DASAR PEMROGRAMAN

# JAVA

Eko Hariyanto
Indri Sulistianingsih

# DASAR PEMROGRAMAN JAVA

**Oleh :**

**Eko Hariyanto**
**Indri Sulistianingsih**

**Fakultas Sains dan Teknologi**
**Universitas Pembangunan Panca Budi**
**2019**

**Dasar Pemrograman Java**
**Oleh : Eko Hariyanto, Indri Sulistianingsih**

Hak Cipta © 2019 pada penulis
Editor : Rusiadi
Desain Cover : Khairil Fahmi Pulungan
Penata Letak : LPPM UNPAB
Halaman : 203 Halaman
ISBN : 978-602-51956-9-3

Terbitan Pertama 2019

Hak cipta dilindungi undang-undang.
Dilarang memperbanyak atau memindahkan sebagian atau seluruh isi buku ini dalam bentuk apapun, baik secara elektronis maupun mekanis, termasuk memfotocopy, merekam atau dengan sistem penyimpanan lainnya, tanpa izin tertulis dari penulis.

Penerbit : Fakultas Ekonomi Universitas Panca Budi
Jalan Jenderal Gatot Subroto Km. 4,5, Telp. (061) 845-5571, Fax. (061) 845-8077
Medan 20122

# Kata Pengantar

Alhamdulillaahirobbil'aalamiin….
Segala puji hanya kepada Allah, Sesembahan seluruh alam, atas segala nikmat, rahmat dan karunia-Nya sehingga Penulis dapat menyelesaikan buku yang berjudul "Dasar Pemrograman Java".

Melalui buku ini, penulis membahas dasar Bahasa Pemrograman Java menggunakan IDE NetBeans untuk tingkat pemula dalam bentuk modul-modul. Beberapa modul memiliki keterkaitan dalam pembahasannya. Dalam buku ini juga dilengkapi dengan latihan-latihan yang disusun langkah demi langkah sehingga para pembaca pemula yang ingin belajar Pemrograman Java dapat mengikuti dengan baik serta dilengkapi pula dengan *link* untuk mengunduh beberapa sumber (resources) yang dibutuhkan dalam latihan.

Terima kasih yang sebesar-besarnya penulis ucapkan kepada keluarga penulis atas atensi dan doanya, serta segenap *civitas academica* Universitas Pembangunan Panca Budi terkhusus kepada Program Studi Sistem Komputer dan Teknik Komputer. Penghargaan yang setinggi-tinggi penulis berikan kepada Kementerian Riset Teknologi dan Pendidikan Tinggi atas kepercayaan yang diberikan melalui program Penelitian Dosen Pemula.

Dengan materi dan sajian yang masih terbatas, penulis berharap buku ini dapat bermanfaat bagi pembaca. Saran dan kritik sangat membantu dalam penyusunan buku ini yang lebih baik lagi.

Medan, Mei 2019

Penulis

# Daftar Isi

Kata Pengantar ..... iii
Daftar Isi ..... iv
Daftar Gambar ..... vii
Daftar Tabel ..... viii
MODUL 1 PENGENALAN PEMROGRAMAN JAVA ..... 1
1.1 Bahasa Pemrograman Java ..... 2
1.1.1 Teknologi Java ..... 2
1.1.2 Fitur dari Java ..... 3
1.1.3 Fase – Fase Pemrograman Java ..... 4
1.1.4 Lingkungan Kerja IDE Netbeans ..... 5
1.2 Percobaan ..... 9
1.2.1 Instalasi Java + Netbeans ..... 9
1.2.2 Menambah Projek Baru (New Project) ..... 16
1.2.3 Menambah Paket Baru (*New Package*) ..... 18
1.2.4 Menambah Frame Form ..... 20
1.2.5 Merancang Aplikasi ..... 21
MODUL 2 DASAR PEMROGRAMAN JAVA ..... 25
2.1 Tipe Data ..... 26
2.1.1 Tipe Data Primitif/Sederhana ..... 26
2.1.2 Tipe Data Referensi ..... 27
2.1.3 Enumerasi ..... 27
2.2 Identifier ..... 28
2.3 Operator ..... 29
2.3.1 Operator Aritmatika ..... 29
2.3.2 Operator *Increment/Decrement* ..... 29
2.3.3 Operator Penugasan (*Assignment*) ..... 29
2.3.4 Operator Pembanding ..... 30

2.3.5 Operator Logika ..... 31
2.3.6 Operator Bitwise ..... 31
2.3.7 Operator Ternary ..... 35
2.4 Latihan ..... 36
2.4.1 Aplikasi Kalkulator1 ..... 36
MODUL 3 STRUKTUR PENSELEKSIAN ..... 41
3.1 Pernyataan *if* ..... 42
3.2 Pernyataan *if-else* ..... 42
3.3 Pernyataan *if-else-if* ..... 43
3.4 Pernyataan *switch* ..... 46
3.5 Penseleksian Bersarang ..... 47
3.6 Latihan ..... 49
3.6.1 Aplikasi Warna ..... 49
3.6.2 Aplikasi Kasir ..... 55
MODUL 4 STRUKTUR PERULANGAN ..... 59
4.1 Struktur *for* ..... 60
4.2 Struktur *while* ..... 61
4.3 Struktur do…while ..... 62
4.4 Perulangan Bersarang ..... 64
4.5 Latihan ..... 65
MODUL 5 ARRAY ..... 69
5.1 Array Satu Dimensi ..... 70
5.2 Array Multi Dimensi ..... 72
5.3 Perulangan Array ..... 73
5.4 Class Array Pada Java ..... 76
5.4.1 Class Arrays ..... 77
5.4.2 Class ArrayList ..... 80
5.5 Latihan ..... 84
MODUL 6 STRUKTUR METODE (*METHOD*) ..... 94

6.1 Metode Tanpa Nilai ..... 95
6.2 Metode Dengan Nilai ..... 95
6.3 Modifier ..... 99
*keyword* ..... 99
*keterangan* ..... 99
*keyword* ..... 100
*keterangan* ..... 100
6.4 Latihan ..... 101
6.4.1 Aplikasi Warna ..... 101
6.4.2 Aplikasi Payroll ..... 105
MODUL 7 PENGOLAH DATA ..... 113
7.1 Data Karakter ..... 114
7.2 Data String ..... 116
7.3 Data Tanggal dan Waktu ..... 121
7.4 Latihan ..... 126
7.4.1 Aplikasi Tarif Parkir ..... 126
7.4.2 Aplikasi Belajar Kriptografi ..... 133
MODUL 8 PROJEK APLIKASI ..... 144
8.1 Aplikasi MDI ..... 145
8.2 Aplikasi Input-Data1 ..... 154
8.3 Aplikasi Input-Data2 ..... 161
8.4 Aplikasi CRUD ..... 168
8.5 Aplikasi Hangman ..... 175
8.6 Aplikasi Alarm ..... 183
8.7 Aplikasi Download Manager ..... 191
DAFTAR PUSTAKA

# Daftar Gambar

Gambar 1.1 Fase dari sebuah Program Java ..... 5
Gambar 1.2 Lingkungan Kerja IDE NetBeans ..... 6
Gambar 1.3 Cara Menampilkan Jendela Projects ..... 6
Gambar 1.4 Cara Menampilkan Jendela Navigator ..... 7
Gambar 1.5 Tampilan Perancangan Antar Muka ..... 7
Gambar 1.6 Tampilan Perancangan Kode Program ..... 8
Gambar 1.7 Cara Menampilkan Jendela Palette ..... 8
Gambar 1.8 Cara Menampilkan Jendela Properties ..... 9

# Daftar Tabel

Tabel 1.1 Fase Sebuah Pemrograman Java ..... 5
Tabel 2.1 Tipe Data Primitif..... 26
Tabel 2.2 Operator Pembanding ..... 30
Tabel 2.3 Operator Logika ..... 31
Tabel 2.4 Operator Bitwise..... 32
Tabel 6.1 Non-Access Modifier ..... 99
Tabel 6.2 Access Modifier..... 100
Tabel 7.1 Escape Sequences ..... 114
Tabel 7.2 Pola Huruf Untuk Format Tanggal dan Waktu ..... 121

# MODUL 1
## PENGENALAN PEMROGRAMAN JAVA

**Pokok Pembahasan :**

- **Teknologi pemrograman Java**
- **Fitur pemrograman Java**
- **Lingkungan Kerja IDE Netbeans**
- **Latihan/Percobaan**

## 1.1 Bahasa Pemrograman Java

Java dikembangkan oleh sebuah tim yang dipimpin oleh James Gosling di perusahaan Sun Microsystems. Sebelumnya bernama "Oak" yang dirancang pada tahun 1991 untuk digunakan dalam chip tertanam di peralatan elektronik. Pada tahun 1995, berganti nama menjadi Java dan didesain ulang untuk mengembangkan aplikasi web. Pada tahun 2010, Sun Microsystems diambil-alih oleh perusahaan Oracle. Java merupakan bahasa pemrograman yang sangat populer. Pengembangan pemrograman Java yang cepat dan kompatibel dapat dilihat dari karakteristik desainnya, dimana Anda dapat menulis program sekali dan dapat dijalankan/dieksekusi dimana saja.

### 1.1.1 Teknologi Java

- **Sebuah Bahasa Pemrograman**

  Sebagai sebuah bahasa pemrograman, Java dapat membuat seluruh bentuk aplikasi, *desktop, web* dan lainnya, sebagaimana dibuat dengan menggunakan bahasa pemrograman konvensional yang lain. Java adalah bahasa pemrograman yang berorientasi objek (dibahas secara khusus pada Buku Pemrograman Berorientasi Objek) dan dapat dijalankan pada berbagai *platform* sistem operasi. Perkembangan Java tidak hanya terfokus pada satu sistem operasi, tetapi dikembangkan untuk berbagai sIstem operasi dan bersifat *open source*.

- **Sebuah *Development Environment***

  Sebagai sebuah peralatan pembangun, teknologi Java menyediakan banyak *tools* : *compiler, interpreter,* penyusun dokumentasi, paket kelas dan sebagainya.

- **Sebuah Aplikasi**

  Aplikasi dengan teknologi Java secara umum adalah aplikasi serba guna yang dapat dijalankan pada seluruh mesin yang memiliki *Java Runtime Environment* (JRE).

- **Sebuah *Deployment Environment***

  Terdapat dua komponen utama dari *Deployment Environment*. Yang pertama adalah JRE, yang terdapat pada paket J2SDK, mengandung kelas–kelas untuk semua paket teknologi Java yang meliputi kelas dasar dari Java, komponen GUI dan sebagainya. Komponen yang lain terdapat pada Web Browser. Hampir

seluruh Web Browser komersial menyediakan *interpreter* dan *runtime environment* dari teknologi Java.

### 1.1.2 Fitur dari Java

- **Java Virtual Machine (JVM)**

  JVM adalah sebuah mesin imajiner (maya) yang bekerja dengan menyerupai aplikasi pada sebuah mesin nyata. JVM menyediakan spesifikasi hardware dan platform dimana kompilasi kode Java terjadi. Spesifikasi inilah yang membuat aplikasi berbasis Java menjadi bebas dari *platform* manapun karena proses kompilasi diselesaikan oleh JVM. Aplikasi program Java diciptakan dengan *file* teks berekstensi *.java*. Program ini dikompilasi menghasilkan satu berkas *bytecode* berekstensi *.class* atau lebih. *Bytecode* adalah serangkaian instruksi serupa instruksi kode mesin. Perbedaannya adalah kode mesin harus dijalankan pada sistem komputer dimana kompilasi ditujukan, sementara *bytecode* berjalan pada *java interpreter* yang tersedia di semua *platform* sistem komputer dan sistem operasi.

- **Garbage Collection**

  Banyak bahasa pemrogaman lain yang mengijinkan seorang programmer mengalokasikan memori pada saat dijalankan. Namun, setelah menggunakan alokasi memori tersebut, harus terdapat cara untuk menempatkan kembali blok memori tersebut supaya program lain dapat menggunakannya. Dalam C, C++ dan bahasa lainnya, adalah programmer yang mutlak bertanggung jawab akan hal ini. Hal ini dapat menyulitkan bilamana programmer tersebut alpa untuk mengembalikan blok memori sehingga menyebabkan situasi yang dikenal dengan nama *memory leaks.* Program Java melakukan *garbage collection* yang berarti program tidak perlu menghapus sendiri objek–objek yang tidak digunakan lagi. Fasilitas ini mengurangi beban pengelolaan memori oleh programmer dan mengurangi atau mengeliminasi sumber kesalahan terbesar yang terdapat pada bahasa yang memungkinkan alokasi dinamis.

- ***Code Security***

  *Code Security* terimplementasi pada Java melalui penggunaan Java Runtime Environment (JRE). Java menggunakan model pengamanan 3 lapis untuk melindungi sistem dari *untrusted Java Code*.

  a. *Class-loader*

Menangani pemuatan kelas Java ke *runtime interpreter.* Proses ini menyediakan pengamanan dengan memisahkan kelas–kelas yang berasal dari *local disk* dengan kelas–kelas yang diambil dari jaringan. Hal ini membatasi aplikasi Trojan karena kelas–kelas yang berasal dari *local disk* yang dimuat terlebih dahulu.

b. *Bytecode verifier*
   Membaca *bytecode* sebelum dijalankan dan menjamin *bytecode* memenuhi aturan–aturan dasar bahasa Java.

c. Manajemen keamanan
   Menangani keamanan tingkat aplikasi dengan mengendalikan apakah program berhak mengakses sumber daya seperti sistem file, *port* jaringan, proses eksternal dan sistem *windowing*. Setelah seluruh proses tersebut selesai dijalankan, barulah kode program di eksekusi.

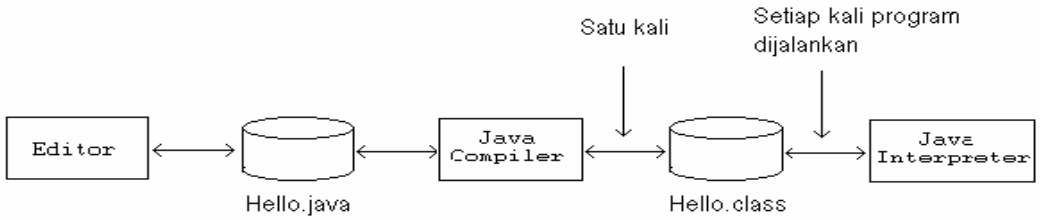
Java juga menyediakan beragam teknik pengamanan lain :

a. Bahasa dirancang untuk mempersulit eksekusi kode perusak.
   Peniadaan *pointer* merupakan langkah besar pengamanan. Java tidak mengenal operasi *pointer*. Di tangan programmer handal, operasi pointer merupakan hal yang luar biasa untuk optimasi dan pembuatan program yang efisien serta mengagumkan. Namun mode ini dapat menjadi petaka di hadapan programmer jahat. Pointer merupakan sarana luar biasa untuk pengaksesan tak diotorisasi. Dengan peniadaan operasi *pointer*, Java dapat menjadi bahasa yang lebih aman.

b. Java memiliki beberapa pengaman terhadap *applet*.
   Untuk mencegah program bertindak mengganggu media penyimpanan, maka *applet* tidak diperbolehkan melakukan *open, read* ataupun *write* terhadap berkas secara sembarangan. Karena Java *applet* dapat membuka jendela *browser* yang baru, maka jendela mempunyai logo Java dan teks identifikasi terhadap jendela yang dibuka. Hal ini mencegah jendela *pop-up* menipu sebagai permintaan keterangan *username* dan *password*.

### 1.1.3 Fase – Fase Pemrograman Java

Gambar dibawah ini menjelaskan aliran proses kompilasi dan eksekusi sebuah program Java :




**Gambar 1.1** Fase dari sebuah Program Java

Langkah pertama dalam pembuatan sebuah program berbasis Java adalah menuliskan kode program pada *text editor*. Contoh *text editor* yang dapat digunakan antara lain : notepad, vi, emacs, sub limes dan lain sebagainya. Kode program yang dibuat kemudian tersimpan dalam sebuah berkas berekstensi *.java.* Setelah membuat dan menyimpan kode program, kompilasi file yang berisi kode program tersebut dengan menggunakan Java Compiler. Hasil dari kompilasi berupa berkas *bytecode* dengan ekstensi *.class.* Berkas yang mengandung *bytecode* tersebut kemudian akan dikonversikan oleh Java Interpreter menjadi bahasa mesin sesuai dengan jenis dan *platform* yang digunakan. Tabel berikut menggambarkan fase dari sebuah program java

**Tabel 1.1** Fase Sebuah Pemrograman Java

| PROSES | TOOL | HASIL |
|---|---|---|
| Menulis kode program | *Text editor* | Berkas berekstensi *.java* |
| Kompilasi program Java | Compiler | Berkas berekstensi *.class* (Java Bytecodes) |
| Menjalankan program Java | Interpreter | Program Output |

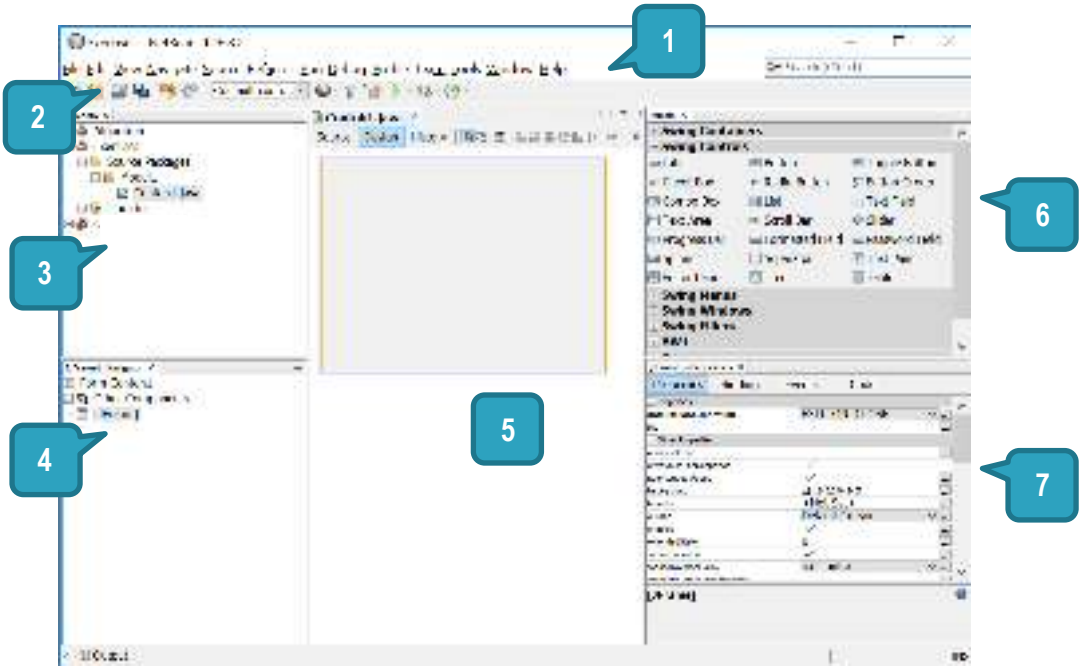
### 1.1.4 Lingkungan Kerja IDE Netbeans

Lingkungan kerja dari IDE NetBeans secara umum dapat dilihat pada gambar 1.2. Adapun keterangan mengenai lingkungan kerja IDE NetBeans adalah sebagai berikut :

1. Menu bar
   Merupakan kumpulan menu yang menyediakan fitur atau fungsi, dan fasilitas tertentu.

2. Toolbar

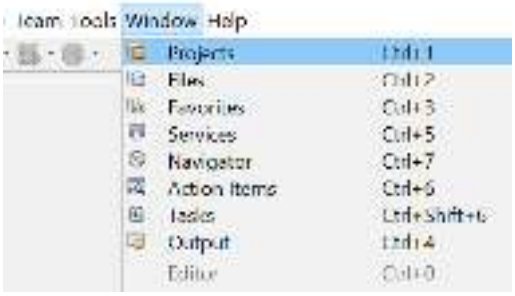
   Merupakan kumpulan tombol pintas untuk menjalankan fungsi atau fitur tertentu berupa gambar icon.




**Gambar 1.2** Lingkungan Kerja IDE NetBeans

3. Jendela Projects

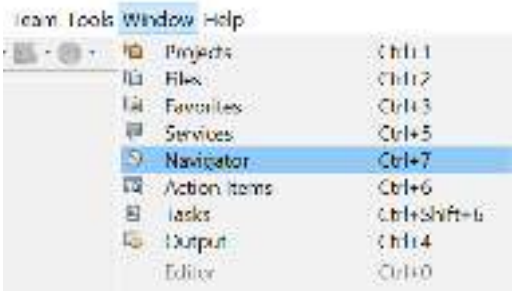
   Merupakan area yang menampilkan seluruh daftar projek yang berada dalam IDE NetBeans. Untuk menampilkan jendela 'Projects' (Gambar 1.3), klik menu 'Window' → 'Projects'.




**Gambar 1.3** Cara Menampilkan Jendela Projects

4. Jendela Navigator

   Merupakan area navigasi baik susunan kode program maupun susunan komponen pada lembar perancangan. Untuk menampilkan jendela 'Navigator' (Gambar 1.4), klik menu 'Window' → 'Navigator'.



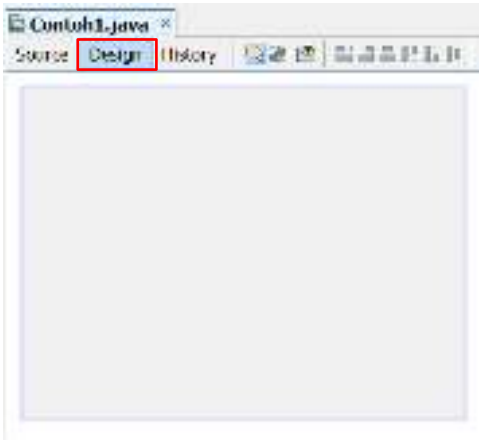

**Gambar 1.4** Cara Menampilkan Jendela Navigator

5. Lembar Perancangan

   Merupakan tempat perancangan antar muka (Gambar 1.5), perancangan kode program (Gambar 1.6), dan riwayat pekerjaan.

6. Jendela Palette

   Merupakan area yang menyediakan komponen untuk merancang antar muka berbasis *Graphic User Interface* (GUI) pada lembar perancangan. Untuk menampilkan jendela 'Palette' (Gambar 1.7), Klik menu 'Windows' → 'IDE Tools' → 'Palette'.

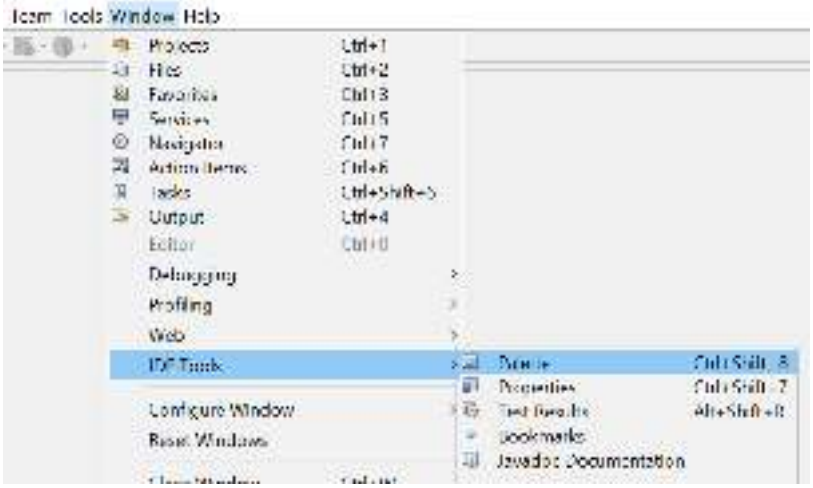



**Gambar 1.5** Tampilan Perancangan Antar Muka

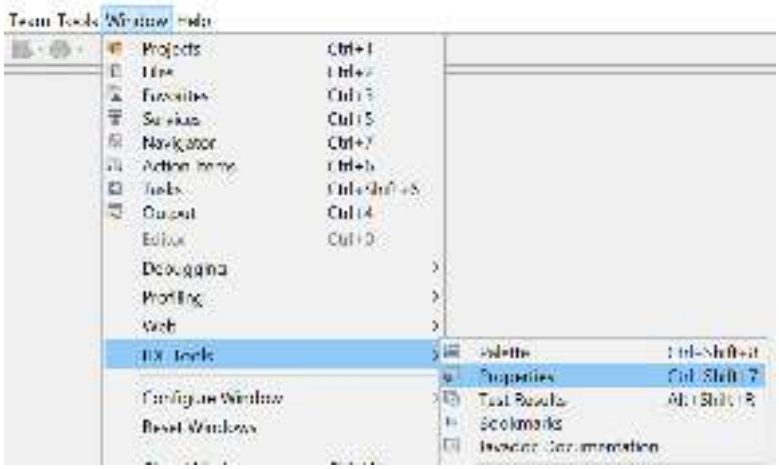
**Gambar 1.6** Tampilan Perancangan Kode Program

7. Jendela Properties

   Merupakan area yang berisi item-item properti dari suatu komponen yang digunakan pada lembar perancangan. Pada area ini pula dapat ditentukan nilai atau data suatu komponen agar pengolahan komponen atau nilai menjadi lebih mudah. Untuk menampilkan jendela 'Properties' (Gambar 1.8), Klik menu 'Windows' → 'IDE Tools' → 'Properties'.



**Gambar 1.7** Cara Menampilkan Jendela Palette




**Gambar 1.8** Cara Menampilkan Jendela Properties

## 1.2 Percobaan

### 1.2.1 Instalasi Java + Netbeans

Dalam percobaan berikut ini, akan dijelaskan langkah-langkah dalam melakukan instalasi Java dan Netbeans. File-file yang dibutuhkan yaitu Java Development Kit (JDK) dan Netbeand IDE yang dapat diunduh secara gratis di https://www.oracle.com/technetwork/java/index.html untuk JDK dan https://netbeans.org/downloads/index.html untuk Netbeans IDE.

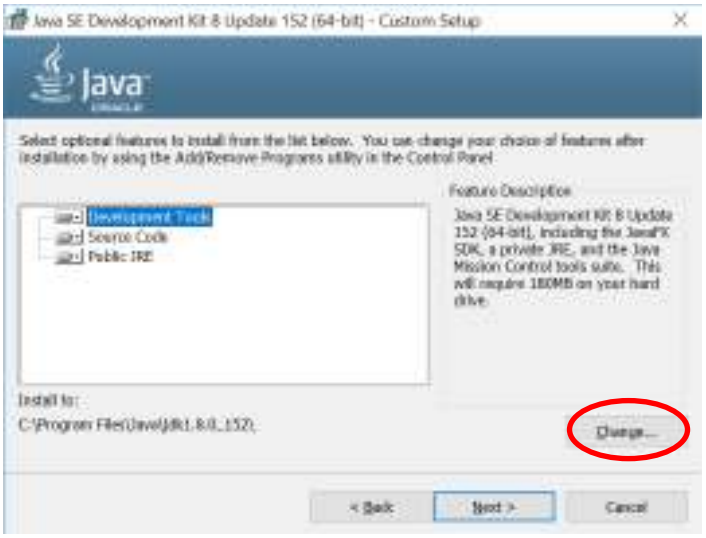
1. Menginstall software Java Development Kit (pada contoh di bawah menggunakan jdk-8) di windows komputer dengan langkah-langkahnya sebagai berikut :
   a. Setelah JDK berhasil diunduh, double click pada 'jdk installer'.

Java Development Kit Installer

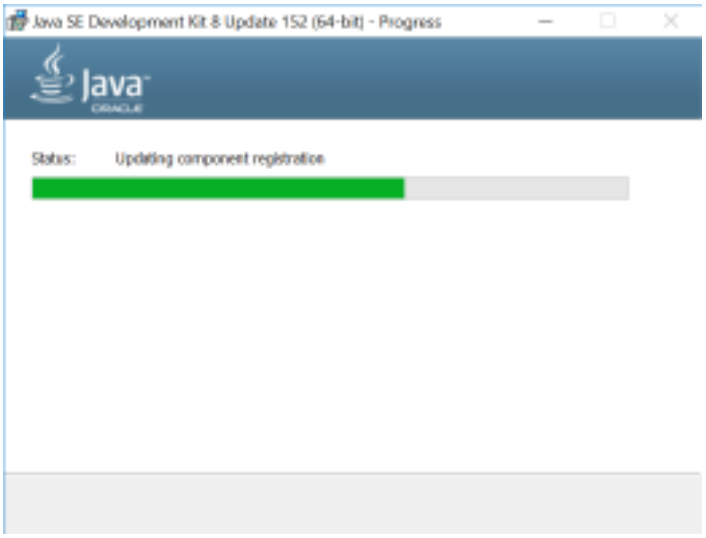
   b. Pada halaman 'Welcome', klik tombol 'Next >' untuk melanjutkan ke proses berikutnya.

Halaman Welcome

c. Pada halaman ‘Custom Setup’, lokasi instalasi dapat diubah dengan klik tombol ‘Change’. klik tombol ‘Next >’ untuk melanjutkan ke proses instalasi JDK. Pada halaman tersebut menampilkan informasi progres instalasi JDK.




Halaman Custom Setup




Halaman Proses Instalasi JDK

d. Setelah selesai proses instalasi JDK, proses berikutnya adalah instalasi Java Runtime Environment (JRE). Halaman instalasi JRE, lokasi instalasi dapat diubah dengan klik tombol 'Change…'. Klik tombol 'Next >' untuk melanjutkan ke proses instalasi. Pada halaman tersebut menampilkan informasi progres instalasi JRE. Setelah selesai proses instalasi maka akan tampil halaman hasil yang berisi informasi status instalasi.

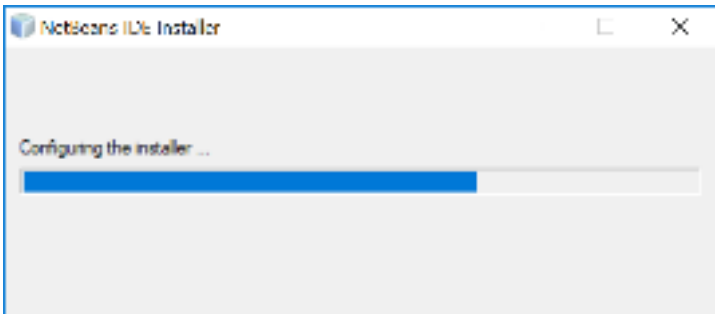
Halaman Instalasi JRE

Halaman Proses Instalasi JRE

Halaman Hasil

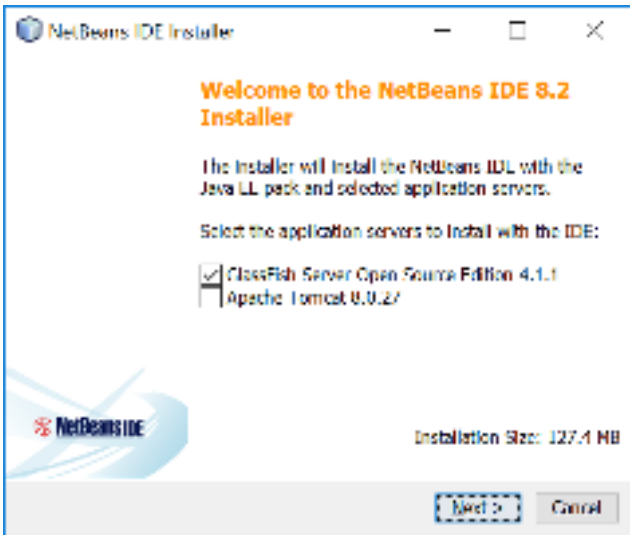
2. Menginstall text editor dan console NetBeans IDE dengan langkah-langkah adalah sebagai berikut :
   a. Setelah NetBeans berhasil diunduh, double click pada ‘netbeans’.

NetBeans Installer




Configuring the Installer

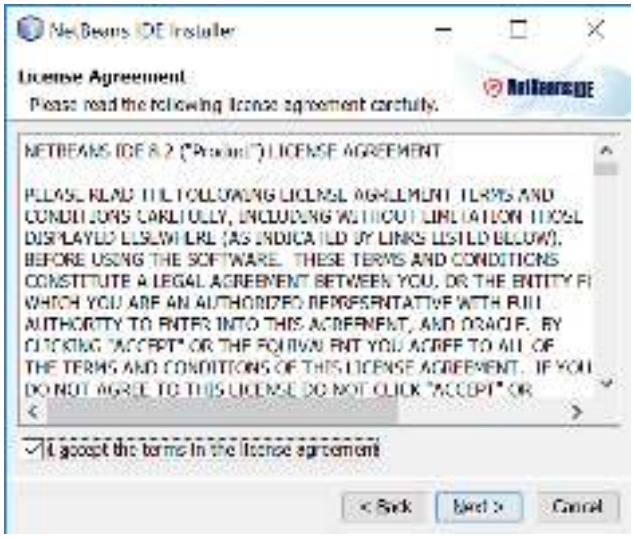
b. Halaman pertama yang tampil adalah halaman 'Welcome'. Pada halaman ini, terdapat pilihan untuk instalasi 'application servers'. Klik tombol 'Next >' untuk melanjutkan ke proses berikutnya.



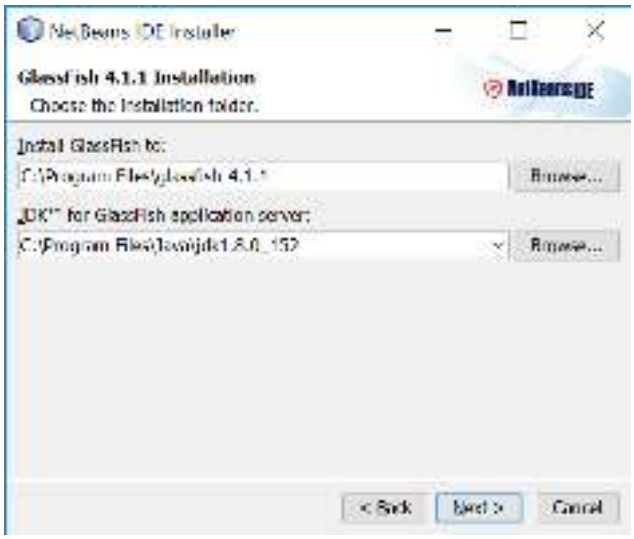

Halaman Welcome

c. Halaman selanjutnya adalah halaman 'License Agreement'. Centang pada bagian **"I accept the terms in the license agreement"**, kemudian klik tombol 'Next >' untuk melanjutkan ke proses berikutnya.

d. Apabila pada halaman 'Welcome' memilih *application server*, tahap berikutnya adalah menentukan lokasi instalasi dari *application server*. Untuk merubah lokasi instalasi yang telah ada, klik tombol 'Browse…'. Klik tombol 'Next >' untuk melanjutkan ke proses berikutnya.

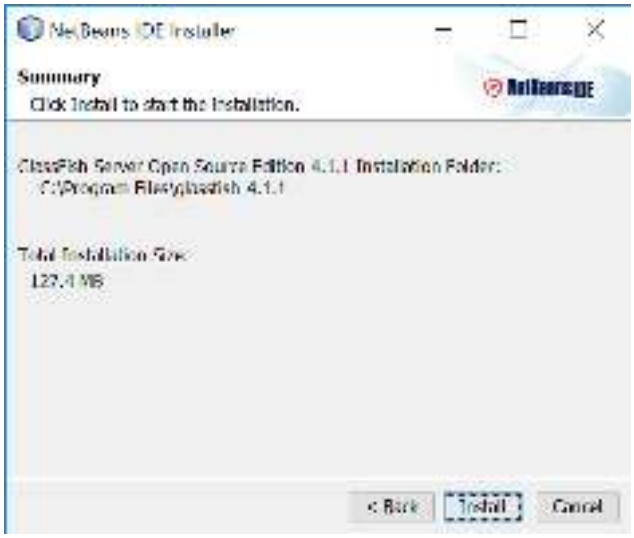



Halaman License Agreement




Halaman Application Server Installation Folder

e. Halaman berikutnya yang tampil adalah halaman 'Summary'. Klik tombol 'Install' untuk melanjutkan ke proses installasi.




Halaman Summary

Halaman Proses Instalasi NetBeans IDE

f. Setelah proses instalasi selesai, akan tampil halaman hasil yang berisi status hasil instalasi. Klik tombol ‘Finish’.

Halaman Hasil

### 1.2.2 Menambah Projek Baru (New Project)

Untuk membuat aplikasi menggunakan Bahasa Pemrograman Java, terlebih dahulu membuat projek baru (*new project*). Berikut langkah-langkah membuat projek baru :

1. Pada layar dekstop, double click NetBeans IDE shortcut untuk menjalankan software NetBeans IDE. Selain itu, dapat pula melalui menu windows :

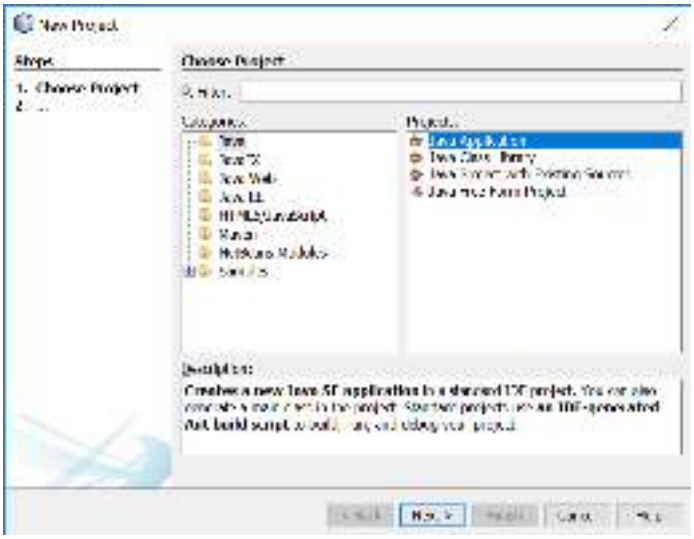
NetBeans IDE shortcut

Menu windows

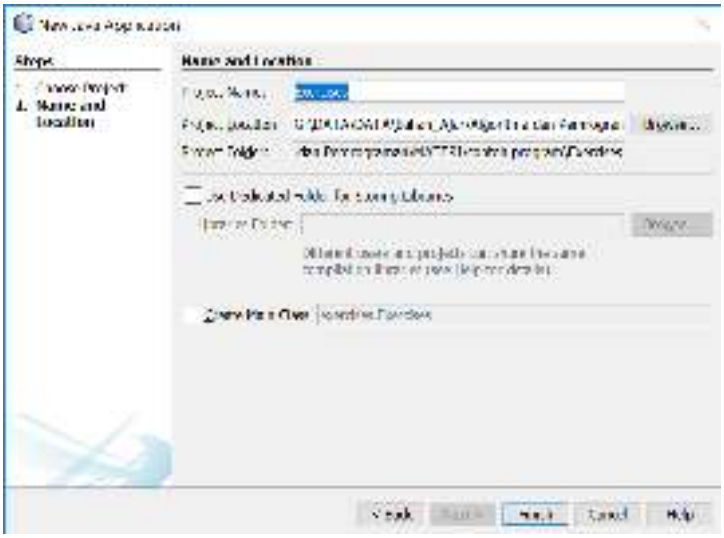
2. Pada tampilan utama software NetBeans, pilih menu ‘File’ → ‘New Project’.

Menu ‘New Project’

3. Pada halaman ‘New Project’**,** pilih ‘Java’ dibagian Categories dan pilih ‘Java Applications’ dibagian Projects kemudian klik tombol ‘Next >’.



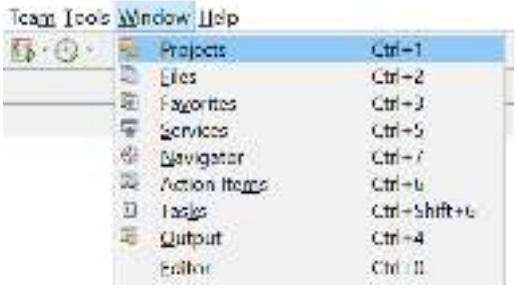

4. Pada Project Name Isikan nama projek akan dibuat contoh ‘Exercises’. Hilangkan centang pada ‘Create Main Class’, kemudian klik ‘Finish’.



***Catatan : nama projek tidak boleh menggunakan spasi***

5. Projek baru akan muncul pada jendela 'Projects' (umumnya disisi sebelah kiri).

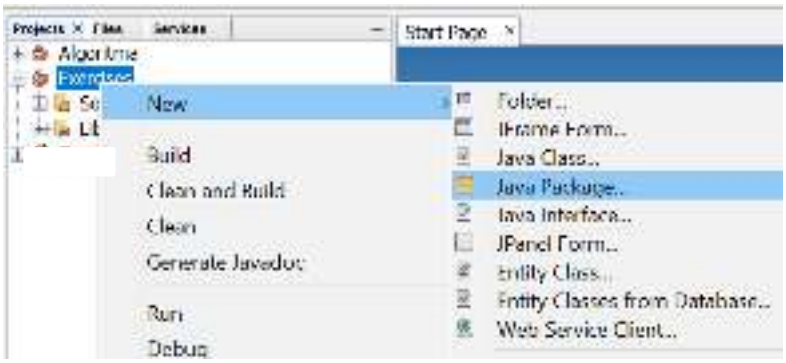
6. Apabila jendela 'Projects' tidak muncul/tampil pada IDE Netbeans, jendela 'Projects' dapat ditampilkan dengan cara : klik menu 'Window' → 'Projects'



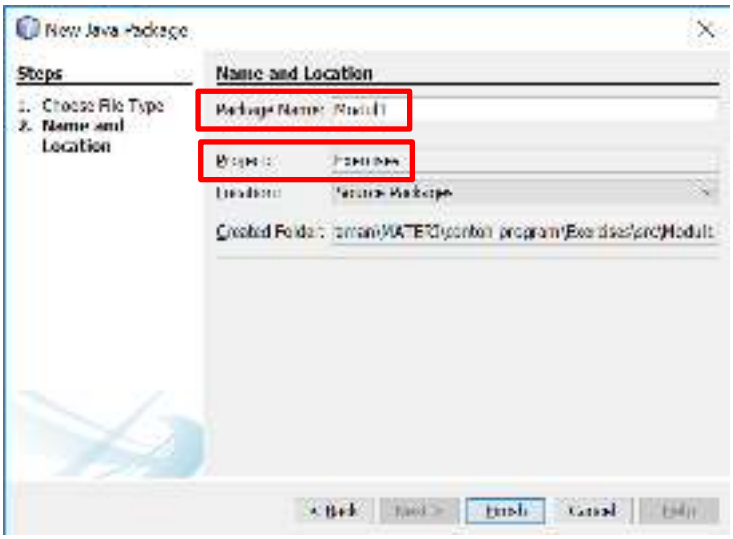

### 1.2.3 Menambah Paket Baru (*New Package*)

Paket (*Package*) digunakan untuk membuat folder yang dapat mempaketkan file-file yang sejenis. Berikut langkah-langkah untuk menambah paket baru :

1. Klik kanan pada projek 'Exercises' → New → 'Java Package'.

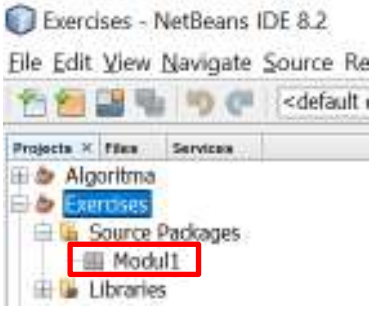



2. Tentukan Package Name, sebagai contoh 'Modul1'. Projek 'Exercises' yang tampil untuk memastikan bahwa paket terdapat dalam projek tersebut. Klik tombol 'Finish'.




