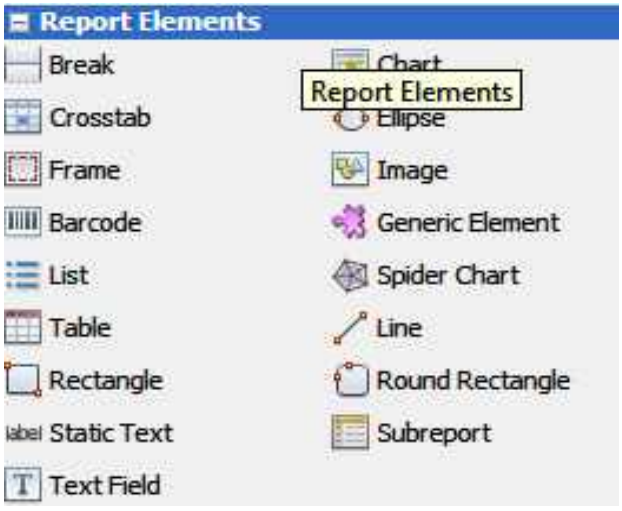
- Klik **Finish** untuk mengakhiri pengaturan file Jrxml.
- Aktifkan netbeans untuk proses design lebih lanjut.
- Pada package Report_prof klik ganda file TiketProfesional.jrxml Maka akan tampil jendela untuk mendesign report seperti gambar berikut.



Gambar 7.36 Tampilan area design Ireport 4.0.2

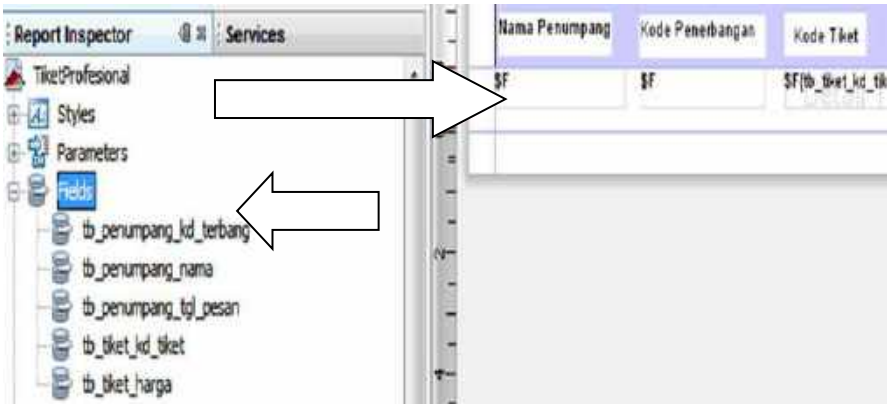
- Pada jendela report terdapat beberapa band yang dapat digunakan untuk keperluan tertentu. Jika ingin dihilangkan, klik kanan pada band yang tidak dibutuhkan lalu klik **Delete Band.**

- Gunakan beberapa komponen tambahan yang diinginkan pada jendela Report Element seperti gambar berikut :



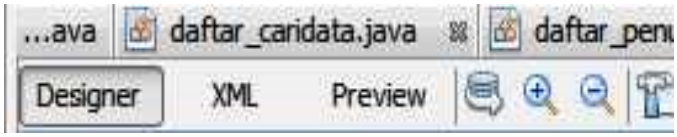
Gambar 7.37 Tampilan report element

- Untuk memasukan field yang telah disiapkan melalui sintaks query sebelumnya, pada jendela Report Inspector bagian **Fields** klik dan seret field yang diinginkan ke jendela design dan tempatkan pada band **Details**. Seperti gambar berikut :



Gambar 7.38 Proses design report

- Untuk melihat hasil design klik tombol **Preview**
- Untuk kembali pada jendela desin klik **Designer**