***Catatan : nama paket tidak boleh menggunakan spasi.***

3. Paket 'Modul1' akan muncul didalam projek 'Exercises'.

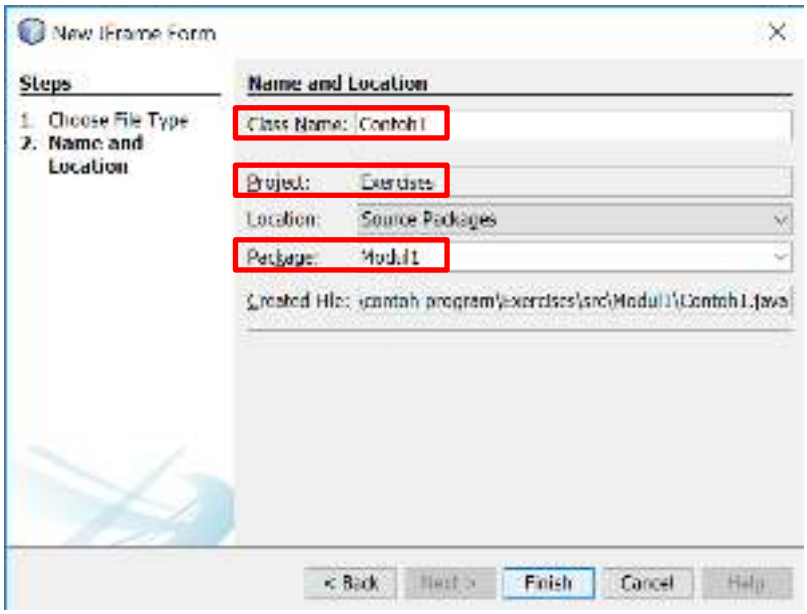


### 1.2.4 Menambah Frame Form

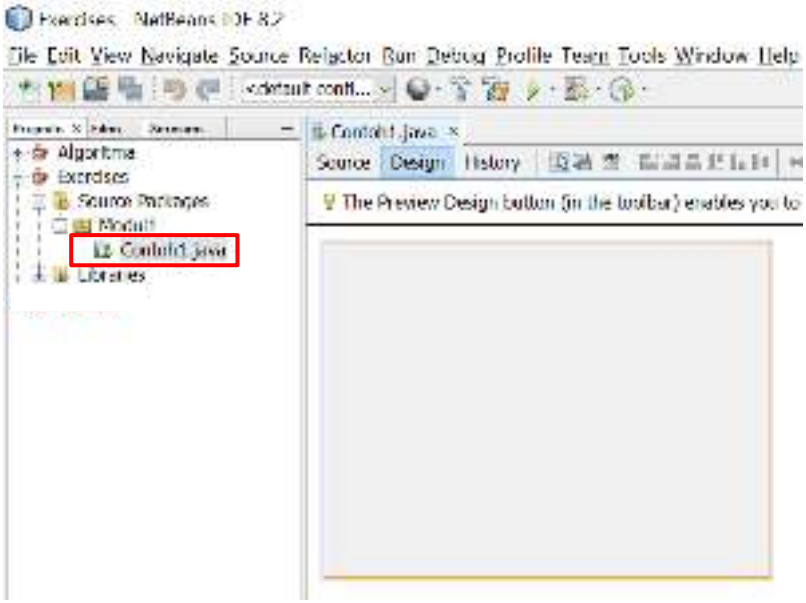
Menambah Frame Form dapat dilakukan pada paket (package) tertentu atau *default package*. Pada contoh di bawah ini, Frame Form akan ditambahkan pada paket "Modul1". Langkah-langkah untuk menambah Frame Form adalah sebagai berikut :

1. Klik kanan pada package "Modul1" → New → JFrame Form…

2. Tentukan Class Name, sebagai contoh 'Contoh1'. Project 'Exercises' dan package 'Modul1' yang tampil untuk memastikan bahwa Frame Form terdapat dalam projek dan paket tersebut. Klik tombol 'Finish'. ***Catatan : nama class tidak boleh menggunakan spasi.***

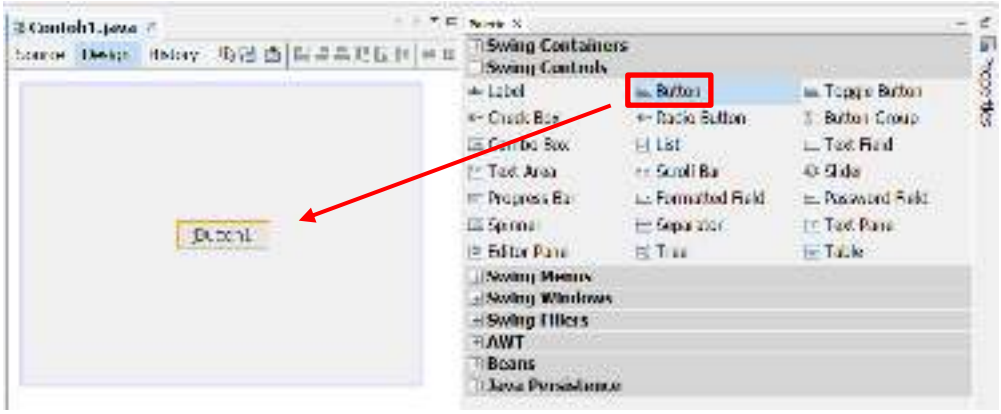


3. Form ‘Contoh1’ akan muncul didalam paket ‘Modul1’ dengan nama ‘Contoh1.java’.




### 1.2.5 Merancang Aplikasi

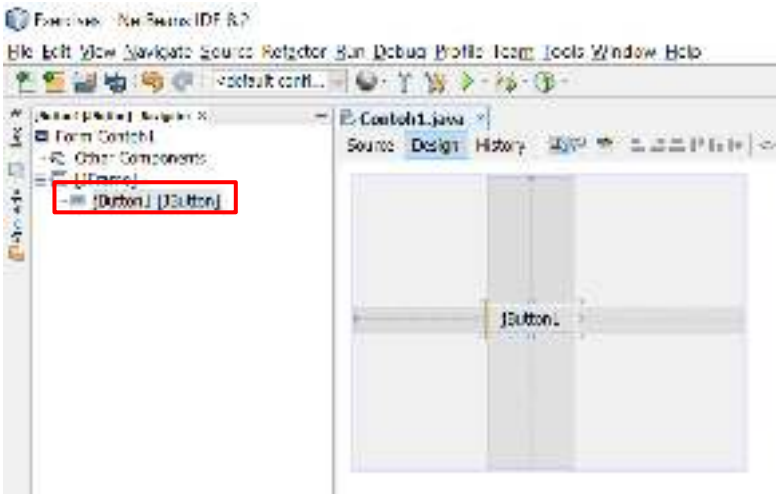
1. Masukkan komponen ‘Button’ kedalam form. Pada jendela ‘Palette’ pilih kategori ‘Swing Controls’, kemudian klik kiri ‘Button’. Arahkan pointer mouse kedalam form untuk menentukan posisi ‘Button’, kemudian klik kiri sekali lagi.




Catatan!

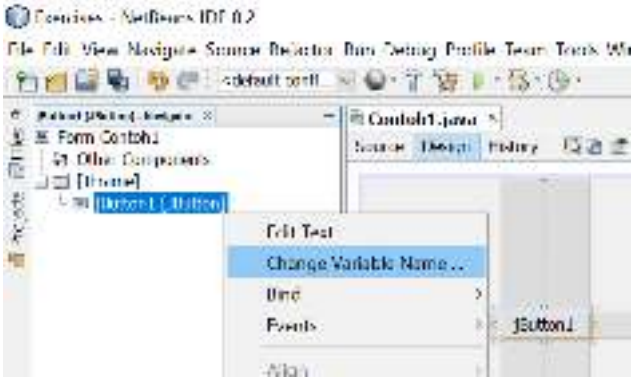
Komponen ‘Button’ (tombol) merupakan komponen eksekutor yang digunakan untuk menjalankan suatu proses tertentu pada aplikasi yang dibangun dengan cara klik kiri.

2. Pada jendela ‘Navigator’, muncul komponen ‘Button’ yang dimasukkan kedalam form dengan nama standar ‘jButton1’.

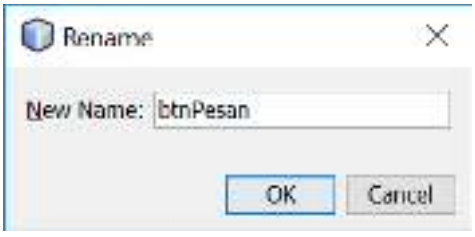


Setiap komponen didalam form memiliki nama (*variable name*) untuk membedakan komponen satu dengan komponen lainnya dan memudahkan programer untuk mengoperasikan komponen melalui kode program.

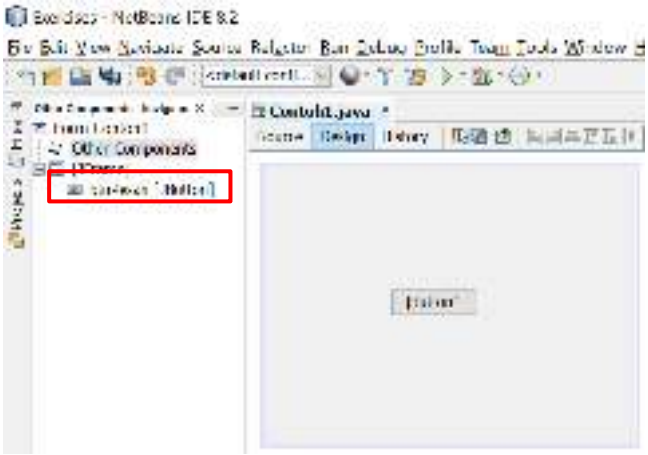
3. Untuk mengubah nama variabel dari komponen, klik kanan 'jButton1' pada jendela navigator kemudian klik 'Change Variable Name…'




Ubah menjadi 'btnPesan'. Klik tombol OK.



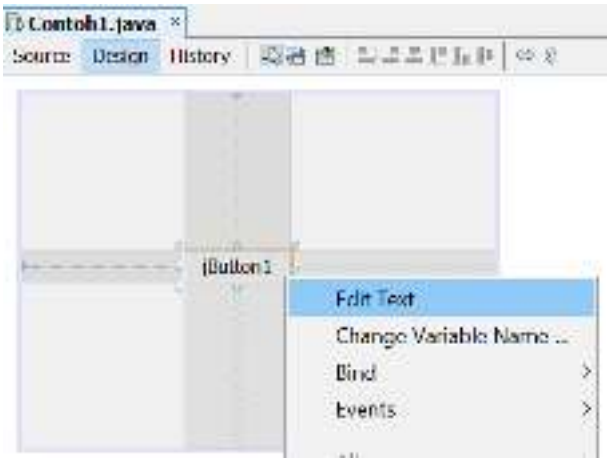

Nama variabel komponen akan berubah pada jendela navigator



Pemberian nama variabel pada komponen bermanfaat untuk memudahkan programer dalam :

- Mengenali dan membedakan setiap komponen yang digunakan.
- Menulis kode program didalam suatu komponen.
- Memanipulasi data yang dimasukkan dari komponen atau untuk ditampilkan ke komponen melalui kode program.

4. Untuk mengubah teks button, klik kanan ‘btnPesan’ pada jendela navigator atau klik kanan komponen pada form, kemudian klik ‘Edit Text’.




Ubah menjadi ‘Pesan’ kemudian tekan enter.

5. Double click button ‘Pesan’ pada form sehingga muncul tampilan kode program untuk menulis kode program pada button ‘btnPesan’

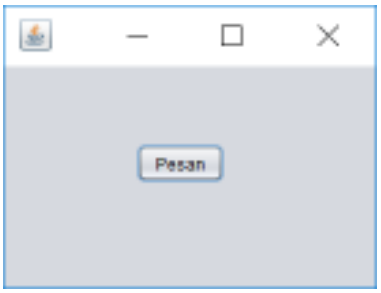
```
    private void btnPesanActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:

    }
```

6. Tuliskan kode program berikut:

```
private void btnPesanActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    javax.swing.JOptionPane.showMessageDialog(null,"Selamat Belajar Pemrograman Java");
}
```

7. Untuk menjalankan program, tekan tombol ‘shift + F6’ pada keyboard atau tombol run project pada toolbar.

8. Pada program yang berjalan (runtime) klik tombol ‘Pesan’, kemudian muncul kotak pesan yang berisi ‘Selamat Belajar Pemrograman Java’.



# MODUL 2
## DASAR PEMROGRAMAN JAVA

**Pokok Pembahasan :**

- **Tipe Data**
- **Identifier**
- **Operator**
- **Latihan**

## 2.1 Tipe Data

Tipe data merupakan klasifikasi data dengan bertujuan untuk mengenalkan data atau nilai yang digunakan kepada kompiler. Dalam pemrograman Java, terdapat dua kelompok tipe data yaitu tipe data primitif/sederhana dan tipe data referensi/komposit.

### 2.1.1 Tipe Data Primitif/Sederhana

Tipe data primitif/sederhana yaitu tipe data dasar yang tersedia/tertanam dan merupakan keyword dalam pemrograman Java. Tipe data primitif hanya dapat menampung satu nilai. Contoh penggunaan tipe data primitif adalah sebagai berikut :

```
char huruf = 'p';
short nilai = 100;
int angka = 20000;
```

→ ada 1 nilai yaitu "p"
→ ada 1 nilai yaitu 100
→ ada 1 nilai yaitu 20000

Adapun tipe data primitif dalam pemrograman Java dapat dilihat pada tabel 2.1 di bawah ini.

**Tabel 2.1** Tipe Data Primitif

| Tipe Data | Deskripsi | Ukuran (bit) | Nilai Minimum | Maksimum |
|---|---|---|---|---|
| boolean | true / false | 1-bit | | |
| char | Karakter Unicode | 16-bit | | |
| byte | Bilangan bulat | 8-bit | -127 | 128 |
| short | Bilangan bulat | 16-bit | -32768 | 32767 |
| int | Bilangan bulat | 32-bit | -2147483648 | 2147483647 |
| long | Bilangan bulat | 64-bit | -9223372036854775808 | 922337203685477580 7 |
| float | Bilangan riil | 32-bit | 1.4012984643248 1707e-45 | 3.4028234663852886 0e+38 |
| double | Bilangan riil | 64-bit | 4.9406564584124 6544e-324 | 1.7976931348623157 0e+308 |

### 2.1.2 Tipe Data Referensi

Tipe data referensi secara umum adalah tipe data buatan yang mereferensikan *Object* atau *Class* dan bukan merupakan *keyword* dalam pemrograman Java. Tipe data ini dapat menampung lebih dari satu nilai. Contoh penggunaan tipe data referensi adalah sebagai berikut :

```
String kampus = "Panca Budi";
short[] ganjil = {1,3,5,7,9};
int[] genap = {2,4,6,8,10};
```

→ 10 nilai (termasuk spasi) bertipe char
→ 5 nilai bertipe short
→ 5 nilai bertipe int

### 2.1.3 Enumerasi

Enumerasi (*enum*) merupakan tipe data buatan dimana nilai enum dapat didefenisikan sendiri sesuai kebutuhan. Nama enum harus dituliskan pada saat variabel dideklarasi dan variabel hanya dapat menggunakan nilai yang sudah didefenisikan dalam enum tersebut sebagai konstanta. Berikut sintaks atau bentuk penulisan enum.

```
modifier enum namaEnum {
        nilai1, nilai2, ..., nilai-n
}
```

Contoh :

```
public class enumerasi {

    enum DaftarHari{
        Senin, Selasa, Rabu, Kamis, Jumat, Sabtu, Minggu
    }

    enum DaftarWarna{
        Hitam, Putih, Merah, Biru, Hijau, Kuning
    }

    public static void main (String[] args){
        DaftarHari hari = null;
        DaftarWarna warna = null;

        System.out.println("Hari  : " + hari.Senin);
        System.out.println("Warna : " + warna.Merah);
        System.out.println("Hari  : " + hari.Jumat);
        System.out.println("Warna : " + warna.Putih);
    }
}
```

Hasil

```
run:
Hari : Senin
Warna : Merah
Hari : Jumat
Warna : Putih
```

Keterangan :

```
enum DaftarHari{
    Senin, Selasa, Rabu, Kamis, Jumat, Sabtu, Minggu
}
```

→ defenisi enum

→ nilai enum

```
DaftarHari hari = null;
```

→ deklarasi variabel menggunakan enum

```
System.out.println("Hari : " + hari.Senin);
```

→ akses nilai enum

## 2.2 Identifier

Identifier merupakan pengenal atau identitas yang berupa nama untuk variabel, sub rutin (prosedur atau fungsi), kelas, interface, maupun namespace. Berikut aturan penamaan identifier :

1. Terdiri dari huruf (a-z / A-Z) atau kombinasi dengan angka (0-9). Namun, tidak boleh diawali dengan angka.
2. Bersifat *case sensitive* yaitu membedakan penggunaan huruf besar dan huruf kecil.
3. Tidak diperbolehkan menggunakan spasi. Untuk membuat pemisah dapat menggunakan simbol *underscore* (_).
4. Tidak diperbolehkan menggunakan kata *keyword* dan simbol-simbol operator yang dikenal oleh pemrograman Java.

Contoh :

```
short nilai_uas = 100;
```

→ identifier *nilai_uas* berupa variabel

```
public class DeretAngka {

}
```

→ identifier *DeretAngka* berupa class

```
static boolean CekGanjil(){

    return true;
}
```

→ identifier *CekGanjil* berupa fungsi

## 2.3 Operator

Secara umum, operator dalam bahasa pemrograman berupa simbol yang menjalankan operasi tertentu. Terdapat beberapa operator dalam pemrograman java yaitu :

### 2.3.1 Operator Aritmatika

Operator ini menjalankan operasi bilangan dasar (aritmatika) yaitu penjumlahan (+), pengurangan (-), perkalian (*), pembagian (/), dan modulus atau modulo (%). Contoh :

```
short A=5+3;
short B=5-3;
short C=5*3;
short D=5/3;
short E=5%3;
```

Hasil

```
run:
A = 8
B = 2
C = 15
D = 1
E = 2
```

### 2.3.2 Operator *Increment/Decrement*

Operator ini menjalankan operasi penjumlahan dengan kelipatan 1 (*increment*) dan operasi pengurangan dengan kelipatan 1 (*decrement*) dari nilai variabel sebelumnya. Operator *increment* (++) dan *decrement* (--) dapat ditempat pada awal maupun akhir variabel. Contoh :

```
short A=5;
short B=3;

A++;
B--;
```

Hasil

```
run:
hasil A++ = 6
hasil B-- = 2
```

Pembahasan :

*A++* → *A = A + 1*
→ *A = 5 + 1*
→ *A = 6*

*B--* → *B = B - 1*
→ *B = 3 - 1*
→ *B = 2*

### 2.3.3 Operator Penugasan (*Assignment*)

Operator ini mirip seperti operator *increment/decrement* namun lebih variatif. Operator penugasan dapat menggunakan seluruh operasi aritmatika dengan kelipatan angka yang berbeda-beda. Contoh :

```
short A=5;
short B=5;
short C=5;
short D=5;
short E=5;

A+=2;
B-=3;
C*=4;
D/=5;
E%=6;
```

Hasil

```
run:
hasil A+=2 adalah 7
hasil B-=3 adalah 2
hasil C*=4 adalah 20
hasil D/=5 adalah 1
hasil E%=6 adalah 5
```

Pembahasan :

*A+=2* → *A = A + 2*
→ *A = 5 + 2*
→ *A = 7*

*B-=3* → *B = B - 3*
→ *B = 5 - 3*
→ *B = 2*

*C*=4* → *C = C * 4*
→ *C = 5 * 4*
→ *C = 20*

*D/=5* → *D = D / 5*
→ *D = 5 / 5*
→ *D = 1*

*E%=6* → *E = E % 6*
→ *E = 5 % 6*
→ *E = 5*

### 2.3.4 Operator Pembanding

Operator ini disebut operator relasi yaitu operator yang berfungsi untuk mengetahui kebenaran dari hubungan nilai operan kanan dan kiri dengan cara membandingkannya. Operator ini akan menghasilkan nilai bertipe boolean yaitu *True/False*.

**Tabel 2.2** Operator Pembanding

| Operator | Keterangan |
|---|---|
| == | Sama dengan |
| != | Tidak sama dengan |
| > | Lebih besar |
| >= | Lebih besar sama dengan |
| < | Lebih kecil |
| <= | Lebih kecil sama dengan |

Contoh :

```
boolean A=(5!=7);
boolean B=(10<=7);
```

Hasil

```
run:
pernyataan 5!=7 adalah true
pernyataan 10<=7 adalah false
```

### 2.3.5 Operator Logika

Seperti halnya operator pembanding, operator logika juga termasuk kedalam operasi relasi yang berfungsi untuk mengetahui kebenaran dari hubungan nilai operan kanan dan kiri serta menghasilkan nilai bertipe boolean. Namun, perbedaannya adalah operator logika hanya menghubungkan nilai operan bertipe boolean saja.

**Tabel 2.3** Operator Logika

| Operator | Nama | Keterangan |
|---|---|---|
| && | dan (and) | Menghasilkan *true* jika hanya kedua nilai operan bernilai *true* |
| \|\| | atau (or) | Menghasilkan *true* salah satu nilai operan bernilai *true* |
| ! | tidak (not) | Menghasilkan kebalikan nilai operan |

Contoh :

Operator logika and (&&)

```
boolean A=(true && true);
boolean B=(true && false);
boolean C=(false && true);
boolean D=(false && false);
```

Hasil

```
run:
logika A adalah true
logika B adalah false
logika C adalah false
logika D adalah false
```

Operator logika or (||)

```
boolean A=(true || true);
boolean B=(true || false);
boolean C=(false || true);
boolean D=(false || false);
```

Hasil

```
run:
logika A adalah true
logika B adalah true
logika C adalah true
logika D adalah false
```

Operator logika not (!)

```
boolean A=true;
boolean B=false;
```

Hasil

```
run:
logika !A adalah false
logika !B adalah true
```

### 2.3.6 Operator Bitwise

Operasi bitwise adalah operasi pengolahanan/manipulasi bit atau bilangan biner (0/1). Angka 0 menjadi nilai *false* dan angka 1 menjadi nilai *true*. Setiap nilai baik angka (0-9) maupun huruf (a-z dan A-Z) akan diubah kedalam bentuk bilangan biner kemudian dioleh sesuai dengan operator bitwise yang digunakan. Adapun operator bitwise dapat dilihat pada tabel 2.4 di bawah ini.

**Tabel 2.4** Operator Bitwise

| Operator | Nama | Keterangan |
|---|---|---|
| & | And | Sama seperti operator logika and |
| \| | Or | Sama seperti operator logika or |
| ^ | XOR (Exclusive Or) | Menghasilkan *true* jika salah satu nilai operan bernilai *true* dan salah satu operan bernilai *false* |
| ~ | Not | Sama seperti operator logika not |
| << | Left Shift | Menggeser bit beberapa digit ke kiri |
| >> | Right Shift | Menggeser bit beberapa digit ke kanan |

Contoh :

a) Operator AND (&)

```
short A=19;
short B=85;
int C=A&B;
System.out.println("hasil AND pada C adalah "+C);
```

Hasil

```
run:
hasil AND pada C adalah 17
```

Pembahasan :

- Nilai 19 dan 85 (merupakan bilangan desimal dalam tabel ASCII) dikonversi menjadi bilangan biner.
- Bilangan biner dari masing-masing angka tersebut akan direlasikan dengan logika AND (lihat operator logika AND) sesuai dengan urutan digitnya.
- Hasil relasi logika AND dikonversikan kembali menjadi bilangan desimal yaitu 17.

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| A=19 | → | 0 | 0 | 0 | **1** | 0 | 0 | 1 | **1** | bentuk biner |
| B=85 | → | 0 | 1 | 0 | **1** | 0 | 1 | 0 | **1** | |
| C= | → | 0 | 0 | 0 | **1** | 0 | 0 | 0 | **1** | |
| C= | | | | | **16** | | + | | **1** | = **17** |

b) Operator OR (|)

```
short A=19;
short B=85;
int C=A|B;
System.out.println("hasil OR pada C adalah "|C);
```

```
run:
hasil OR pada C adalah 87
```

Pembahasan :

- Nilai 19 dan 85 (merupakan bilangan desimal dalam tabel ASCII) dikonversi menjadi bilangan biner.
- Bilangan biner dari masing-masing angka tersebut akan direlasikan dengan logika OR (lihat operator logika OR) sesuai dengan urutan digitnya.
- Hasil relasi logika OR dikonversikan kembali menjadi bilangan desimal yaitu 87.

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| A=19 | → | 0 | **0** | 0 | **1** | 0 | **0** | **1** | **1** | bentuk biner |
| B=85 | → | 0 | **1** | 0 | **1** | 0 | **1** | **0** | **1** | |
| C= | → | 0 | **1** | 0 | **1** | 0 | **1** | **1** | **1** | |
| C= | | | **64** | + | **16** | + | **4** | + **2** + **1** | = **87** | |

c) Operator XOR (^)

```
short A=19;
short B=85;
int C=A^B;
System.out.println("hasil XOR pada C adalah "+C);
```

```
run:
hasil XOR pada C adalah 70
```

Pembahasan :

- Nilai 19 dan 85 (merupakan bilangan desimal dalam tabel ASCII) dikonversi menjadi bilangan biner.
- Bilangan biner dari masing-masing angka tersebut akan direlasikan dengan logika XOR sesuai dengan urutan digitnya.
- Logika XOR akan menghasilkan nilai true jika kedua operan memiliki nilai yang berbeda.
- Hasil relasi logika XOR dikonversikan kembali menjadi bilangan desimal yaitu 70.

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| A=19 | → | 0 | **0** | 0 | 1 | 0 | **0** | **1** | 1 | bentuk biner |
| B=85 | → | 0 | **1** | 0 | 1 | 0 | **1** | **0** | 1 | |
| C= | → | 0 | **1** | 0 | 0 | 0 | **1** | **1** | 0 | |

C= **64** + **4** + **2** = **70**

d) Operator Left Shift (<<)

```
short A=19;
int C=A<<3;
System.out.println("hasil left shift pada C adalah "+C);
```

Hasil

```
run:
hasil left shift pada C adalah 152
```

Pembahasan :

- Nilai 19 (merupakan bilangan desimal dalam tabel ASCII) dikonversi menjadi bilangan biner.
- Urutan bilangan biner digeser ke kiri sejauh 3 langkah.
- Hasil pergeseran bilangan binar tersebut dikonversikan kembali menjadi bilangan desimal yaitu 152.

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| A= 19 | → | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |
| pergeseran ke-1 | | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| pergeseran ke-2 | | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| pergeseran ke-3 | | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |

C= **128** + **16** + **8** = **152**

e) Operator Right Shift (>>)

```
short B=85;
int C=B>>3;
System.out.println("hasil right shift pada C adalah "+C);
```

Hasil

```
run:
hasil right shift pada C adalah 10
```

Pembahasan :

- Nilai 85 (merupakan bilangan desimal dalam tabel ASCII) dikonversi menjadi bilangan biner.
- Urutan bilangan biner digeser ke kanan sejauh 3 langkah.
- Hasil pergeseran bilangan biner tersebut dikonversikan kembali menjadi bilangan desimal yaitu 10.

B= 85

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |

pergeseran ke-1
pergeseran ke-2
pergeseran ke-3

C= **8 + 2 = 10**

### 2.3.7 Operator Ternary

Konsep kerja operator ini mirip seperti pernyataan IF dengan 2 kondisi, namun disederhanakan menjadi 1 baris berupa pertanyaan. Contoh

```
short A=9;
int B=A%2;
String bilangan;
bilangan = (B==1)? "Ganjil" :"Genap";
System.out.println(A+" adalah bilangan "+bilangan);
```

Hasil

```
run:
9 adalah bilangan Ganjil
```

Pembahasan :

```
bilangan = (B==1)? "Ganjil":"Genap";
           1       2        3
```

Operator Ternary terdiri dari 3 parameter yaitu :
1. Variabel yaitu bilangan dan Kriteria/Kondisi yaitu (B==1).
2. Nilai jika kriteria/kondisi terpenuhi (bernilai *true*).
3. Nilai jika kriteria/kondisi tidak terpenuhi (bernilai *false*).

Adapun nilai yang terpilih akan disimpan pada variabel yang tertulis pada parameter 1 yaitu bilangan.

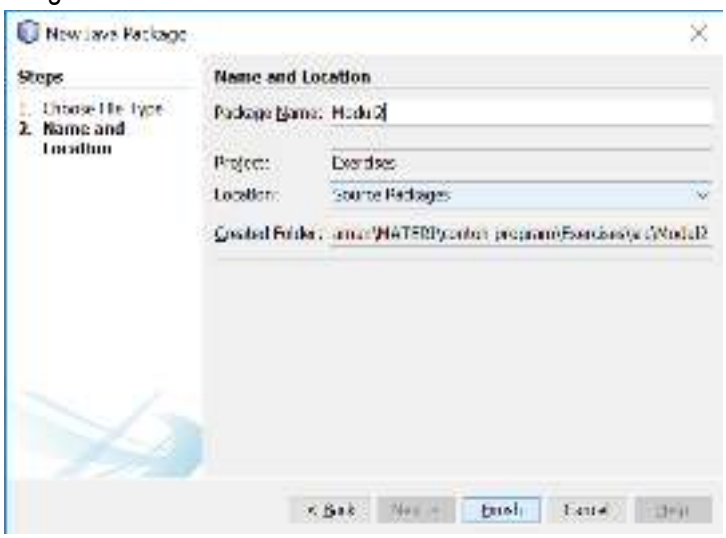
## 2.4 Latihan

### 2.4.1 Aplikasi Kalkulator1

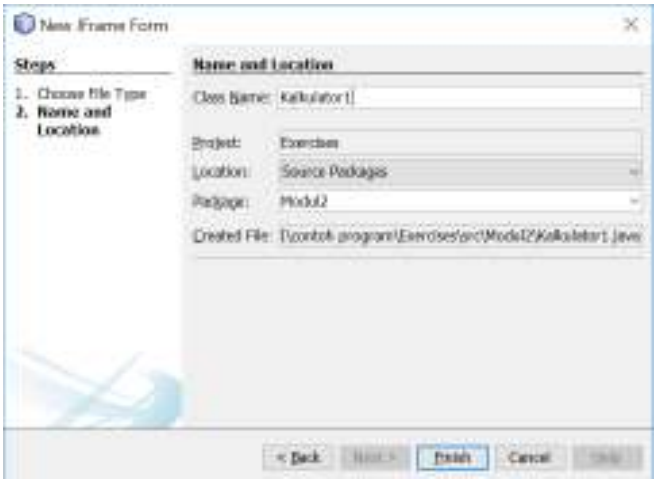
Berikut adalah percobaan membuat aplikasi kalkulator sederhana untuk menggunakan operator aritmatika.

1. Masih menggunakan projek sebelumnya pada modul 1 yaitu ‘Exercises’.
2. Apabila projek tersebut tidak ada, maka buatlah projek pada NetBeans IDE dengan cara yang sama seperti membuat projek (New Project) pada percobaan modul 1**.**

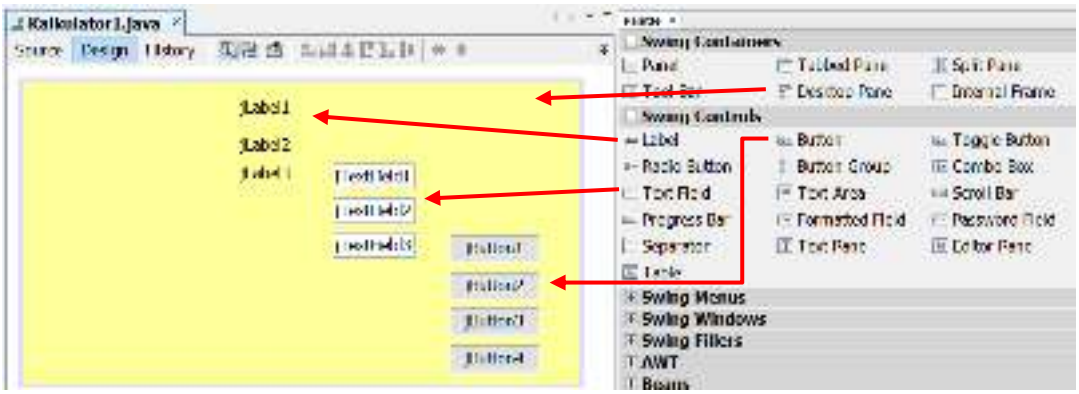
3. Tambahkan paket (*Java Package*) pada projek ‘Exercises’ dengan langkah-langkah seperti pada percobaan modul 1 (*New Package*). Beri nama paket dengan ‘Modul2’.




4. Tambahkan ‘JFrame Form’ pada paket ‘Modul2’ dengan langkah-langkah seperti pada percobaan modul 1 (menambah frame form). Beri nama JFrame Form dengan ‘Kalkulator1’.



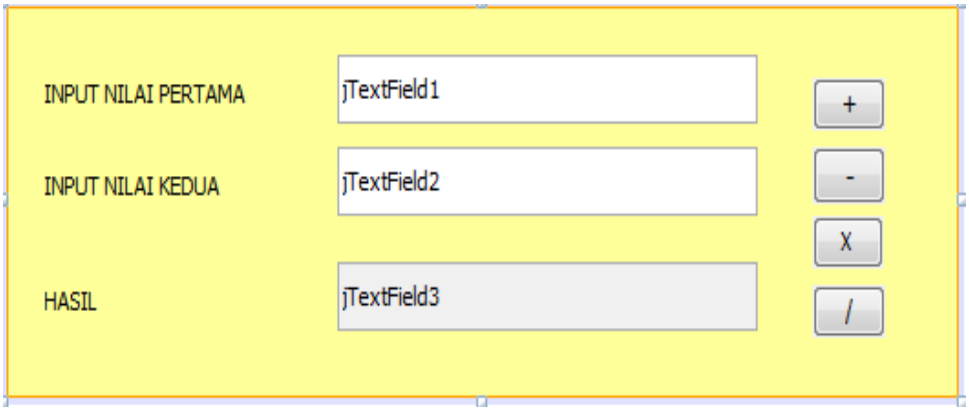

5. Desain form menggunakan beberapa komponen dari jendela palette seperti berikut ini :




Catatan!

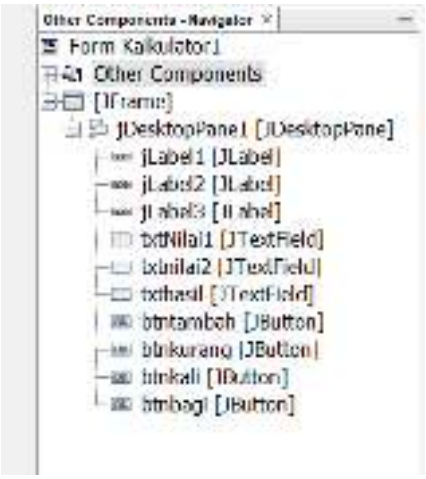
- ✓ Komponen ‘Label’ merupakan komponen keterangan yang digunakan untuk memberikan keterangan pada komponen lain atau menampilkan informasi tertentu.
- ✓ Komponen ‘Text Filed’ merupakan komponen entri data yang digunakan untuk memasukkan nilai/data dengan cara diketik pada saat program dijalankan/running. Selain memasukkan nilai/data, komponen ini juga dapat menampilkan nilai/data yang bersifat *non-editable*.

6. Ubah susunan dan nilai properti masing-masing komponen seperti berikut ini.




| Komponen | Nama Variabel | Properties |
|---|---|---|
| Desktop Pane | - | background : Yellow |
| jLabel1 | - | text = INPUT NILAI PERTAMA |
| jLabel2 | - | text = INPUT NILAI KEDUA |
| jLabel3 | - | text = HASIL |
| jTextField1 | txtnilai1 | text = <kosongkan> |
| jTextField2 | txtnilai2 | text = <kosongkan> |
| jTextField3 | txthasil | text = <kosongkan><br>editable = False |
| jButton1 | btntambah | text = + |
| jButton2 | btnkurang | text = - |
| jButton3 | btnkali | text = x |
| jButton4 | btnbagi | text = / |

7. Seluruh komponen yang digunakan (beserta nama variabel dari komponen) dapat dilihat pada jendela Navigator.




8. Tulis kode program dibawah pada masing-masing button dengan cara seperti berikut :
   Klik kanan komponen btntambah pada jendela navigator → Event → Action → klik actionPerformed. Ketikan kode program dibawah ini

```
private void btntambahActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    Double nilai1 = Double.valueOf(txtnilai1.getText());
    Double nilai2 = Double.valueOf(txtnilai2.getText());
    Double hasil = nilai1 + nilai2 ;
    String tampil = String.valueOf(hasil);
    txthasil.setText(tampil);
```

   Klik kanan komponen btnkurang pada jendela navigator → Event → Action → klik actionPerformed. Ketikan kode program dibawah ini

```
private void btnkurangActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    Double nilai1 = Double.valueOf(txtnilai1.getText());
    Double nilai2 = Double.valueOf(txtnilai2.getText());
    Double hasil = nilai1 - nilai2 ;
    String tampil = String.valueOf(hasil);
    txthasil.setText(tampil);
```

   Klik kanan komponen btnbagi pada jendela navigator → Event → Action → klik actionPerformed. Ketikan kode program dibawah ini

```
private void btnbagiActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    Double nilai1 = Double.valueOf(txtnilai1.getText());
    Double nilai2 = Double.valueOf(txtnilai2.getText());
    Double hasil = nilai1 / nilai2;
    String tampil = String.valueOf(hasil);
    txthasil.setText(tampil);
}
```

Klik kanan komponen btnkali pada jendela navigator → Event → Action → klik actionPerformed. Ketikan kode program dibawah ini

```
private void btnkaliActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    Double nilai1 = Double.valueOf(txtnilai1.getText());
    Double nilai2 = Double.valueOf(txtnilai2.getText());
    Double hasil = nilai1 * nilai2 ;
    String tampil = String.valueOf(hasil);
    txthasil.setText(tampil);
```

Apabila terdapat kode program dengan garis bawah berwarna merah dan tanda seperti pada gambar berarti terdapat kesalahan pada baris dan kode program tersebut.

9. Untuk menjalankan program, klik menu Run → Run File (Shift + F6).

# MODUL 3
## STRUKTUR PENSELEKSIAN

**Topik Pembahasan :**

- **Pernyataan *if***
- **Pernyataan *if-else***
- **Pernyataan *if-else-if***
- **Pernyataan *switch***
- **Penseleksian Bersarang**
- **Latihan**

### 3.1 Pernyataan *if*

Pernyataan *if* digunakan untuk menyeleksi 1 kondisi/keadaan dengan 1 kriteria/persyaratan (*boolean_expression*). Kondisi (dapat berupa blok kode program) akan dieksekusi jika dan hanya jika kriteria terpenuhi atau bernilai benar (*true*). Sintaks atau bentuk penulisan pernyataan *if* adalah sebagai berikut :

```
if( boolean_expression )
      statement;          → kondisi

atau

if( boolean_expression ){
    statement1;
    statement2;           } kondisi
      . . . ;
}
```

Keterangan :
*boolean_expression* merupakan kriteria dari suatu kondisi berupa operator pembanding.

### 3.2 Pernyataan *if-else*

Pernyataan *if-else* digunakan untuk menyeleksi 2 kondisi/keadaan dengan 1 kriteria/persyaratan. Apabila kriteria pertama terpenuhi atau bernilai benar (*true*) maka kondisi ke-1 akan dieksekusi. Namun, jika kriteria pertama tidak terpenuhi atau bernilai salah (*false*) maka secara otomatis kondisi ke-2 akan dieksekusi. Sintaks atau bentuk penulisan pernyataan *if-else* adalah sebagai berikut :

```
if( boolean_expression )
    statement;          → kondisi ke-1
else
    statement;          → kondisi ke-2

atau

if( boolean_expression ) {
    statement1;
    statement2;           } kondisi ke-1
      …;
} else {
    statement3;
    statement4;           } kondisi ke-2
    …;
}
```

Contoh :

```
short pwd = 123;

if (pwd == 123)
    System.out.println("Password Anda Benar!");
else
    System.out.println("Password Anda Salah!");
```

Hasil

```
run:
Password Anda Benar!
```

Pembahasan:
Operator pembanding yang digunakan sebagai kriteria adalah sama dengan (==). Nilai variabel '*pwd*' adalah 123 sehingga hasil operator pembanding menjadi 123==123 (dibaca 'benarkah 123 sama dengan 123') dan hasilnya adalah benar. Oleh karena hasil dari operator pembanding bernilai benar atau kriteria terpenuhi maka program akan memilih kondisi ke-1. Apabila hasil operator pembanding bernilai salah atau kriteria tidak terpenuhi maka program akan memilih kondisi ke-2.

### 3.3 Pernyataan *if-else-if*

Pernyataan *if-else if* dapat digunakan untuk menyeleksi beberapa kondisi (2 atau lebih) dengan beberapa kriteria (2 atau lebih) pula. Kondisi struktur seperti ini mengijinkan kita untuk membuat seleksi kondisi yang lebih kompleks. Sintaks atau bentuk penulisan pernyataan *if-else if* adalah sebagai berikut :

```
if( boolean_expression1 ){
    statement1;
    statement2;
} else if( boolean_expression2) {
    statement3;
    statement4;
} else {
    statement5;
    statement6;
}
```

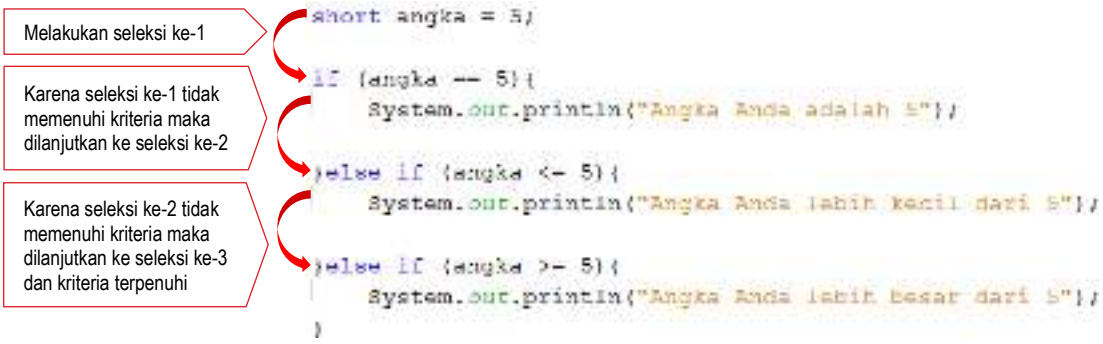
Kondisi ke-1

Kondisi ke-2

Kondisi ke-3

Keterangan :
Kondisi ke-3 akan dieksekusi secara otomatis apabila seluruh kriteria (`boolean_expression`) sebelumnya tidak terpenuhi.

Contoh 1:




Hasil

```
run:
Angka Anda lebih besar dari 5
```

Pembahasan :

Ada 3 kriteria yang digunakan pada contoh di atas dengan operator pembanding yang berbeda-beda. Program akan memulai seleksi dari pernyataan *if* pertama atau kriteria-1.

- Kriteria-1 menggunakan operator sama dengan (==) yaitu angka == 5 (dibaca '*benarkah 8 = 5*') dan hasilnya adalah salah (*false*) sehingga kriteria-1 tidak terpenuhi. Oleh karena itu, program akan melanjutkan proses seleksi ke pernyataan *if* kedua atau kriteria-2.

- Kriteria-2 menggunakan operator lebih kecil sama dengan (<=) yaitu angka <= 5 (dibaca '*benarkah 8<=5*') dan hasilnya adalah salah (*false*) sehingga kriteria-2 tidak terpenuhi. Oleh karena itu, program akan melanjutkan proses seleksi ke pernyataan *if* ketiga atau kriteria-3.

- Kriteria-3 menggunakan operator lebih besar sama dengan (>=) yaitu angka >= 5 (dibaca '*benarkah 8>=5*') dan hasilnya adalah benar (*true*) atau kriteria-3 terpenuhi. Oleh karena itu, program akan memilih kondisi ke 3 dan menghentikan proses seleksi.

Contoh 2:

Melakukan seleksi ke-1 dan kriteria terpenuhi

```
short angka = 5;

if (angka == 5){
    System.out.println("Angka Anda adalah 5");

}else if (angka <= 5){
    System.out.println("Angka Anda lebih kecil dari 5");

}else if (angka >= 5){
    System.out.println("Angka Anda lebih besar dari 5");
}
```

Hasil

```
run:
Angka Anda adalah 5
```

Pembahasan :

Operator pembanding pada seleksi ke-1 bernilai benar (*true*) atau kriteria terpenuhi. Oleh karena itu, program akan memilih kondisi ke 1 dan menghentikan proses seleksi.

Pada kedua contoh di atas, pernyataan *if* yang digunakan merupakan satu struktur bangunan *if*. Hal ini berarti bahwa apabila proses seleksi pada kriteria pertama telah terpenuhi (operator pembanding bernilai benar/*true*) maka proses seleksi pada kriteri-kriteria berikut akan dihentikan (proses seleksi telah selesai).

Contoh 3:

Kriteri pada masing-masing pernyataan *if* terpenuhi atau bernilai benar (*true*)

```
short angka = 5;

if (angka == 5)
    System.out.println("Angka Anda adalah 5");

if (angka <= 5)
    System.out.println("Angka Anda lebih kecil dari 5");

if (angka >= 5)
    System.out.println("Angka Anda lebih besar dari 5");
```

Hasil

```
run:
Angka Anda adalah 5
Angka Anda lebih kecil dari 5
Angka Anda lebih besar dari 5
```

Pembahasan :

Pada contoh 3 di atas, terdapat 3 pernyataan if dengan struktur bangunan yang berdiri sendiri-sendiri, sehingga proses seleksi akan terjadi pada setiap pernyataan *if* atau terjadi 3x proses seleksi.

### 3.4 Pernyataan *switch*

Cara lain untuk membuat cabang seleksi adalah dengan menggunakan pernyataan *switch*. Pernyataan *switch* mengkonstruksikan cabang seleksi untuk beberapa kondisi dengan beberapa kriteria. Berbeda dengan pernyataan *if*, pernyataan switch hanya menggunakan operator pembanding sama dengan (==) saja dan tidak dapat menggunakan operator logika. Sintaks atau bentuk penulisan pernyataan switch adalah sebagai berikut :

```
switch(switch_expression){
   case case_selector1:
      statement1;
      statement2;          kondisi ke-1
      ...;
      break;               menghentikan proses seleksi/keluar dari switch
   case case_selector2:
      statement3;
      statement4;          kondisi ke-2
      . . . ;
      break;
   case case_selector-n
      statemen-n;
      ….;
      break;
   default:                memilih kondisi secara otomatis apabila kriteria sebelumnya tidak
      statement1;          terpenuhi
      statement2;
      . . . ;
      break;
}
```

Contoh :

```
short pilihan = 2;
String menu;
int harga;

switch(pilihan){
    case 1:
        menu="Nasi Ayam Saus Mentega";
        harga=20000;
        break;
```

```
        case 2:
            menu="Nasi Gurame Asam Manis";
            harga=35000;
            break;

        case 3:
            menu="Nasi Soup Iga";
            harga=35000;
            break;

        default:
            menu="Tidak Ada";
            harga=0;
            break;
    }
    System.out.println("Pilihan: " + pilihan);
    System.out.println("Menu: " + menu);
    System.out.println("Harga: " + harga);
```

```
run:
Pilihan: 2
Menu: Nasi Gurame Asam Manis
Harga: 35000
```

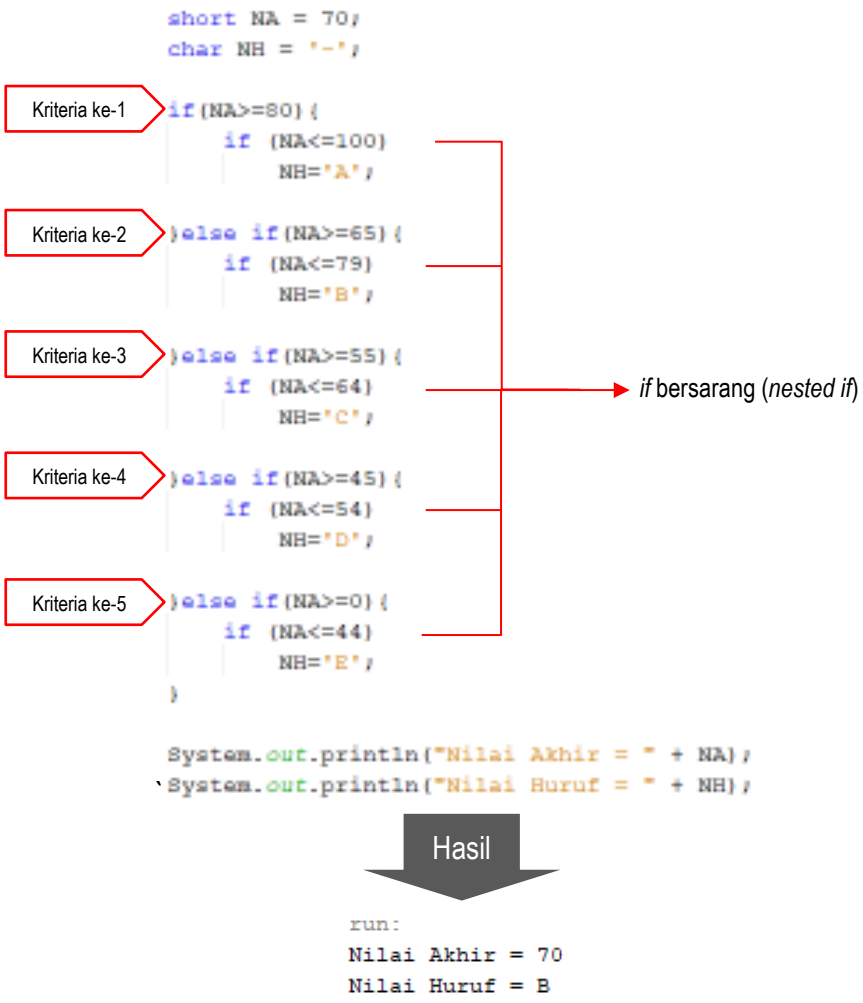
### 3.5 Penseleksian Bersarang

Penseleksian bersarang merupakan proses seleksi suatu kondisi yang berlapis atau penseleksian didalam penseleksian. Untuk membuat penseleksian bersarang dapat menggunakan pernyataan seleksi yang sejenis (*if* didalam *if* atau *switch* didalam *switch*) atau kombinasi diantara keduanya (*if* didalam *switch* atau sebaliknya).

Contoh 1 :
Konversi nilai akhir menjadi nilai huruf dengan ketentuan sebagai berikut :

| *Nilai Akhir* | | *Nilai Huruf* |
|---|---|---|
| 80 | 100 | A |
| 65 | 79 | B |
| 55 | 64 | C |
| 45 | 54 | D |
| 0 | 44 | E |

dapat disajikan dalam kode program sebagai berikut ini :




Pembahasan :

Contoh di atas menggunakan *if* bersarang (*nested if*) yaitu *if* didalam *if*. Seleksi variabel `NA = 70` memenuhi kriteria *if* yang kedua yaitu `else if(NA>=65)` dan juga memenuhi kriteria *if* yang ada didalamnya yaitu `if(NA<=79)` sehingga nilai huruf yang didapat adalah 'B'. Oleh karena itu, untuk mendapatkan nilai huruf maka kedua kriterianya harus terpenuhi atau bernilai benar (*true*).

Kode program di atas dapat disederhanakan menggunakan operator logika and seperti berikut ini.

```
short NA = 70;
char NH = '-';

if(NA>=80 && NA<=100){
        NH='A';

}else if(NA>=65 && NA<=79){
        NH='B';

}else if(NA>=55 && NA<=64){
        NH='C';

}else if(NA>=45 && NA<=54){
        NH='D';

}else if(NA>=0 && NA<=44){
        NH='E';
}

System.out.println("Nilai Akhir = " + NA);
System.out.println("Nilai Huruf = " + NH);
```

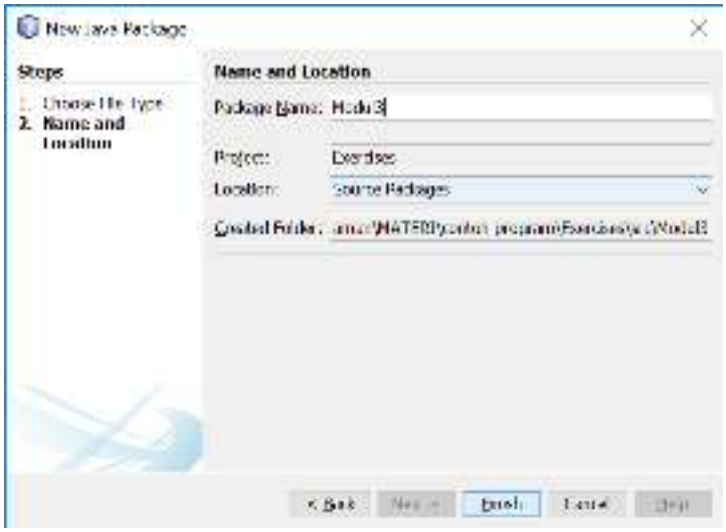
Pembahasan :

Kode program ini menggunakan operator pembanding dan operator logika '*and*' (&&) secara bersamaan. Nilai variabel `NA = 70`, sesuai dengan kriteria kedua karena memenuhi persyaratan operator '*and*' (&&) (`NA>=65 && NA<=79`) dimana nilai operan kiri bernilai bernilai *true* (`70>=65` → `true`) dan nilai operan kanan bernilai *true* (`70<=79` → `true`).

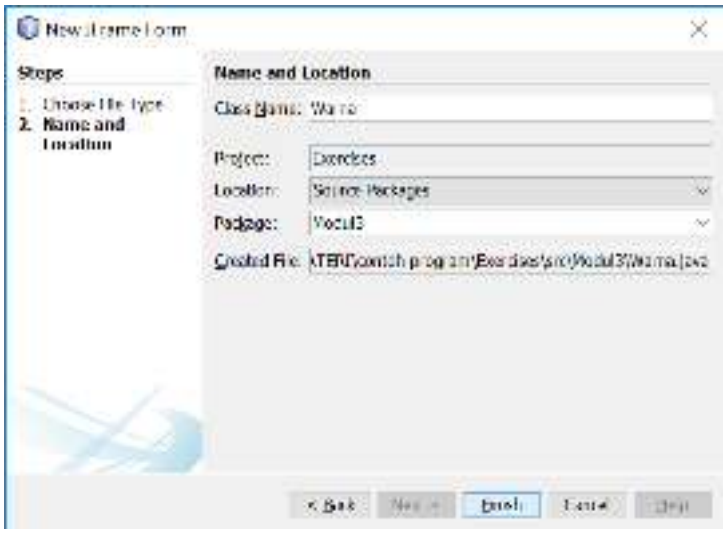
### 3.6 Latihan

#### 3.6.1 Aplikasi Warna

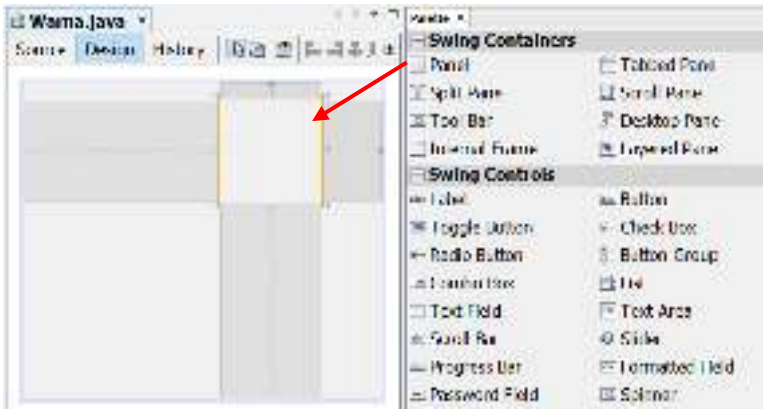
1. Tambahkan paket (*Java Package*) pada projek 'Exercises' dengan langkah-langkah seperti pada percobaan modul 1 (*New Package*). Beri nama paket dengan 'Modul3'.




2. Tambahkan ‘JFrame Form’ dalam paket ‘Modul3’. Beri nama JFrame Form dengan ‘Warna’.

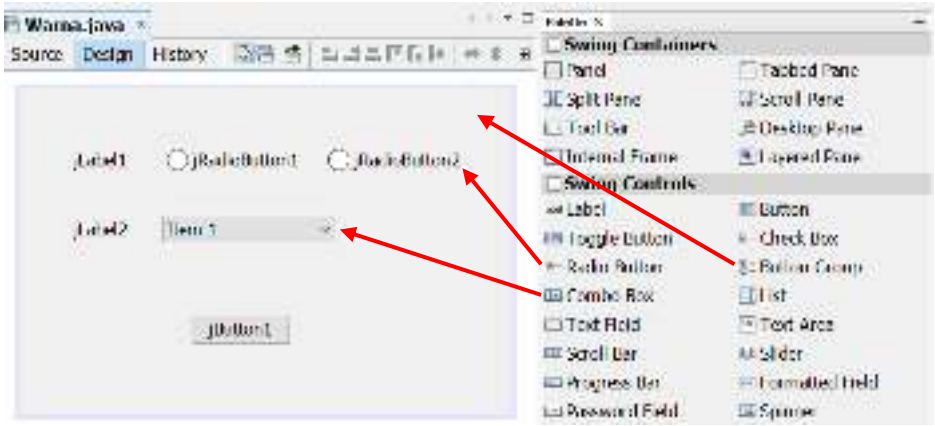



3. Masukkan komponen Panel dari jendela palette Swing Containers kedalam form.

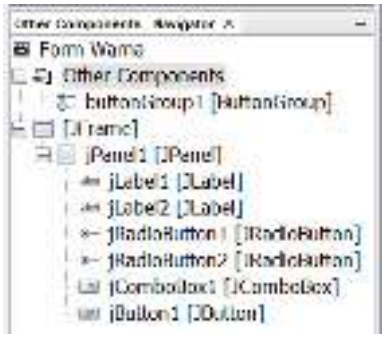



Kemudian ubah posisi dan ukurannya menutupi seluruh form.

4. Masukkan komponen Label, Radio Button, Button Group, Combo Box, dan Button pada palette Swing Controls kedalam komponen Panel sebelumnya seperti berikut ini :




Khusus untuk komponen Button Group tidak tampak pada JFrameForm namun dapat dilihat pada jendela Navigator. Komponen Button Group digunakan untuk mengelompokkan komponen Radio Button agar hanya dapat dipilih salah satunya saja.



Catatan!

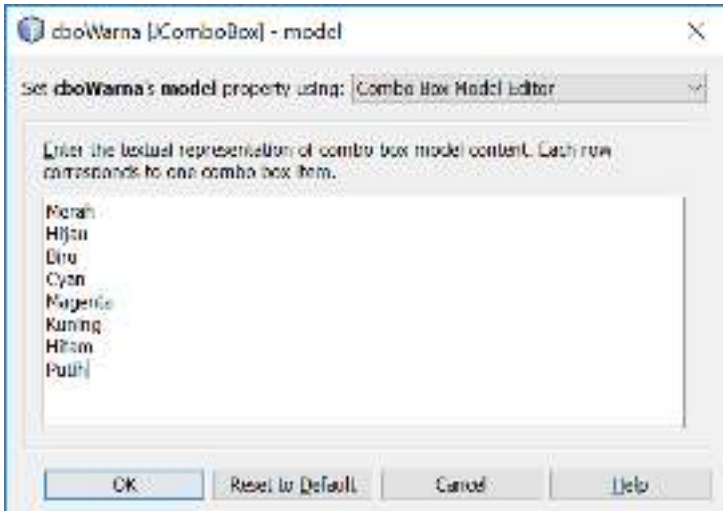
- ✓ Komponen 'Combo Box' merupakan komponen entri data yang digunakan untuk memberikan/memasukkan nilai/data dalam bentuk pilihan-pilihan. Komponen ini mampu menyimpan lebih dari satu pilihan nilai/data dimana setiap nilai/data memiliki nomor urut atau indeks. Nilai/data yang tampil/tampak pada komponen Combo Box adalah nilai/data yang terpilih (*selectedItem*) dengan indeks sesuai dengan urutannya (*selectedIndex*).
- ✓ Komponen 'Radio Button' merupakan komponen entri data yang digunakan untuk memberikan/memasukkan nilai/data dalam bentuk pilihan tunggal. Apabila menggunakan lebih dari satu komponen ini maka disarankan untuk menggunakan komponen 'Button Group'.
- ✓ Komponen 'Button Group' merupakan komponen untuk mengelompokkan komponen 'Radio Button' sesuai dengan kelompok data. Contoh kelompok 'Jenis Kelamin' menggunakan dua komponen 'Radio Button' yaitu laki-laki dan perempuan.

5. Ubah nama variabel dan nilai properti pada masing-masing komponen seperti berikut ini.

| Komponen | Nama Variabel | Properti |
|---|---|---|
| buttonGroup1 | grupWarna | - |
| jLabel1 | - | text = Bagian : |
| jLabel2 | - | text = Warna : |
| jButton1 | btnUbah | text = Ubah |
| jRadioButton1 | rbTeks | text = Warna Teks |
| | | buttonGroup = grupWarna |
| jRadioButton2 | rbLatar | text = Warna Latar |
| | | buttonGroup = grupWarna |
| jComboBox1 | cboWarna | model = Merah, Hijau, Biru |
| | | |

6. Untuk mengisi nilai/data pada komponen JComboBox, klik komponen JComboBox terlebih dahulu kemudian pada jendela properties, klik tombol pada properti 'model'.

7. Isi nilai/data yang dibutuhkan, kemudian klik tombol OK.




8. Klik kanan komponen btnUbah pada jendela navigator → Event → Action → klik ‘actionPerformed’. Ketikan kode program dibawah ini

```
    private void btnUbahActionPerformed(java.awt.event.ActionEvent evt) {

        if (rbTerar.isSelected()){
            if (cboWarna.getSelectedItem()== "Merah"){
                jPanel1.setBackground(Color.red);

            }else if (cboWarna.getSelectedItem()== "Hijau"){
                jPanel1.setBackground(Color.green);

            }else if (cboWarna.getSelectedItem()== "Biru"){
                jPanel1.setBackground(Color.blue);

            }else if (cboWarna.getSelectedItem()== "Cyan"){
                jPanel1.setBackground(Color.cyan);

            }else if (cboWarna.getSelectedItem()== "Magenta"){
                jPanel1.setBackground(Color.magenta);

            }else if (cboWarna.getSelectedItem()== "Kuning"){
                jPanel1.setBackground(Color.yellow);

            }else if (cboWarna.getSelectedItem()== "Hitam"){
                jPanel1.setBackground(Color.black);

            }else if (cboWarna.getSelectedItem()== "Putih"){
                jPanel1.setBackground(Color.white);

            }
```

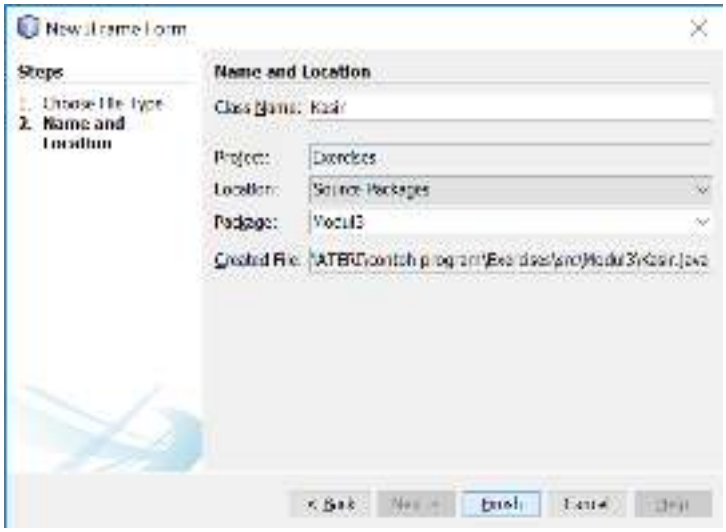
```
147             }else if (rbTeks.isSelected()){
148                 switch (cbWarna.getSelectedIndex()){
149                     case 0:
150                         jLabel1.setForeground(Color.red);
151                         jLabel2.setForeground(Color.red);
152                         rbLatar.setForeground(Color.red);
153                         rbTeks.setForeground(Color.red);
154                         break;
155
156                     case 1:
157                         jLabel1.setForeground(Color.green);
158                         jLabel2.setForeground(Color.green);
159                         rbLatar.setForeground(Color.green);
160                         rbTeks.setForeground(Color.green);
161                         break;
162
163                     case 2:
164                         jLabel1.setForeground(Color.blue);
165                         jLabel2.setForeground(Color.blue);
166                         rbLatar.setForeground(Color.blue);
167                         rbTeks.setForeground(Color.blue);
168                         break;
169
170                     case 3:
171                         jLabel1.setForeground(Color.cyan);
172                         jLabel2.setForeground(Color.cyan);
173                         rbLatar.setForeground(Color.cyan);
174                         rbTeks.setForeground(Color.cyan);
175                         break;
176
177                     case 4:
178                         jLabel1.setForeground(Color.magenta);
179                         jLabel2.setForeground(Color.magenta);
180                         rbLatar.setForeground(Color.magenta);
181                         rbTeks.setForeground(Color.magenta);
182                         break;
183
184                     case 5:
185                         jLabel1.setForeground(Color.yellow);
186                         jLabel2.setForeground(Color.yellow);
187                         rbLatar.setForeground(Color.yellow);
188                         rbTeks.setForeground(Color.yellow);
189                         break;
190
191                     case 6:
192                         jLabel1.setForeground(Color.black);
193                         jLabel2.setForeground(Color.black);
194                         rbLatar.setForeground(Color.black);
195                         rbTeks.setForeground(Color.black);
196                         break;
197
198                     case 7:
199                         jLabel1.setForeground(Color.white);
200                         jLabel2.setForeground(Color.white);
201                         rbLatar.setForeground(Color.white);
202                         rbTeks.setForeground(Color.white);
203                         break;
204                 }
205             }
206         }
```

9. Pastikan tidak ada tanda seru merah pada kode program Anda. Selanjutnya, jalankan program dengan cara klik kanan pada file program (frame) pilih Run File atau tekan Shift + F6 pada keyboard.

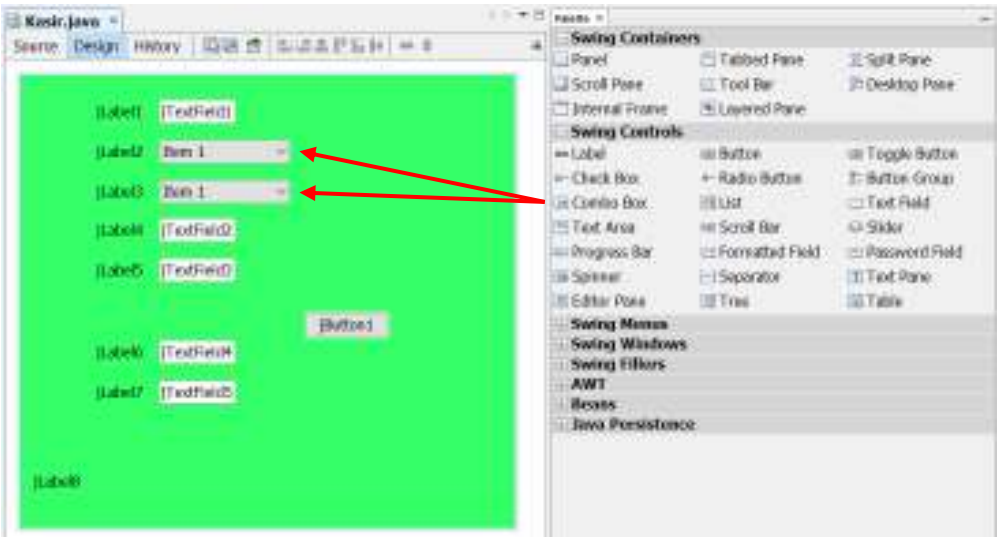
### 3.6.2 Aplikasi Kasir

Berikut adalah percobaan membuat aplikasi kasir sederhana untuk menghitung tagihan transaksi.

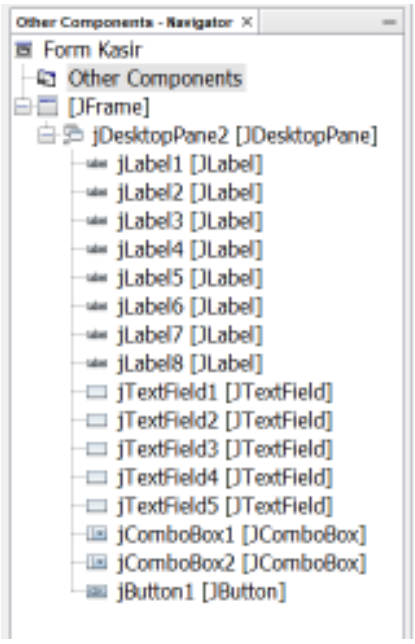
1. Tambahkan 'JFrame Form' dalam paket 'Modul3'. Beri nama JFrame Form dengan 'Kasir'.




2. Desain form menggunakan komponen Dekstop Pane dari palette Swing Containers dan komponen Label, Text Field, Combo Box, serta Button dari jendela palette (Swing Controls) seperti berikut ini :



Komponen yang digunakan dapat dilihat pada jendela Navigator

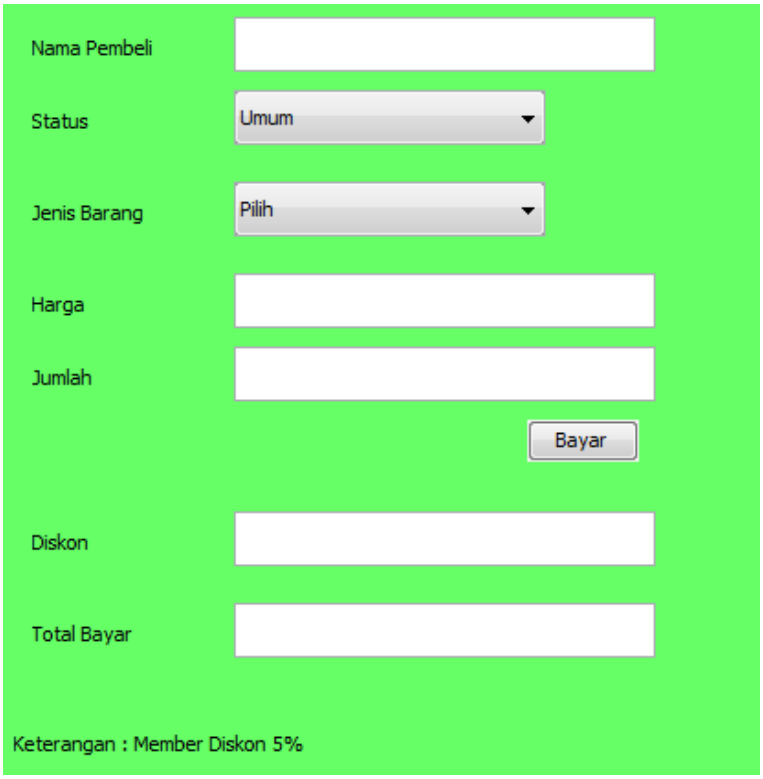



3. Ubah nama variabel dan nilai properti pada masing-masing komponen seperti berikut ini.

| Komponen | Nama Variabel | Properties |
|---|---|---|
| Desktop Pane | - | background = Green |
| jLabel1 | - | text = Nama Pembeli |
| jLabel2 | - | text = Status |
| jLabel3 | - | text = Jenis Barang |

| | | |
|---|---|---|
| jLabel4 | - | text = Harga |
| jLabel5 | - | text = Jumlah |
| jLabel6 | - | text = Diskon |
| jLabel7 | - | text = Total Bayar |
| jLabel8 | - | text = Keterangan : Member Diskon 5% |
| jButton1 | btnbayar | text = Bayar |
| jTextField1 | txtnama | text = <kosongkan> |
| jTextField2 | txtharga | text = <kosongkan> |
| | | editable = False |
| jTextField3 | txtjumlah | text = <kosongkan> |
| jTextField4 | txtdiskon | text = <kosongkan> |
| | | editable = False |
| jTextField5 | txttotal | text = <kosongkan> |
| | | editable = False |
| jComboBox1 | cbostatus | model = Umum, Member |
| jComboBox2 | cbojenis | model = Pilih, iPad, iPhone, iPod |

4. Ubah susunan masing-masing komponen seperti berikut ini.




5. Klik kanan komponen cbojenis pada jendela navigator → Event → Action → klik ‘actionPerformed’. Ketikan kode program dibawah ini

```
private void cbojenisActionPerformed(java.awt.event.ActionEvent
    // TODO add your handling code here:
    if(cbojenis.getSelectedItem()=="iPad"){
        txtharga.setText("4999000");
    }else if (cbojenis.getSelectedItem()=="iPhone"){
        txtharga.setText("7999000");
    }else if (cbojenis.getSelectedItem()=="iPod"){
        txtharga.setText("2999000");
    }else {
        txtharga.setText("0");
    }
}
```

6. Klik kanan komponen btnbayar pada jendela navigator → Event → Action → klik 'actionPerformed'. Ketikan kode program dibawah ini

```
private void btnbayarActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    long harga = Integer.parseInt(txtharga.getText());
    int jumlah = Integer.parseInt(txtjumlah.getText());
    long diskon, bayar, total;

    bayar = harga * jumlah ;

    if (cbostatus.getSelectedItem()=="Member"){
        diskon = (5 * bayar)/100 ;
        total = bayar - diskon ;
    } else {
        diskon = 0 ;
        total = bayar - diskon ;
    }
    txtdiskon.setText(""+diskon);
    txttotal.setText(""+total);
}
```

7. Perhatikan penulisan source code huruf besar & huruf kecil berpengaruh (Case Sensitif).

8. Pastikan tidak ada tanda seru merah pada kode program Anda. Selanjutnya, jalankan program dengan cara klik kanan pada file program (frame) pilih Run File atau tekan Shift + F6 pada keyboard.

# MODUL 4
## STRUKTUR PERULANGAN

**Topik Pembahasan :**

- **Struktur *for***
- **Struktur *while***
- **Struktur *do…while***
- **Perulangan Bersarang**
- **Latihan**

Perulangan (looping) merupakan proses eksekusi blok kode program secara berulang-ulang (putaran) sampai batas tertentu (kriteria terpenuhi). Secara umum, perulangan menggunakan operator pembanding sebagai suatu kriteria untuk menentukan batas akhir sehingga dalam proses kerja perulangan seolah-olah terdapat atau menggunakan pernyataan *if*. Dalam modul ini, struktur perulangan yang dibahas adalah *for*, *do…while*, dan *while*.

### 4.1 Struktur *for*

Struktur perulangan *for* memiliki 3 parameter. Sintaks atau bentuk penulisan struktur perulangan *for* adalah sebagai berikut :

```
for(inisialisasi; kriteria_kondisi; proses_iterasi){
    statement1;
    statement2;
    statement-n;
}
```

Keterangan :

- Parameter 1 adalah inisialisasi yaitu menentukan nilai awal untuk memenuhi kriteria yang digunakan.
- Parameter 2 adalah kriteria_kondisi yaitu persyaratan yang harus terpenuhi (bernilai benar) agar proses perulangan dapat berjalan. Secara umum, parameter ini menggunakan operator pembanding.
- Parameter 3 adalah proses_iterasi yaitu proses perulangan nilai (berupa angka). Secara umum, parameter ini menggunakan operator increment/decrement atau operator penugasan (*assignment*).

Contoh :

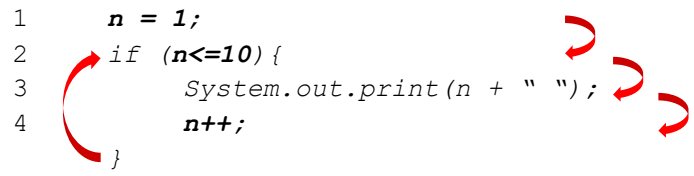
```
short n;
for (n=1; n<=10; n++){
    System.out.print(n + " ");
}
```

Hasil

```
run:
1 2 3 4 5 6 7 8 9 10
```

Pembahasan :

Tahapan proses eksekusi kode program pada contoh perulangan *for* di atas, digambarkan pada ilustrasi di bawah ini.




- Tahap 1 merupakan parameter pertama yaitu menentukan nilai awal dari variabel *n*.
- Tahap 2 merupakan parameter kedua yaitu proses seleksi dengan kriteria/syarat ***n<=10*** (operator pembanding lebih kecil sama dengan). Apabila kriteria terpenuhi atau bernilai benar (*true*) maka eksekusi kode program akan dilanjutkan ke tahap 3.
- Tahap 3 merupakan pernyataan atau kode program didalam perulangan *for*.
- Tahap 4 merupakan parameter ketiga yaitu proses iterasi dengan operator increment.
- Selanjutnya proses eksekusi akan diulang kembali dari tahap 2 dan dilanjutkan lagi ke tahap-tahap berikut.
- Putaran eksekusi dalam perulangan tetap terjadi selama kriteria terpenuhi atau bernilai benar (*true*).
- Apabila kriteria tidak terpenuhi maka eksekusi perulangan akan berhenti (selesai).

### 4.2 Struktur *while*

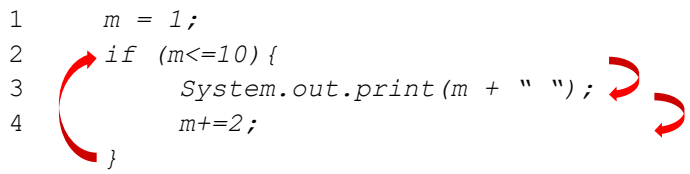
Perulangan ini akan bekerja selama (*while*) kriteria kondisi terpenuhi atau bernilai benar (*true*). Berikut adalah sintaks atau bentuk penulisan perulangan while.

```
while(kriteria_kondisi) {
    statement1;
    statement2;
    statement-n;
    proses_iterasi;
}
```

Contoh :

```
short m=1;
while (m <= 10){
    System.out.print(m + " ");
    m+=2;
}
```

Pembahasan :
Tahapan proses eksekusi kode program pada contoh perulangan *while* di atas, digambarkan pada ilustrasi di bawah ini.




- Tahap 1, menentukan nilai awal dari variabel *m*.
- Tahap 2, melakukan proses seleksi dengan kriteria/syarat ***m<=10*** (operator pembanding lebih kecil sama dengan). Apabila kriteria terpenuhi atau bernilai benar (*true*) maka eksekusi kode program akan dilanjutkan ke tahap 3.
- Tahap 3, menjalankan eksekusi pernyataan atau blok kode program didalam perulangan *while*.
- Tahap 4, melakukan proses iterasi dengan operator penugasan (*assignment*).
- Selanjutnya proses eksekusi akan diulang kembali dari tahap 2 dan dilanjutkan lagi ke tahap-tahap berikut.
- Putaran eksekusi dalam perulangan tetap terjadi selama kriteria terpenuhi atau bernilai benar (*true*).
- Apabila kriteria tidak terpenuhi maka eksekusi perulangan akan berhenti (selesai).

**Peringatan**
Perulangan *while* tanpa menggunakan proses iterasi mengakibatkan proses eksekusi kode program terjadi tanpa akhir karena kriteria kondisi akan tetap terpenuhi.

### 4.3 Struktur do…while

Seperti halnya perulangan *while*, perulangan ini juga akan bekerja selama (*while*) kriteria kondisi terpenuhi atau bernilai benar (*true*). Namun, berbeda dengan 2 struktur perulangan sebelumnya (*for* dan *while*), *do…while* akan melakukan eksekusi

pernyataan terlebih dahulu (*do*) sebelum melakukan seleksi kriterianya (*while*). Berikut adalah sintaks atau bentuk penulisan perulangan *do…while*

```
do{
    statement1;
    statement2;
    statement-n;
    proses_iterasi
}while(kriteria_kondisi);
```

Contoh :

```
int p=2;
do {
    System.out.print(p + " ");
    p = p + 2;

} while (p <= 10);
```

Hasil

```
run:
2 4 6 8 10
```

Pembahasan :

Tahapan proses eksekusi kode program pada contoh perulangan *do…while* di atas, digambarkan pada ilustrasi di bawah ini.

```
1     p = 2;
2     System.out.print(p + " ");
3     p = p + 2;
4     if (p<=10){
5         <kriteria terpenuhi>
6     } else {
          <kriteria tidak terpenuhi>
      }
```

- Tahap 1, menentukan nilai awal dari variabel *p*.
- Tahap 2, menjalankan eksekusi pernyataan atau blok kode program didalam perulangan *do…while*.
- Tahap 3, melakukan proses iterasi dengan operator penugasan (*assignment*).
- Tahap 4, melakukan proses seleksi dengan kriteria/syarat ***p<=10*** (operator pembanding lebih kecil sama dengan).

- Apabila kriteria terpenuhi atau bernilai benar (*true*) maka eksekusi kode program akan diulang kembali dari tahap 2 dan dilanjutkan lagi ke tahap-tahap berikut.
- Putaran eksekusi dalam perulangan tetap terjadi selama kriteria terpenuhi atau bernilai benar (*true*).
- Apabila kriteria tidak terpenuhi maka eksekusi perulangan akan berhenti (selesai).

**Peringatan**

Perulangan *do…while* tanpa menggunakan proses iterasi mengakibatkan proses eksekusi kode program terjadi tanpa akhir karena kriteria kondisi akan tetap terpenuhi.

### 4.4 Perulangan Bersarang

Perulangan bersarang adalah penggunaan perulangan didalam perulangan. Perulangan yang digunakan dapat berupa perulangan sejenis atau kombinasi jenis perulangan satu dengan jenis perulangan yang lainnya.

Contoh :

```
short n;
short m;
for (n=1; n<=2; n++){
    m=n;
    while (m<=10) {
        System.out.print(m + " ")
        m+=2;
    }
    System.out.println("");
}
```

Hasil

```
run:
1 3 5 7 9
2 4 6 8 10
```

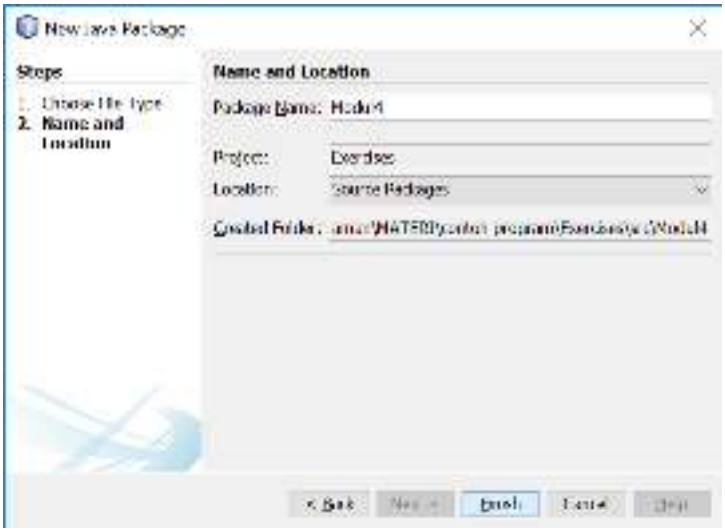
Pembahasan :

Contoh di atas menggunakan 2 buah jenis perulangan yang berbeda yaitu perulangan *while* didalam perulangan *for*. Dalam putaran perulangan *for* yang pertama, perulangan while akan menyelesaikan putaran proses eksekusinya sebelum melanjutkan ke putaran perulangan *for* berikutnya.

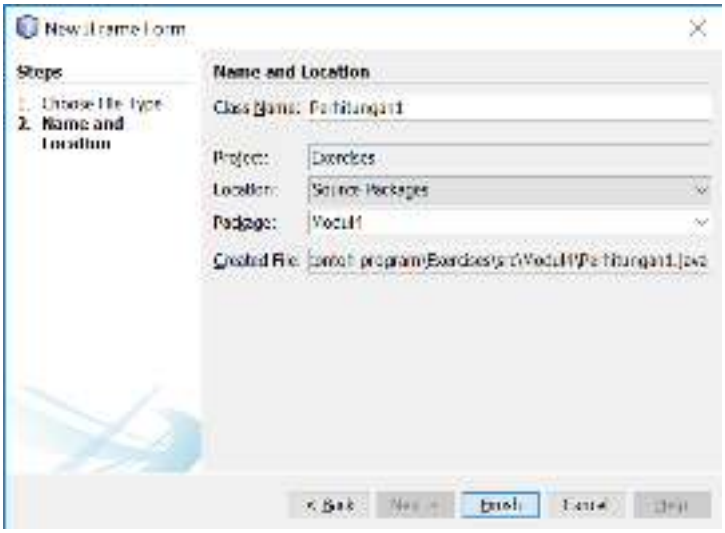
### 4.5 Latihan

Berikut adalah percobaan membuat aplikasi sederhana untuk menampilkan daftar perkalian menggunakan perulangan.

1. Tambahkan paket (*Java Package*) pada projek 'Exercises' dengan langkah-langkah seperti pada percobaan modul 1 (*New Package*). Beri nama paket dengan 'Modul4'.




2. Tambahkan 'JFrame Form' dengan langkah-langkah seperti pada percobaan modul 1 (menambah frame form). Beri nama JFrame Form dengan 'Kasir'.

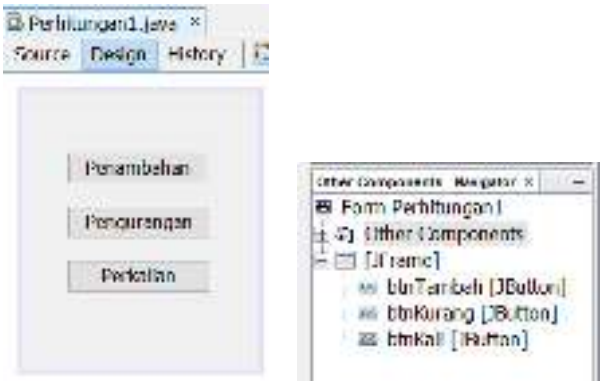



3. Desain form menggunakan komponen Button dari jendela palette seperti berikut ini :

4. Ubah nama variabel dan nilai properti pada masing-masing komponen seperti berikut ini.

| Komponen | Nama Variabel | Properti |
|---|---|---|
| jButton1 | btnTambah | text = Penjumlahan |
| jButton2 | btnKurang | text = Pengurangan |
| jButton3 | btnKali | text = Perkalian |

sehingga menjadi




5. Klik kanan komponen btnTambah pada jendela navigator → Event → Action → klik ‘actionPerformed’. Ketikan kode program dibawah ini

```
private void btnTambahActionPerformed(java.awt.event.ActionEvent
    // TODO add your handling code here:
    short n;
    short m=1;
    String hasil = "";

    for (n=1; n<=10; n++){
        hasil = hasil + m + " + " + n;
        hasil = hasil + " = " + (m+n);
        hasil = hasil + "\n";
    }

    javax.swing.JOptionPane.showMessageDialog(null, hasil);
}
```

6. Klik kanan komponen btnKurang pada jendela navigator → Event → Action → klik 'actionPerformed'. Ketikan kode program dibawah ini

```
private void btnKurangActionPerformed(java.awt.event.ActionEvent
    // TODO add your handling code here:
    short n;
    short m=1;
    String hasil = "";

    for (n=10; n>=1; n--){
        hasil = hasil + n + " - " + m;
        hasil = hasil + " = " + (n-m);
        hasil = hasil + "\n";
    }

    javax.swing.JOptionPane.showMessageDialog(null, hasil);

}
```

7. Klik kanan komponen btnKali pada jendela navigator → Event → Action → klik 'actionPerformed'. Ketikan kode program dibawah ini

```
private void btnKaliActionPerformed(java.awt.event.ActionEvent
    // TODO add your handling code here:
    short n=1;
    short m=2;
    String hasil = "";

    do {
        hasil = hasil + n + " x " + m;
        hasil = hasil + " = " + (m*n);
        hasil = hasil + "\n";

        n++;
    } while (n<=10);

    javax.swing.JOptionPane.showMessageDialog(null, hasil);
}
```

8. Pastikan tidak ada tanda seru merah pada kode program Anda. Selanjutnya, jalankan program dengan cara klik kanan pada file program (frame) pilih Run File atau tekan Shift + F6 pada keyboard.

# MODUL 5
# ARRAY

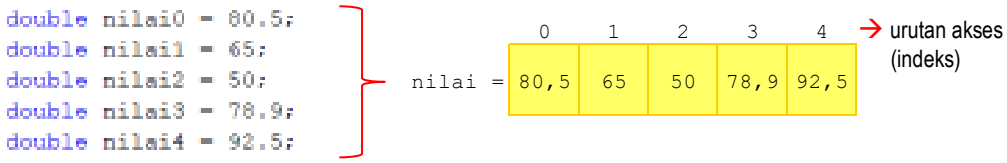
**Pokok Pembahasan :**

- **Array Satu Dimensi**
- **Array Multi Dimensi**
- **Perulangan Array**
- **Class Array**
- **Latihan**

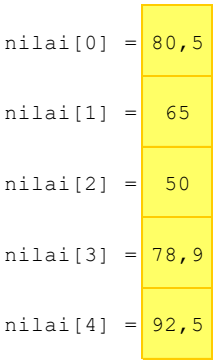
Array atau disebut juga larik merupakan struktur kumpulan variabel secara berurutan yang menyimpan data/nilai (isi variabel) bertipe sama. Satu variabel dalam array disebut sebagai elemen dan urutan akses elemen disebut indeks. Indeks elemen dalam array selalu dimulai dari angka 0. Dalam pemrograman Java, deklarasi array ditandai dengan kurung siku [ ].

### 5.1 Array Satu Dimensi

Array satu dimensi adalah array atau larik dengan satu ruang penyimpanan. Untuk lebih jelas mengenai perbedaan variabel biasa dengan array satu dimensi dapat dilihat pada gambar di bawah ini.

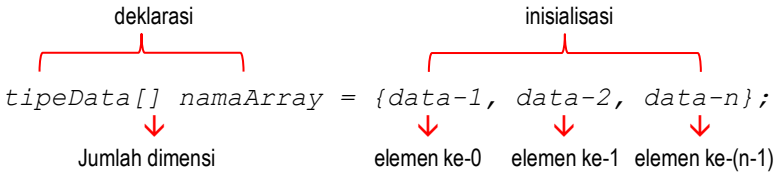



Untuk menyimpan 5 nilai yang berbeda dibutuhkan 5 variabel biasa untuk masing-masing nilai tersebut. Dengan array, kelima nilai tersebut dapat disimpan dalam 1 variabel dengan ruang penyimpanan yang berbeda sehingga seolah-olah hanya menggunakan 1 variabel saja. Untuk mengakses ruang penyimpanannya, serta urutan akses (indeks) setelah nama variabel.