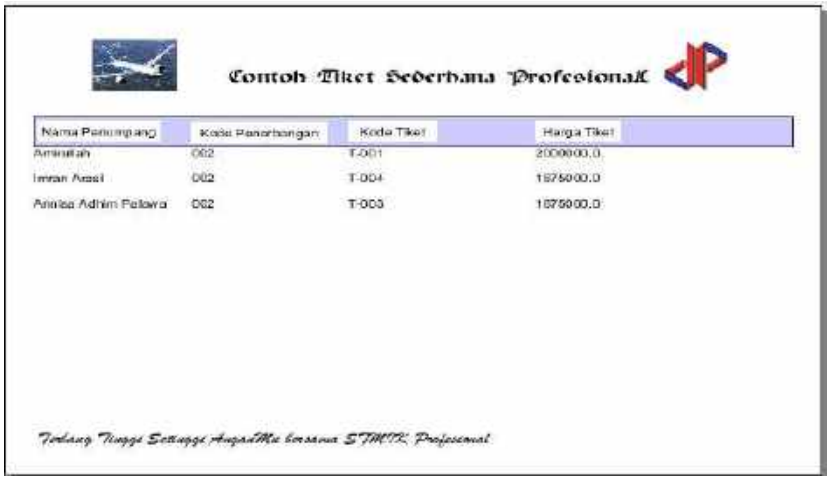


Gambar 7.39 Tombol preview

- Hasil perancangan report seperti gambar berikut.

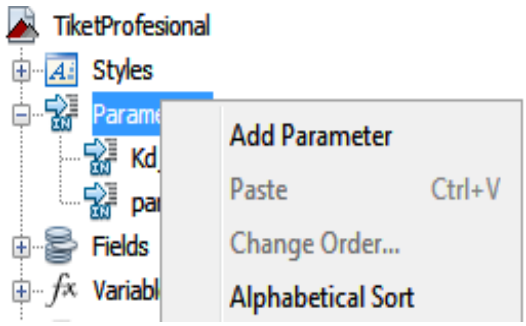
Gambar 7.40 Tampilan hasil perancangan report

- Hasil preview perancangan report seperti gambar berikut :



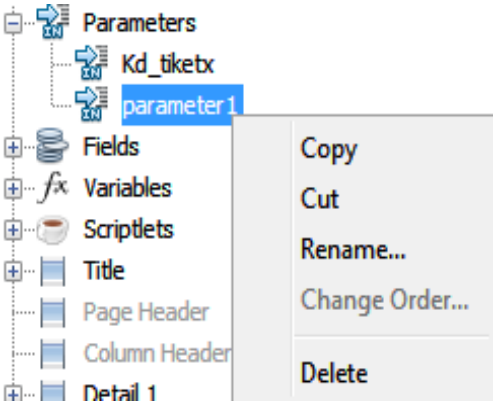
Gambar 7.41 Tampilan hasil preview perancangan report

- Selanjutnya akan dibuat parameter pada report, yang nantinya dapat digunakan untuk kebutuhan saat melakukan filter data. Prosedur pembuatan parameter sebagai berikut :
- Pada report inspector klik kanan **Parameter** =>**add parameter**



Gambar 7.42 Pembuatan parameter query

- Setelah terbentuk, klik kanan parameter lalu pilih R**ename**., lalu isi dengan field yg akan menjadi parameter didalam sintaks query sql.

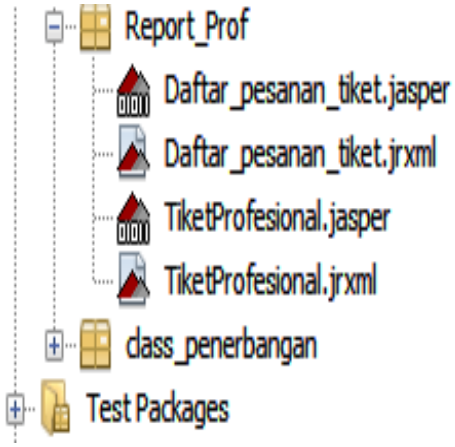


Gambar 7.43 Rename parameter

- Setelah semua komponen dianggap cukup, lakukan **compile** file report jrxml agar terbentuk file Jasper.
- File jasper inilah yang nantinya dipanggil melalui sintaks melalui form cetak.