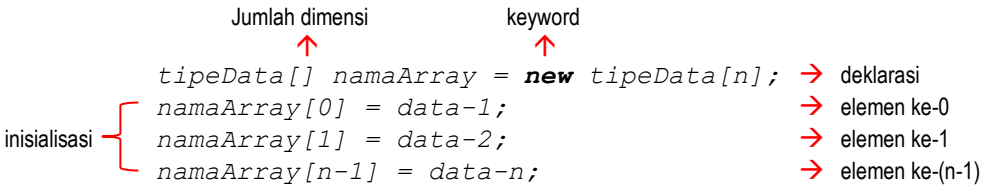



Berikut bentuk deklarasi array satu dimensi.

Bentuk 1 :



Bentuk 2 :




Keterangan :

- *tipeData* disesuaikan dengan jenis data/nilai yang akan disimpan dalam array.
- *namaArray* adalah nama variabel.
- *n* adalah jumlah/banyaknya elemen array.

Pada bentuk 1, data/nilai ditentukan secara langsung (inisialisasi) setelah deklarasi array. Data/nilai satu dengan yang lainnya dipisahkan dengan simbol koma (,). Banyaknya data/nilai yang di-inisialisasi-kan sama dengan banyaknya elemen array yang akan terbentuk. Indeks elemennya dimulai dari angka 0 sampai dengan n-1

Pada bentuk 2, inisialisasi dilakukan setelah proses deklarasi array. Banyaknya data/nilai yang di-inisialisasikan tergantung kepada nilai *n* pada deklarasi array dengan indeks elemennya dimulai dari angka 0 sampai dengan n-1.

Contoh 1:
Berikut adalah contoh array dimensi satu (bentuk 1) :

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

    double[] nilai = {80.5, 65, 50, 78.9, 92.5};

    System.out.println("elemen ke-0 = " + nilai[0]);
    System.out.println("elemen ke-1 = " + nilai[1]);
    System.out.println("elemen ke-2 = " + nilai[2]);
    System.out.println("elemen ke-3 = " + nilai[3]);
    System.out.println("elemen ke-4 = " + nilai[4]);
}
```

Hasil

```
run:
elemen ke-0 = 80.5
elemen ke-1 = 65.0
elemen ke-2 = 50.0
elemen ke-3 = 78.9
elemen ke-4 = 92.5
```

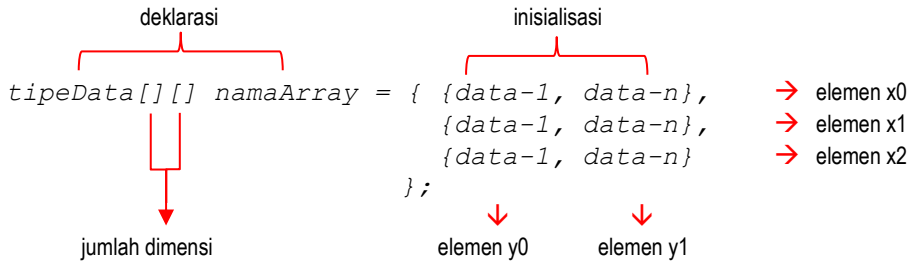
Pembahasan :

Banyaknya data yang di-inisialisasi-kan berjumlah 5 (*n=5*) yaitu {80.5, 65, 50, 78.9, 92.5} dengan urutan akses elemen dari 0 sampai dengan n-1.

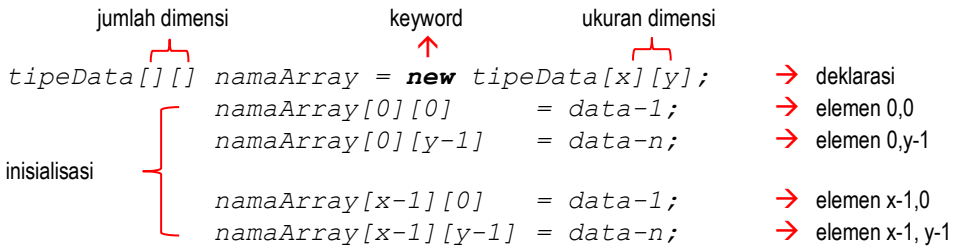
## 5.2 Array Multi Dimensi

Array multi dimensi memiliki lebih dari satu dimensi atau ruang penyimpanan data/nilai. Pada array dua dimensi, ruang penyimpanan berupa matriks atau tabel. Berikut bentuk deklarasi array multi dimensi.

Bentuk 1 :




Bentuk 2 :




Keterangan :

- Jumlah kurung siku menyatakan jumlah dimensi yang digunakan dimana (`[][]`) berarti menggunakan array 2 dimensi.
- Pada bentuk 2, `[x][y]` menyatakan ukuran dimensi yang digunakan.

Contoh 1:

Berikut adalah contoh array multi dimensi (bentuk 1) :

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

    String[][] mhs = { {"1700001", "Tora"},
                       {"1700002", "Tompel"},
                       {"1700003", "Toyo"}
                     };

    System.out.println("NPM \t\t Nama Mahasiswa");

    System.out.println(mhs[0][0] + "\t\t" + mhs[0][1]);
    System.out.println(mhs[1][0] + "\t\t" + mhs[1][1]);
    System.out.println(mhs[2][0] + "\t\t" + mhs[2][1]);
}
```

Hasil

```
run:
NPM             Nama Mahasiswa
1700001         Tora
1700002         Tompel
1700003         Toyo
```

Pembahasan :
Contoh di atas menggunakan array 2 dimensi dengan ukuran 3x2 yaitu 3 baris (0-2) dan 2 kolom (0-1). Array 2 dimensi ditandai dengan kurung siku sebanyak 2 kali. Karakter '\t' akan mencetak 'tab' sebanyak 1x.

### 5.3 Perulangan Array

Perulangan pada array menggunakan pernyataan *for* dan hanya berlaku untuk array satu dimensi. Salah satu pernyataan *for* yang dapat digunakan pada array adalah *for each* dimana akan mengiterasi dan menyimpan nilai setiap elemen array dalam variabel tunggal. Berikut sintaks atau bentuk penulisan pernyataan *for each* pada array.

```
for(tipeData namaVariabel : namaArray){
     kode_program1;
     kode_program2;          } blok kode program
     …;
}
```

Keterangan :

- `tipeData` yang digunakan harus sama dengan tipe pada array.

Contoh 1 :
Pada contoh array satu dimensi di atas,

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

    double[] nilai = {80.5, 65, 50, 78.9, 92.5};

    System.out.println("elemen ke-0 = " + nilai[0]);
    System.out.println("elemen ke-1 = " + nilai[1]);
    System.out.println("elemen ke-2 = " + nilai[2]);
    System.out.println("elemen ke-3 = " + nilai[3]);
    System.out.println("elemen ke-4 = " + nilai[4]);
}
```

dapat diubah menggunakan perulangan array menjadi :

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

    short m = 0;
    double[] nilai = {80.5, 65, 50, 78.9, 92.5};

    for (double n : nilai){
        System.out.println("elemen ke-" + m + " = " + n);
        m++;
    }
}
```

tetap dengan hasil eksekusi (*run*) yang sama.

```
run:
elemen ke-0 = 80.5
elemen ke-1 = 65.0
elemen ke-2 = 50.0
elemen ke-3 = 78.9
elemen ke-4 = 92.5
```

Contoh 2 :

Berikut adalah contoh array dimensi satu (bentuk 2) :

```
public static void main(String[] args){
    short i;
    String[] terbilang = new String[10];
    terbilang[0] = "nol";
    terbilang[1] = "satu";
    terbilang[2] = "dua";
    terbilang[3] = "tiga";
    terbilang[4] = "empat";
    terbilang[5] = "lima";
    terbilang[6] = "enam";
    terbilang[7] = "tujuh";
    terbilang[8] = "delapan";
    terbilang[9] = "sembilan";
```

```
    for(i=0; i<terbilang.length; i++){
        System.out.println(i + " = " + terbilang[i]);
    }
}
```

Hasil

```
run:
0 = nol
1 = satu
2 = dua
3 = tiga
4 = empat
5 = lima
6 = enam
7 = tujuh
8 = delapan
9 = sembilan
```

Pembahasan :

Fungsi `length` pada array menghasilkan nilai banyaknya array yang terbentuk sehingga dapat digunakan pada perulangan *for* sebagai batas akhir perulangan untuk mengakses setiap elemen pada array.

Contoh 3:

Berikut adalah contoh array multi dimensi (bentuk 2) dengan contoh kasus jadwal railink Medan – Kualanamu sesuai tabel di bawah ini.

| MEDAN - KUALANAMU AIRPORT | | |
|---|---|---|
| NO KERETA | KEBERANGKATAN | KETIBAAN |
| U2 | 03:30 | 04:01 |
| U4 | 05:15 | 05:46 |
| U6 | 06:00 | 06:31 |

(sumber : https://www.railink.co.id/index.php/jadwal/en, diakses 1/12/2018)

```
public static void main(String[] args){
    short bar, kol;
    String[][] mdn_kno = new String[3][3];

    mdn_kno[0][0]="U2";
    mdn_kno[0][1]="03:30";
    mdn_kno[0][2]="04:01";
```

```
        mdn_kno[1][0]="U4";
        mdn_kno[1][1]="05:15";
        mdn_kno[1][2]="05:46";

        mdn_kno[2][0]="U6";
        mdn_kno[2][1]="06:00";
        mdn_kno[2][2]="06:31";

        System.out.println("Jadwal Railink Bandara (WIB)");
        System.out.println("No Kereta \t Berangkat \t Tiba");

        for (bar=0; bar < mdn_kno.length; bar++){
            for (kol=0; kol < mdn_kno[bar].length; kol++){
                System.out.print(mdn_kno[bar][kol] + "\t\t");
            }
            System.out.println("");
        }
    }
```

Hasil

```
run:
Jadwal Railink Bandara (WIB)
No Kereta        Berangkat        Tiba
U2               03:30            04:01
U4               05:15            05:46
U6               06:00            06:31
```

Pembahasan :

Contoh di atas menggunakan array 2 dimensi dengan ukuran 3x3 yaitu 3 baris (0-2) dan 3 kolom (0-2). Array 2 dimensi ditandai dengan kurung siku sebanyak 2 kali dan ukuran array dita`new String[3][3];` . Karakter '\t' akan mencetak 'tab' sebanyak 1x.

Perintah `mdn_kno.length;` menghasilkan nilai banyaknya array pada ruang dimensi x dan pe`mdn_kno[bar].length;` menghasilkan nilai banyaknya array pada ruang dimensi y.

### 5.4 Class Array Pada Java

Untuk melakukan operasi lebih lanjut pada array, pemrograman Java telah menyediakan class untuk membantu programer mengolah data/nilai pada array.

### 5.4.1 Class Arrays

Class Arrays merupakan turunan dari `java.util.Arrays`. Class Arrays melakukan operasi terhadap variabel array, yang berarti class ini tidak dapat digunakan tanpa adanya variabel array (baik array satu dimensi maupun array multidimensi). Dengan Class Arrays, programer dapat melakukan operasi terhadap variabel array seperti menyalin (*copying*), mengurutkan (*sorting*), mencari (*searching*) dan lain-lain. Sebelum menggunakan class arrays, terlebih dahulu programer harus menambahkan class tersebut melalui import ( `import java.util.Arrays;` ). Berikut beberapa operasi (*method*) Class Arrays yang akan dibahas.

- *`copyOf(arraySumber, ukuranArray)`* adalah fungsi untuk menyalin array sumber (baik struktur maupun isi) secara keseluruhan ke array tujuan. Contoh :

```
public static void main(String[] args){
    double[] lama = {80.5, 65, 50, 78.9, 92.5};
    double[] baru = Arrays.copyOf(lama, lama.length);

    for (double n : baru){
        System.out.print(n + " ");
    }
}
```

Hasil

```
run:
80.5 65.0 50.0 78.9 92.5
```

*`copyOfRange(arraySumber, indeksAwal, indeksAkhir)`* sama seperti *`copyOf`*, namun dapat ditentukan/dikostumasi ukuran dan isi array sumber ke array tujuan. Contoh :

```
public static void main(String[] args){
    double[] lama = {80.5, 65, 50, 78.9, 92.5};
    double[] baru = Arrays.copyOfRange(lama, 2, lama.length);

    for (double n : baru){
        System.out.print(n + " ");
    }
}
```

Hasil

```
run:
50.0 78.9 92.5
```

- *`sort(arraySumber, indeksAwal, indeksAkhir)`* adalah metode untuk melakukan pengurutan pada seluruh elemen array. Parameter *`indeksAwal`* dan *`indeksAkhir`* bersifat opsional. Contoh :

```
public static void main(String[] args){
    double[] nilai = {80.5, 65, 50, 78.9, 92.5};
    Arrays.sort(nilai);

    for (double n : nilai){
        System.out.print(n + " ");
    }
}
```

Hasil

```
run:
50.0 65.0 78.9 80.5 92.5
```

untuk melakukan pengurutan pada elemen tertentu pada array, dapat menambahkan indeks awal dan indeks akhir dari elemen array yang ingin diurutkan pada pada parameter kedua dan ketiga. Contoh :

```
double[] nilai = {80.5, 65, 50, 78.9, 92.5};
Arrays.sort(nilai,2,4);
```

- `binarySearch(arraySumber, nilaiCari)` adalah fungsi untuk melakukan pencarian biner berdasarkan nilai yang ada didalam array. Fungsi ini menghasilkan nilai bertipe *int* yang menyatakan indeks elemen. Apabila nilai yang dicari tidak ditemukan, fungsi akan menghasilkan nilai lebih kecil 0. Contoh :

```
public static void main(String[] args){
    int indeks;
    double cari = 78.9;
    double[] nilai = {80.5, 65, 50, 78.9, 92.5};
    indeks = Arrays.binarySearch(nilai, cari);

    if (indeks < 0){
        System.out.println(cari + " tidak ditemukan");
    }else{
        System.out.println(cari + " ditemukan pada indeks ke-" + indeks);
    }
}
```

Hasil

```
run:
78.9 ditemukan pada indeks ke-3
```

- `equals(array1, array2)` adalah fungsi untuk melakukan perbandingan terhadap seluruh nilai elemen array. Fungsi ini menghasilkan nilai bertipe *boolean* yang menyatakan `true` apabila seluruh nilai elemen sama dan `false` apabila salah satu nilai elemen tidak sama. Contoh :

```
public static void main(String[] args){
    boolean result;

    char[] huruf1 = {'m', 'k', 'o'};
    char[] huruf2 = {'m', 'k', 'o'};
    char[] huruf3 = {'h', 'm', 'r', 'i'};

    result = Arrays.equals(huruf1, huruf2);
    System.out.println("nilai array huruf1 & huruf2 sama?"+result);

    result = Arrays.equals(huruf1, huruf3);
    System.out.println("nilai array huruf1 & huruf3 sama?"+result);
}
```

Hasil

```
run:
nilai array huruf1 & huruf2 sama?true
nilai array huruf1 & huruf3 sama?false
```

- `fill(arraySumber, nilaiBaru)` merupakan metode (*method*) untuk mengisi nilai kedalam elemen array. Contoh 1:

```
public static void main(String[] args){
    double[] hasil = {80.5, 65, 50, 78.9, 92.5};
    Arrays.fill(hasil, 55);

    System.out.println("nilai elemen array:");
    for (double n : hasil){
        System.out.print(n + " ");
    }
}
```

Hasil

```
run:
nilai elemen array:
55.0 55.0 55.0 55.0 55.0
```

Terdapat 5 elemen dalam array 'hasil' yang telah memiliki nilai awal. Dengan *method fill(hasil, 55)* pada class Arrays, seluruh nilai pada setiap elemen array 'hasil' diubah menjadi '55'.

Contoh 2:

```
public static void main(String[] args){
    double[] hasil = {80.5, 65, 50, 78.9, 92.5};
    Arrays.fill(hasil, 2, 4, 55);

    System.out.println("nilai elemen array:");
    for (double n : hasil){
        System.out.print(n + " ");
    }
}
```

Hasil

```
run:
nilai elemen array:
80.5 65.0 55.0 55.0 92.5
```

Untuk mengubah sebagian nilai elemen dalam array, tentukan parameter ke-2 yaitu indeks awal dan parameter ke3 yaitu banyaknya elemen array.

### 5.4.2 Class ArrayList

Class ArrayList yang dibahas pada modul ini hanya untuk array satu dimensi. Pengolahan data/nilai array menggunakan Class ArrayList menjadi lebih dinamis yang berarti bahwa pengolahan data/nilai tidak hanya dilakukan melalui rancangan kode program (sebelum program dijalankan/*running*) namun juga dapat dilakukan pada saat program dijalankan/*running*. Beberapa perbedaan Class Arrays dan Class ArrayList antara lain adalah :

1. Arrays tidak dideklarasikan sedangkan ArrayList harus dideklarasikan menjadi sebuah objek.
2. Arrays mengolah nilai elemen array dari variabel array sedangkan ArrayList dapat mengolah nilai elemen array secara langsung.
3. Data/nilai pada ArrayList bukan merupakan data/nilai bertipe primitif melainkan objek sehingga ArrayList dapat menampung data/nilai dengan berbagai tipe yang berbeda-beda.

Class ArrayList juga merupakan turunan dari `java.util` yaitu `java.util.ArrayList`. Oleh karena itu, sebelum menggunakan ArrayList terlebih dahulu programer harus menambahkan class tersebut melalui import ( `import java.util.ArrayList;`. Adapun sintaks atau bentuk deklarasi objek Class ArrayList adalah sebagai berikut.

```
ArrayList namaObjek = new ArrayList();
```

Untuk lebih memahami pengolahan data/nilai array menggunakan Class ArrayList, berikut akan dijelaskan beberapa operasi (*method*) pada Class ArrayList.

- *`add(indeks, nilai)`* merupakan metode (*method*) untuk menambah elemen array dan nilainya. Parameter *`indeks`* bersifat opsional. Contoh :

```
public static void main(String[] args){
    ArrayList Mhs = new ArrayList();

    Mhs.add("Tompel");  → tipe String
    Mhs.add('L');       → tipe char
    Mhs.add(3.95);      → tipe double
    Mhs.add(true);      → tipe boolean

    System.out.println("ArrayList = "+ Mhs.toString());

    Mhs.add(2, 2017);
    System.out.println("ArrayList - add(2, 2017) = "+ Mhs.toString());
}
```

Hasil

```
run:
ArrayList = [Tompel, L, 3.95, true]
ArrayList - add(2, 2017) = [Tompel, L, 2017, 3.95, true]
```

Keterangan :
Jenis data/nilai yang ditambahkan (*add*) pada objek ArrayList berbeda-beda. Perintah `Mhs.add(2, 2017);` berarti menambah nilai '2017' pada indeks ke-2.

- `set(indeks, nilai)` merupakan metode (*method*) untuk merubah nilai elemen pada indeks tertentu. Contoh :

```
public static void main(String[] args){
    ArrayList Mhs = new ArrayList();

    Mhs.add("Tompel");
    Mhs.add('L');
    Mhs.add(3.95);
    Mhs.add(true);

    System.out.println("ArrayList = "+ Mhs.toString());

    Mhs.add(2, 2017);
    System.out.println("ArrayList - add(2, 2017) = "+ Mhs.toString());

    Mhs.set(3, 3.33);
    System.out.println("ArrayList - set(3, 3.33) = "+ Mhs.toString());
}
```

Hasil

```
run:
ArrayList = [Tompel, L, 3.95, true]
ArrayList - add(2, 2017) = [Tompel, L, 2017, 3.95, true]
ArrayList - set(3, 3.33) = [Tompel, L, 2017, 3.33, true]
```

Keterangan :
Perintah `Mhs.set(3, 3.33);` merubah data/nilai pada indeks elemen ke-3 dengan nilai '3.33'.

- `remove(indeks)` merupakan metode (*method*) untuk menghapus elemen pada indeks tertentu. Contoh :

```
public static void main(String[] args){
    ArrayList Mhs = new ArrayList();

    Mhs.add("Tompel");
    Mhs.add('L');
    Mhs.add(3.95);
    Mhs.add(true);

    System.out.println("ArrayList = "+ Mhs.toString());
```

```
    Mhs.add(2, 2017);
    System.out.println("ArrayList - add(2, 2017) = "+ Mhs.toString());

    Mhs.set(3, 3.33);
    System.out.println("ArrayList - set(3, 3.33) = "+ Mhs.toString());

    Mhs.remove(1);
    System.out.println("ArrayList - remove(1) = "+ Mhs.toString());
}
```

Hasil

```
run:
ArrayList = [Tompel, L, 3.95, true]
ArrayList - add(2, 2017) = [Tompel, L, 2017, 3.95, true]
ArrayList - set(3, 3.33) = [Tompel, L, 2017, 3.33, true]
ArrayList - remove(1) = [Tompel, 2017, 3.33, true]
```

Keterangan :

Perintah `Mhs.remove(1);` menghapus elemen pada indeks ke-1.

- `clear()` merupakan metode (*method*) untuk menghapus seluruh elemen pada objek ArrayList. Contoh :

```
public static void main(String[] args){
    ArrayList Mhs = new ArrayList();

    Mhs.add("Tompel");
    Mhs.add('L');
    Mhs.add(3.95);
    Mhs.add(true);

    System.out.println("ArrayList = "+ Mhs.toString());

    Mhs.clear();
    System.out.println("ArrayList - clear = "+ Mhs.toString());
}
```

Hasil

```
run:
ArrayList = [Tompel, L, 3.95, true]
ArrayList - clear = []
```

- `size()` merupakan metode (*method*) yang menghasilkan jumlah/banyak elemen pada objek ArrayList. Contoh :

```
public static void main(String[] args){
    ArrayList Mhs = new ArrayList();

    Mhs.add("Tompel");
    Mhs.add('L');
    Mhs.add(3.95);
    Mhs.add(true);

    System.out.println("Banyak elemen = "+ Mhs.size());
    System.out.println("Nilai elemen = "+ Mhs.toString());
}
```

Hasil

```
run:
Banyak elemen = 4
Nilai elemen = [Tompel, L, 3.95, true]
```

- `get(index)` merupakan metode (*method*) untuk mendapatkan data/nilai dari elemen ArrayList berdasarkan nomor indeks. Contoh :

```
public static void main(String[] args){
    ArrayList Mhs = new ArrayList();

    Mhs.add("Tompel");
    Mhs.add('L');
    Mhs.add(3.95);
    Mhs.add(true);

    for (short x=0; x<Mhs.size(); x++){
        System.out.println("elemen " + x + " = " + Mhs.get(x).toString());
    }
}
```

Hasil

```
run:
elemen-0 = Tompel
elemen-1 = L
elemen-2 = 3.95
elemen-3 = true
```

- `indexOf(arrayValue)` merupakan metode (*method*) untuk mendapatkan nomor indeks dari elemen ArrayList berdasarkan data/nilai. Contoh :

```
public static void main(String[] args){
    ArrayList Mhs = new ArrayList();

    Mhs.add("Tompel");
    Mhs.add('L');
    Mhs.add(3.95);
    Mhs.add(true);

    System.out.print("Nilai 'L' terdapat pada indeks ke ");
    System.out.println(Mhs.indexOf('L'));
}
```

Hasil

```
run:
Nilai 'L' terdapat pada indeks ke 1
```

- `contains(arrayValue)` merupakan metode (*method*) untuk memeriksa keberadaan data/nilai tertentu (`arrayValue`) dalam objek ArrayList. *Method* ini akan menghasilkan nilai *true* jika ditemukan dan menghasilkan nilai *false* jika data/nilai tidak ditemukan. Contoh :

```
public static void main(String[] args){
    ArrayList Mhs = new ArrayList();

    Mhs.add("Tompel");
    Mhs.add('L');
    Mhs.add(3.95);
    Mhs.add(true);

    System.out.println("Ada nilai 'L'? " + Mhs.contains('L'));
    System.out.println("Ada nilai 'Tom?' " + Mhs.contains("Tom"));
}
```

Hasil

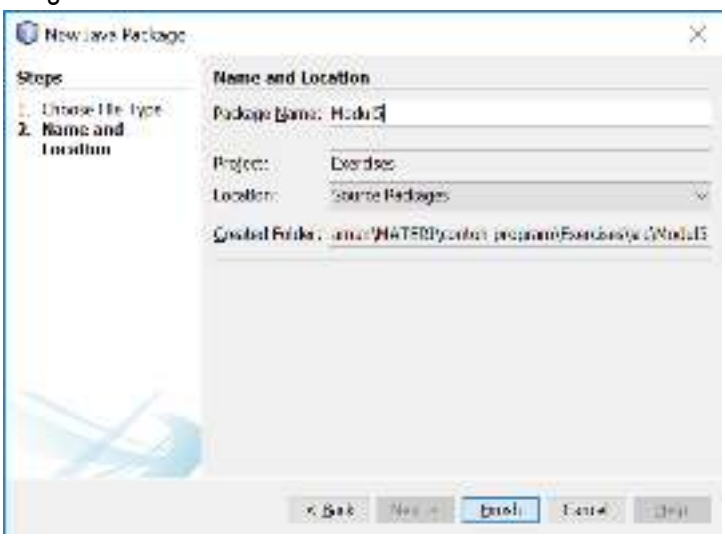
```
run:
Ada nilai 'L'? true
Ada nilai 'Tom?' false
```

- `isEmpty()` merupakan metode (*method*) untuk memeriksa ada atau tidaknya data/nilai dalam objek ArrayList. *Method* ini akan menghasilkan nilai *true* jika tidak ada data/nilai dan menghasilkan nilai *false* jika ada data/nilai.

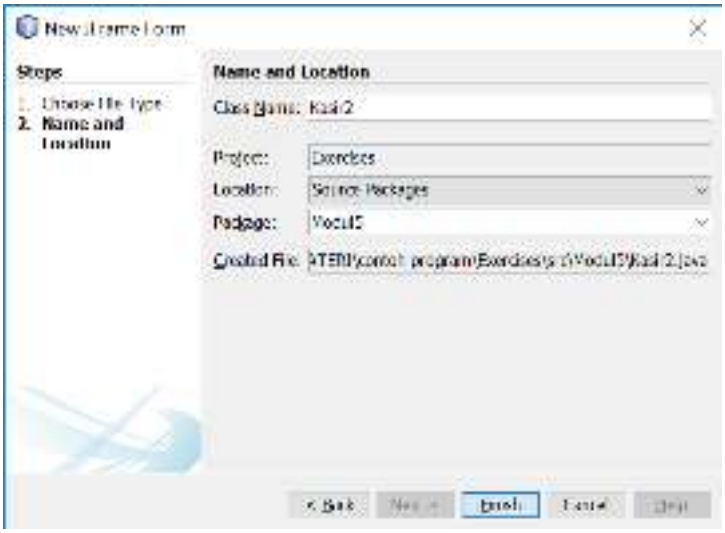
## 5.5 Latihan

Berikut adalah percobaan membuat aplikasi sederhana untuk menampilkan daftar perkalian menggunakan perulangan.

1. Tambahkan paket (*Java Package*) pada projek 'Exercises' dengan langkah-langkah seperti pada percobaan modul 1 (*New Package*). Beri nama paket dengan 'Modul5'.

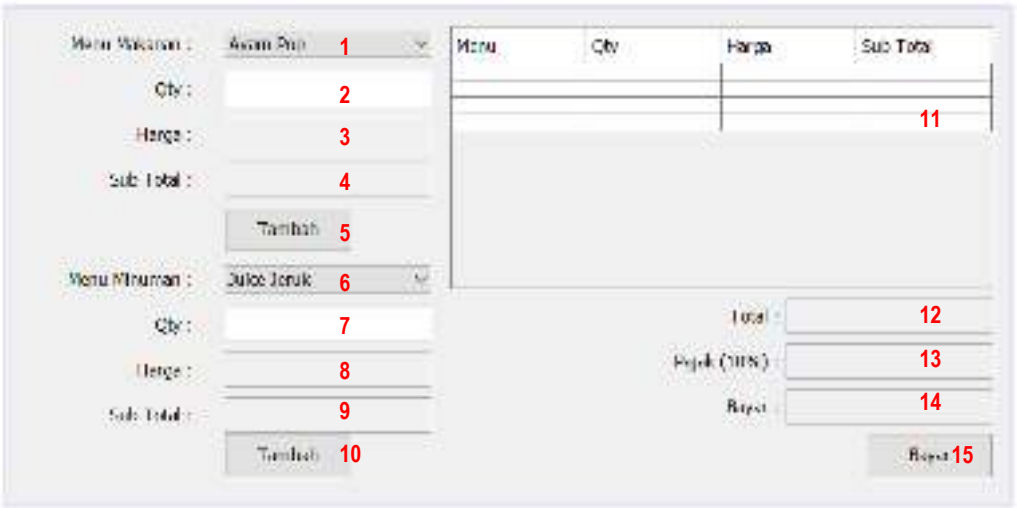


2. Tambahkan ‘JFrame Form’ dengan langkah-langkah seperti pada percobaan modul 1 (menambah frame form) pada paket ‘Modul5’. Beri nama JFrame Form dengan ‘Kasir2’.



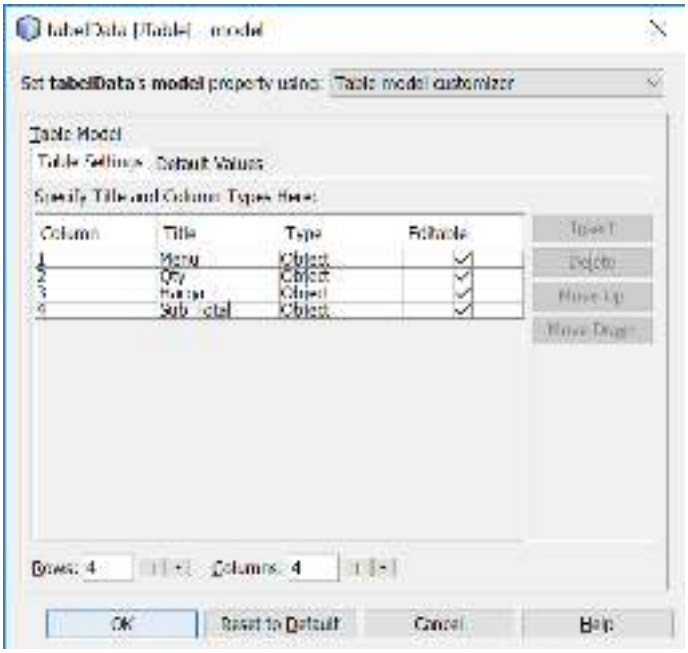

3. Masukkan komponen ‘Panel’ dari palette swing container kedalam form dan atur ukurannya sesuai dengan ukuran form.

4. Masukkan komponen ‘Label’, ‘Text Field’, ‘Combo Box’, ‘Button’ dan ‘Table’ dari palette swing controls kedalam komponen ‘Panel’. Susun posisi masing-masing komponen seperti pada gambar di bawah ini.

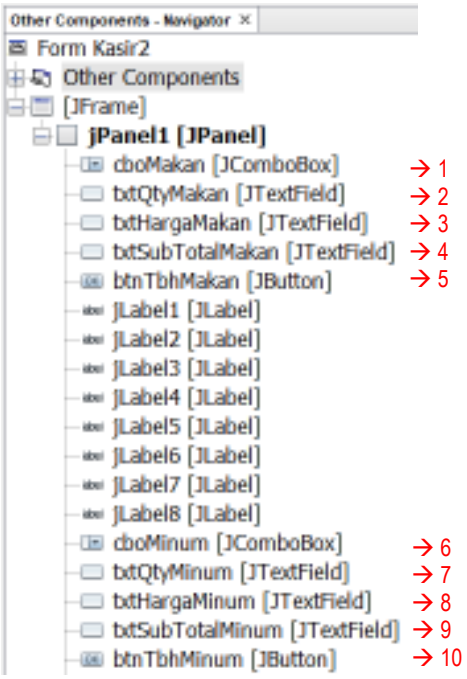


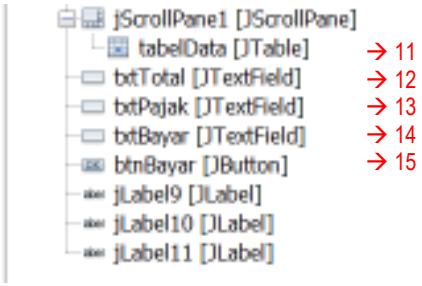

Catatan!
Komponen ‘Table’ merupakan komponen yang digunakan untuk menampilkan data dari objek yang lain.

5. Untuk mengubah nama-nama kolom pada komponen jTable1, klik kanan pada komponen 'Tabel' (atau pada jendela navigator jTable1) → Properties. Klik tombol ... pada properti model. *Double click* pada kolom 'Title' dan ubah nama kolom sesuai gambar berikut ini. Klik tombol 'OK'.

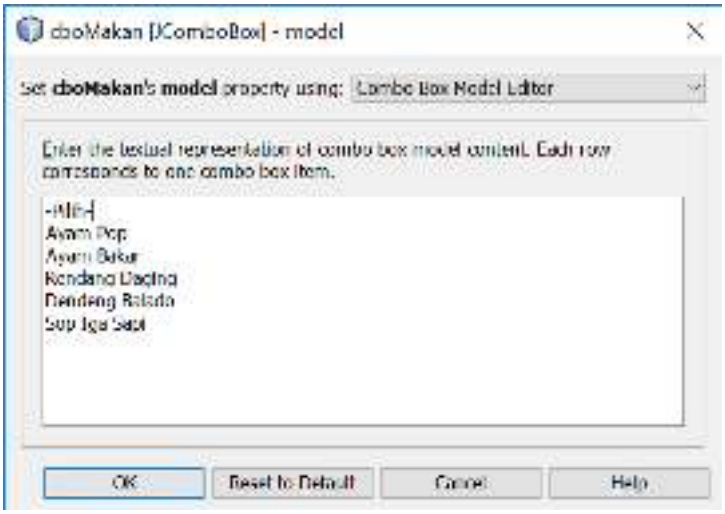



6. Ubah nama variabel dari beberapa komponen sesuai dengan nomor komponen pada poin 4 di atas.

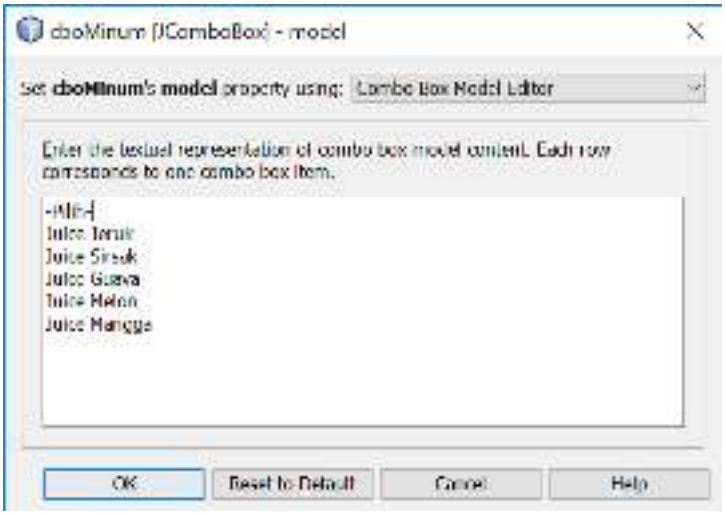





7. Klik kanan pada 'cboMakan' → Properties. Klik tombol ... pada model kemudian isi data sesuai gambar berikut ini.




8. Klik kanan pada 'cboMinum' → Properties. Klik tombol ... pada model kemudian isi data sesuai gambar berikut ini.



9. Ubah properti ‘editable’ menjadi *false* pada komponen ‘txtHargaMakan’, ‘txtSubTotalMakan’, ‘txtHargaMinum’, ‘txtSubTotalMinum’, ‘txtTotal’, ‘txtPajak’, dan ‘txtBayar’.

10. Klik tab ‘Source’ kemudian ketikkan kode program berikut ini untuk memasukkan (*import*) paket Java yang dibutuhkan.

```
6   package Modul5;
7
8   import java.util.ArrayList;
9   import javax.swing.JOptionPane;
10  import javax.swing.table.DefaultTableModel;
11
12
```

11. Ketikkan deklarasi objek ArrayList seperti di bawah ini agar dapat digunakan didalam kelas.

```
13  public class Kasir2 extends javax.swing.JFrame {
14
15      public ArrayList menu = new ArrayList();
16      public ArrayList harga = new ArrayList();
17      public ArrayList qty = new ArrayList();
18      public ArrayList subtotal = new ArrayList();
```

12. Ketikkan *method* ‘cekKosong’ berikut ini.

```
24      private boolean cekKosong(boolean makan){
25          boolean hasil=false;
26
27          if (makan==true){
28              if (txtQtyMakan.getText().isEmpty()==true){
29                  JOptionPane.showMessageDialog(null, "Qty makanan harus diisi!");
30                  hasil=true;
31              }
32          }else{
33              if (txtQtyMinum.getText().isEmpty()==true){
34                  JOptionPane.showMessageDialog(null, "Qty minuman harus diisi!");
35                  hasil=true;
36              }
37          }
38          return hasil;
39      }
```

*Method* ‘cekKosong’ digunakan untuk memeriksa komponen entri data yang masih kosong. Apabila terdapat komponen entri data yang masih kosong maka akan menampilkan pesan peringatan.

13. Ketikkan *method* 'cekDataTabel' berikut ini.

```
41      private boolean cekDataTabel(String data){
42          boolean hasil=menu.contains(data);
43          if (hasil==true)
44              JOptionPane.showMessageDialog(null, "Menu sudah ditambahkan!");
45          return hasil;
46      }
```

*Method* 'cekDataTabel' digunakan untuk memeriksa keberadaan nilai/data yang sudah ada pada objek ArrayList. Apabila terdapat nilai/data yang sudah ada didalam objek ArrayList maka akan menampilkan pesan peringatan dan nilai/data tersebut tidak ditambahkan kedalam objek ArrayList.

14. Ketikkan *method* 'setHargaMakanan' berikut ini.

```
48      private void setHargaMakanan(){
49          int harga=0;
50          int pil = cboMakan.getSelectedIndex();
51          switch(pil){
52              case 1:
53                  harga=25000;
54                  break;
55              case 2:
56                  harga=25000;
57                  break;
58              case 3:
59                  harga=35000;
60                  break;
61              case 4:
62                  harga=35000;
63                  break;
64              case 5:
65                  harga=40000;
66                  break;
67          }
68          txtHargaMakan.setText(""+harga);
69          try{
70              int qty = Integer.parseInt(txtQtyMakan.getText());
71              txtSubTotalMakan.setText(""+(harga*qty));
72          } catch (NumberFormatException e){
73              txtSubTotalMakan.setText("0");
74          }
75      }
```

*Method* 'setHargaMakanan' digunakan untuk menentukan harga makanan yang dipilih dari komponen 'cboMakan' berdasarkan nomor urut (indeks) nilai/data terpilih. Harga akan ditampilkan pada komponen 'txtHargaMakan' dan menghitung harga sub total makanan.

15. Ketikkan *method* 'setHargaMinuman' berikut ini.

```
77      private void setHargaMinuman(){
            int harga;
79          if (cboMinum.getSelectedIndex()==0)
80              harga=0;
81          else
82              harga=20000;
83
84          txtHargaMinum.setText(""+harga);
85          try{
                int qty = Integer.parseInt(txtQtyMinum.getText());
87              txtSubTotalMinum.setText(""+(harga*qty));
88          } catch (NumberFormatException e){
89              txtSubTotalMinum.setText("0");
90          }
91      }
```

*Method* 'setHargaMinuman' digunakan untuk menentukan harga minuman yang dipilih dari komponen 'cboMinum'. Harga akan ditampilkan pada komponen 'txtHargaMinum' dan menghitung harga sub total minuman.

16. Ketikkan *method* 'setTabel' berikut ini.

```
93      private void setTabel(){
94          int total=0;
            double pajak=0;
            int bayar=0;
97
98          DefaultTableModel tbl = new DefaultTableModel();
99          tbl.addColumn("Menu");
100         tbl.addColumn("Qty");
101         tbl.addColumn("Harga");
102         tbl.addColumn("Sub Total");
103
104         for(int n = 0; n<menu.size(); n++){
105             total = total + Integer.parseInt(subtotal.get(n).toString());
106
107             tbl.addRow(new Object[]{
108             menu.get(n),
109             qty.get(n),
110             harga.get(n),
111             subtotal.get(n)
112             });
113         }
114         pajak = total * 0.1;
115         bayar = (int) (total + pajak);
116
117         tabelData.setModel(tbl);
118         txtTotal.setText(""+total);
119         txtPajak.setText(""+pajak);
120         txtBayar.setText(""+bayar);
121     }
```

*Method* 'setTabel' digunakan untuk menampilkan seluruh nilai/data dari objek ArrayList melalui komponen 'Table', menghitung nilai pajak yang dikenakan,

menghitung seluruh biaya yang harus dibayarkan, dan menampilkannya pada komponen ‘txtTotal’, ‘txtPajak’, dan ‘txtBayar’.

17. Klik kanan pada komponen ‘btnTbhMakan’ → Events → Action → actionPerformed. Ketikkan kode program berikut ini.

```
422     private void btnTbhMakanActionPerformed(java.awt.event.ActionEvent ev
423         if (cekKosong(true) == false){
424             String menuItem = cboMakan.getSelectedItem().toString();
425             if (cekDataTabel(menuItem) == false){
426                 int hrg = Integer.parseInt(txtHargaMakan.getText());
427                 int kuantiti = Integer.parseInt(txtQtyMakan.getText());
428                 txtSubTotalMakan.setText(""+(hrg*kuantiti));
429 
430                 menu.add(menuItem);
431                 harga.add(txtHargaMakan.getText());
432                 qty.add(txtQtyMakan.getText());
433                 subtotal.add(txtSubTotalMakan.getText());
434 
435                 setTabel();
436             }
437         }
438     }
```

Komponen Button ‘btnTbhMakan’ akan menjalankan proses yang berkenaan dengan menu makanan antara lain menambah item makanan terpilih kedalam tabel, menghitung nilai sub total, total, pajak, dan nilai biaya yang harus dibayarkan.

18. Klik kanan pada komponen ‘btnTbhMinum’ → Events → Action → actionPerformed. Ketikkan kode program berikut ini.

```
444     private void btnTbhMinumActionPerformed(java.awt.event.ActionEvent e
445         if (cekKosong(false) == false){
446             String menuItem = cboMinum.getSelectedItem().toString();
447             if (cekDataTabel(menuItem) == false){
448                 int hrg = Integer.parseInt(txtHargaMinum.getText());
449                 int kuantiti = Integer.parseInt(txtQtyMinum.getText());
450                 txtSubTotalMinum.setText(""+(hrg*kuantiti));
451 
452                 menu.add(menuItem);
453                 harga.add(txtHargaMinum.getText());
454                 qty.add(txtQtyMinum.getText());
455                 subtotal.add(txtSubTotalMinum.getText());
456 
457                 setTabel();
458             }
459         }
460     }
```

Komponen Button ‘btnTbhMinum’ akan menjalankan proses yang berkenaan dengan menu minuman antara lain menambah item minuman terpilih kedalam

tabel, menghitung nilai sub total, total, pajak, dan nilai biaya yang harus dibayarkan.

19. Klik kanan pada komponen 'btnBayar' → Events → Action → actionPerformed. Ketikkan kode program berikut ini.

```
    private void btnBayarActionPerformed(java.awt.event.ActionEvent evt) {
        JOptionPane.showMessageDialog(null, "Terima Kasih atas Kunjungannya!");
        cboMakan.setSelectedIndex(0);
        txtQtyMakan.setText("0");
        txtHargaMakan.setText("0");
        txtsubtotalmakan.setText("0");

        cboMinum.setSelectedIndex(0);
        txtQtyMinum.setText("0");
        txtHargaMinum.setText("0");
        txtSubtotalMinum.setText("0");

        menu.clear();
        harga.clear();
        qty.clear();
        subtotal.clear();

        setTabel();
    }
```

20. Klik kanan pada komponen 'cboMakan' → Events → Action → actionPerformed. Ketikkan kode program berikut ini.

```
    private void cboMakanActionPerformed(java.awt.event.A
        setHargaMakanan();
    }
```

21. Klik kanan pada komponen 'cboMinum' → Events → Action → actionPerformed. Ketikkan kode program berikut ini.

```
    private void cboMinumActionPerformed(java.awt.ev
        setHargaMinuman();
    }
```

22. Klik kanan pada komponen 'txtQtyMakan' → Events → Key → keyTyped. Ketikkan kode program berikut ini.

```
    private void txtQtyMakanKeyTyped(java.awt.event.
        if(!Character.isDigit(evt.getKeyChar())){
            evt.consume();
        }
    }
```

Perintah ini digunakan hanya untuk menerima nilai masukan dari komponen txtQtyMakan berupa karakter angka saja.