Gambar 7.44 Compile file jasper

- File jrxml dan jasper sudah berada dalam package **Report_Prof.**



Gambar 7.45 Daftar file jrxml dan jasper

E. Perancangan form cetak.

- Contoh yang pertama akan dilakukan perancangan form cetak tiket dengan nama **Form_Cetak.java** seperti gambar berikut :

Gambar 7.46 Perancangan form cetak tiket

- Untuk mengisi item combobox kode tiket dan kode penerbangan, seperti yang dilakukan pada form penumpang dimana data akan diambil dari tabel tb_tiket dan tb_penerbangan dengan meload data dari tabel tersebut melalui sintaks seperti berikut

```
try {
 KoneksiDatabase k_cetak = new KoneksiDatabase();
 kon_cetak  = k_cetak.getConnection();
 String sqlku ="select kd_tiket from tb_tiket";
 Statement stm_kdtiket = kon_cetak.createStatement();
 ResultSet panggil_sql = stm_kdtiket.executeQuery(sqlku);
 String sqlku2 ="select kd_terbang from tb_penerbangan";
 Statement stm_kdterbang = kon_cetak.createStatement();
 ResultSet panggil_sql2 = stm_kdterbang.executeQuery(sqlku2);
 while (panggil_sql.next())
    {
          jComboBox1.addItem(panggil_sql.getString("kd_tiket"));
 }
 while (panggil_sql2.next())
 {
jComboBox2.addItem(panggil_sql2.getString("kd_terbang"));
}
      panggil_sql.close();
      panggil_sql2.close();
 } catch (Exception e)
 { JOptionPane.showConfirmDialog(null,"Proses koneksi GAGAL....",
          "Informasi",JOptionPane.YES_OPTION);   }
```

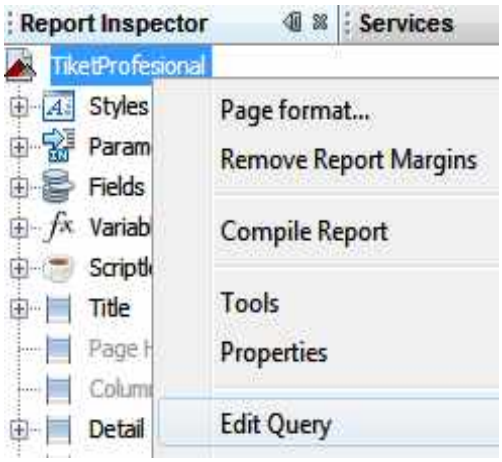
- Agar sintak untuk menampilkan report dapat tereksekusi dengan baik, terlebih dahulu masukan library yang dibutuhkan dalam sintaks cetak seperti gambar berikut :
- Selanjutnya dalam tombol **PreView** ketik sintaks berikut :

```
try   {
       String kdT=jComboBox1.getSelectedItem().toString();
       String kdTr=jComboBox2.getSelectedItem().toString();
         System.out.println("Compiling report...");
         System.out.println("Selesai!");
 String alamatFile="C:\\Users\\SatelliteL510\\”+
Documents\\NetBeansProjects\\”+
ujiReport\\src\\Report_Prof\\TiketProfesional.jasper";
         Map parameter = new HashMap();
         parameter.put ("Kd_tiketx",kdT);
         parameter.put ("kd_terbang", kdTr);
         System.out.println("get a database connection!");
         String userName = "root";
         String password = "";
         String url = "jdbc:mysql://localhost/tiket_profesional";
         Class.forName ("org.gjt.mm.mysql.Driver");
```

Lanjutan...

```
        Connection conx =
DriverManager.getConnection(url,userName,password);
        JasperPrint jasperPrint =
JasperFillManager.fillReport(alamatfile, parameter, conx);
       System.out.println("view report in the JasperViewer!");
        JasperViewer.viewReport(jasperPrint, false);
        System.out.println("Done!");
      } catch (JRException e)
      {
 JOptionPane.showMessageDialog(this, "Gagal tampilkan
report1!\n" + e);
      } catch (Exception e)
      {
JoptionPane  .showMessageDialog
(this, "Gagal tampilkan report!\n" + e);  }
```

- Selanjutnya akan dilakukan pengeksekusian terhadap file jasper yang menggunakan parameter.
- Hal yang perlu diperhatikan adalah pada sintak query dalam file jrxml terlebih dahulu disiapkan variabel yang nantinya akan diisi dengan nilai dari componen input seperti jCombobox jTextField dll
- Setelah terbentuk file jrxml pada jendela Report Inspector klik kanan nama file lalu pilih **Edit Query** seperti gambar berikut :

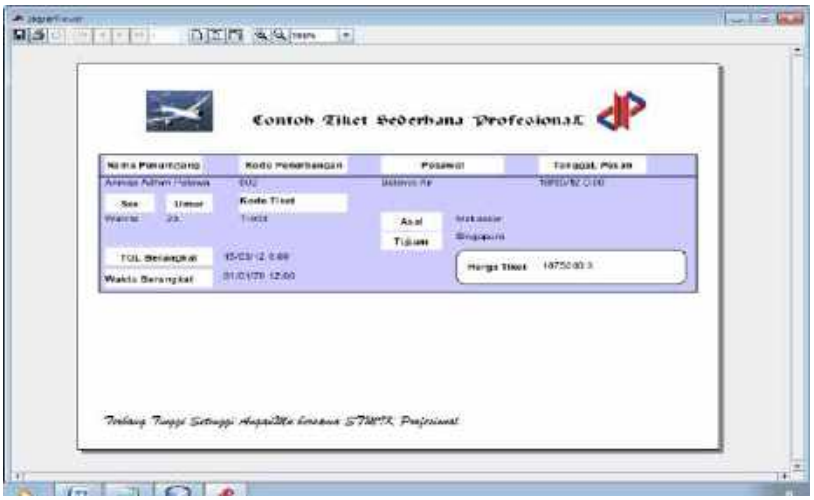


Gambar 7.47 Prosedure edit query

- Lalu ketik sintaks berikut dalam editor design query :

```
SELECT
tb_penerbangan.`Pesawat` AS tb_penerbangan_Pesawat,
    tb_penerbangan.`asal` AS tb_penerbangan_asal,
    tb_penerbangan.`tujuan` AS tb_penerbangan_tujuan,
    tb_penerbangan.`tgl_berangkat` AS
tb_penerbangan_tgl_berangkat,
    tb_penerbangan.`wkt` AS tb_penerbangan_wkt,
    tb_penumpang.`nama` AS tb_penumpang_nama,
    tb_penumpang.`jk` AS tb_penumpang_jk,
    tb_penumpang.`umur` AS tb_penumpang_umur,
    tb_penumpang.`telp` AS tb_penumpang_telp,
    tb_penumpang.`tgl_pesan` AS tb_penumpang_tgl_pesan,
    tb_tiket.`jns_tiket` AS tb_tiket_jns_tiket,
    tb_tiket.`harga` AS tb_tiket_harga,
    tb_penumpang.`kd_tiket` AS tb_penumpang_kd_tiket,
    tb_penumpang.`kd_terbang` AS tb_penumpang_kd_terbang,
    tb_tiket.`kd_tiket` AS tb_tiket_kd_tiket,
    tb_penerbangan.`kd_terbang` AS tb_penerbangan_kd_terbang
FROM
    `tb_penumpang` tb_penumpang INNER JOIN `tb_tiket` tb_tiket
ON tb_penumpang.`kd_tiket` = tb_tiket.`kd_tiket`
    INNER JOIN `tb_penerbangan` tb_penerbangan ON
tb_penumpang.`kd_terbang` = tb_penerbangan.`kd_terbang`
where tb_penumpang.kd_tiket =$P{Kd_tiketx} and
tb_penumpang.`kd_terbang` =$P{kd_terbang}
```