23. Klik kanan pada komponen ‘txtQtyMinum’ → Events → Key → keyTyped. Ketikkan kode program berikut ini.

```
504     private void txtQtyMinumKeyTyped(java.awt.event.
505         if(!Character.isDigit(evt.getKeyChar())){
506             evt.consume();
507         }
508     }
```

24. Pastikan tidak ada tanda seru merah pada kode program Anda. Selanjutnya, jalankan program dengan cara klik kanan pada file program (frame) pilih Run File atau tekan Shift + F6 pada keyboard.

# MODUL 6
## STRUKTUR METODE (*METHOD*)

**Pokok Pembahasan :**

- **Metode Tanpa Nilai**
- **Metode Dengan Nilai**
- **Modifier**
- **Latihan**

Metode (*method*) merupakan sekumpulan atau blok kode program yang menjadi sub kode program. Kode program pada metode ditulis seluruhnya (modifier, tipe nilai kembalian, nama, parameter, dan isi metode atau blok kode program) oleh programer diluar dari kode program utama. Oleh karena itu untuk menjalankan sebuah metode, nama metode harus ditulis didalam kode program utamanya. Melalui penggunaan metode, programer dapat mengurangi penulisan kode program yang sama atau berulang.

### 6.1 Metode Tanpa Nilai

Metode (*method*) tanpa nilai berarti tidak dapat menampung nilai kembalian (*return value*) dari metode itu sendiri. Metode tanpa nilai juga disebut sebagai prosedur (*procedure*). Sintaks atau bentuk penulisan metode tanpa nilai pada Java adalah sebagai berikut :

```
modifier void methodName(parameter){
    kode_program1;
    kode_program2;
    kode_program3;
}
```

blok kode program (isi metode)

Keterangan :

- ***void*** merupakan ciri utama dari metode tanpa nilai yang harus ditulis sebelum nama metode.
- `parameter` berupa variabel untuk menampung nilai dari luar metode yang dapat digunakan didalam metode. Penggunaan paramater bersifat opsional.

### 6.2 Metode Dengan Nilai

Metode (*method*) dengan nilai adalah metode yang dapat menampung nilai kembalian (*return value*) dari metode itu sendiri. Oleh karena itu, metode ini harus menggunakan tipe nilai (tipe data) kembaliannya yaitu tipe data primitif. Metode ini juga disebut sebagai fungsi (*function*). Sintaks atau bentuk penulisan metode dengan nilai pada Java adalah sebagai berikut :

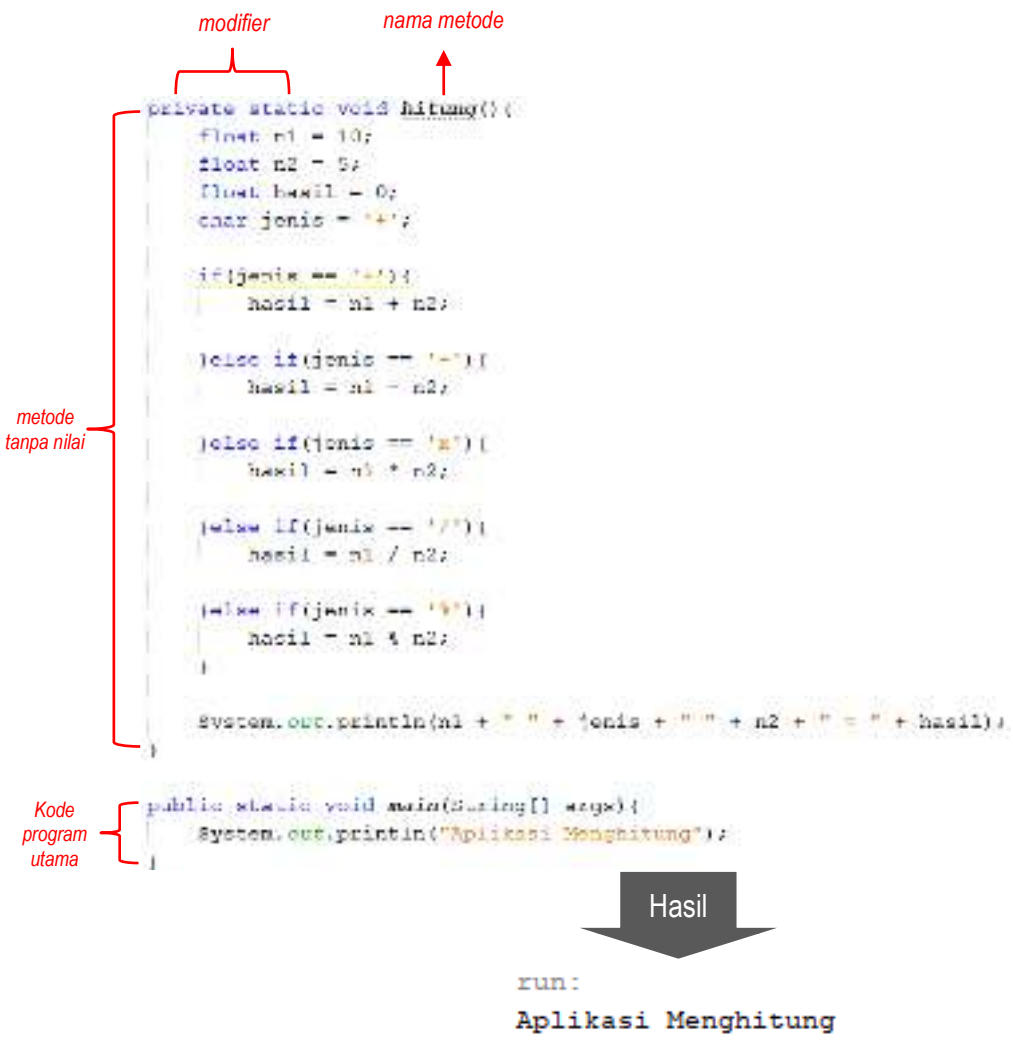
```
modifier returnValueTipe methodName(parameter){
    kode_program1;
    kode_program2;
    kode_program3;
    return value;
}
```

blok kode program (isi metode)

Keterangan :

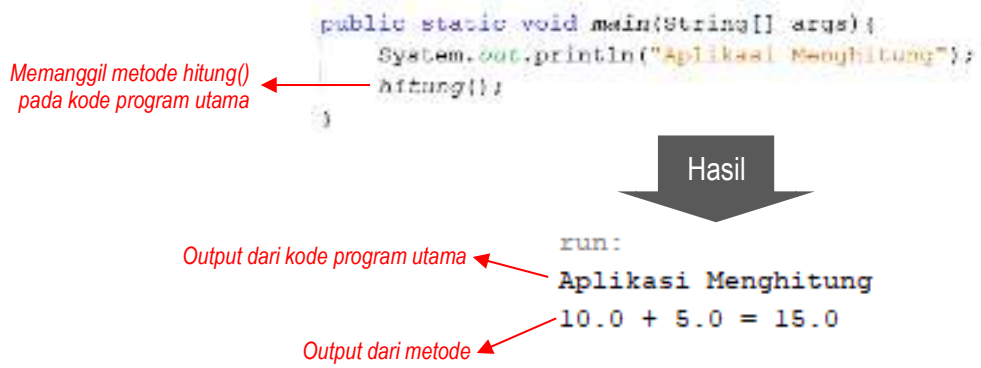
- `returnValueTipe` merupakan tipe data primitif yang harus ditulis sebelum nama metode.
- `return` merupakan pernyataan yang wajib digunakan untuk mengembalikan nilai/data dari metode. Pernyataan `return` diikuti dengan nilai/data sesuai dengan `returnValueTipe`.
- `parameter` berupa variabel untuk menampung nilai dari luar metode yang dapat digunakan didalam metode. Penggunaan paramater bersifat opsional.

Contoh metode tanpa nilai - I (tidak menggunakan parameter) :

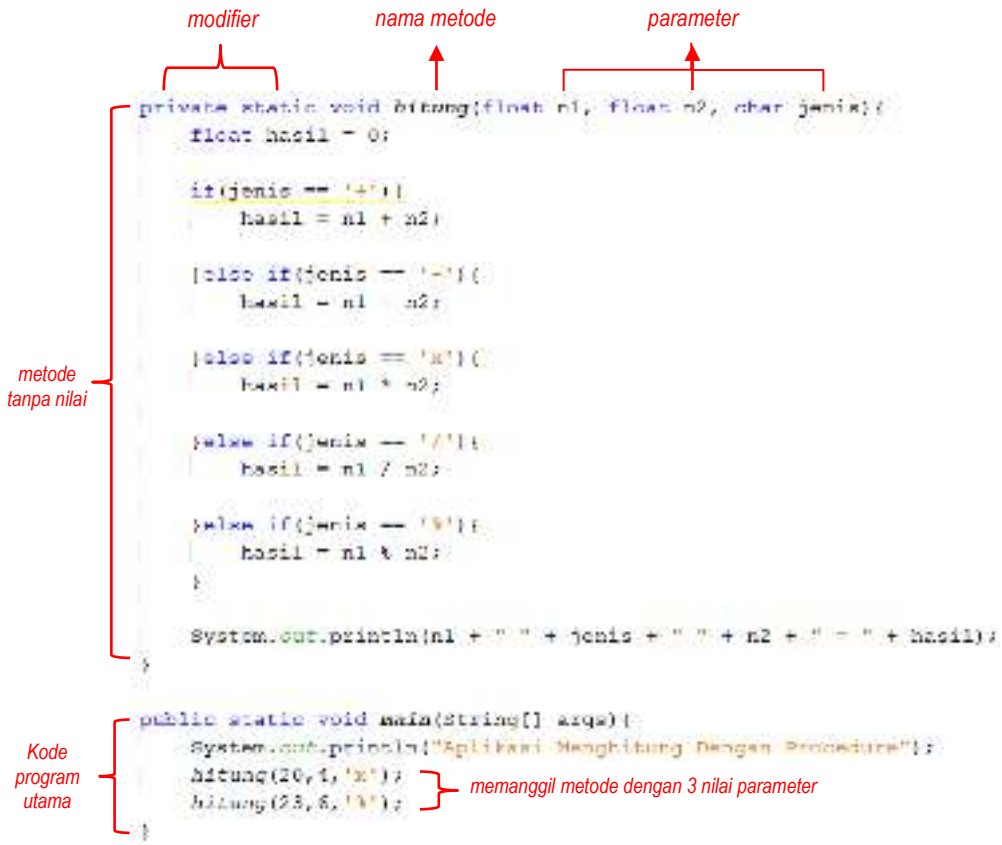


Pembahasan :

Pada kode program utama, hanya terdapat perintah untuk mencetak kalimat “Aplikasi Menghitung” (`System.out.println("Aplikasi Menghitung");`) sehingga metode ‘*hitung()*’ tidak dijalankan/dieksekusi. Untuk mengeksekusi metode, tulis nama metode tersebut pada kode program utama.




Contoh metode tanpa nilai - II (menggunakan parameter):



Hasil

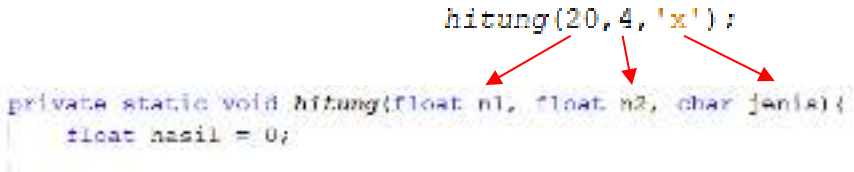
```
run:
Aplikasi Menghitung Dengan Procedure
20.0 x 4.0 = 80.0
23.0 % 6.0 = 5.0
```

Pembahasan :

Metode '*hitung*' di atas menggunakan 3 parameter ( `float n1, float n2, char jenis`) dengan tipe data float dan char. Tipe data yang digunakan pada parameter disesuaikan dengan nilai/data yang akan diberikan/dimasukkan.

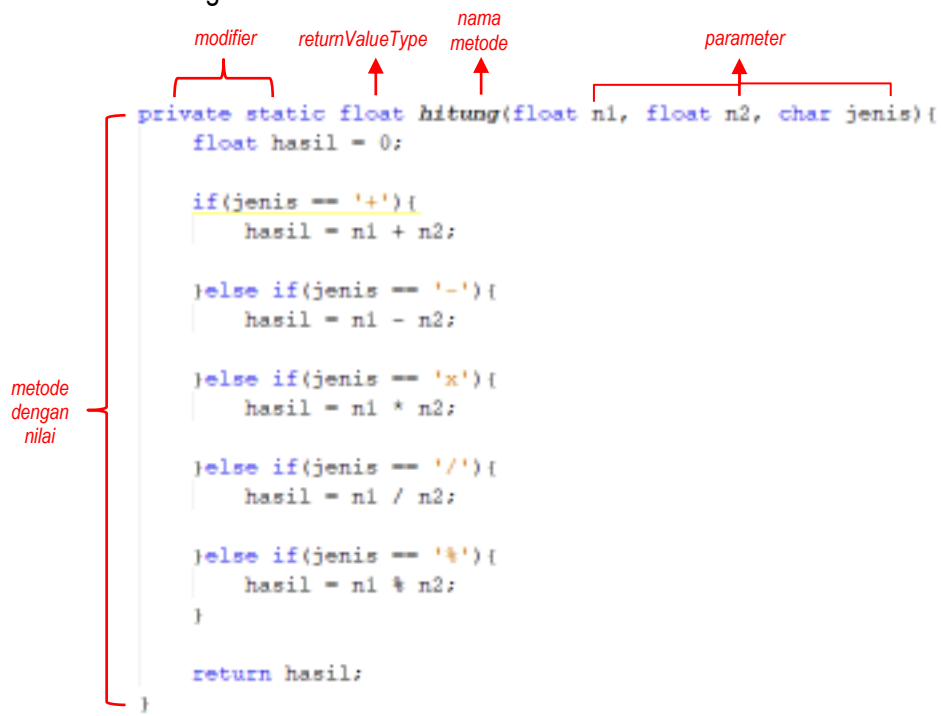
```
hitung(20,4,'x');
```

Nilai/data yang diberikan/dimasukkan kedalam metode ditentukan pada saat pemanggilan metode.




Nilai/data akan ditampung oleh masing-masing parameter. Ketiga parameter tersebut juga dapat digunakan didalam metode itu sendiri sebagai pengolahan nilai/data.

Contoh metode dengan nilai :



kode program utama

```
public static void main(String[] args){
    System.out.println("Aplikasi Menghitung Dengan Function");

    float nilai = hitung(20,4,'/');
    System.out.println("20 / 4 = " + nilai);

    System.out.println("30 - 10 = " + hitung(30,10,'-'));
}
```

Hasil

```
run:
Aplikasi Menghitung Dengan Function
20 / 4 = 5.0
30 - 10 = 20.0
```

Pembahasan :

Ciri metode dengan nilai adalah penggunaan tipe nilai kembalian sebelum nama metode (`private static float hitung(float n1, float n2, char jenis)`) dan pernyataan *return* (`return hasil;`).

Metode dengan nilai dapat dipanggil didalam variabel. Nilai kembalian dari metode akan disimpan pada variabel (`float nilai = hitung(20,4,'/');`) yang memanggil metode tersebut. Oleh karena itu, tipe data variabel harus sesuai dengan tipe nilai kembalian.

Selain itu, metode dengan nilai juga dapat dipanggil didalam suatu perintah (`System.out.println("30 - 10 = " + hitung(30,10,'-'));`) sehingga nilai kembali dari metode akan langsung dieksekusi oleh perintah tersebut.

## 6.3 Modifier

Modifier pada pemrograman Java terbagi menjadi dua kriteria yaitu access modifier dan non-access modifier. Modifier merupakan pengaturan level akses untuk membatasi ruang lingkup penggunaan dari variabel, method, *data member*, class, atau konstruktor. *Keyword* yang termasuk non-access modifier dapat dilihat pada tabel 6.1.

**Tabel 6.1** Non-Access Modifier

| ***keyword*** | ***keterangan*** |
|---|---|
| static | Modifier static dapat diterapkan pada variabel atau *method*. Pada variabel, modifier ini akan mempertahankan nilai yang ada (nilai terakhir) pada variabel atau *method*. |

| | |
|---|---|
| | Pada *method*, modifier ini akan mempertahankan *method* yang sudah ada dalam setiap *instance* sebuah class. |
| final | *Keyword* final dapat digunakan pada variabel, *method*, atau class. Pada variabel, modifier ini membuat nilai tetap konstan. Pada method, modifier ini mencegah terjadinya override atau modifikasi dalam sebuah subclass. |
| abstract | Modifier abstract dapat digunakan pada *method* dan class. Jika class berisi method abstaract maka kelas juga harus dideklarasikan sebagai abstract. |
| synchronized | Java menyediakan cara untuk membuat thread dan mensinkronkan tugas (thread) menggunakan blok yang tersinkronisasi. Modifier synchronized biasa digunakan pada program multi-threading. Modifier ini mensinkronkan blok thread pada satu waktu. |
| transient | Modifier ini digunakan pada serialisasi. Variabel yang menggunakan *keyword* transient dapat mengabaikan nilai asli dari variabel dan menyimpan nilai default dari tipe data variabel tersebut. |
| volatile | Modifier volatile digunakan untuk mengizinkan JVM mengetahui thread yang sedang mengakses variabel untuk menggabungkan salinan variabel itu sendiri dengan salinan utama pada memori. |

*Keyword* yang termasuk kedalam access modifer dapat dilihat pada tabel 6.2.

**Tabel 6.2** Access Modifier

| ***keyword*** | ***keterangan*** |
|---|---|
| <default> | Modifier <default> tidak memiliki keyword sehingga tidak ada yang dituliskan secara eksplisit pada deklarasi modifier ini. Modifier <default> dapat diakses oleh class lainnya dalam satu paket (*package*) |
| private | *Keyword* private harus ditulis secara eksplisit pada deklarasi. Modifier private tidak dapat diterapkan pada class dan interface. *Method* atau *data member* yang menggunakan modifier private hanya dapat diakses didalam class dimana tempat pendeklarasiannya. |
| protected | *Keyword* protected harus ditulis secara eksplisit pada deklarasi. Modifier protected tidak dapat diterapkan pada class dan interface. Variabel, method dan konstruktor yang menggunakan modifier protected dapat diakses pada paket (*package*) yang sama atau sub class pada paket yang berbeda. |

| | |
|---|---|
| public | Keyword public harus ditulis secara eksplisit pada deklarasi. Variabel, method, konstruktor, class dan lain sebagainya yang menggunakan modifier public dapat diakses diseluruh ruang lingkup kerja Java. |

Access modifier akan dibahas lebih lanjut pada materi pemrograman berorientasi objek.

## 6.4 Latihan

### 6.4.1 Aplikasi Warna

Aplikasi Warna pada percobaan modul 3, terdapat beberapa kode program yang berulang-ulang dengan nilai yang berbeda satu dengan yang lainnya seperti pada potongan kode program di bawah ini.

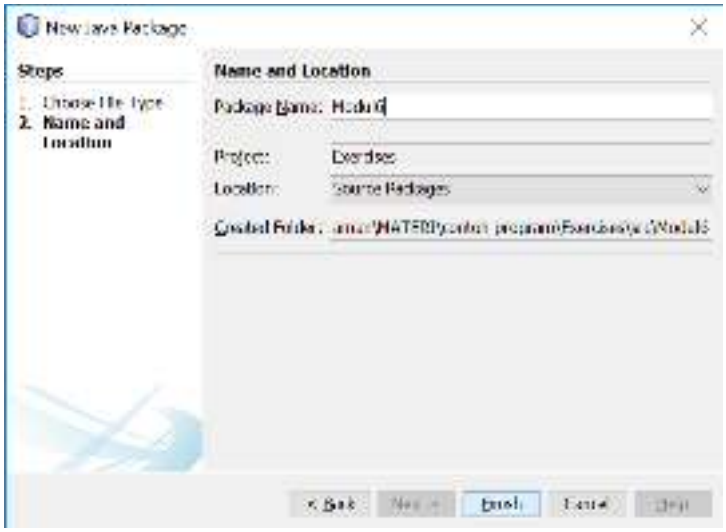
```
switch (cboWarna.getSelectedIndex()){
    case 0:
        jLabel1.setForeground(Color.red);
        jLabel2.setForeground(Color.red);
        rbLatar.setForeground(Color.red);
        rbTeks.setForeground(Color.red);
        break;

    case 1:
        jLabel1.setForeground(Color.green);
        jLabel2.setForeground(Color.green);
        rbLatar.setForeground(Color.green);
        rbTeks.setForeground(Color.green);
        break;

    case 2:
        jLabel1.setForeground(Color.blue);
        jLabel2.setForeground(Color.blue);
        rbLatar.setForeground(Color.blue);
        rbTeks.setForeground(Color.blue);
        break;

    case 3:
        jLabel1.setForeground(Color.cyan);
        jLabel2.setForeground(Color.cyan);
        rbLatar.setForeground(Color.cyan);
        rbTeks.setForeground(Color.cyan);
        break;
```

Pola antara ‘case 0’ dengan case-case yang lainnya memiliki kode program yang sama yaitu :

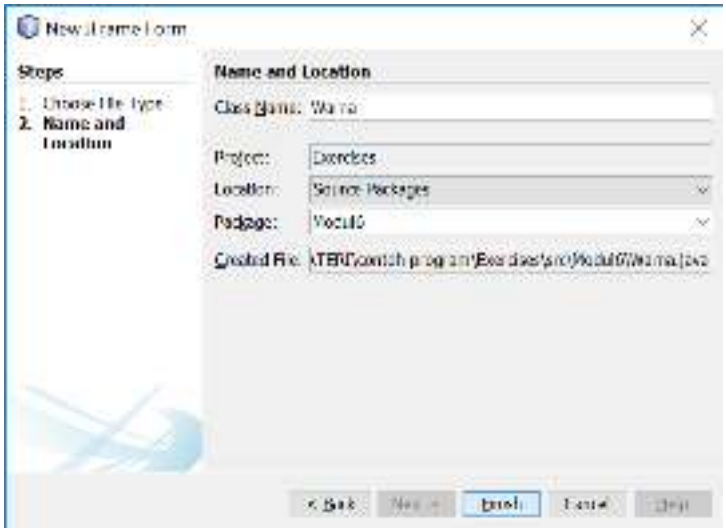
```
jLabel1.setForeground
jLabel2.setForeground
rbLatar.setForeground
rbTeks.setForeground
```

namun dengan nilai warna yang berbeda-beda sesuai dengan ketentuan warna. Oleh karena itu, kode program di atas dapat disederhanakan menggunakan *method*. 4 baris kode program dapat dipisahkan kedalam sub kode program menggunakan *method* dengan satu parameter yaitu nilai warna sehingga didalam kode program pernyataan switch, programer hanya memanggil nama *method* dan memberikan nilai warna sesuai pilihan pada *case_selector*. Berikut adalah percobaan membuat aplikasi warna menggunakan *method*.

1. Tambahkan paket (*Java Package*) pada projek 'Exercises' dengan langkah-langkah seperti pada percobaan modul 1 (*New Package*). Beri nama paket dengan 'Modul6'.




2. Tambahkan 'JFrame Form' dengan langkah-langkah seperti pada percobaan modul 1 (menambah frame form). Beri nama JFrame Form dengan 'Warna'.




3. Desainlah JFrame Form seperti aplikasi warna pada percobaan modul 3.

4. Klik tab 'Source' , kemudian ketikkan perintah *import* seperti berikut ini.

```
1
2   package Modul6;
3
4   import java.awt.Color;
5
```

5. Ketikkan method 'setWarna' dalam class Warna seperti berikut ini.

```
6   public class Warna extends javax.swing.JFrame {
7
8       public Warna() {
9           initComponents();
10      }
11
12      private void setWarna(Color warna){
13          jLabel1.setForeground(warna);
14          jLabel2.setForeground(warna);
15          rbLatar.setForeground(warna);
16          rbTeks.setForeground(warna);
17      }
```

*Method* 'setWarna' digunakan untuk merubah warna latar depan (warna teks) dari komponen jLabel1, jLabel2, rbLatar, dan rbTeks. *Method* ini menggunakan satu parameter untuk melewatkan nilai warna yang dikirim/dimasukkan dari luar *method* dan diterapkan untuk merubah warna komponen.

6. Ketikkan kode program berikut pada komponen 'btnUbah' dengan Event → Action → actionPerformed.

```
private void btnUbahActionPerformed(java.awt.event.ActionEvent.evt) {

    if (rbtstar.isSelected()){
        if (cboWarna.getSelectedItem()== "Merah"){
            jPanel1.setBackground(Color.red);

        }else if (cboWarna.getSelectedItem()== "Hijau"){
            jPanel1.setBackground(Color.green);

        }else if (cboWarna.getSelectedItem()== "Biru"){
            jPanel1.setBackground(Color.blue);

        }else if (cboWarna.getSelectedItem()== "Cyan"){
            jPanel1.setBackground(Color.cyan);

        }else if (cboWarna.getSelectedItem()== "Magenta"){
            jPanel1.setBackground(Color.magenta);

        }else if (cboWarna.getSelectedItem()== "Kuning"){
            jPanel1.setBackground(Color.yellow);

        }else if (cboWarna.getSelectedItem()== "Hitam"){
            jPanel1.setBackground(Color.black);

        }else if (cboWarna.getSelectedItem()== "Putih"){
            jPanel1.setBackground(Color.white);

        }

    }else if (rbricks.isselected()){
        switch (cboWarna.getSelectedIndex()){
            case 0:
                setWarna(Color.red);
                break;

            case 1:
                setWarna(Color.green);
                break;

            case 2:
                setWarna(Color.blue);
                break;

            case 3:
                setWarna(Color.cyan);
                break;

            case 4:
                setWarna(Color.magenta);
                break;

            case 5:
                setWarna(Color.yellow);
                break;

            case 6:
                setWarna(Color.black);
                break;

            case 7:
                setWarna(Color.white);
                break;
        }
    }
}
```

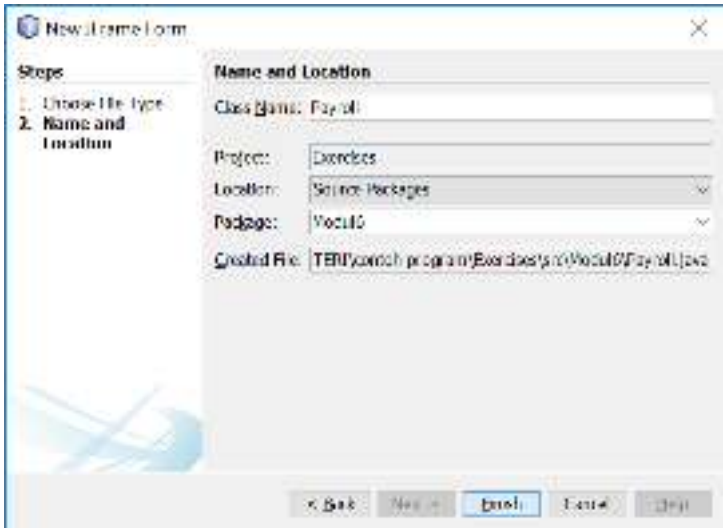
*Method* ‘setWarna’ dipanggil dalam komponen btnUbah pada event actionPerformed. Nilai warna diberikan/dimasukkan ke *method* ‘setWarna’ sesuai dengan pilihan dari komponen cboWarna.

7. Pastikan tidak ada tanda seru merah pada kode program Anda. Selanjutnya, jalankan program dengan cara klik kanan pada file program (frame) pilih Run File atau tekan Shift + F6 pada keyboard.

### 6.4.2 Aplikasi Payroll

Berikut adalah percobaan membuat aplikasi sederhana untuk menampilkan daftar perkalian menggunakan perulangan.

1. Tambahkan ‘JFrame Form’ dengan langkah-langkah seperti pada percobaan modul 1 (menambah frame form). Beri nama JFrame Form dengan ‘Payroll’.




2. Desain form menggunakan komponen Desktop Pane, Label, Text Field, Combo Box, Radio Button, Button Group, dan Button dari jendela palette. Susun posisi komponen pada form dan ubah nilai properti pada Label dan Text Field seperti pada gambar di bawah ini.

3. Ubah nama variabel dan nilai properti komponen seperti berikut ini.

| Komponen | Nama Variabel | Properties |
|---|---|---|
| buttonGroup1 | grupStatus | - |
| jTextField1 | txtnip | text = <kosong> |
| jTextField2 | txtnama | text = <kosong> |
| jComboBox1 | cmbjabatan | model = Direktur, Manager, Kabag, Sekretaris, Karyawan, Office Boy |
| jRadioButton1 | rbmenikah | text = Menikah |
| | | buttonGroup = grupStatus |
| jRadioButton2 | rbbelummenikah | text = Belum Menikah |
| | | buttonGroup = grupStatus |
| jTextField3 | txtjumlahanak | text = <kosong> |
| jTextField4 | txtgajipokok | text = <kosong> |
| jTextField5 | txttunjanganistri | text = <kosong> |
| jTextField6 | txttunjangananak | text = <kosong> |
| jTextField7 | txtgajikotor | text = <kosong> |
| jTextField8 | txtpajak | text = <kosong> |
| jTextField9 | txtgajibersih | text = <kosong> |
| jButton1 | btnhapus | text = Hapus |
| jButton2 | btnhitung | text = Hitung |

4. Klik tab 'Source'

, kemudian tulis deklarasi variabel seperti berikut ini.

```
11  public class Payroll extends javax.swing.JFrame {
12      int gajipokok, gajikotor, pajak, gajibersih;
13      int tunjanganistri, tunjangananak, jumlahanak;
14      String t_istri, t_anak;
15      String s_gajikotor, s_pajak, s_gajibersih;
```

5. Ketikkan kode program berikut didalam konstruktor class.

```
19      public Payroll() {
            setTitle("Form Perhitungan Gaji Pegawai");
21          initComponents();
22      }
```

6. Selanjutnya, ketikkan method ‘Gaji’ sesuai kode program berikut ini.

```
24  private void Gaji(){
25          int pilihan=cmbjabatan.getSelectedIndex();
26          switch(pilihan){
27              case 0:
28                  txtgajipokok.setText("7500000");
29              break;
30              case 1:
31                  txtgajipokok.setText("5000000");
32              break;
33              case 2:
34                  txtgajipokok.setText("3000000");
35              break;
36              case 3:
37                  txtgajipokok.setText("2000000");
38              break;
39              case 4:
40                  txtgajipokok.setText("1200000");
41              break;
42              case 5:
43                  txtgajipokok.setText("750000");
44              break;
45          }
46      }
```

7. Ketikkan method ‘TunjanganAnak’ sesuai kode program berikut ini.

```
47
48      private void TunjanganAnak(){
49          gajipokok = Integer.valueOf(txtgajipokok.getText());
50          jumlahanak = Integer.valueOf(txtjumlahanak.getText());
51            if(jumlahanak <= 2){
52                    tunjangananak= ((15*gajipokok)/100) * jumlahanak;
53                }else if(jumlahanak > 2){
54                    tunjangananak=((15*gajipokok)/100) * 2;
55                }else{
56                    tunjangananak=0;
57                }
58                t_anak = String.valueOf(tunjangananak);
59                txttunjangananak.setText(t_anak);
60            }
```

8. Ketikkan method ‘Menikah’ sesuai kode program berikut ini.

```
62      private void Menikah(){
63            txttunjanganistri.setEnabled(true);
64            txttunjangananak.setEnabled(true);
65            txtjumlahanak.setEnabled(true);
66            gajipokok = Integer.valueOf(txtgajipokok.getText());
67            tunjanganistri=(20*gajipokok)/100;
68            t_istri=String.valueOf(tunjanganistri);
69            txttunjanganistri.setText(t_istri);
70            TunjanganAnak();
71        }
```

9. Ketikkan method ‘BelumMenikah’ sesuai kode program berikut ini.

```
74      private void BelumMenikah(){
75            tunjanganistri=0;
76            tunjangananak=0;
77            jumlahanak=0;
78            txttunjanganistri.setText("0");
79            txttunjangananak.setText("0");
80            txtjumlahanak.setText("0");
81            txttunjanganistri.setEnabled(false);
82            txttunjangananak.setEnabled(false);
83            txtjumlahanak.setEnabled(false);
84        }
```

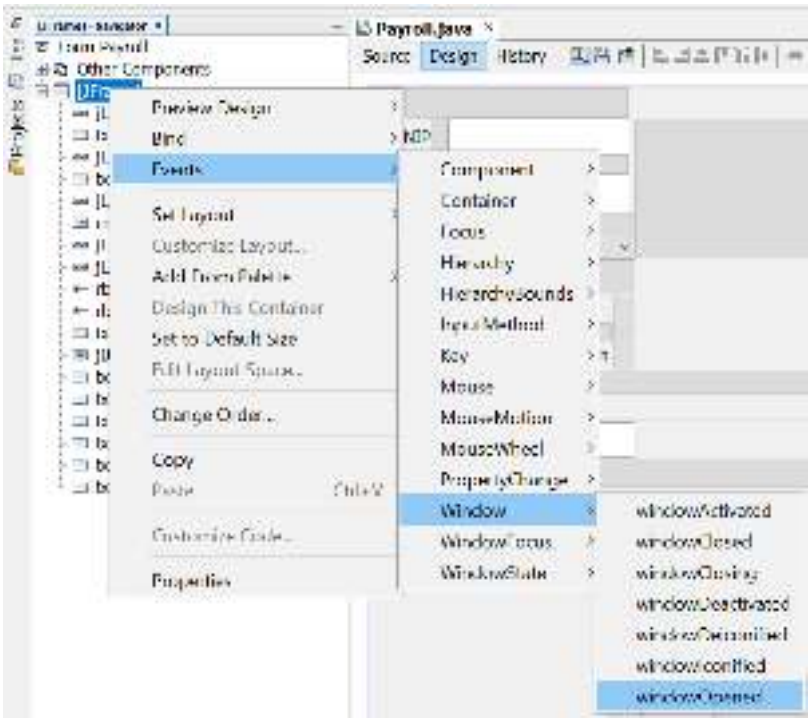
10. Ketikkan method ‘Hitung’ sesuai kode program berikut ini.

11. Ketikkan method ‘Hapus’ sesuai kode program berikut ini.

```
98      private void Hapus(){
99          gajipokok=0;
100         gajikotor=0;
101         pajak=0;
102         gajibersih=0;
103         tunjanganistri=0;
104         tunjangananak=0;
105         jumlahanak=0;
106         txtgajipokok.setText("0");
107         txttunjanganistri.setText("0");
108         txtjumlahanak.setText("0");
109         txttunjangananak.setText("0");
110         txtgajikotor.setText("0");
111         txtpajak.setText("0");
112         txtgajibersih.setText("0");
113     }
```

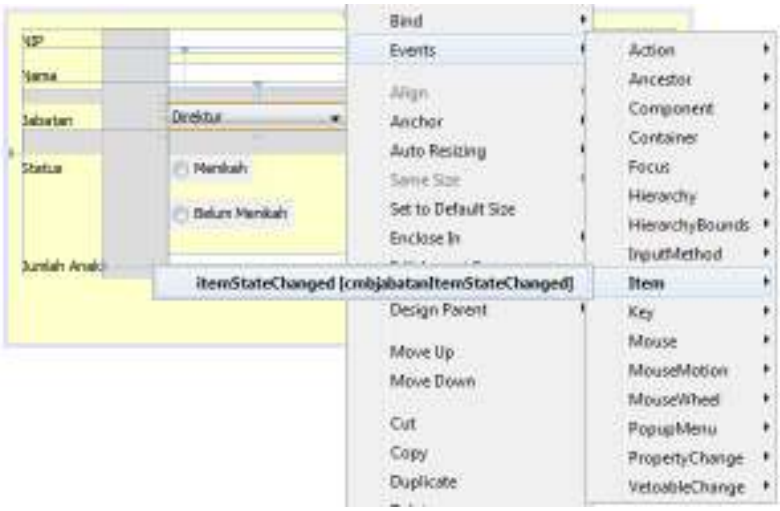
12. Selanjutnya kembali lagi ke tab Desain.

13. Pada panel navigator, klik kanan pada JFrame → Events → Windows → windowOpened.




Ketikkan kode program : BelumMenikah();

14. Klik kanan pada ‘cmbjabatan’, kemudian pilih Events → Item → itemStateChanged.



kemudian ketikkan kode berikut ini:

```
349    private void cmbjabatanItemStateChanged(java.awt.event.ItemEv
350        // TODO add your handling code here:
351        Kapus();
352        Guji();
353        if(rbmenikah.isSelected()==true){
354            Menikah();
355        }else{
356            BelumMenikah();
357        }
358    }
```

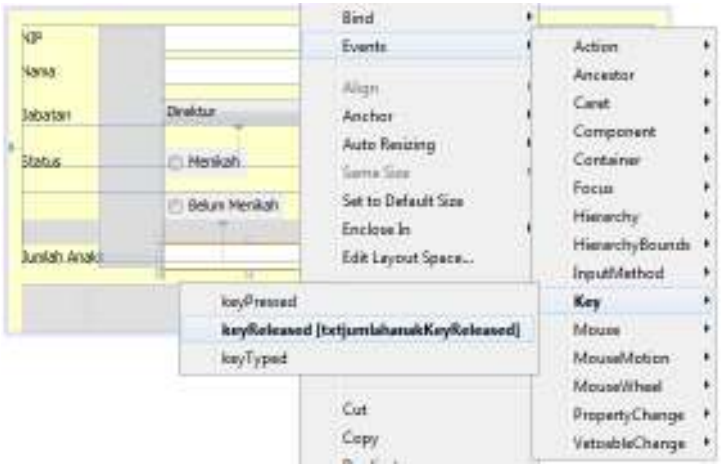
15. Klik kanan 'rbmenikah' kemudian pilih Events → Action → actionPerformed. Ketikkan kode program berikut ini.

```
370    private void rbmenikahActionPerformed(java.awt.event.ActionEv
371        // TODO add your handling code here:
372        Menikah();
373    }
```

16. Klik kanan 'rbbelummenikah' kemudian pilih Events → Action → actionPerformed. Ketikkan kode program berikut ini.

```
375    private void rbbelummenikahActionPerformed(java.awt.event.Ac
376        // TODO add your handling code here:
377        BelumMenikah();
378    }
```

17. Klik kanan pada 'txtjumlahanak' kemudian pilih Events → Key → keyRealeased.



kemudian ketikkan kode berikut ini:

```
424     private void txtjumlahanakKeyReleased(java.awt.event.
425         // TODO add your handling code here:
426         TunjanganAnak();
427     }
```

18. Klik kanan pada 'btnhitung' kemudian pilih Events → Action → actionPerformed. Ketikkan kode program berikut ini.

```
401     private void btnhitungActionPerformed(java.awt.event.
402         // TODO add your handling code here:
403         Hitung();
404     }
```

19. Klik kanan pada 'btnhapus' kemudian pilih Events → Action → actionPerformed. Ketikkan kode program berikut ini.

```
406     private void btnhapusActionPerformed(java.awt.event.
407         // TODO add your handling code here:
408         Hapus();
409     }
```

20. Compile (F9) dan Run Program melalui menu Run → Run File (Shift + F6).

21. Isikan NIP, Nama, dan Pilih Jabatan, Status, Jika sudah Menikah isikan Jumlah anak. Setelah itu klik tombol Hitung maka total gaji akan ditampilkan.

22. Klik Tombol Hapus untuk mengosongkan semua TextField menjadi nilai 0.

# MODUL 7
# PENGOLAH DATA

**Pokok Pembahasan :**

- **Data Karakter**
- **Data String**
- **Data Tanggal dan Waktu**
- **Latihan**

## 7.1 Data Karakter

Tipe data char pada pemrograman Java digunakan untuk merepresentasikan sebuah nilai karakter tunggal. Nilai char ditandai dengan tanda kutip tunggal seperti contoh dibawah ini.

```
char huruf = 'M';
char angka = '2';
```

Tidak hanya huruf, angka, dan simbol yang dapat digunakan pada data char namun juga nilai unicode dari karakter ASCII (*American Standart Code for Information Interchange*) seperti

```
char huruf = '\u004D';
char angka = '\u0032';
```

Nilai char yang diawali dengan karakter *backslash* (\) disebut dengan *escape sequence*. Hal ini menjadi pengecualian khusus bagi nilai char yang hanya dapat menyimpan nilai tunggal dimana nilai `'\u004D'` tidak dianggap sebagai nilai String melainkan tetap nilai tunggal karena nilai `u004D` merupakan nilai unicode untuk karakter 'M'. Dalam pemrograman Java terdapat karakter *escape sequence* yang umum digunakan dan dapat dilihat pada tabel di bawah ini.

**Tabel 7.1** Escape Sequences

| ***Escape Sequence*** | ***Keterangan*** | ***Unicode*** |
|---|---|---|
| \b | Backspace | \u0008 |
| \t | Tab | \u0009 |
| \n | Baris baru | \u000A |
| \f | Form Feed | \u000C |
| \r | Carriage Return | \u000D |
| \' | Kutip tunggal | \u0027 |
| \" | Kutip ganda | \u0022 |
| \\ | Backslash | \u005C |

Contoh:

```
public static void main (String[] args){
    char huruf = '\u004D';
    char angka = '\u0032';

    System.out.println("Menggunakan unicode dan escape sequence");
    System.out.println(huruf + "\n \t \"" + angka + "\"");
}
```

Hasil

```
run:
Menggunakan unicode dan escape sequence
M
        "2"
```

Untuk mengolah data karakter lebih lanjut, pemrograman Java telah menyediakan Class Character. Berikut beberapa method yang terdapat dalam Class Character.

- `isLetter(charValue)` merupakan fungsi untuk mengetahui apakah nilai char adalah huruf. Fungsi menghasilkan nilai *true* jika nilai char adalah huruf dan sebaliknya.
- `isDigit(charValue)` merupakan fungsi untuk mengetahui apakah nilai char adalah angka (digit). Fungsi menghasilkan nilai *true* jika nilai char adalah angka dan sebaliknya.
- `isUpperCase(charValue)` merupakan fungsi untuk mengetahui apakah nilai char adalah huruf besar (kapital). Fungsi menghasilkan nilai *true* jika nilai char adalah huruf besar (kapital) dan sebaliknya.
- `toUpperCase(charValue)` merupakan fungsi untuk merubah nilai char menjadi huruf besar (kapital).
- `toLowerCase(charValue)` merupakan fungsi untuk merubah nilai char menjadi huruf kecil.

Contoh :

```
public static void main (String[] args){
    System.out.println(Character.isLetter('E'));
    System.out.println(Character.isUpperCase('K'));
    System.out.println(Character.isLowerCase('O'));
    System.out.println(Character.isDigit('7'));
    System.out.println(Character.toLowerCase('R'));
    System.out.println(Character.toUpperCase('n'));
}
```

Hasil

```
run:
true
true
false
true
r
N
```

## 7.2 Data String

String merupakan kumpulan karakter (baik huruf, angka, maupun simbol) yang berurutan. Dalam pemrograman Java, String termasuk tipe data referensi (sebuah class) karena mengolah tipe data primitif yaitu *char*. Pada semua bahasa pemrograman terdapat fungsi untuk memanipulasi data String. Oleh karena itu, pemrograman Java menyediakan sebuah class khusus untuk mengolah atau memanipulasi kata atau kalimat. Berikut beberapa method pada Class String yang akan dibahas.

- `charAt(indeks)` adalah fungsi untuk mendapatkan/menghasilkan karakter dalam sebuah String berdasarkan indeks (nomor urut posisi karakter dalam String). Contoh :

```
public static void main (String[] args){
    String str = "Sumatera Utara";
    System.out.println("Karakter 0 = " + str.charAt(0));
    System.out.println("Karakter 9 = " + str.charAt(9));
}
```

Hasil

```
run:
Karakter 0 = S
Karakter 9 = U
```

- `concat(stringLain)` adalah fungsi untuk menggabungkan String dengan String yang lainnya (`stringLain`). Contoh :

```
public static void main (String[] args){
    String str1 = "Kotamadya";
    String str2 = "Medan";
    String str3 = str1.concat(str2) ;

    System.out.println("Hasil concat = " + str3);
}
```

Hasil

```
run:
Hasil concat = KotamadyaMedan
```

- `endWith(akhiranKarakter)` adalah fungsi untuk menemukan keberadaan karakter-karakter pada bagian akhiran dari String. Fungsi akan menghasilkan nilai *true* jika karakter-karakter pada `akhiranKarakter` sama dengan akhiran karakter pada String dan menghasilkan nilai *false* jika sebaliknya. Contoh :

```
public static void main (String[] args){
    String str = "Deli Serdang";

    System.out.println("sufiks 'dang'? " + str.endsWith("dang"));
    System.out.println("sufiks 'Del'? " + str.endsWith("Del"));
}
```

Hasil

```
run:
sufiks 'dang'? true
sufiks 'Del'? false
```

- `equals(stringLain)` adalah fungsi yang membandingkan String dengan String yang lainnya (`stringLain`). Proses perbandingannya dipengaruhi oleh urutan karakter dan besar-kecil huruf pada kedua String. Fungsi ini akan menghasilkan nilai *true* jika urutan karakter dan besar-kecil huruf adalah sama pada kedua String, menghasilkan nilai *false* jika sebaliknya. Contoh :

```
public static void main (String[] args){
    String kota1 = "Binjai";
    String kota2 = "BiNjAi";

    System.out.print(kota1 + " = " + kota2 + "? " );
    System.out.println(kota1.equals(kota2));
}
```

Hasil

```
run:
Binjai = BiNjAi? false
```

- `equalsIgnoreCase(stringLain)` adalah fungsi yang membandingkan String dengan String yang lainnya (`stringLain`). Namun, proses perbandingannya tidak dipengaruhi oleh besar-kecil huruf pada kedua String. Fungsi ini akan menghasilkan nilai *true* jika karakter pada kedua String adalah sama dan menghasilkan nilai *false* jika sebaliknya. Contoh :

```
public static void main (String[] args){
    String kota1 = "Langkat";
    String kota2 = "LaNgKaT";

    System.out.print(kota1 + " = " + kota2 + "? " );
    System.out.println(kota1.equalsIgnoreCase(kota2));
}
```

Hasil

```
run:
Langkat = LaNgKaT? true
```

- *indexOf(char_String, beginIndex)* adalah fungsi yang menghasilkan nomor indeks dalam sebuah String berdasarkan karakter. Apabila karakter tidak ditemukan dalam String maka fungsi akan menghasilkan nilai -1. Contoh :

```
public static void main (String[] args){
    String kabupaten = "Karo";

    System.out.print("Karakter 'r' pada indeks ke-");
    System.out.println(kabupaten.indexOf('r'));
}
```

Hasil

```
run:
Karakter 'r' pada indeks ke-2
```

Posisi awal proses pencarian karakter pada fungsi *indexOf* juga dapat ditentukan. Apabila posisi awal pencari tidak ditentukan maka proses pencarian dimulai dari indeks ke-0. Contoh :

```
public static void main (String[] args){
    String kabupaten = "Serdang Bedagai";

    System.out.print("Karakter 'd' pada indeks ke ");
    System.out.println(kabupaten.indexOf('d', 4));
}
```

Hasil

```
run:
Karakter 'd' pada indeks ke-10
```

Selain karakter, fungsi *indexOf* juga dapat melakukan pencarian berdasarkan String.

- *length()* adalah fungsi yang menghasilkan panjang atau banyaknya karakter dalam sebuah String.
- *replace(oldchar, newchar)* adalah fungsi untuk menggantikan seluruh karakter lama (parameter pertama) dengan karakter baru (parameter kedua). Contoh :

```
public static void main (String[] args){
    String kabupaten = "Batu Bara";

    System.out.print(kabupaten + " menjadi ");
    System.out.println(kabupaten.replace('a', 'e'));
}
```

Hasil

```
run:
Batu Bara menjadi Betu Bere
```

- *replaceAll(old_subString, new_subString)* adalah fungsi untuk menggantikan sub string lama (parameter pertama) dengan sub string baru (parameter kedua). Contoh :

```
public static void main (String[] args){
    String kabupaten = "Labuhanbatu Selatan";

    System.out.print(kabupaten + " menjadi ");
    System.out.println(kabupaten.replaceAll("batu", "Hati"));
}
```

Hasil

```
run:
Labuhanbatu Selatan menjadi LabuhanHati Selatan
```

- *split(subString)* adalah fungsi untuk memisah-misahkan string menjadi array string. Parameter *subString* menjadi acuan atau titik pemisah. Contoh :

```
public static void main (String[] args){
    String kabupaten = "Padang-Lawas-Utara";

    for (String kata : kabupaten.split("-")){
        System.out.println(kata);
    }
}
```

Hasil

```
run:
Padang
Lawas
Utara
```

- *substring(beginIndex)* adalah fungsi untuk mendapatkan/menghasilkan sub string dari sebuah string yang dimulai dari indeks tertentu (*beginIndex*). Contoh :

```
public static void main (String[] args){
    String kota = "Pematangsiantar";
    System.out.println("Kota " + kota.substring(8));
}
```

Hasil

```
run:
Kota siantar
```

Selain indeks awal, fungsi *substring* juga dapat menggunakan paramter untuk menentukan indeks akhir (*endIndex*). Contoh :

```
public static void main (String[] args){
    String kabupaten = "Toba Samosir";
    System.out.println("substring " + kabupaten);
    System.out.println("menjadi '" + kabupaten.substring(2, 9) + "'");
}
```

Hasil

```
run:
substring Toba Samosir
menjadi 'ba Samo'
```

- *toLowerCase()* adalah fungsi untuk merubah seluruh huruf dalam string menjadi huruf kecil. Contoh :

```
public static void main (String[] args){
    String kota = "Tebing Tinggi";
    System.out.println(kota);
    System.out.println("Lower Case : " + kota.toLowerCase());
}
```

Hasil

```
run:
Tebing Tinggi
Lower Case : tebing tinggi
```

- *toUpperCase()* adalah fungsi untuk merubah seluruh huruf dalam string menjadi huruf besar (kapital). Contoh :

```
public static void main (String[] args){
    String kabupaten = "Mandailing Natal";
    System.out.println(kabupaten);
    System.out.println("Upper Case : " + kabupaten.toUpperCase());
}
```

Hasil

```
run:
Mandailing Natal
Upper Case : MANDAILING NATAL
```

### 7.3 Data Tanggal dan Waktu

Data tanggal dan waktu sering digunakan dalam aplikasi. Umumnya digunakan pada tanggal lahir, penjadwalan dan sebagainya. Pemrograman Java menyediakan Class Date yang terdapat pada paket (*package*) `java.util.Date` sehingga memudahkan penggunanya untuk memanipulasi data tanggal dan waktu. Mulai pemrograman Java versi 8 telah dilengkapi dengan paket baru untuk memanipulasi tanggal dan waktu yaitu `java.time`. Berikut beberapa manipulasi data tanggal dan waktu yang akan dibahas.

- Tanggal dan waktu saat ini

  Untuk mendapatkan tanggal dan waktu saat ini, dapat melalui objek yang dideklarasikan sebagai sebagai Date (paket `java.util.Date`). Contoh :

```
public static void main (String[] args){
    Date tgl = new Date();
    System.out.println(tgl.toString());
}
```

Hasil

```
run:
Wed Dec 05 10:11:45 ICT 2018
```

- Format penulisan tanggal dan waktu

  Pemrograman Java telah menyediakan beberapa format tanggal dan waktu dengan pola karakter tertentu menggunakan paket (*package*) `java.util.Date`. dan paket `java.text.SimpleDateFormat`. Adapun pola karakter tersebut dapat dilhat pada tabel 7.1 di bawah ini.

**Tabel 7.2** Pola Huruf Untuk Format Tanggal dan Waktu

| ***Huruf*** | ***Komponen*** | ***Format Hasil*** | ***Contoh*** |
|---|---|---|---|
| G | Era designator | Text | AD |
| y | Year | Year | 1996; 96 |
| Y | Week year | Year | 2009; 09 |
| M | Month in year (context sensitive) | Month | July; Jul; 07 |

| | | | |
|---|---|---|---|
| L | Month in year (standalone form) | Month | July; Jul; 07 |
| w | Week in year | Number | 27 |
| W | Week in month | Number | 2 |
| D | Day in year | Number | 189 |
| d | Day in month | Number | 10 |
| F | Day of week in month | Number | 2 |
| E | Day name in week | Text | Tuesday; Tue |
| u | Day number of week (1 = Monday, ..., 7 = Sunday) | Number | 1 |
| a | am/pm marker | Text | PM |
| H | Hour in day (0-23) | Number | 0 |
| k | Hour in day (1-24) | Number | 24 |
| K | Hour in am/pm (0-11) | Number | 0 |
| h | Hour in am/pm (1-12) | Number | 12 |
| m | Minute in hour | Number | 30 |
| s | Second in minute | Number | 55 |
| S | Millisecond | Number | 978 |
| z | Time zone | General time zone | Pacific Standard Time; PST; GMT-08:00 |
| Z | Time zone | RFC 822 time zone | -800 |
| X | Time zone | ISO 8601 time zone | -08; -0800; -08:00 |

(sumber : https://docs.oracle.com/javase/8/docs/api/java/text/SimpleDateFormat.html)

Contoh 1 – format tanggal dan waktu :

```
public static void main (String[] args){
    Date tgl = new Date( );
    SimpleDateFormat pola;

    pola = new SimpleDateFormat("dd/MMM/yyyy");
    System.out.println("Tanggal: " + pola.format(tgl));
```

```
        pola = new SimpleDateFormat("EEEE");
        System.out.println("Hari: " + pola.format(tgl));

        pola = new SimpleDateFormat("kk:mm:ss");
        System.out.println("Waktu: " + pola.format(tgl));

        pola = new SimpleDateFormat("zzz");
        System.out.println("Zona Waktu: " + pola.format(tgl));
    }
```

Hasil

```
run:
Tanggal: 05/Dec/2018
Hari: Wednesday
Waktu: 11:10:50
Zona Waktu: ICT
```

Contoh 2 – selisih tanggal menggunakan paket *java.util.Date*

```
public static void main (String[] args) throws ParseException{
    SimpleDateFormat pola = new SimpleDateFormat("dd/MM/yyyy");
    Date tglAwal = pola.parse("01/12/2018");
    Date tglAkhir = pola.parse("05/12/2018");

    long selisih = Math.abs(tglAkhir.getTime() - tglAwal.getTime());

    System.out.println("Tanggal pinjam : " + pola.format(tglAwal));
    System.out.println("Tanggal kembali : " + pola.format(tglAkhir));
    System.out.print("lama pinjam : ");
    System.out.println(TimeUnit.MILLISECONDS.toDays(selisih)+" hari");
}
```

Hasil

```
run:
Tanggal pinjam : 01/12/2018
Tanggal kembali : 05/12/2018
lama pinjam : 4 hari
```

Keterangan :

Untuk menggunakan perintah `TimeUnit.MILLISECONDS.toDays` , terlebih dahulu masukkan paket java.util.concurrent.TimeUnit; melalui import

```
import java.util.concurrent.TimeUnit;
```

Contoh 3 – selisih tanggal menggunakan paket *java.time*

```
public static void main (String[] args) throws ParseException{
    LocalDate pinjam = LocalDate.parse("2018-12-01");
    LocalDate kembali = LocalDate.now();

    Period period = Period.between(pinjam, kembali);
    int lama = period.getDays();

    System.out.println("Tanggal pinjam : " + pinjam);
    System.out.println("Tanggal kembali : " + kembali);
    System.out.println("lama pinjam : " + lama + " hari");
}
```

Hasil

```
run:
Tanggal pinjam : 2018-12-01
Tanggal kembali : 2018-12-05
lama pinjam : 4 hari
```

Contoh 4 – selisih waktu menggunakan paket *java.util.Date*

```
public static void main(String[] args) throws ParseException{

    SimpleDateFormat pola = new SimpleDateFormat("HH:mm:ss");
    Date mulai = pola.parse("11:05:15");
    Date selesai = new Date();
    long selisih = Math.abs(selesai.getTime() - mulai.getTime()) ;

    long menit = selisih / (60 * 1000) % 60;
    long jam = selisih / (60 * 60 * 1000) % 24;

    System.out.println("Mulai \t\t: " + pola.format(mulai) );
    System.out.println("Selesai \t: " + pola.format(selesai));

    System.out.print("Lama Proses \t: ");
    System.out.println(jam + " jam, " + menit + " menit");
}
```

Hasil

```
run:
Mulai           : 11:05:15
Selesai         : 14:16:03
Lama Proses     : 3 jam, 10 menit
```

Contoh 5 – selisih waktu menggunakan paket *java.time*

```
public static void main(String[] args) throws ParseException{
    LocalTime mulai = LocalTime.parse("11:02:10"); //hh:mm:ss
    LocalTime selesai = LocalTime.now();
    Duration lama = Duration.between(mulai, selesai);

    System.out.println("Mulai \t\t: " + mulai);
    System.out.println("Selesai \t: " + selesai);

    System.out.print("Lama Proses \t: ");
    System.out.print(lama.toHours() + " jam, ");
    System.out.println(lama.toMinutes() % 60 + " menit");
}
```

Hasil

```
run:
Mulai           : 11:02:10
Selesai         : 13:19:40.336
Lama Proses     : 2 jam, 17 menit
```

Contoh 6 – selisih umur menggunakan paket *java.time*

```
public static void main(String[] args){
    LocalDate tglLahir = LocalDate.of(1985, Month.JUNE, 23);
    LocalDate tglSkrg = LocalDate.now();

    Period p = Period.between(tglLahir, tglSkrg);

    System.out.println("Tanggal Lahir : " + tglLahir);
    System.out.println("Hari ini      : " + tglSkrg);
    System.out.print("Usia          : ");
    System.out.print(p.getYears() + " tahun, ");
    System.out.print(p.getMonths() + " bulan, ");
    System.out.println(p.getDays() + " hari");
}
```

Hasil

```
run:
Tanggal Lahir : 1985-06-23
Hari ini      : 2018-12-05
Usia          : 33 tahun, 5 bulan, 12 hari
```

- Contoh 7 – perbandingan kedua tanggal yang berbeda menggunakan paket *java.util.Date*

```
public static void main(String[] args) throws ParseException{
    SimpleDateFormat sdf = new SimpleDateFormat("dd-MM-yyyy");
    Date hariGuru = sdf.parse("25-11-2018");
    Date hariIni = new Date();
    long selisih;
```

```
        System.out.println("Hari Guru \t: " + sdf.format(hariGuru));
        System.out.println("Hari ini \t: " + sdf.format(hariIni));

        if (hariGuru.after(hariIni)) {
            selisih = Math.abs(hariGuru.getTime() - hariIni.getTime());
            System.out.print("Hari Guru masih ");
            System.out.print(TimeUnit.MILLISECONDS.toDays(selisih));
            System.out.println(" hari lagi dari hari ini");

        }else if (hariGuru.before(hariIni)) {
            selisih = Math.abs(hariIni.getTime() - hariGuru.getTime());
            System.out.print("Hari Guru telah lewat ");
            System.out.print(TimeUnit.MILLISECONDS.toDays(selisih));
            System.out.println(" hari yang lalu dari hari ini");

        }else if (hariGuru.equals(hariIni)) {
            System.out.println("Hari ini adalah hari Guru");
        }
    }
```

Hasil

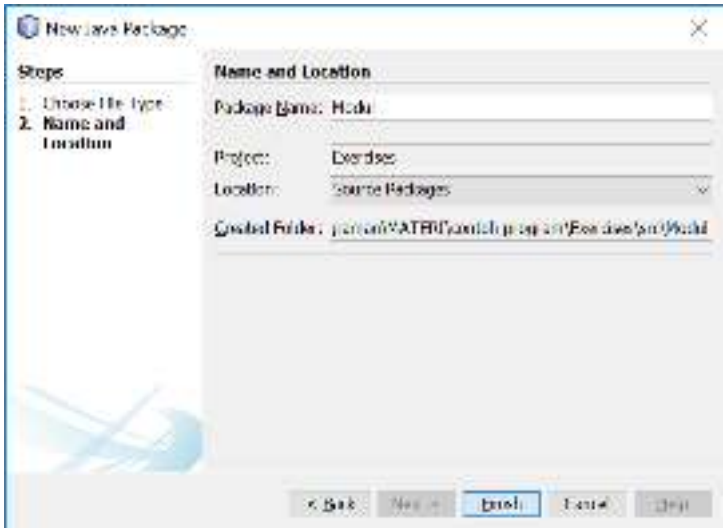
```
run:
Hari Guru       : 25-11-2018
Hari ini        : 05-12-2018
Hari Guru telah lewat 10 hari yang lalu dari hari ini
```

Konsep perbandingan kedua tanggal yang berbeda menggunakan paket `java.time` sama dengan paket `java.util.Date`. Pada paket `java.time` menggunakan *method* `isAfter` untuk tanggal setelah tanggal pembanding, *method* `isBefore` untuk tanggal sebelum tanggal pembanding dan method `isEqual` untuk tanggal sama dengan tanggal pembanding.

## 7.4 Latihan

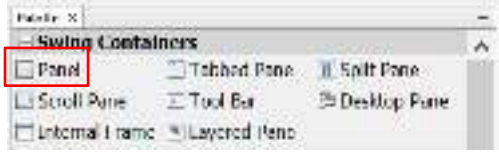
### 7.4.1 Aplikasi Tarif Parkir

1. Tambahkan paket (*Java Package*) pada projek 'Exercises'. Beri nama paket dengan 'Modul'.




2. Tambahkan ‘JFrame Form’ didalam paket ‘Modul’ dengan langkah-langkah seperti pada percobaan modul 1 (menambah frame form). Beri nama JFrame Form dengan ‘Parkir’.

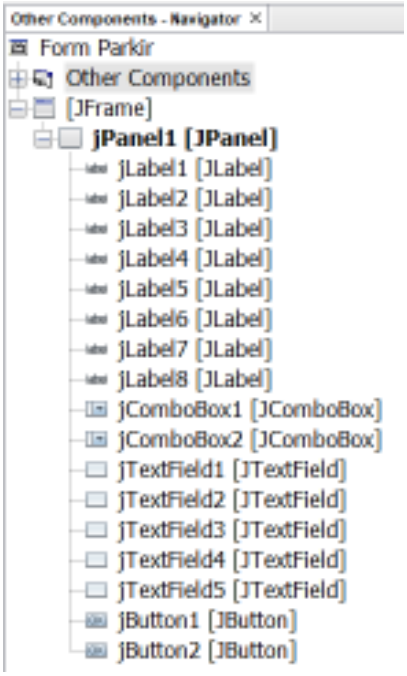
3. Masukkan komponen Panel dari palette ‘Swing Containers’ kedalam form. Ubah ukuran komponen Panel sesuai dengan ukuran form.




4. Masukkan komponen Label, Text Field, Combo Box, dan Button dari palette Swing Controls kedalam komponen Panel.

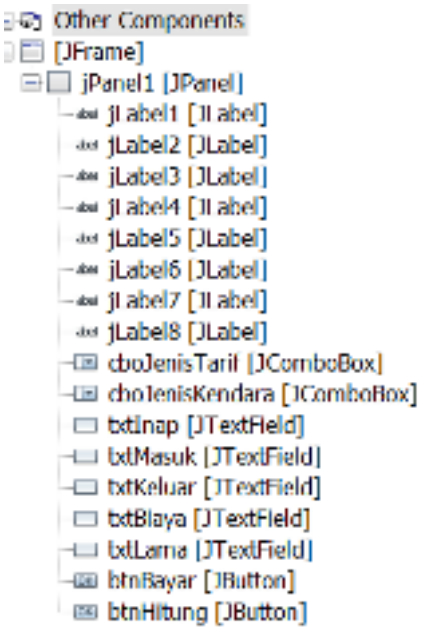
5. Susun posisi masing-masing komponen tersebut dan ubah nilai properti pada komponen Label dan Text Field seperti pada gambar di bawah ini.

Seluruh komponen yang digunakan dapat dilihat pada jendela navigator



6. Ubah nama variabel beberapa komponen seperti pada tabel di bawah ini.

| Komponen | Nama Variabel | Properties |
|---|---|---|
| jTextField1 | txtInap | text = <kosong> |
| jTextField2 | txtMasuk | text = <kosong> |
| jTextField3 | txtKeluar | text = <kosong> |
| jTextField4 | txtBiaya | text = <kosong> |
| | | editable = false |
| jTextField5 | txtLama | text = <kosong> |
| | | editable = false |
| jComboBox1 | cboJenisTarif | model = Umum, Inap, Pegawai |
| jComboBox2 | cboJenisKendara | model = Sepeda Motor, Mobil, Bus |
| jButton1 | btnBayar | text = Bayar |
| jButton2 | btnHitung | text = Hitung |




7. Klik tab 'Source', ketikkan kode program berikut ini untuk *import* paket-paket yang dibutuhkan.

```
6   package Modul;
7
8   //import paket yang dibutuhkan
9   import java.text.ParseException;
10  import java.text.SimpleDateFormat;
11  import java.util.Date;
12  import javax.swing.JOptionPane;
```

8. Ketikkan kode program *method* 'bersihEntry' berikut ini.

```
20      private void bersihEntry(){
21          txtTnap.setText("");
22          txtMasuk.setText("");
23          txtKeluar.setText("");
24          txtLama.setText("");
25          txtBiaya.setText("");
26      }
```

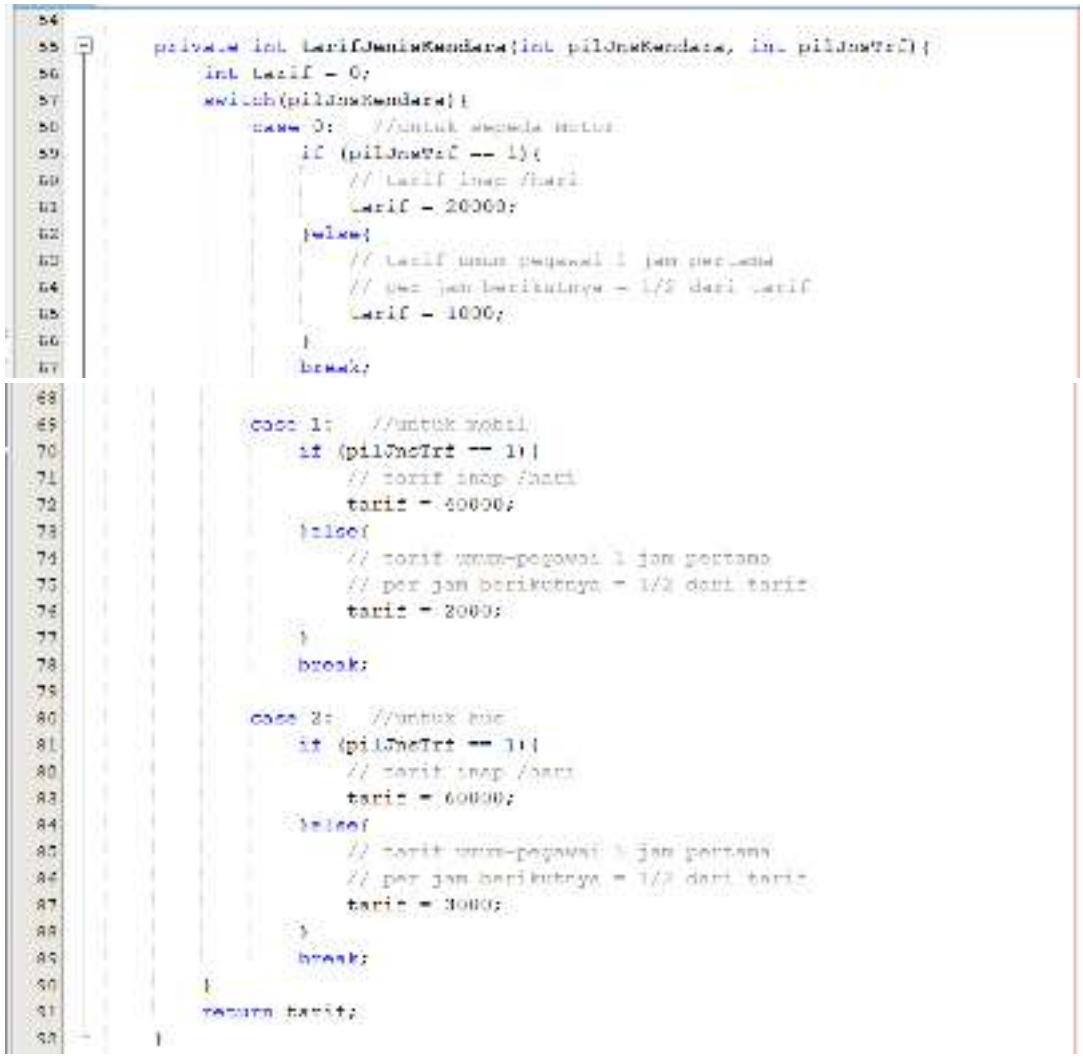
Kode program *method* 'bersihEntry' digunakan untuk membersihkan (mengosongkan) komponen entri data dari data sebelumnya.

9. Ketikkan kode program *method* 'validasiData' berikut ini.

```
28      private boolean validasiData(int pilJnsTrf){
29          boolean hasil = true;
            String pesan = null;
31
32          if(pilJnsTrf == 1){
33              //untuk inap
34              if (txtTnap.getText().isEmpty()){
35                  pesan = "Masukkan Lama Hari";
36                  hasil=false;
37                  JOptionPane.showMessageDialog(null, pesan);
38              }
39          }else{
40              //untuk umum & pegawai
41              if (txtMasuk.getText().isEmpty()){
42                  pesan = "Masukkan Waktu Masuk";
43                  hasil=false;
44                  JOptionPane.showMessageDialog(null, pesan);
45
46              }else if (txtKeluar.getText().isEmpty()){
47                  pesan = "Masukkan Waktu Keluar";
48                  hasil=false;
49                  JOptionPane.showMessageDialog(null, pesan);
50              }
51          }
52          return hasil;
53      }
```

Kode program *method* 'validasiData' digunakan untuk memeriksa tempat entri data yang masih kosong. Apabila masih komponen entri data masih kosong maka akan menampilkan pesan.

10. Ketikkan kode program *method* 'tarifJenisKendara' berikut ini.



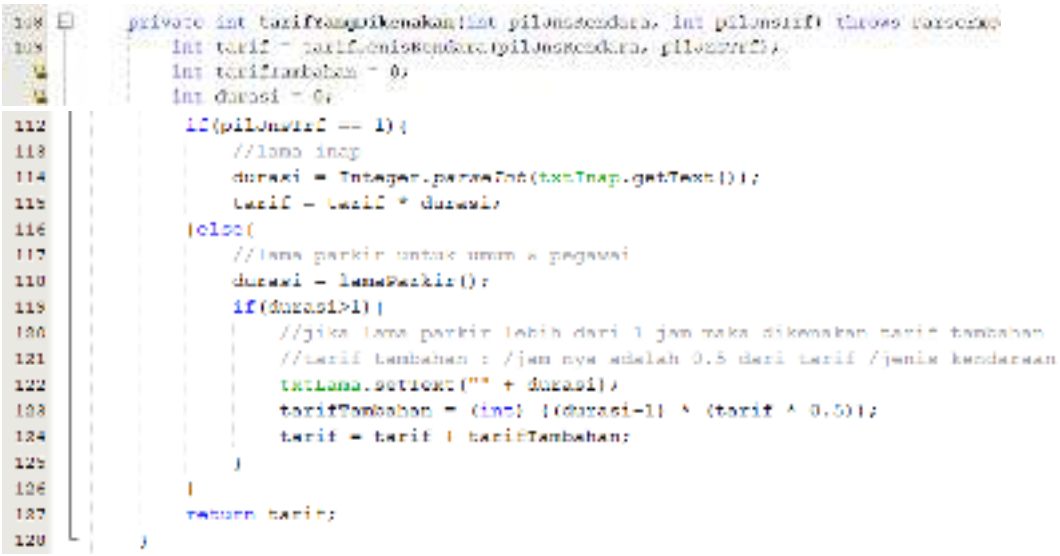
Kode program *method* 'tarifJenisKendara' untuk menentukan tarif parkir berdasakan jenis kendaraan dan jenis tarif.

11. Ketikkan kode program *method* 'lamaParkir' berikut ini.

```
94      private int lamaParkir() throws ParseException{
95          int lama=0;
96          SimpleDateFormat pola = new SimpleDateFormat("HH:mm:ss");
97          try{
98              Date mulai = pola.parse(txtMasuk.getText());
99              Date selesai = pola.parse(txtKeluar.getText());
100             long selisih = Math.abs(selesai.getTime() - mulai.getTime()) ;
101
102             lama = (int) (selisih / (60 * 60 * 1000) % 24);
103         }catch( ParseException ex) {
104         }
105         return lama;
106     }
```

Kode program *method* 'tarifJenisKendara' untuk menentukan tarif parkir berdasakan jenis kendaraan dan jenis tarif.

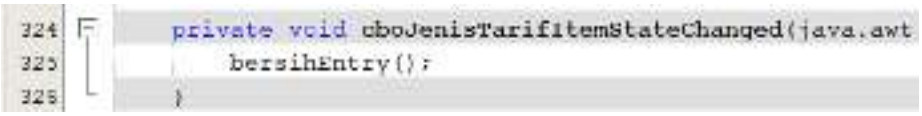
12. Ketikkan kode program *method* 'tarifYangDikenakan' berikut ini.



```
    private int tarifYangDikenakan(int pilJnsKendara, int pilJnsTrf) throws ParseExc
        int tarif = tarifJenisKendara(pilJnsKendara, pilJnsTrf);
        int tarifTambahan = 0;
        int durasi = 0;
        if(pilJnsTrf == 1){
            //lama inap
            durasi = Integer.parseInt(txtInap.getText());
            tarif = tarif * durasi;
        }else{
            //lama parkir untuk umum & pegawai
            durasi = lamaParkir();
            if(durasi>1){
                //jika lama parkir lebih dari 1 jam maka dikenakan tarif tambahan
                //tarif tambahan : /jam nya adalah 0.5 dari tarif /jenis kendaraan
                txtLama.setText("" + durasi);
                tarifTambahan = (int) ((durasi-1) * (tarif * 0.5));
                tarif = tarif + tarifTambahan;
            }
        }
        return tarif;
    }
```

Kode program *method* 'tarifYangDikenakan' digunakan untuk mengkalkulasi tarif parkir yang dikenakan dan ditampilkan.

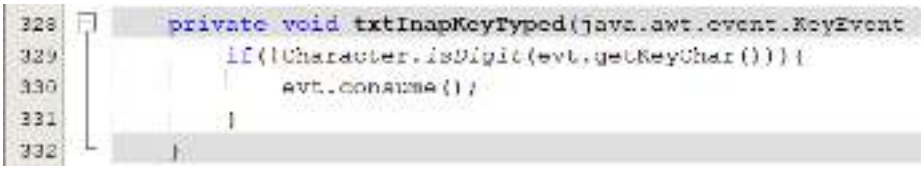
13. Klik kanan komponen 'cboJenisTarif' pada jendela navigator → Events → Item → itemStateChanged. Ketikkan kode program berikut ini.



```
    private void cboJenisTarifItemStateChanged(java.awt
        bersihEntry();
    }
```

Memanggil *method* 'bersihEntry' pada saat terjadi perubahan nilai/data yang dipilih pada komponen cboJenisTarif.

14. Klik kanan komponen 'txtInap' pada jendela navigator → Events → Key → keyTyped. Ketikkan kode program berikut ini.



```
    private void txtInapKeyTyped(java.awt.event.KeyEvent
        if(!Character.isDigit(evt.getKeyChar())){
            evt.consume();
        }
    }
```

Menolak karakter yang diinput selain karakter angka.

15. Klik kanan komponen 'btnHitung' pada jendela navigator → Events → Action → actionPerformed. Ketikkan kode program berikut ini.

```
334     private void btnHitungActionPerformed(java.awt.event.ActionEvent e
335         int pilJnsKendara = cboJenisKendara.getSelectedIndex();
336         int pilJnsTrf = cboJenisTarif.getSelectedIndex();
337         boolean valid = validasiData(pilJnsTrf);
            int tarif = 0;
339
340         if (valid == true){
341             try {
342                 tarif = tarifYangDikenakan(pilJnsKendara, pilJnsTrf);
343                 txtBiaya.setText("" + tarif);
344             } catch (ParseException ex) {
345             }
346         }
347     }
```

Menjalankan seluruh poses perhitungan tarif parkir.

16. Klik kanan komponen 'btnBayar' pada jendela navigator → Events → Action → actionPerformed. Ketikkan kode program berikut ini.

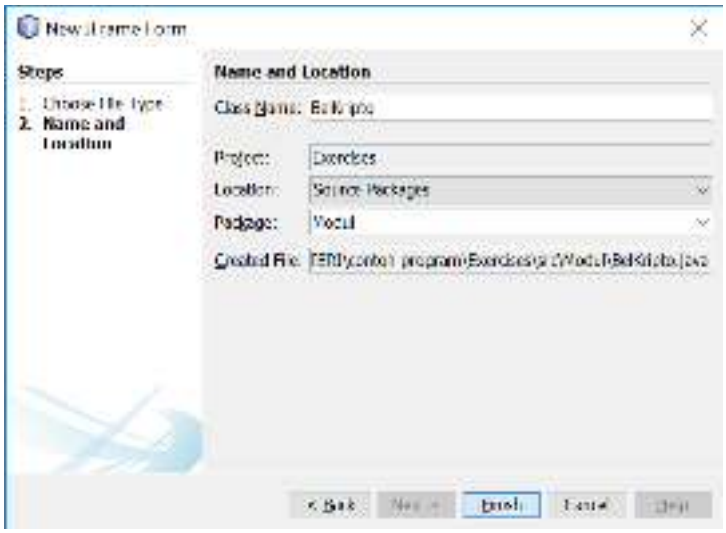
```
348
349     private void btnBayarActionPerformed(java.awt.event.ActionEv
350         JOptionPane.showMessageDialog(null, "Terima Kasih!");
351         bersihEntry();
352     }
```

Menampilkan pesan dan memanggil *method* 'bersihEntry'.

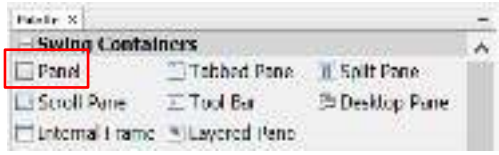
17. Compile (F9) dan Run Program melalui menu Run → Run File (Shift + F6).

### 7.4.2 Aplikasi Belajar Kriptografi

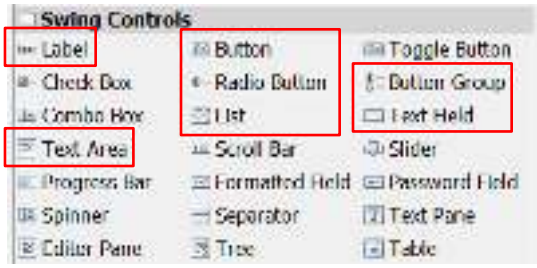
1. Tambahkan 'JFrame Form' didalam paket 'Modul'. Beri nama JFrame Form dengan 'BelKripto'.



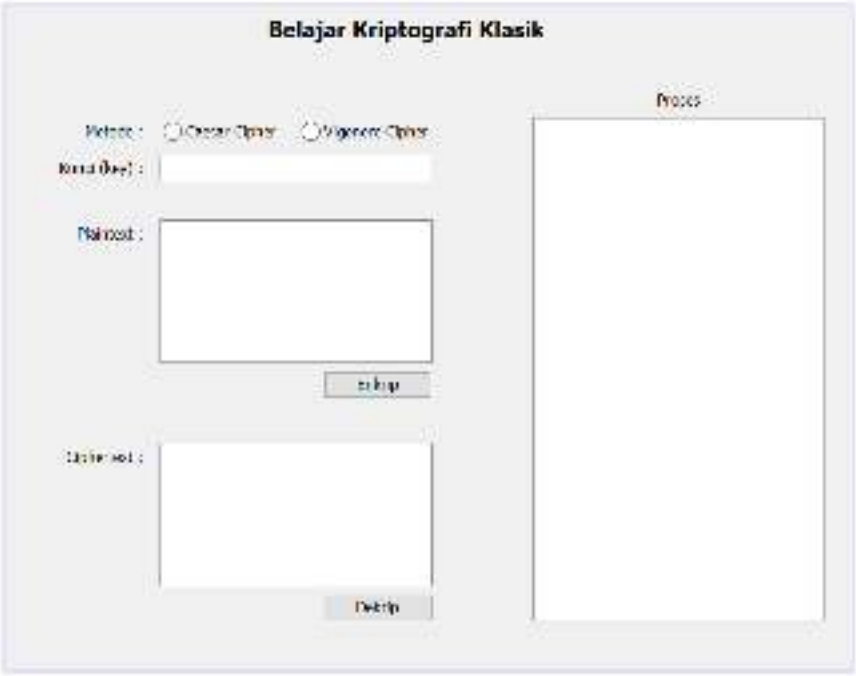
2. Masukkan komponen Panel dari palette 'Swing Containers' kedalam form. Ubah ukuran komponen Panel sesuai dengan ukuran form.



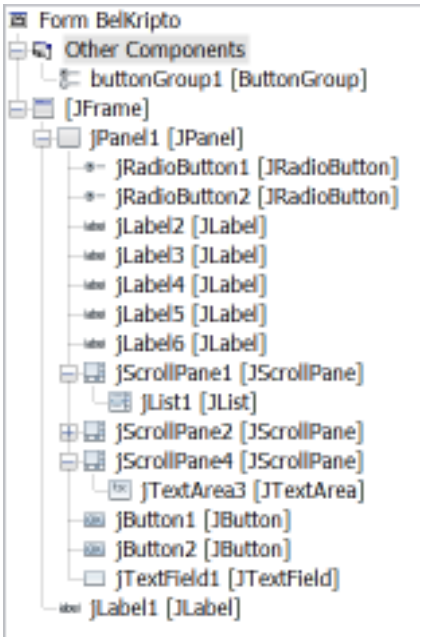

3. Masukkan komponen Label, Text Field, Text Area, List, Radio Button, Button Group, dan Button dari palette Swing Controls kedalam komponen Panel.




Susun posisi masing-masing komponen tersebut dan ubah nilai properti pada komponen Label dan Text Field seperti pada gambar di bawah ini.



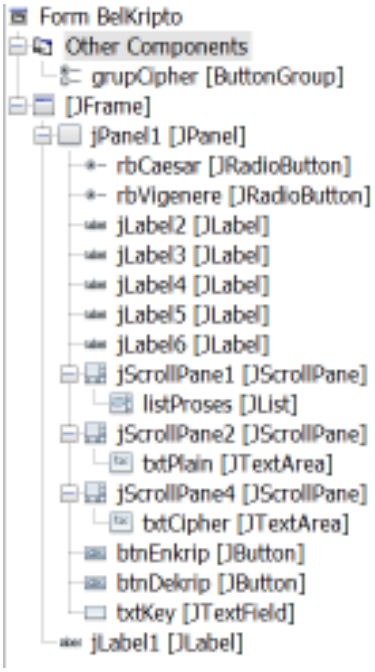

Seluruh komponen yang digunakan dapat dilihat pada jendela navigator




4. Ubah nama variabel beberapa komponen seperti pada tabel di bawah ini.

| Komponen | Nama Variabel | Properties |
|---|---|---|
| jTextArea1 | txtPlain | text = <kosong> |
| | | lineWrap = true |
| | | wrapStyleWord = true |
| jTextArea2 | txtCipher | text = <kosong> |
| | | lineWrap = true |
| | | wrapStyleWord = true |
| jTextField1 | txtKey | text = <kosong> |
| jList1 | listProses | model = <kosong> |
| ButtonGroup1 | grupCipher | - |
| jRadioButton1 | cbCaesar | text = Caesar Cipher |
| | | buttonGroup = grupCipher |
| jRadioButton2 | cbVigenere | text = Vigenere Cipher |
| | | buttonGroup = grupCipher |
| jButton1 | btnEnkrip | text = Enkrip |
| jButton2 | btnDekrip | text = Dekrip |

Nama Komponen yang telah diubah dapat dilihat pada jendela navigator seperti berikut ini.




5. Klik tab 'Source', ketikkan kode program berikut ini untuk *import* paket-paket yang dibutuhkan.

```
1   package Modul;
2
3   import java.util.ArrayList;
4   import javax.swing.DefaultListModel;
5   import javax.swing.JOptionPane;
```

6. Ketikkan kode program berikut ini untuk mendeklarasikan variabel global yang dibutuhkan.

```
8   public class BelKripto extends javax.swing.JFrame {
9       String pesan = null;
10      ArrayList alfabet = new ArrayList();
```

7. Ketikkan kode program berikut ini untuk menjalankan/mengeksekusi method 'inisialisasi' pada konstruktor.

```
12      public BelKripto() {
13          initComponents();
14          inisialisasi();
15      }
```

Menjalankan kode program *method* 'iniComponents()' dan 'inisialisasi()'

8. Ketikkan kode program *method* 'inisialisasi' berikut ini.

```
17      private void inisialisasi(){
18          alfabet.add('A'); alfabet.add('B'); alfabet.add('C');
19          alfabet.add('D'); alfabet.add('E'); alfabet.add('F');
20          alfabet.add('G'); alfabet.add('H'); alfabet.add('I');
21          alfabet.add('J'); alfabet.add('K'); alfabet.add('L');
22          alfabet.add('M'); alfabet.add('N'); alfabet.add('O');
23          alfabet.add('P'); alfabet.add('Q'); alfabet.add('R');
24          alfabet.add('S'); alfabet.add('T'); alfabet.add('U');
25          alfabet.add('V'); alfabet.add('W'); alfabet.add('X');
26          alfabet.add('Y'); alfabet.add('Z');
27      }
```

Menyimpan nilai/data karakter alfabet (A-Z) dalam objek ArrayList (inisialisasi).

9. Ketikkan kode program *method* 'enkripCaesar' berikut ini.

```
29      private String enkripCaesar(String plain){
30          DataLinkedList items = new DataLinkedList();
31
32          short kunci = 0;
33          short indeks;
34          short hasilRumus;
35
36          String cipher  = null;
37          String arrChar = null;
38          String steps   = null;
39
40          int pjgPlain = 0;
41          char charPlain;
42
43          try{
44              listProses.removeAll();
45              pjgPlain = plain.length();
46              kunci = Short.parseShort(txtKey.getText());
47
48              for(int x=0; x<pjgPlain; x++){
49                  charPlain = plain.charAt(x);
50                  steps = Character.toString(charPlain);
51                  steps = steps.concat(" ( " + kunci + " % 25");
52                  steps = steps.concat("    )  ");
53
54                  indeks = (short) alfabet.indexOf(charPlain);
55                  steps = steps.concat("" + indeks + " + " + kunci + " % 25");
56                  steps = steps.concat("  -->  ");
57
58                  hasilRumus = (short) ((indeks + kunci) % 25);
59                  steps = steps.concat("" + hasilRumus + " - ");
60
61                  arrChar =  alfabet.get(hasilRumus).toString();
62                  steps = steps.concat(arrChar);
63
64                  if (cipher == null)
65                      cipher = arrChar;
66                  else
67                      cipher = cipher.concat(arrChar);
68
69                  items.addElement(steps);
70              }
71              listProses.setModel(items);
```

```
        }catch (Exception ex){
            JOptionPane.showMessageDialog(null, ex);
        }
        return cipher;
    }
```

*Method* ‘enkripCaesar’ digunakan untuk menjalankan proses enkripsi menggunakan teknik enkripsi ‘Caesar’. Teknik enkripsi ini mengganti (substitusi) setiap karakter pesan asli (*plaintext*) dengan cara menambahkan indeks karakter alfabet (A-Z) dengan nilai kunci.

10. Ketikkan kode program *method* ‘enkripVigenere’ berikut ini.

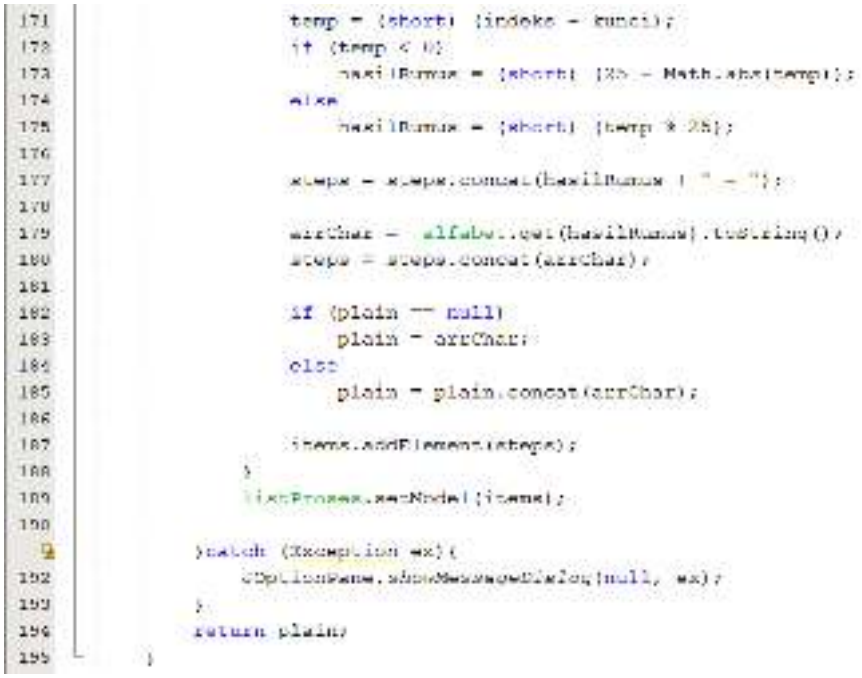
```
    private String enkripVigenere(String plain){
        DefaultTableModel tModel = new DefaultTableModel();

        int pjgPlain = 0;
        int pjgKunci = 0;
        int itrPlain = 0;
        int itrKunci = 0;

        String cipher  = null;
        String kunci   = null;
        String arrChar = null;
        String steps   = null;

        char charPlain;
        char charKunci;

        short indeksPlain = 0;
        short indeksKunci = 0;
        short hasilAkhir  = 0;

        try{
            tblPesan.removeAll();
            pjgPlain = plain.length();
            kunci = txtKey.getText().toUpperCase();
            pjgKunci = kunci.length();

            for (itrPlain=0; itrPlain<pjgPlain; itrPlain++){
                charPlain = plain.charAt(itrPlain);
                steps = Character.valueOf(charPlain);

                charKunci = kunci.charAt(itrKunci);
                steps = steps.concat(" + " + charKunci + " % 26");
                steps = steps.concat("  -->  ");

                itrKunci++;
                if (itrKunci >= pjgKunci)
                    itrKunci = 0;

                indeksPlain = (short) alfabet.indexOf(charPlain);
                indeksKunci = (short) alfabet.indexOf(charKunci);
                steps = steps.concat(indeksPlain + " + " + indeksKunci + " % 26");
                steps = steps.concat("  -->  ");

                hasilAkhir = (short) ((indeksPlain + indeksKunci) % 26);
                arrChar =  alfabet.get(hasilAkhir).toString();
                steps = steps.concat(hasilAkhir + " = " + arrChar);
```

```
                if (cipher == null)
                    cipher = arrChar;
                else
                    cipher = cipher.concat(arrChar);

                items.addElement(steps);
            }
            listProses.setModel(items);

        }catch (Exception ex){
            JOptionPane.showMessageDialog(null, ex);
        }
        return cipher;
    }
```

*Method* ‘enkripVigenere’ digunakan untuk menjalankan proses enkripsi menggunakan teknik enkripsi ‘Vigenere’. Teknik enkripsi ‘Vigenere’ sama seperti teknik enkripsi ‘Caesar’ namun kunci yang digunakan adalah karakter huruf. Indeks setiap karakter pesan asli ditambah dengan indeks karakter kunci.