- Didadalm sintaks query diatas terdapat pemanggilan parameter sesuai dengan yang telah dibuat sebelumnya.
- Hasil eksekusi tombol **PreView** seperti gambar berikut :



Gambar 7.48 Hasil eksekusi tombol preview

F. Perancangan Form Menu Utama dan Login.

- Untuk keamanan aplikasi akan dirancang form login yang akan memproteksi setiap pengguna, dengan melakukan autentikasi pengguna tersebut setiap akan masuk ke menu utama.
- gambar form login seperti berikut :

Gambar 7.49 Perancangan form login

Palette : Swing Controls				
Komponen	**Text**	**icon**	**Variabel name**	**Border**
jButton1	Login	User.png	jButton1	Ecthed Border
jTextField1	-	-	jTextField1	Bevel Border
jPasswordField1	-	-	jPasswordField1	-

- Klik ganda pada tombol **LOGIN** lalu ketik sintaks berikut :

```
try {   String usernm=jTextField1.getText().trim();
        String passwd=jPasswordField1.getText().trim();
        KoneksiDatabase K_Log = new KoneksiDatabase();
        konek_login = K_Log.getConnection();
 String sql="select pass from tb_user where user='"+usernm+"'";
         Statement stm = konek_login.createStatement();
         ResultSet rs = stm.executeQuery(sql);
         if (rs.next())
         {    String pass =rs.getString("Pass");
              if (pass.equals(passwd)){
 JOptionPane.showMessageDialog(null,"SELAMAT DATANG","STMIK
PROFESIONAL MAKASSAR",JOptionPane.WARNING_MESSAGE);
Menu_Utama utama = new Menu_Utama();    utama.setVisible(true);
this.dispose();    }   else    {
JOptionPane.showMessageDialog(null,"Username atau Password
salah","Peringatan",JOptionPane.WARNING_MESSAGE);
                    ClearForm();     }            }
 Else  {      JOptionPane.showMessageDialog(null,"Username atau
Password salah","Peringatan",JOptionPane.WARNING_MESSAGE);
              ClearForm();        }          stm.close();
      }catch(Exception e)      { System.out.println("salah ");   }
```

- Klik ganda pada tombol **Reset** lalu ketik sintaks berikut :

```
jTextField1.setText("");
jPasswordField1.setText("");
jTextField1.requestFocus();
```

- Klik ganda pada tombol **Reset** lalu ketik sintaks berikut :

```
int tutup;
tutup=JOptionPane.showConfirmDialog(null,"Akan membatalkan
Login??",
          "peringatan",JOptionPane.YES_NO_OPTION);
      if (jwb==JOptionPane.YES_OPTION) {
         this.setDefaultCloseOperation(EXIT_ON_CLOSE);
         System.exit(0);       }
```

- Tahap selanjutnya, untuk memudahkan pemanggilan setiap form dapat dilakukan pengontrolan melalui form menu utama. Langkah pembuatannya adalah dengan mengatur penempatan proses input, pencarian dan report seperti gambar berikut :

Gambar 7.50 Perancangan menu utama

- Import library jasper melalui sintaks berikut :

```
import javax.swing.JOptionPane;
import net.sf.jasperreports.view.*;
import net.sf.jasperreports.engine.*;
import java.sql.DriverManager;
import java.sql.Connection;
import java.util.*;
```

- Pada menu File => Input Data Tiket **event ActionPerformed** ketik sintak berikut :

```
Form_tiket tampil = new Form_tiket();
tampil.setVisible(true);
```

- Pada menu File =>Input Data Penerbangan **event ActionPerformed** ketik sintak berikut :

```
Form_penerbangan  tampil = new Form_penerbangan();
tampil.setVisible(true);
```

- Pada menu File => Input Data Penumpang **event ActionPerformed** ketik sintak berikut :

```
Form_penumpang  tampil = new Form_penumpang();
tampil.setVisible(true);
```