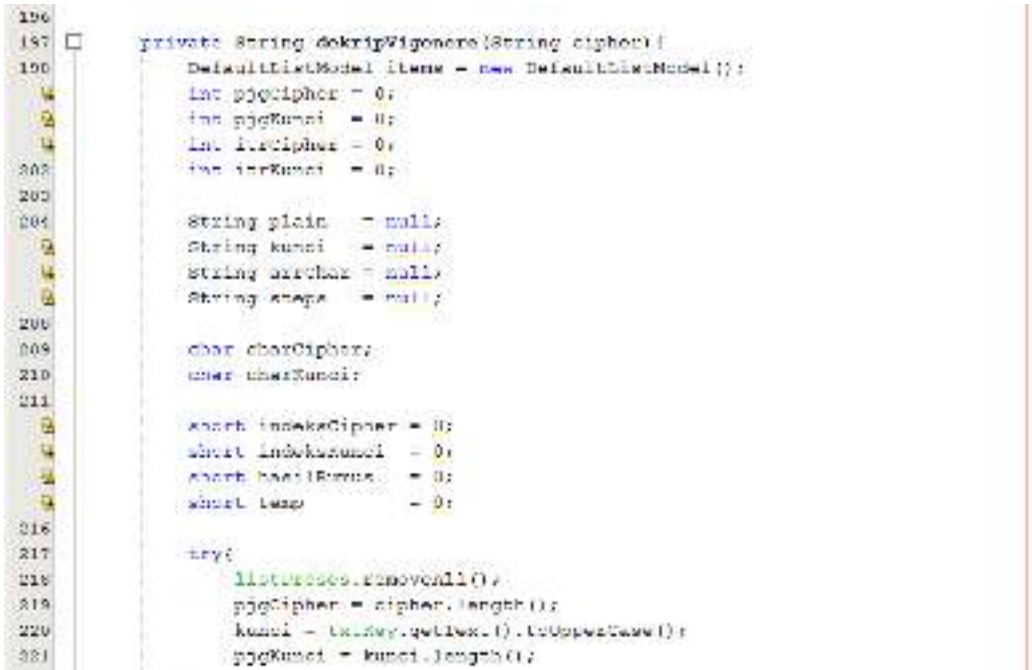
11. Ketikkan kode program *method* ‘dekripCaesar’ berikut ini.

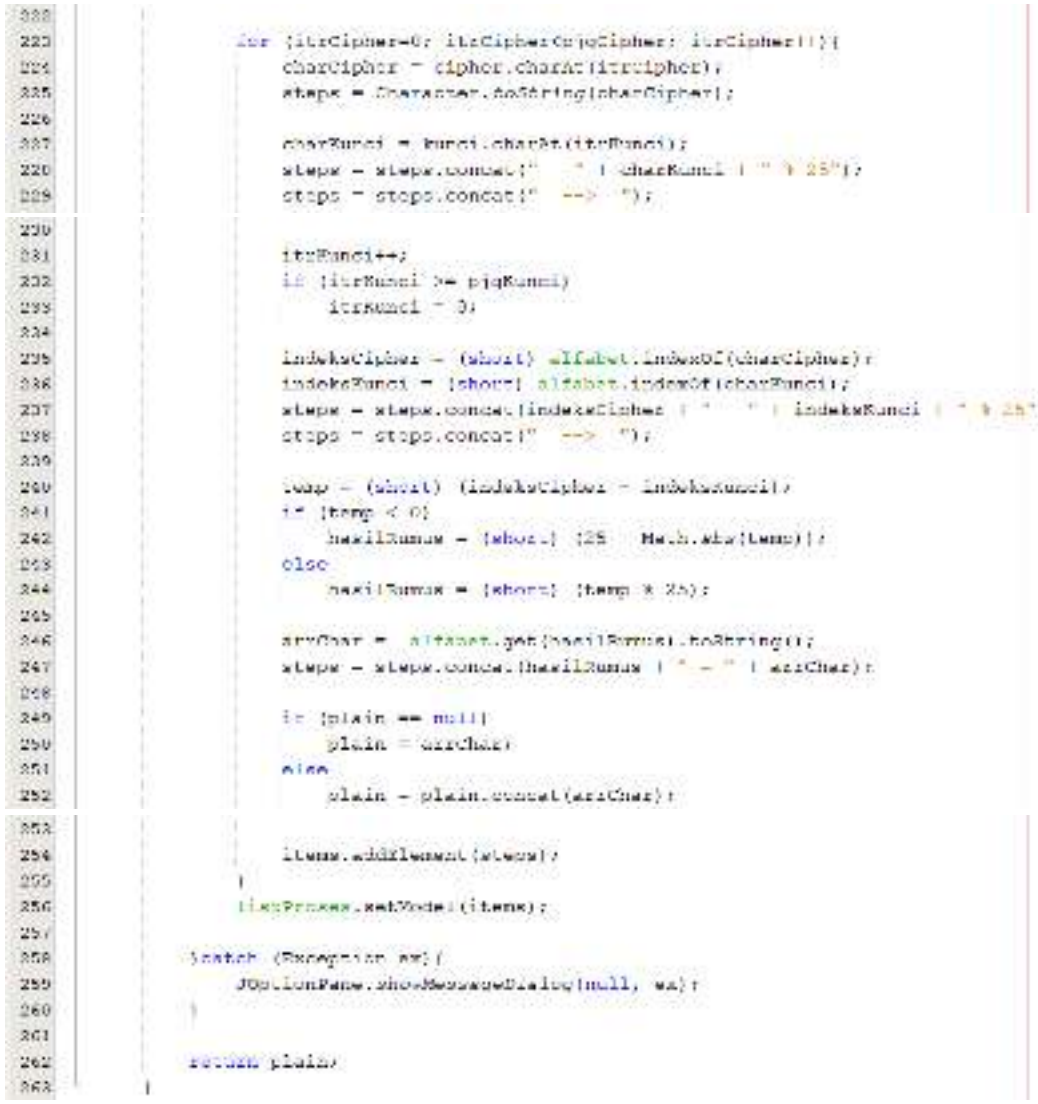
```
    private String dekripCaesar(String cipher){
        DefaultListModel items = new DefaultListModel();

        short kunci = 0;
        short indeks;
        short hasilKunci;
        short temp;

        String plain  = null;
        String arrChar = null;
        String steps  = null;

        int pjgCipher = 0;
        char charCipher;

        try{
            listProses.removeAll();
            pjgCipher = cipher.length();
            kunci = Short.parseShort(txtKey.getText());

            for(int x=0; x<pjgCipher; x++){
                charCipher = cipher.charAt(x);
                steps = Character.toString(charCipher);
                steps = steps.concat(" - " + kunci + " % 26");
                steps = steps.concat("  -->  ");

                indeks = (short) alfabet.indexOf(charCipher);
                steps = steps.concat("" + indeks + " - " + kunci + " % 26");
                steps = steps.concat("  -->  ");
```



*Method* ‘dekripCaesar’ digunakan untuk menjalankan proses dekripsi menggunakan teknik dekripsi ‘Caesar’ yaitu indeks pesan tersandi (*ciphertext*) dikurangi dengan nilai kunci.

12. Ketikkan kode program *method* ‘dekripVigenere’ berikut ini.





*Method* 'dekripsiVigenere' digunakan untuk menjalankan proses dekripsi menggunakan teknik dekripsi 'Vigenere' yaitu mengurangi indeks setiap karakter dari pesan tersandi (*ciphertext*) dengan indeks karakter kunci.

13. Ketikkan kode program *method* 'prosesEnkrip' berikut ini.

```
274
275             } else if (vigen == true){
276                 hasil = enkripVigenere(plain);
277                 txtCipher.setText(hasil);
278                 lblProses.setText("Proses Enkripsi Vigenere Cipher");
279
280             } else {
281                 pesan = "Pilih salah satu metode Enkripsi";
282                 JOptionPane.showMessageDialog(null, pesan);
283             }
284         }
```

*Method* 'prosesEnkrip' digunakan untuk menjalankan proses enkripsi sesuai dengan pilihan dari komponen RadioButton.

14. Ketikkan kode program *method* 'prosesDekrip' berikut ini.

```
285
286     private void prosesDekrip(String cipher){
287         boolean caesar = rbCaesar.isSelected();
288         boolean vigen = rbVigenere.isSelected();
            String hasil = null;
290
291         if (caesar == true){
292             hasil = dekripCaesar(cipher);
293             txtPlain.setText(hasil);
294             lblProses.setText("Proses Dekrip Caesar Cipher");
295
296         } else if (vigen == true){
297             hasil = dekripVigenere(cipher);
298             txtPlain.setText(hasil);
299             lblProses.setText("Proses Dekrip Vigenere Cipher");
300
301         } else {
302             pesan = "Pilih salah satu metode Enkripsi";
303             JOptionPane.showMessageDialog(null, pesan);
304         }
305     }
306
```

*Method* 'prosesDekripsi' digunakan untuk menjalankan proses dekripsi sesuai dengan pilihan dari komponen RadioButton.

15. Klik kanan komponen btnEnkrip pada panel navigator → Events → Actioin --. actionPerformed. Ketikkan kode program berikut ini.

```
484     private void btnEnkripActionPerformed(java.awt.event.ActionEve
            String plain = null;
486         if(txtPlain.getText().isEmpty() == true){
487             pesan = "Masukkan data plaintext!";
488             JOptionPane.showMessageDialog(null, pesan);
489
```

```
490             } else if(txtKey.getText().isEmpty() == true) {
491                 pesan = "Masukkan kunci (key)!";
492                 JOptionPane.showMessageDialog(null, pesan);
493 
494             } else {
495                 plain = txtPlain.getText().toUpperCase();
496                 prosesEnkrip(plain);
497             }
498         }
499 
```

16. Klik kanan komponen btnEnkrip pada panel navigator → Events → Actioin --. actionPerformed. Ketikkan kode program berikut ini.

```
499 
500     private void btnDekripActionPerformed(java.awt.event.Act
501         String cipher = null;
502         if(txtCipher.getText().isEmpty()){
503             pesan = "Masukkan data ciphertext!";
504             JOptionPane.showMessageDialog(null, pesan);
505         } else if(txtKey.getText().isEmpty() == true) {
506             pesan = "Masukkan kunci (key)!";
507             JOptionPane.showMessageDialog(null, pesan);
508         } else {
509             cipher = txtCipher.getText().toUpperCase();
510             prosesDekrip(cipher);
511         }
512     }
```

17. Compile (F9) dan Run Program melalui menu Run → Run File (Shift + F6).

# MODUL 8
# PROJEK APLIKASI

**Pokok Pembahasan :**

- **Aplikasi MDI**
- **Aplikasi Input-Data1**
- **Aplikasi Input-Data2**
- **Aplikasi CRUD**
- **Aplikasi Hangman**
- **Aplikasi Alarm**
- **Aplikasi Download Manager**

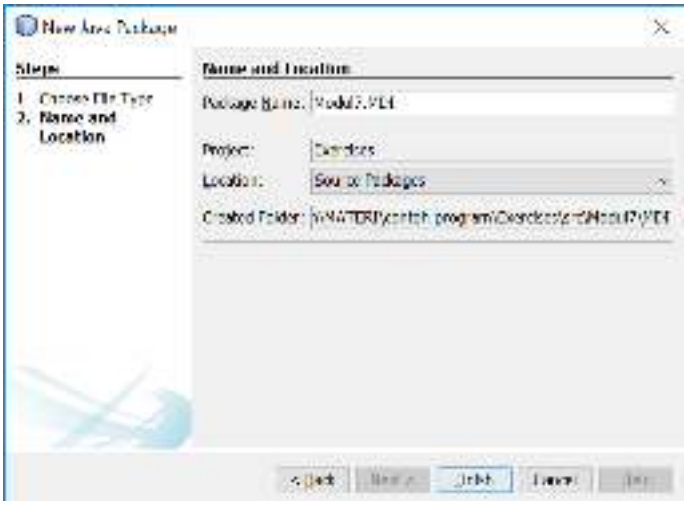
## 8.1 Aplikasi MDI

Sesuai namanya, aplikasi MDI (Multiple Document Interface) merupakan aplikasi yang bisa menampilkan beberapa dokumen sekaligus. Contoh aplikasi MDI adalah program aplikasi Micrososft word, Microsoft Excel dan program aplikasi lainnya. Di dalam aplikasi MDI, ada 2 pembagian Form.

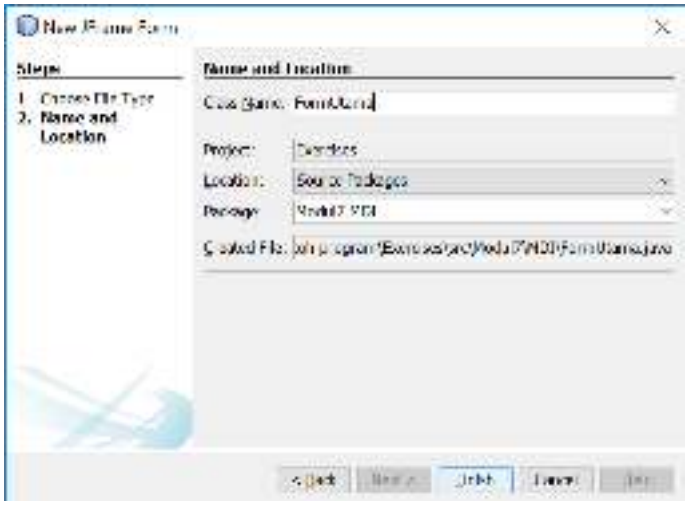
- **Form Induk** : Form terluar yang dijadikan tempat (wadah) untuk menampilkan form yang lain.
- **Form Anak** : Form yang ditampilkan di dalam form Induk. Form ini terpasang seolah-olah menempel di dalam Form induk dan tidak dapat keluar dari tampilan Form induk.

Berikut contoh projek menggunakan MDI :

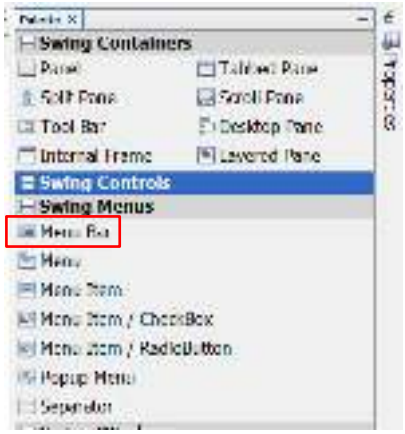
1. Tambahkan paket (*Java Package*) pada projek 'Exercises'. Beri nama paket dengan 'Modul7.MDI'.



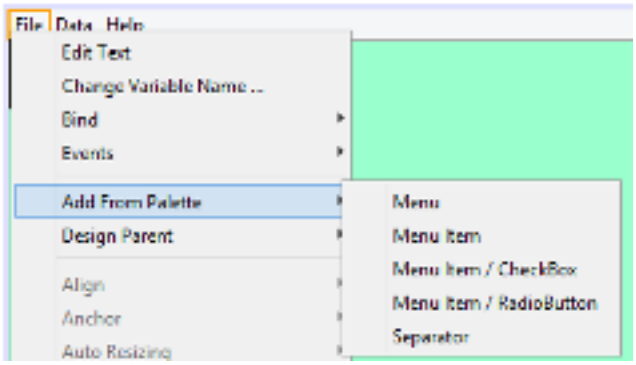

2. Tambahkan 'JFrame Form' dengan langkah-langkah seperti pada percobaan modul 1 (menambah frame form). Beri nama JFrame Form dengan 'FormUtama'.




3. Tambahkan komponen Desktop Pane dan Menu Bar pada ‘FormUtama’ seperti pada gambar di bawah ini.

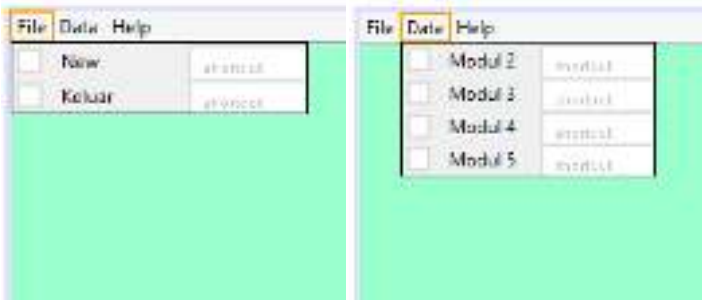


4. Tambahkan JMenuItem di Menu File dengan Cara klik kanan Menu File → Add From Palette → Menu Item




5. Ganti Text JMenuItem dengan 'New' caranya klik kanan JMenuItem1 → Edit Text

6. Ulangi Langkah 4 dan 5, buat tampilannya seperti berikut :



7. Buat Variable Global untuk dapat diakses semua method yang ada didalam class FormUtama.

8. Tambahkan perintah berikut di dalam *constructor* FormUtama :

9. Buat Method baru dengan nama "FormBaru" dengan isi berikut :

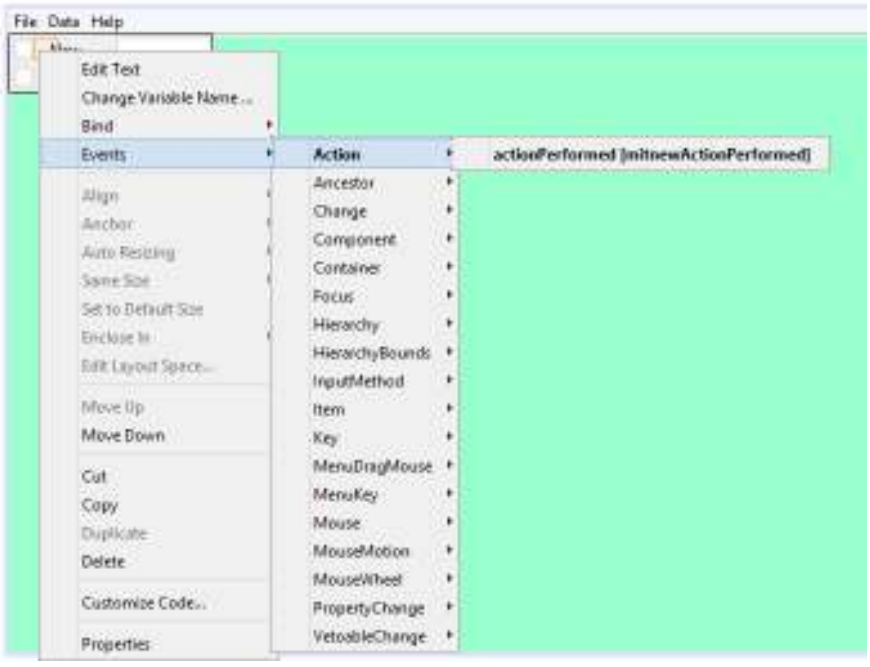
```
17  public class FormUtama extends javax.swing.JFrame {
18  private int jml;
19  private String judul;
20
21      /**
22       * Creates new form FormUtama
23       */
24      public FormUtama() {
25      setTitle("MDI Form");
26      setExtendedState(MAXIMIZED_BOTH);
27          initComponents();
28      }
29
30      private void FormParu(){
31       try{
32         JInternalFrame jin = new JInternalFrame(judul,false, true,true);
33          jin.setDefaultCloseOperation(DISPOSE_ON_CLOSE);
34          JPanel pn = new JPanel();
35          jin.setBounds(10, 10, 500, 500);
36          jDesktopPanel.add(jin);
37          jin.setVisible(true);
38      }catch (Exception e){
39          JOptionPane.showMessageDialog(null, e);
40      }
41  }
```

8

9

10

Isi Perintah pada Menu New ( JMenuItem ) dengan cara Klik kanan pada menu New → Events → Action → actionPerformed.



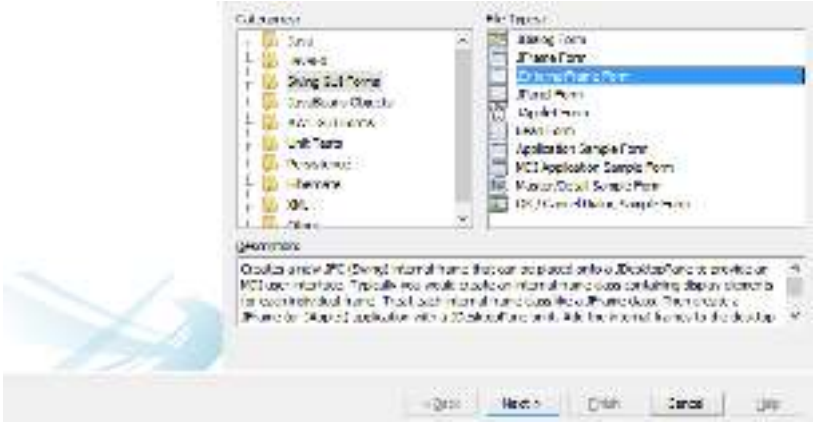
Ketikkan kode program seperti berikut :

```
165     private void mitnewActionPerformed(java.awt.event.ActionEvent
166         // TODO add your handling code here:
167         jml = jml + 1;
168         judul = "Form Baru "+jml;
169         FormBaru();
```

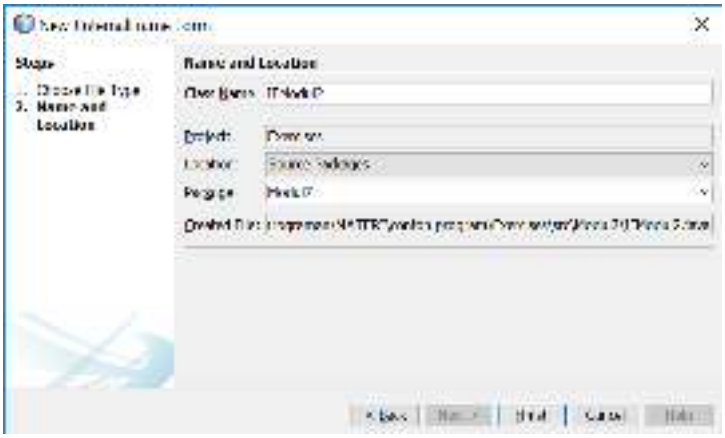
10. Isi Perintah pada Menu Keluar (JMenuItem) dengan cara Klik kanan pada menu New → Events → Action → actionPerformed.

```
193
194     private void mitkeluarActionPerformed(java.awt.event.ActionEvent
195         // TODO add your handling code here:
196         System.exit(0);
197     }
```

11. Tambahkan JInternalFrame Form pada project pilih menu File → New File pilih Project anda, kemudian pada Categories pilih Swing GUI Forms dan pada File Types pilih JInternalFrame Form kemudian klik Next.

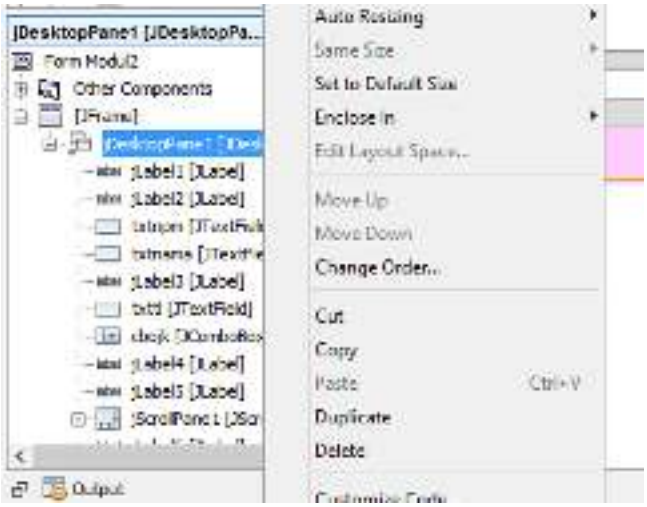


12. Isikan Class name “IFModul2” dan Pakage “Modul7” kemudian klik Finish.

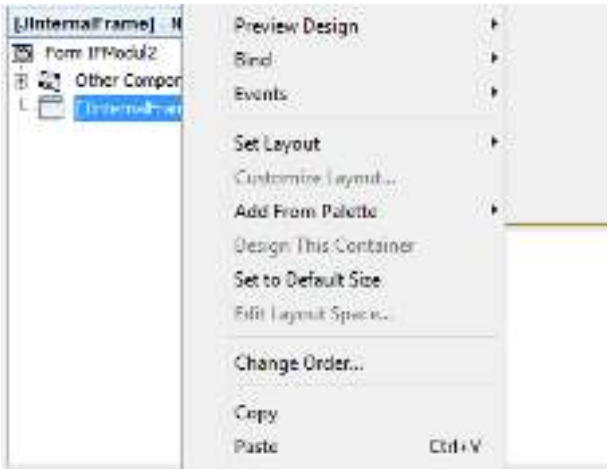


Kemudian desain sama seperti program pada modul2, atau copy paste dengan cara :

Buka Modul2 yang sebelumnya kemudian pilih navigator item jDesktopPane1 → klik kanan → Copy.



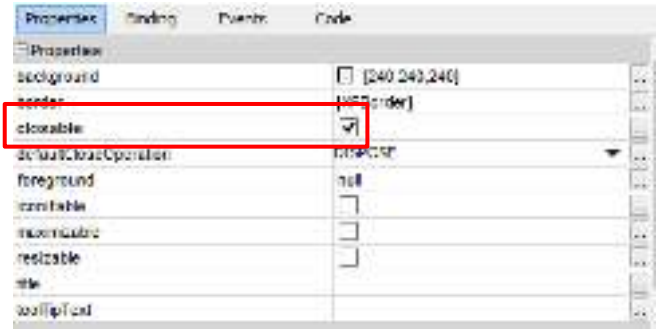

Buka IFModul2 yang sebelumnya kemudian pilih navigator item jInternalFrame → klik kanan → Paste.




13. Lakukan hal yang sama (ikuti langkah-langkah 11 s/d 12) untuk membuat file jInternalFrame Form untuk 'IFModul3', 'IFModul4', 'IFModul5'.

PERHITUNGAN GAJI PEGAWAI

14. Ubah nilai properti pada masing-masing InternalFrame, pilih navigator item jInternalFrame → klik kanan Properties. Centang Closable sehingga bernilai true.




15. Kembali pada FormUtama kemudian berikan perintah-perintah untuk memanggil form-form yang telah dibuat diatas.

16. Isi Perintah pada Menu Modul2 (JMenuItem) dengan cara Klik kanan pada menu Modul 2 → Events → Action → actionPerformed.

```
171
172     private void mimodul2ActionPerformed(java.awt.event.ActionEvent evt) {
173         // TODO add your handling code here:
174         IFModul2 form2 = new IFModul2();
175         jDesktopPane1.add(form2);
176         Dimension parentSize = jDesktopPane1.getSize();
177         Dimension childSize = form2.getSize();
178         form2.setLocation((parentSize.width - childSize.width)/2,
179                 (parentSize.height - childSize.height)/2);
180         form2.setVisible(true);
181     }
```

17. Isi Perintah pada Menu Modul3 (JMenuItem) dengan cara Klik kanan pada menu Modul 3 → Events → Action → actionPerformed.

```
183     private void mimodul3ActionPerformed(java.awt.event.ActionEvent evt) {
184         // TODO add your handling code here:
185         IFModul3 form3 = new IFModul3 ();
186         jDesktopPane1.add(form3);
187         Dimension parentSize = jDesktopPane1.getSize();
188         Dimension childSize = form3.getSize();
189         form3.setLocation((parentSize.width - childSize.width)/2,
190                 (parentSize.height - childSize.height)/2);
191         form3.setVisible(true);
192     }
```

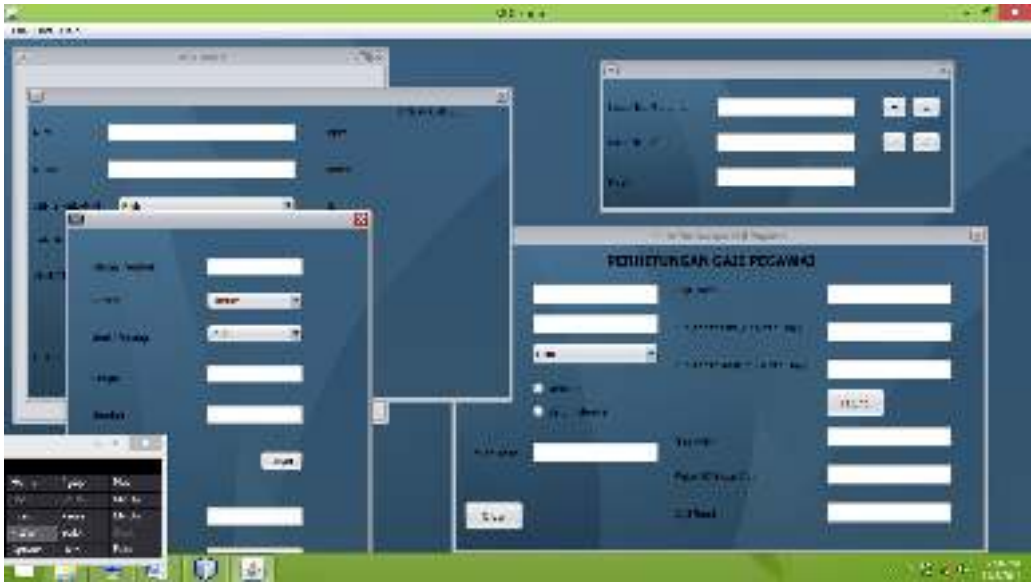
18. Isi Perintah pada Menu Modul4 (JMenuItem) dengan cara Klik kanan pada menu Modul 4 → Events → Action → actionPerformed.

```
    private void mitmodul4ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        IFModul4 form4 = new IFModul4 ();
        jDesktopPanel.add(form4);
        Dimension parentSize = jDesktopPanel.getSize();
        Dimension childSize = form4.getSize();
        form4.setLocation((parentSize.width - childSize.width)/2,
                (parentSize.height - childSize.height)/2);
        form4.setVisible(true);
    }
```

19. Isi Perintah pada Menu Modul5 (JMenuItem) dengan cara Klik kanan pada menu Modul 5 → Events → Action → actionPerformed.

```
    private void mitmodul5ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        IFModul5 form5 = new IFModul5 ();
        jDesktopPanel.add(form5);
        Dimension parentSize = jDesktopPanel.getSize();
        Dimension childSize = form5.getSize();
        form5.setLocation((parentSize.width - childSize.width)/2,
                (parentSize.height - childSize.height)/2);
        form5.setVisible(true);
    }
```

20. Run Program melalui FormUtama menu Run → Run File (Shift + F6).



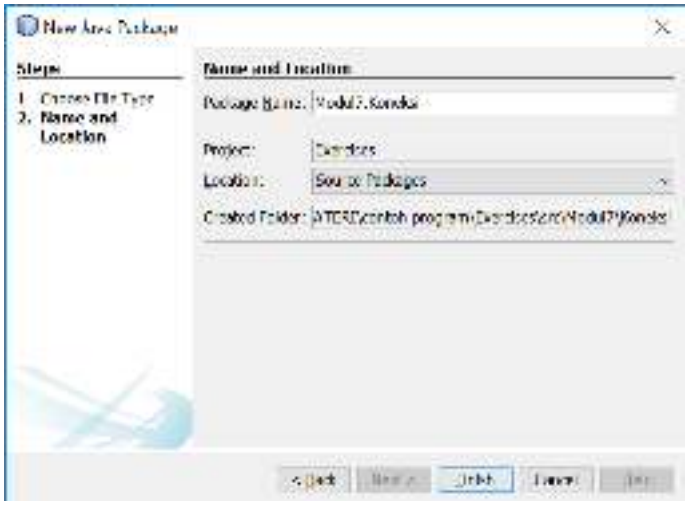
### 8.2 Aplikasi Input-Data1

Berikut ini adalah cara untuk membuat program input data customer dari Java dan Netbeans dan menggunakan database MySQL

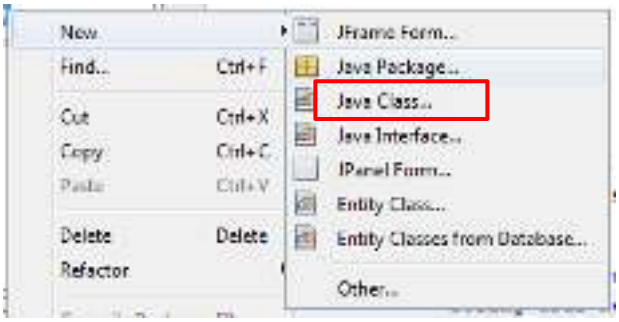
1. Buatlah database dan tabel terlebih dahulu :

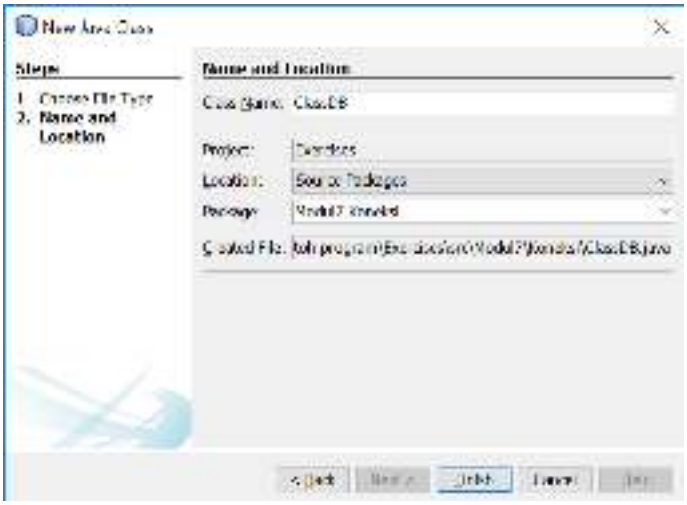
```
CREATE DATABASE dbjava;
USE dbjava;
CREATE TABLE tcustomer (
id VARCHAR(6) PRIMARY KEY,
nama VARCHAR(25),
nohp VARCHAR(14),
alamat VARCHAR(100));
```

2. Tambahkan paket (*Java Package*) pada projek 'Exercises'. Beri nama paket dengan 'Modul7.Koneksi'.



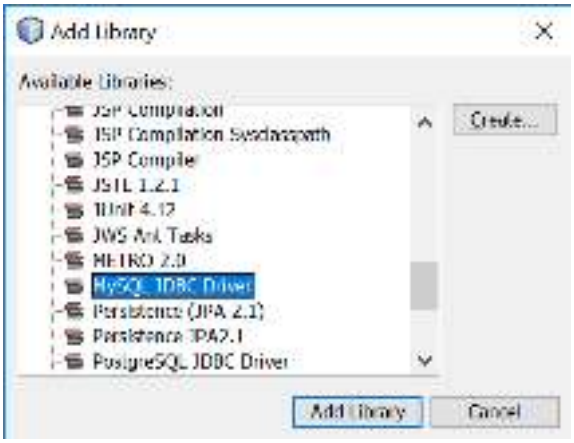

3. Buatlah JClass caranya, klik kanan pakage Modul7.Koneksi, New → Java Class. Beri nama 'ClassDB', kemudian klik Finish.






4. Tambahkan Library konektor MySQL pada Libraries Project ‘Exercises’ dengan cara klik kanan pada folder Libraries → Add Library

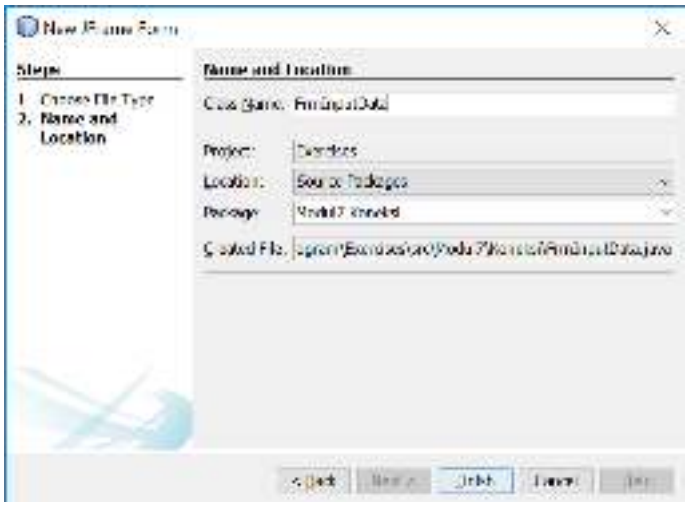
Pilih MySQL JDBC Driver → klik tombol ‘Add Library’




5. Ketikkan kode progam berikut ini didalam ‘ClassDB’.

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;



public class ClassDB {
    private static Connection koneksi;
    public static Connection getkoneksi() {
        if (koneksi==null) {
            try {
                String url=new String();
                String user=new String();
                String password=new String();
                url="jdbc:mysql://localhost:3306/dbjava";
                user="root";
                password="1234";
                DriverManager.registerDriver(new com.mysql.jdbc.Driver());
                koneksi=DriverManager.getConnection(url,user,password);
            }catch (SQLException e) {
                System.out.println("Error membuat koneksi");
            }
        }
     return koneksi;
    }
}
```

6. Tambahkan JFrame Form pada paket 'Modul7.Koneksi' dan beri nama 'FrmInputData'.




7. Tambahkan beberapa komponen seperti yang ada di dalam tabel berikut ini:

| Objek | Properties | Nilai |
|---|---|---|
| jDesktopPane1 | Background | Purple [204,0,255] |
| | Border | Title Border = Data Customer |
| JLabel1 | text | ID |

| | | |
|---|---|---|
| JLabel2 | text | Nama |
| JLabel3 | text | No Hp |
| JLabel4 | text | Alamat |
| JTextField1 | text<br>Variable Name | Txtid |
| JTextField2 | text<br>Variable Name | Txtnama |
| JTextField3 | text<br>Variable Name | Txthp |
| JTextArea | text<br>Variable Name | Txtalamat |
| JButton1 | text<br>Variable Name | Simpan<br>Btnsimpan |
| JButton2 | text<br>Variable Name | Batal<br>Btnbatal |

8. Susunan posisi dan ukuran komponen seperti pada gambar dibawah ini:

9. Deklarasikan class import berikut :

```
6   package Modul7.Koneksi;
7
8   import java.sql.*;
9   import javax.swing.JOptionPane;
10  import Modul7.Koneksi.ClassDB;
11
```

10. Selanjutnya Ketikkan kode berikut ini:

```
27      public FrmInputData() {
28          initComponents();
29          bacaId();
30          txtid.setEnabled(false);
31          btnsavv.setEnabled(false);
32      }
33
34      private void bacaId() {
35          int kode = 0;
36          try {
37              Class.forName("com.mysql.jdbc.Driver");
38              try (Connection cn = DriverManager.getConnection("jdbc:mysql://localhost/dbjava", "root", "");
39                      Statement smt = cn.createStatement()) {
40                  String sql = "SELECT max(Id) as id FROM tcustomer";
41                  ResultSet rs = smt.executeQuery(sql);
42                  if (rs.next()) {
43                      kode = rs.getInt("id");
44                  } else {
45                  }
46                  txtid.setText("000" + (kode + 1));
47              }
48          } catch (ClassNotFoundException | SQLException e) {
49          }
50      }
```

Keterangan Kode:

Method dengan nama 'bacaId', digunakan untuk membaca field id pada tabel tcustomer.

```
private void simpan() {
        String id = this.txtid.getText();
        String nama = this.txtnama.getText();
        String nohp = this.txthp.getText();
        String alamat = this.txtalamat.getText();
        String st = "Free";
```

```
        if ("".equals(this.txtid.getText()) || ("".equals(this.txtnama.getText()))
                || ("".equals(this.txthp.getText())) || ("".equals(this.txtalamat.getText()))) {
            JOptionPane.showMessageDialog(this, "Lengkapi data");
        } else {
            try {
                Connection c = ClassDB.getkoneksi();
                Statement s = c.createStatement();
                String sql = "Insert into tcustomer values (?,?,?,?)";
                try (com.mysql.jdbc.PreparedStatement
                    p= (com.mysql.jdbc.PreparedStatement) c.prepareStatement(sql))
                {
                    p.setString(1, id);
                    p.setString(2, nama);
                    p.setString(3, nohp);
                    p.setString(4, alamat);
                    p.executeUpdate();
                }
                JOptionPane.showMessageDialog(this, "Sukses Tambah data");
            } catch (SQLException e) {
                System.out.println("Terjadi kesalahan " + e);
            } finally {
                bersih();
            }
        }
    }
```

Keterangan Kode:

Method dengan nama simpan, digunakan untuk menyimpan data pada tabel tcustomer.

11. Selanjutnya di bagian bawah kode perintah:

```
private void bersih() {
        bacaId();
        txtnama.setText(null);
        txthp.setText(null);
        txtalamat.setText(null);
        btnsimpan.setEnabled(false);
        txtid.setEnabled(false);
    }
```

Keterangan kode:
Method dengan nama bersih, maksudnya yaitu untuk membersihkan/ mengosongkan nilai pada tiap-tiap JTextField dengan tujuan untuk memulai proses baru.

12. Klik kanan pada btnsimpan setelah itu pilih Events → Action → actionPerformed. Ketikkan kode perintah:

```
simpan();
```

13. Klik kanan pada btnclear setelah itu pilih Events → Action → actionPerformed. Ketikkan kode perintah:

```
bersih();
```

14. Klik kanan pada txtnama setelah itu pilih Events → Key → KeyTyped. Ketikkan kode perintah:

```
btnsimpan.setEnabled(true);
```

15. Kompile dan jalankan dari menu Run → Run File atau tekan Shift + F6.
16. Id akan otomatis terisi. Isikan Nama, No HP, Alamat kemudian pilih tombol Simpan.
17. Klik Tombol Clear untuk mengosongkan TextField.

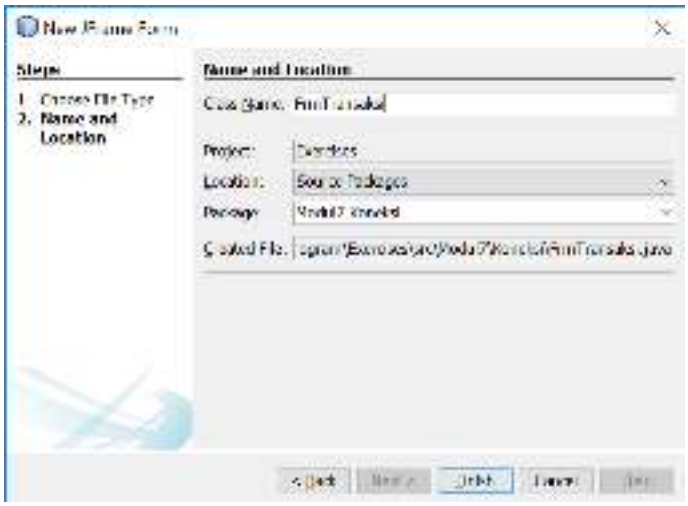
### 8.3 Aplikasi Input-Data2

Berikut ini adalah langkah-langkah untuk membuat program input data transaksi dari Java dan Netbeans dan menggunakan database MySQL. Namun sebelum itu anda telah menyelesaikan aplikasi input-data1 sebelumnya.

1. Buatlah database dan tabel terlebih dahulu :

```
CREATE DATABASE dbjava;
USE dbjava;
CREATE TABLE tpenjualan (
   invoice VARCHAR(12) PRIMARY KEY,
   tanggal DATE,
   id VARCHAR(6),
   nama VARCHAR(30),
   status VARCHAR(15),
   barang VARCHAR(30),
   harga VARCHAR(12),
   jumlah VARCHAR(6),
   diskon VARCHAR(12),
   bayar VARCHAR(12)
 )
```

2. Tambahkan JFrame Form pada paket 'Modul7.Koneksi' dan beri nama 'FrmTransaksi'.




3. Selanjutnya, tambahkan beberapa komponen seperti yang ada di dalam tabel berikut ini:

| Objek | Properties | Nilai |
|---|---|---|
| jDesktopPane1 | Background | Green |
| JLabel1 | text | Tanggal |

| | | |
|---|---|---|
| JLabel2 | text | Invoice |
| JLabel3 | text | Id Customer |
| JLabel4 | text | Nama |
| JLabel5 | text | No Hp |
| JLabel6 | text | Alamat |
| JLabel7 | text | Status Member |
| JLabel8 | text | Jenis Barang |
| JLabel9 | text | Harga |
| JLabel10 | text | Jumlah |
| JLabel11 | text | Diskon |
| JLabel12 | text | Total Bayar |
| JTextField1 | Variable Name | txttanggal |
| JTextField2 | Variable Name | txtinvoice |
| JTextField3 | Variable Name | txtid |
| JTextField4 | Variable Name | txtnama |
| JTextField5 | Variable Name | txthp |
| JTextArea1 | Variable Name | txtalamat |
| JComboBox1 | Variable Name | cbostatus |
| JComboBox2 | Variable Name | cbojenis |
| JTextField6 | Variable Name | txtharga |
| JTextField7 | Variable Name | txtjumlah |
| JTextField8 | Variable Name | txtdiskon |
| JTextField9 | Variable Name | txttotal |
| JButton1 | text<br>Variable Name | Bayar<br>btnbayar |
| JButton2 | text<br>Variable Name | Simpan<br>btnsimpan |

4. Susunan posisi dan ukuran komponen seperti pada gambar dibawah ini:

5. Deklarasikan class import berikut :

```
6   package Modul7.Koneksi;
7
8   import java.awt.HeadlessException;
9   import java.sql.*;
10  import java.text.SimpleDateFormat;
11  import java.util.Date;
12  import javax.swing.JOptionPane;
13  import Modul7.Koneksi.ClassDB;
```

6. Selanjutnya Ketikkan kode program berikut ini:

```
18  public class FrmTransaksi extends javax.swing.JFrame {
19
20      /**
21       * Creates new form FrmTransaksi
22       */
23      public FrmTransaksi() {
24          initComponents();
25          invoice();
26          Date current = new Date ();
27          SimpleDateFormat frmt = new SimpleDateFormat ("dd/MM/yyyy");
28          String dateString1 = frmt.format(current);
29          txttanggal.setText(dateString1);
30      }
```

7. Lanjutkan ketik dibahwahnya kode program berikut ini :

```
private void invoice(){
        String kode = "14000" ;
        try {
        Class.forName("com.mysql.jdbc.Driver");
            try (Connection cn =
DriverManager.getConnection("jdbc:mysql://localhost/dbjava", "root", "");
                Statement smt = cn.createStatement()) {
                String sql = "SELECT max(invoice) as iv FROM tpenjualan";
                ResultSet rs = smt.executeQuery(sql) ;
                if (rs.next()) {
                }  else {
                }
                txtinvoice.setText((kode+1));
            }
        } catch (ClassNotFoundException | SQLException e) {
        }
    }
```

Keterangan Kode:

Method dengan nama invoice, digunakan untuk mencari data id invoice pada tabel tpenjualan.