- Pada menu Cari Data => Penumpang / Tanggal **event ActionPerformed** ketik sintak berikut :

```
Form_cari  tampil = new Form_cari();
tampil.setVisible(true);
```

- Pada menu Cetak => Tiket Penumpang **event ActionPerformed** ketik sintak berikut :

```
Form_cetak  tampil = new Form_cetak();
tampil.setVisible(true);
```

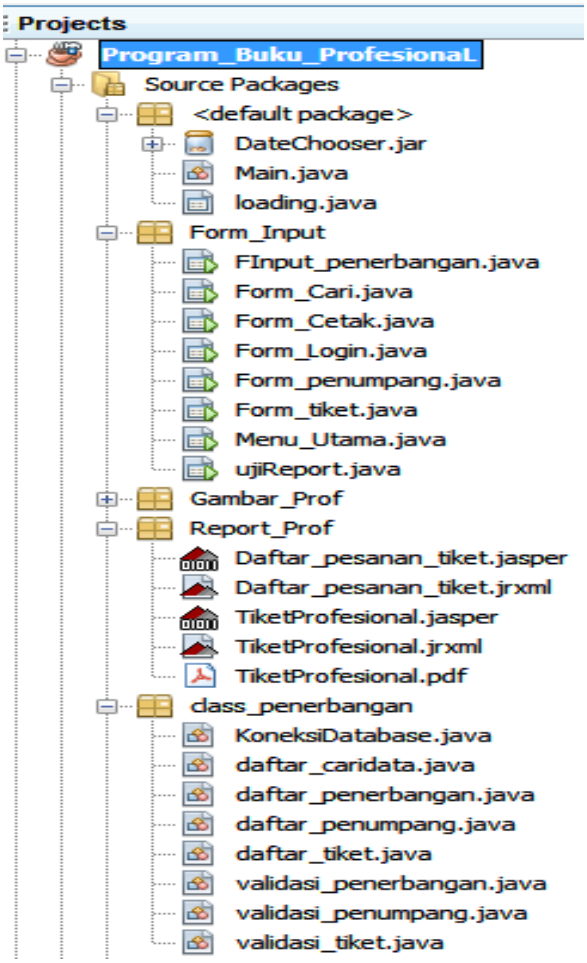
- Pada menu Cetak => Laporan **event ActionPerformed** ketik sintak berikut :

```
try  {      String path="C:\\Users\\Satellite
L510\\Documents\\NetBeansProjects\\ujiReport\\src\\Report_Prof\\Dafta
r_pesanan_tiket.jasper";
          System.out.println(path);
         HashMap parameter = new HashMap();
         String userName = "root";
         String password = "";
         String url = "jdbc:mysql://localhost/tiket_profesional";
         Class.forName ("org.gjt.mm.mysql.Driver");
Connection conx = DriverManager.getConnection(url,userName,password);
         JasperPrint jasperPrint = JasperFillManager.fillReport(path,
parameter, conx);
         JasperViewer.viewReport(jasperPrint, false);
      }       catch (JRException e)
      {          JOptionPane.showMessageDialog(this, "Gagal tampilkan
report1!\n" + e);
      }       catch (Exception e)      {
         JOptionPane.showMessageDialog(this, "Gagal tampilkan report!\n" +
e);       }
```

- Langkah terakhir pada class Main ketik sintaks berikut :

```
public static void main(String[] args) {
      loading tampil = new loading();
     tampil.setVisible(true);    }
```

Hirarki Project Buku_ProfesionaL



Gambar 7.51 Hirarki Project Buku_netbeans_profesionaL

DAFTAR PUSTAKA

Elenia, E. E., S, I. G. N. B. A., Oktavia, Ma., S, M. R. T., Aldisa, N., Widjayanti, P., & Ependi, V. (2020). Modul Praktikum Modul Praktikum. *Akuntansi Keuangan Lanjut 2*, 10.

FT, Mhaisen, & et, al. (2018). 済無No Title No Title No Title. *Angewandte Chemie International Edition, 6(11), 951–952.*, *13*, 10–27.

Litbang, T. (2016). *Webmaster Series JavaScript* (Ignas (ed.); 1st ed.). WAHANA KOMPUTER DAN ANDI.

DESKRIPSI – BIODATA SAMSURIAH

A. Identitas Diri

1.	Nama Lengkap	Samsuriah, S.Kom., M.T.
2.	Jenis Kelamin	Perempuan
3.	Jabatan Fungsional	Lektor
4.	NIDN	0905068005
5.	Tempat dan Tanggal Lahir	Pajalesang, 05 Juni 1980
6.	E-mail	samsuriahagus@gmail.com
7.	Nomor Telepon/HP	08114118115
8.	Alamat Kantor	Jl. AP.Pettarani No.27 Makassar
9.	Nomor Telepon/fax	
10.	Lulusan yang Telah DIhasilkan	Lebih 500 Mahasiswa dari Program Studi Manajemen Informatika dan Sistem Informasi
11.	Mata Kuliah yang Diampu	Teknik Riset Operasi, Sistem Operasi Linux Ubuntu, Pemrograman Berorientasi Objek (Java)

B. Pengalaman Penelitian dan Pengabdian kepada Masyarakat dalam 5 Tahun Terakhir

No.	Tahun	Judul Penelitian	Pendanaan	
			Sumber	Jml (Juta Rp)
1	2019	Analisis Wireless Access Point Pada Fitur Konfigurasi Unifi Ubiquiti Networks	-	-
2	2020	Penentuan Titik Koordinat pada GPS untuk Data Mahasiswa	STMIK Profesional Makassar	1.500.000,-
3	2021	Implementasi Articulate Storyline Dalam Pembuatan Bahan Ajar Digital Pada Mata Kuliah Teknik Riset Operasi	-	-
4	2022	Sistem Pendukung Keputusan Dalam Menentukan Tingkat Kepuasan Pelanggan Terhadap Pelayanan Pada Cafe Algo Makassar	-	-
5	2023	Penerapan Kriptografi Caesar Cipher Pada Fitur Pesan Teks	-	-

C. Publikasi Ilmiah pada Jurnal dalam 5 Tahun Terakhir

No.	Judul Artikel Ilmiah	Nama Jurnal	Volume/Nomor/Tahun
1	Analisis Wireless Access Point Pada Fitur Konfigurasi	Jurnal Informatika	11/1/2019

	Unifi Ubiquiti Networks	Progres	
2	Penentuan Titik Koordinat pada GPS untuk Data Mahasiswa	Jurnal Informatika Progres	12/2/2020
3	Implementasi Articulate Storyline Dalam Pembuatan Bahan Ajar Digital Pada Mata Kuliah Teknik Riset Operasi	Jurnal Informatika Progres	13/1/2021
4	Sistem Pendukung Keputusan Dalam Menentukan Tingkat Kepuasan Pelanggan Terhadap Pelayanan Pada Cafe Algo Makassar	Jurnal Informatika Progres	14/2/2022
5	Penerapan Kriptografi Caesar Cipher Pada Fitur Pesan Teks	Nusantara Hasana Journal	2/9/2023

D. Karya Buku dalam 5 Tahun Terakhir

No.	Judul Buku	Tahun	Jumlah Halaman	Penerbit
-	-	-	-	-

Semua data yang tertulis dan tercantum dalam biodata ini adalah benar.

Buku Ajar

PEMROGRAMAN BERORIENTASI OBJEK

Penulis : Ida
Mhd. Faisal
Samsuriah
Musdalifa
Darniati
Rosnani
Nasir
Dikwan Moeis

Java merupakan salah satu bahasa pemrograman yang banyak diminati antara lain karena java unggul ketika digunakan dalam pembuatan aplikasi berbasis mobile, juga aplikasi yang berskala enterprise. Dengan demikian, ketika Anda menguasai bahasa Java maka dapat membuat berbagai macam aplikasi yang bejalan pada multiplatform, baik pada skala kecil maupun yang berskala besar

PUBLISHER BY
PT. Inovasi Pratama Internasional