```
private void cariId(){
        try {
        Class.forName("com.mysql.jdbc.Driver");
            try (Connection cn =
DriverManager.getConnection("jdbc:mysql://localhost/dbjava", "root", "");
                Statement smt = cn.createStatement()) {
                String sql = "SELECT * FROM tcustomer where id ='"+txtid.getText()+"' " ;
                ResultSet rs = smt.executeQuery(sql) ;
                if (rs.next()) {
                txtnama.setText(rs.getString("Nama"));
                txthp.setText(rs.getString("nohp"));
                txtalamat.setText(rs.getString("alamat"));
                } else {
                JOptionPane.showMessageDialog(null,"Id Customer Tidak
Terdaftar","INFORMASI",JOptionPane.INFORMATION_MESSAGE) ;
                txtid.setText(null);
                txtnama.setText(null);
                txtid.requestFocus() ;
                }
            }
        } catch (ClassNotFoundException | SQLException | HeadlessException e) {
        }
    }
```

```
private void simpan(){
    String iv=this.txtinvoice.getText();
    String tgl=this.txttanggal.getText().toString();
    String id=this.txtid.getText();
    String nama=this.txtnama.getText();
    String status=this.cbostatus.getSelectedItem().toString();
    String barang=this.cbojenis.getSelectedItem().toString();
    String harga=this.txtharga.getText();
    String jumlah=this.txtjumlah.getText();
    String diskon=this.txtdiskon.getText();
    String bayar=this.txttotal.getText();

    String st="Free";
    if ("".equals(this.txtinvoice.getText()) || ("".equals(this.txttanggal.getText()))
            || ("".equals(this.txtid.getText()) || ("".equals(this.txtnama.getText()))
            ||                ("".equals(this.cbostatus.getSelectedItem().toString()))                ||
("".equals(this.cbojenis.getSelectedItem().toString()))
            || ("".equals(this.txtharga.getText())) || ("".equals(this.txtjumlah.getText()))
            || ("".equals(this.txtdiskon.getText())) || ("".equals(this.txttotal.getText())))
            {
            JOptionPane.showMessageDialog(this, "Lengkapi data");
        }else{
    try {
            Connection c=ClassDB.getkoneksi();
            Statement s= c.createStatement();
            String sql="Insert into tpenjualan values (?,?,?,?,?,?,?,?,?,?)";
        try  (com.mysql.jdbc.PreparedStatement  p  =  (com.mysql.jdbc.PreparedStatement)
c.prepareStatement(sql)) {
            p.setString(1, iv);
            p.setString(2, tgl);
            p.setString(3, id);
            p.setString(4, nama);
            p.setString(5, status);
            p.setString(6, barang);
            p.setString(7, harga);
            p.setString(8, jumlah);
            p.setString(9, diskon);
            p.setString(10, bayar);
            p.executeUpdate();
        }
            JOptionPane.showMessageDialog(this, "Invoice Tersimpan");
        }catch(SQLException e) {
            System.out.println("Gagal Menyimpan " + e);
        }finally{
     bersih();
    }
    }
    }
```

```
private void bayar(){
        txtjumlah.setText("1");
        int harga=Integer.parseInt(txtharga.getText());
        int jumlah=Integer.parseInt(txtjumlah.getText());
        int diskon, nilaidiskon, total ;

        if(cbostatus.getSelectedItem()=="Silver"){
        diskon = ((5 * harga)/100);
        nilaidiskon = harga - diskon ;
        total = (nilaidiskon * jumlah) ;

        }else{
         diskon = ((20 * harga)/100);
        nilaidiskon = harga - diskon ;
        total = (nilaidiskon * jumlah) ;
        }
        txtdiskon.setText(""+diskon);
        txttotal.setText(""+total);
    }

    private void bersih(){
        invoice();
        txtid.setText(null);
        txtnama.setText(null);
        txthp.setText(null);
        txtalamat.setText(null);
        cbostatus.setSelectedItem("Pilih");
        txtharga.setText(null);
        txtjumlah.setText("1");
        txtdiskon.setText(null);
        txttotal.setText(null);
    }
```

8. Klik kanan pada cbojenis setelah itu pilih Events → Action → actionPerformed. Ketikkan kode program berikut ini:

```
385     private void cbojenisActionPerformed(java.awt.event.ActionEvent evt) {
386         // TODO add your handling code here:
387         if(cbojenis.getSelectedItem()=="iPad"){
388             txtharga.setText("4999000");
389         }else if (cbojenis.getSelectedItem()=="iPhone"){
390             txtharga.setText("7999000");
391         }else if (cbojenis.getSelectedItem()=="iPod"){
392             txtharga.setText("2999000");
393         }else{
394             txtharga.setText("");
395         }
396     }
```

9. Klik kanan pada btnbayar setelah itu pilih Events → Action → actionPerformed. Ketikkan kode program berikut ini :

```
376     private void btnbayarActionPerformed(java.awt.event.ActionEvent evt) {
377         // TODO add your handling code here:
378         bayar();
379     }
```

10. Klik kanan pada txtnama setelah itu pilih Events → Action → MouseClicked. Ketikkan kode program berikut ini:

```
402     private void txtnamaMouseClicked(java.awt.event.MouseEvent evt) {
403         // TODO add your handling code here:
404         cariId();
405     }
406
407     private void txtnamaMouseReleased(java.awt.event.MouseEvent evt) {
408         // TODO add your handling code here:
409         cariId();
410     }
```

11. Klik kanan pada btnsimpan setelah itu pilih Events → Action → actionPerformed. Ketikkan kode program berikut ini:

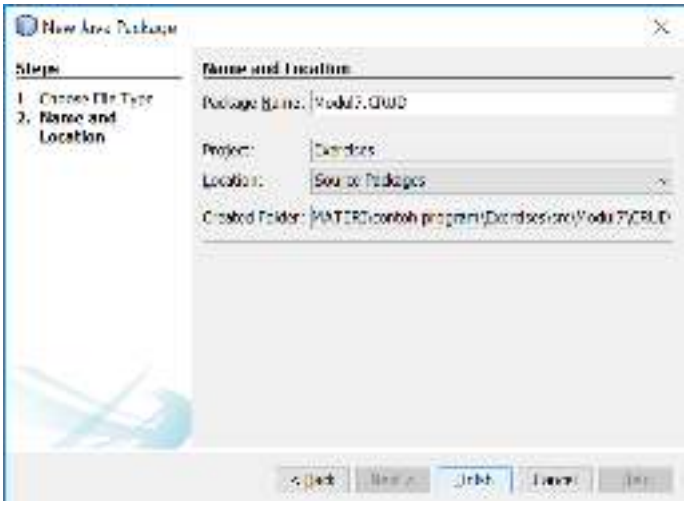
```
430     private void btnsimpanActionPerformed(java.awt.event.ActionEvent evt) {
431         // TODO add your handling code here:
432         simpan();
433         bersih();
434     }
```

12. Compile dan jalankan dari menu Run → Run File atau tekan Shift + F6.

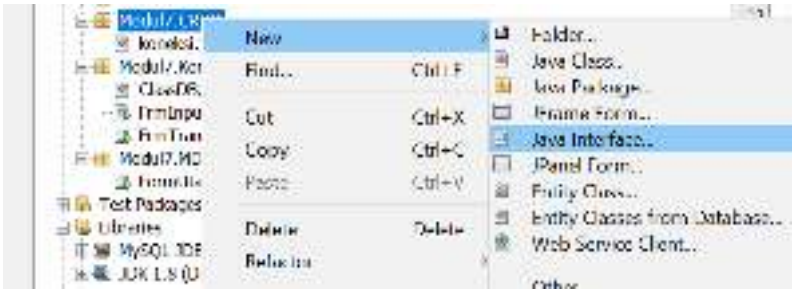
## 8.4 Aplikasi CRUD

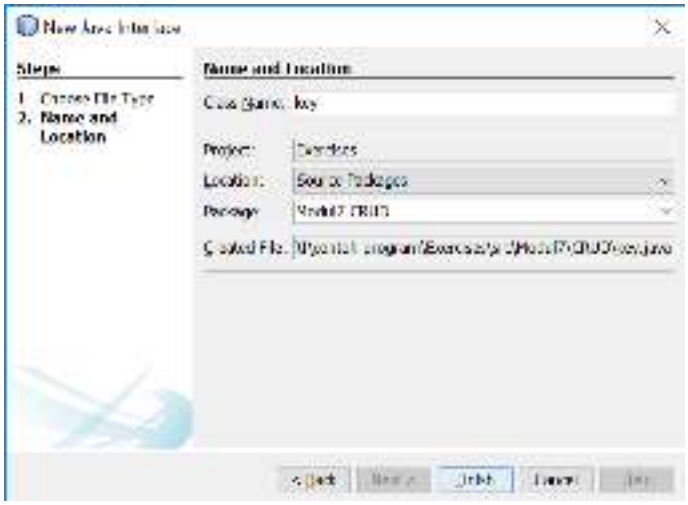
Dalam pemrograman komputer, CRUD (*create, read, update and delete*) adalah empat fungsi dasar dari penyimpanan persisten. Berikut langkah-langkah untuk membuat aplikasi CRUD.

1. Tambahkan paket (*Java Package*) pada projek 'Exercises'. Beri nama paket dengan 'Modul7.CRUD'.




2. Tambahkan Java Interface pada paket 'Modul7.CRUD' dan beri nama dengan nama "key".



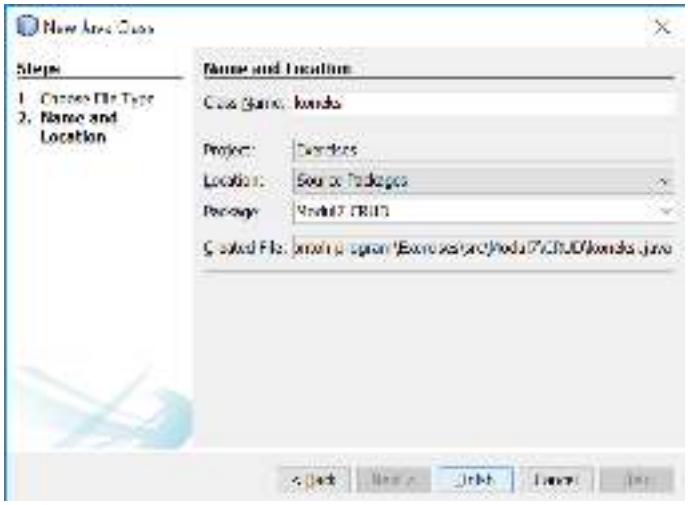



Tulis kode program berikut didalam interface key.

```
interface key= {
    String url ="jdbc:mysql://localhost:3306/dborganisasi";
    String username ="root";
    String password ="";
    String driver ="com.mysql.jdbc.Driver";

}
```

3. Pastikan MySQL JDBC Driver telah ditambahkan pada Libraries projek ‘Exercises’.

4. Tambahkan Java Class dan beri nama dengan “koneksi” pada paket Modul7.CRUD.

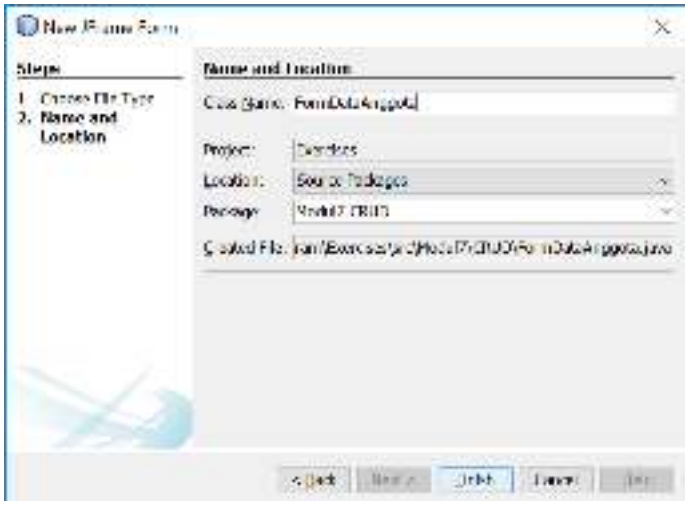



5. Tulis kode program berikut ini didalam class koneksi.

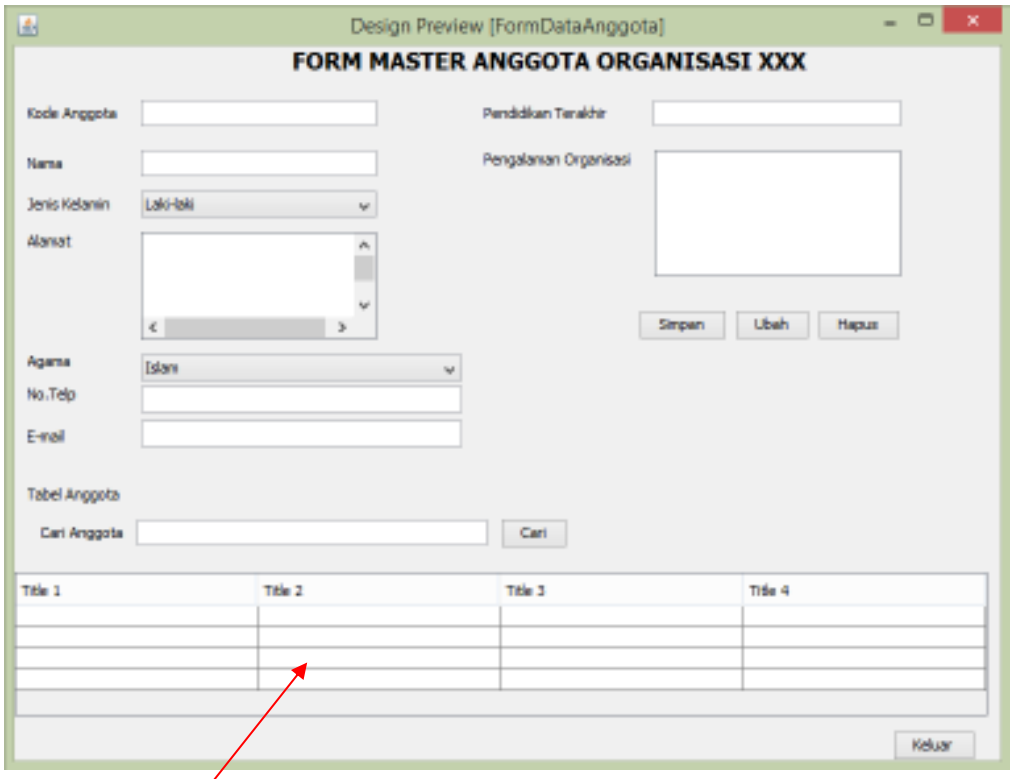
```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
/**...4 lines */
```

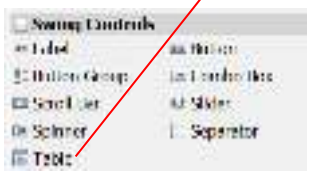
```
public class koneksi implements keys{
    Connection con;
    public koneksi(){
        try{
            Class.forName(driver).newInstance();
            con = DriverManager.getConnection(url, username, password);
            if(con==null){
                System.out.println("Koneksi Gagal");
            }else{
                System.out.println("Koneksi Berhasil");
            }
        }catch (ClassNotFoundException e){
            System.out.println(""+ e.getMessage());
        } catch (InstantiationException e) {
            System.out.println(""+ e.getMessage());
        } catch (IllegalAccessException e) {
            System.out.println(""+ e.getMessage());
        } catch (SQLException e) {
            System.out.println(""+ e.getMessage());
        }
    }
}
```

6. Tambahkan JFrame Form pada paket Modul7.CRUD dan beri nama dengan "FormDataAnggota".




Desainlah form seperti pada gambar di bawah ini :



7. Tulis kode program berikut ini didalam FormDataAnggota.

```
12 import java.sql.Statement;
13 import java.sql.ResultSet;
14 import javax.swing.table.DefaultTableModel;
```

```
20 public class FormDataAnggota extends javax.swing.JFrame {
21     koneksi k;
22     ResultSet resultset;
23     Statement statement;
24
25     /**
26      * Creates new form FormDataAnggota
27      */
28     public FormDataAnggota() {
29         initComponents();
30         k = new koneksi();
31         //memanggil data select()
32         tabel();
33     }
```

8. Tambahkan dan Ketik method-method berikut ini dibawah konstruktor :
   Terdapat 6 method yaitu **: tabel(), hapusForm(), simpan(), cari(), ubah() hapusData()**

   Method table()

```
private void tabel() {
//listing untuk mengganti nama judul didalam tabel
//dan untuk menambahkan jumlah kolom pada tabel
Object header[] = {"Kode Anggota", "Nama", "Jenis Kelamin",
    "Alamat", "Agama", "No.telp", "Email", "Pendidikan",
    "Pengalaman"};

//listing untuk menyesuaikan lebar
//kolom dan lebar baris pada tabel
DefaultTableModel defaultTableModel
        = new DefaultTableModel(null, header);
tblAnggota.setModel(defaultTableModel);

//listing untuk menambahkan jumlah baris pada tabel
int baris = tblAnggota.getRowCount();
for (int i = 0; i < baris; i++) {
    defaultTableModel.removeRow(i);
}
```

```
//listing untuk menghubungkan database dengan java
String sql = "select * from anggota";
try {
    statement = k.con.createStatement();
    resultset = statement.executeQuery(sql);
    while (resultset.next()) {
    //setelah string diambil seluruh dari field
        //yang ada di dalam database tabel anggota
        String kode = resultset.getString(1);
        String nama = resultset.getString(2);
        String jeniskelamin = resultset.getString(3);
        String alamat = resultset.getString(4);
        String agama = resultset.getString(5);
        String notelp = resultset.getString(6);
        String email = resultset.getString(7);
        String pendidikan = resultset.getString(8);
        String pengalaman = resultset.getString(9);
        String kolom[] = {kode, nama, jeniskelamin, alamat,
            agama, notelp, email, pendidikan, pengalaman};
        defaultTableModel.addRow(kolom);
    }
} catch (Exception e) {
    System.out.println("" + e.getMessage());
}
}
```

Method simpan()

```
//listing untuk button simpan
private void simpan() {
    //listing memanggil combo box
    String idJenkel = cmbJenkel.getSelectedItem().toString();
    String Agama = cmbAgama.getSelectedItem().toString();

    //listing koneksi sql dan Java untuk simpan
    String sql_simpan = "insert into anggota values('"
            + txtKodeAnggota.getText() + "','"
            + txtNama.getText() + "','"
            + idJenkel + "','"
            + txtAlamat.getText() + "','"
            + Agama + "','"
            + txtTelepon.getText() + "','"
            + txtEmail.getText() + "','"
            + txtPendidikan.getText() + "','"
            + txtPengalamanOrganisasi.getText() + "')";
```

```
    //sinkronisasi java dan sql
    try {
        statement.executeUpdate(sql_simpan);
    } catch (Exception e) {
        System.out.println(" " + e.getMessage());
    }
}
```

Method hapusForm()

```
private void hapusForm() {
    txtKodeAnggota.setText(null);
    txtNama.setText(null);
    txtAlamat.setText(null);
    txtTelepon.setText(null);
    txtEmail.setText(null);
    txtPendidikan.setText(null);
    txtPengalamanOrganisasi.setText(null);
}
```

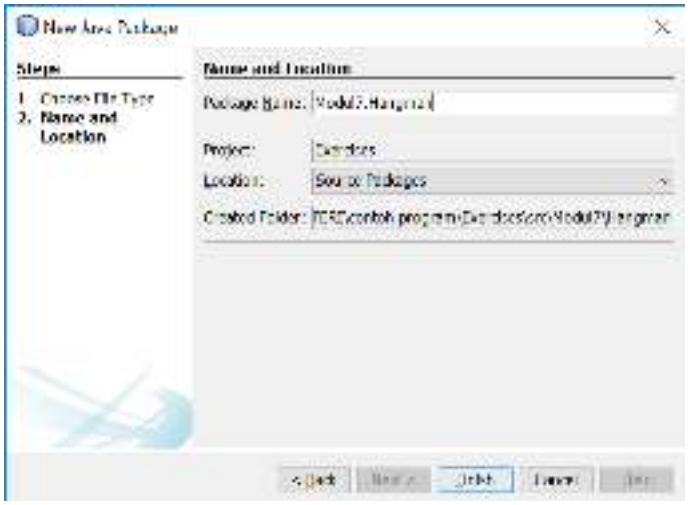
### 7.4.1 Tugas :

Lengkapi listing program form diatas sehingga dapat berfungsi sesuai konsep yang dimaksud.

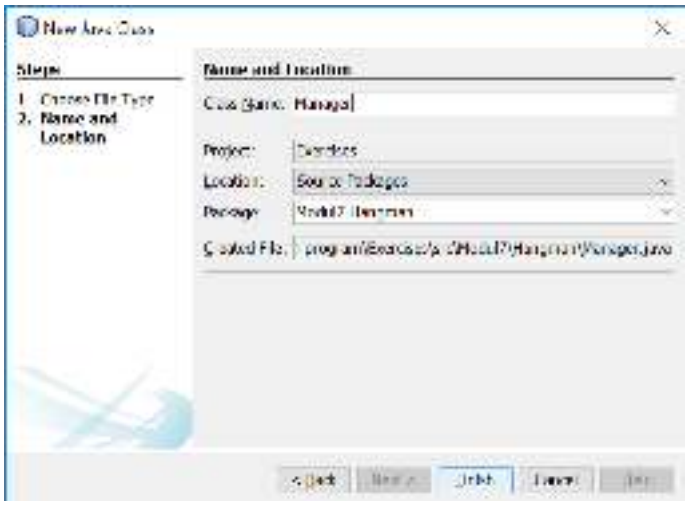
- Buatlah method cari() : listing yang digunakan untuk mengambil data yang ada pada tabel/database.
- Buatlah method ubah() : listing yang digunakan untuk mengubah atau memperbaharui data pada table/database.
- Buatlah method hapusData() : listing yang digunakan untuk menghapus data pada table/database.
- BUAT DATABASE “dborganisasi” dengan table “anggota” dengan field sesuai desai form dengan database MySQL.

## 8.5 Aplikasi Hangman

1. Tambahkan paket (*Java Package*) pada projek 'Exercises'. Beri nama paket dengan 'Modul7.Hangman'.




2. Tambahkan Java Class dengan nama "Manager" pada paket 'Modul7.Hangman'.

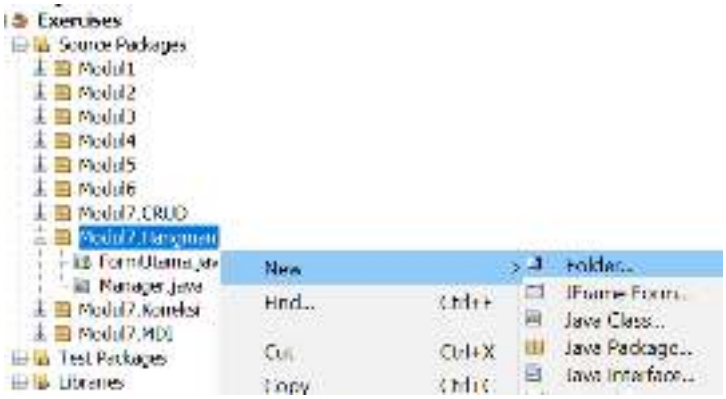


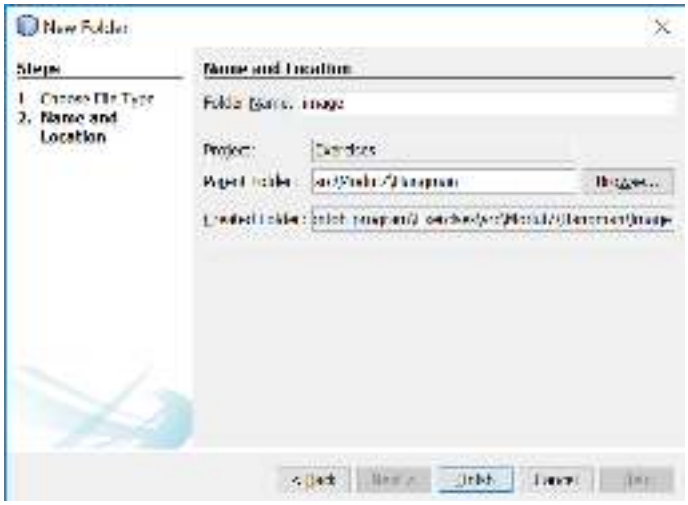

Tulis kode program seperti berikut ini.

```
1
2  import java.io.BufferedReader;
3  import java.io.File;
4  import java.io.FileReader;
5  import java.util.ArrayList;
6
```

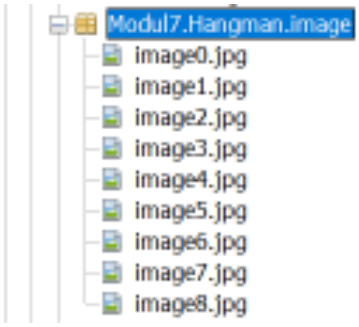
```
*/
public class Manager {

    private ArrayList arl = new ArrayList();
    private File file;

    public Manager() {
        file = new File("Kata.txt");
        int i = 0;
        try {
            BufferedReader br = new BufferedReader(new FileReader(file));

            while (br.readLine() != null) {
                i++;
            }
            br.close();

            br = new BufferedReader(new FileReader(file));
            for (int j = 0; j < i; j++) {
                arl.add( br.readLine());
            }
            br.close();
        } catch (Exception e) {
            System.out.println("Error: " + e);
        }
    }

    public String getRandomWord() throws Exception {
        int random = (int) (Math.random() * arl.size());
        return (String) arl.get(random);
    }
}
```

3. Tambahkan folder baru pada paket ‘Modul7.Hangman’ dan beri nama ‘image’.



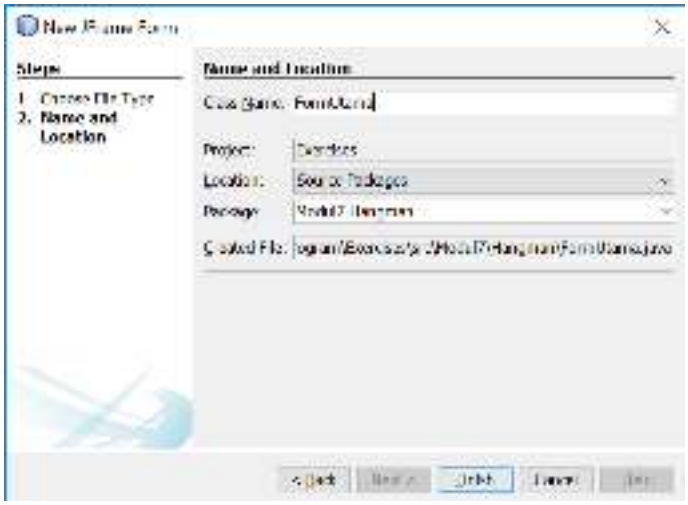



4. Masukkan file citra yang dibutuhkan kedalam folder image tersebut dengan cara *copy* (ctrl+c) seluruh file citra kemudian *paste* (ctrl+v). Seluruh file citra dapat diunduh pada link berikut ini : https://drive.google.com/drive/mobile/folders/1wkx0OrrELfcB5SEiJfc7-y3Hq5XxQsi2/1_7pHs_lkhBtCAKknFfPUFebvlzSGiOCh?usp=sharing&sort=13&direction=a

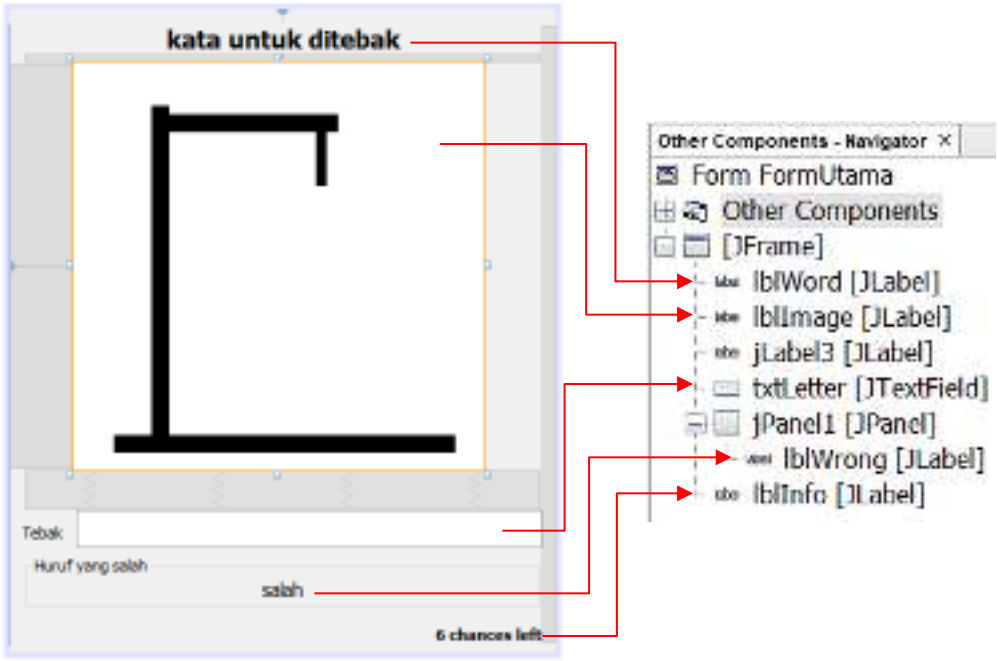



5. Masukkan file 'kata.txt' kedalam paket 'Modul7.Hangman' dengan cara *copy* (ctrl+c) file tersebut kemudian *paste* (ctrl+v). File 'kata.txt' dapat diunduh pada link yang sama di atas.

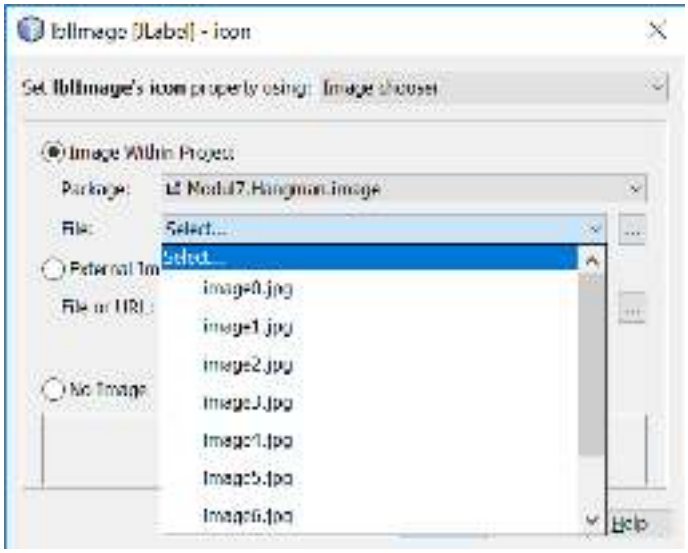
6. Tambahkan JFrame Form dan beri nama dengan 'FormUtama' pada paket 'Modul7.Hangman'.



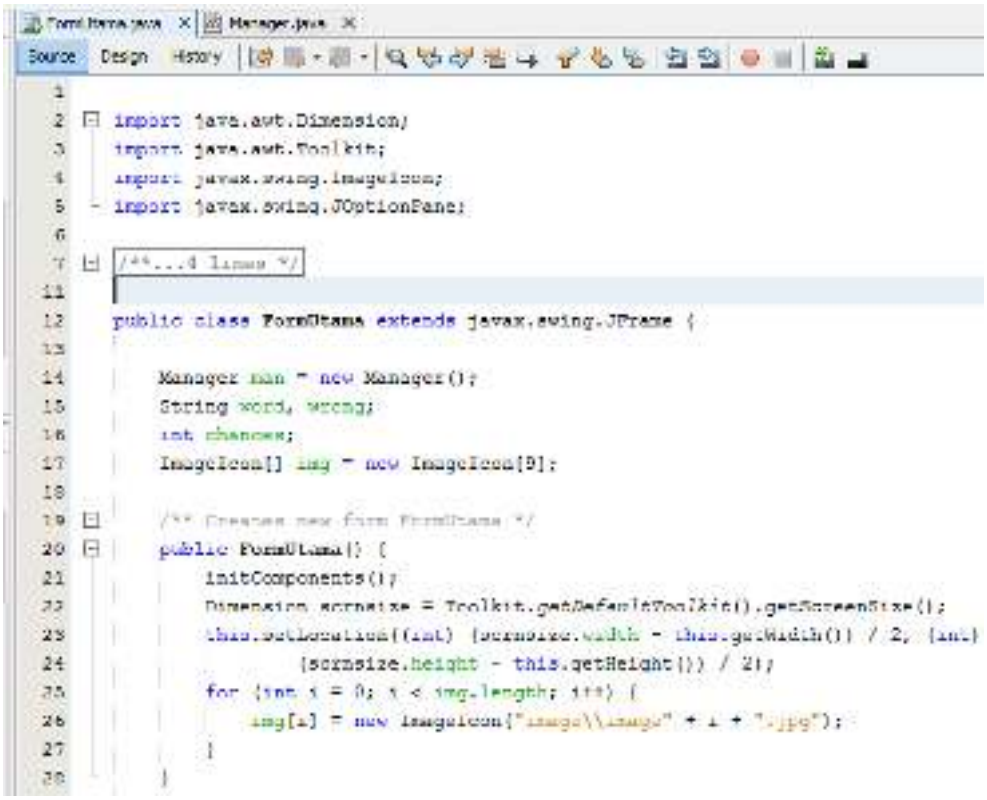
Desain form seperti berikut ini.




7. Masukkan file citra pada lblImage. Klik Kanan lblImage → Properties → Icon → klik tombol . Pilih Package: 'Modul7.Hangman.image' kemudian pilih File: yang diinginkan.



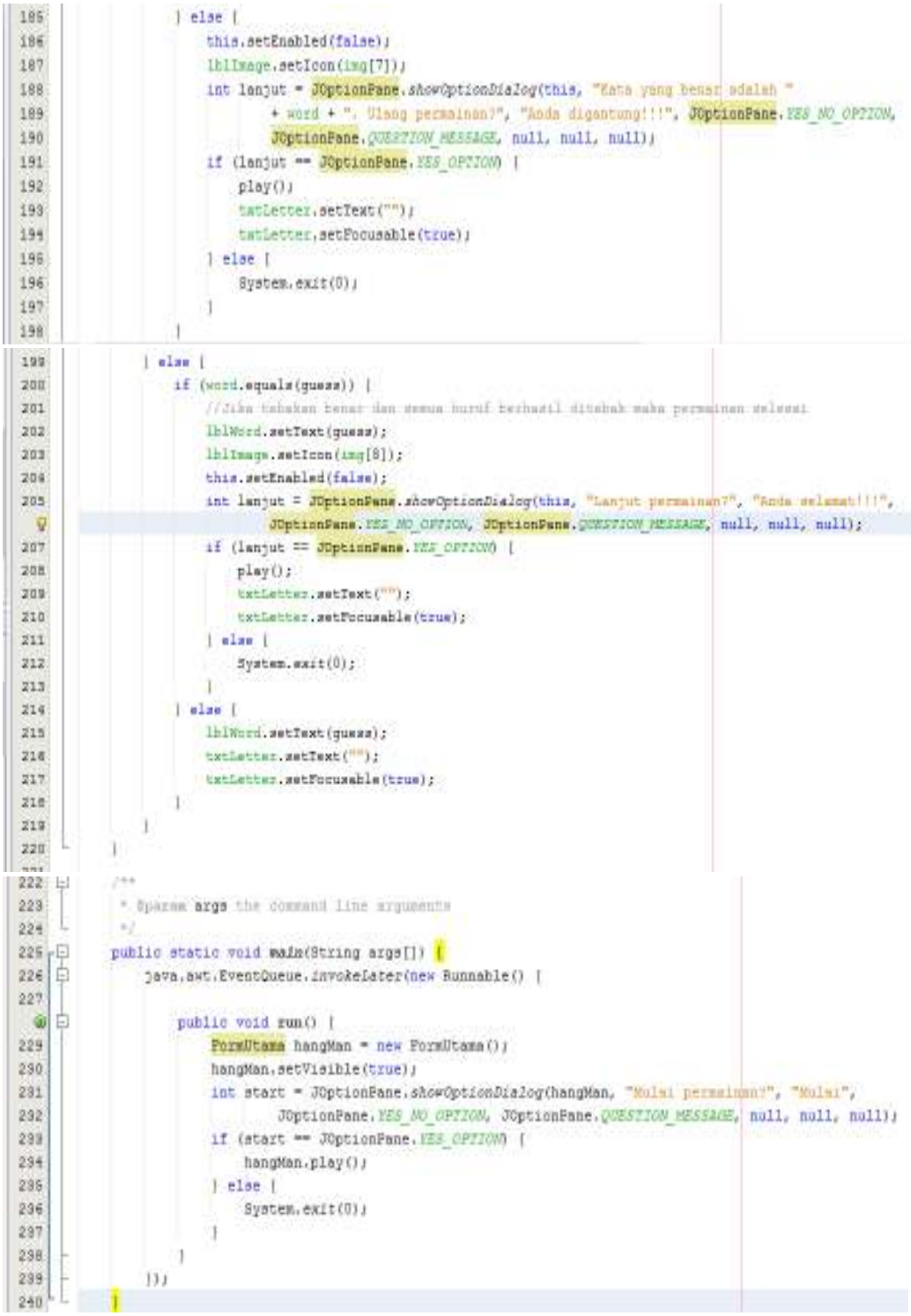

8. Tambahkan import dan variable-variable pada tab ‘Source’ seperti pada gambar di bawah ini.




9. Ketikkan method-method berikut ini pada tab ‘Source’ FormUtama :

```
136     public void play() {
137         this.setEnabled(true);
138         try {
139             word = men.getRandomWord();
140             String str = "";
141             for (int i = 0; i < word.length(); i++) {
142                 str += "-";
143             }
144             lblWord.setText(str);
145             lblWrong.setText(" ");
146             chances = 6;
147             lblInfo.setText(chances + " chances left");
148             lblImage.setIcon(img[0]);
149             wrong = "";
150         } catch (Exception ex) {
151         }
152     }
```

```
154     public void guessWord() {
155         //User menebak huruf
156         String letter = txtLetter.getText();
157         char ltr = letter.toLowerCase().charAt(0);
158 
159         //Memeriksa apakah huruf yang ditebak ada di kata yang disahsiat
160         String guess = "";
161         for (int i = 0; i < word.length(); i++) {
162             if (ltr == word.charAt(i)) {
163                 guess += ltr;
164             } else {
165                 if (lblWord.getText().charAt(i) == '-') {
166                     guess += "-";
167                 } else {
168                     guess += lblWord.getText().charAt(i);
169                 }
170             }
171         }
172 
173         if (lblWord.getText().equals(guess)) {
174             //Jika tebakan salah kesempatan berkurang 1
175             //Jika kesempatan habis permainan selesai
176             int x = 7 - chances;
177             wrong += " " + ltr;
178             if (chances > 0) {
179                 chances--;
180                 lblImage.setIcon(img[x]);
181                 lblInfo.setText(chances + " chances left");
182                 lblWrong.setText(wrong);
183                 txtLetter.setText("");
184                 txtLetter.setFocusable(true);
```

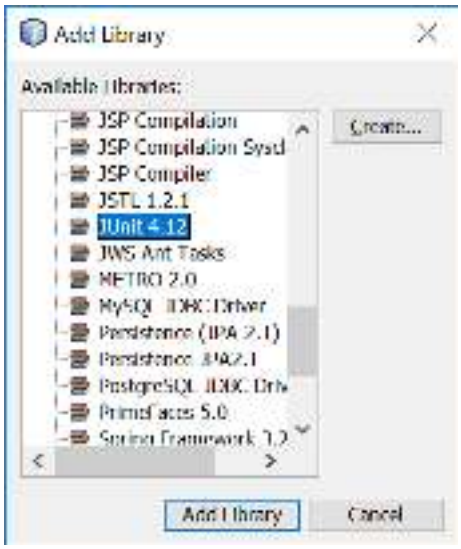


```
            } else {
                this.setEnabled(false);
                lblImage.setIcon(img[7]);
                int lanjut = JOptionPane.showOptionDialog(this, "Kata yang benar adalah "
                        + word + ". Ulang permainan?", "Anda digantung!!!", JOptionPane.YES_NO_OPTION,
                        JOptionPane.QUESTION_MESSAGE, null, null, null);
                if (lanjut == JOptionPane.YES_OPTION) {
                    play();
                    txtLetter.setText("");
                    txtLetter.setFocusable(true);
                } else {
                    System.exit(0);
                }
            }
        } else {
            if (word.equals(guess)) {
                //Jika tebakan benar dan semua huruf berhasil ditebak maka permainan selesai
                lblWord.setText(guess);
                lblImage.setIcon(img[8]);
                this.setEnabled(false);
                int lanjut = JOptionPane.showOptionDialog(this, "Lanjut permainan?", "Anda selamat!!!",
                        JOptionPane.YES_NO_OPTION, JOptionPane.QUESTION_MESSAGE, null, null, null);
                if (lanjut == JOptionPane.YES_OPTION) {
                    play();
                    txtLetter.setText("");
                    txtLetter.setFocusable(true);
                } else {
                    System.exit(0);
                }
            } else {
                lblWord.setText(guess);
                txtLetter.setText("");
                txtLetter.setFocusable(true);
            }
        }
    }
    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(new Runnable() {

            public void run() {
                FormUtama hangMan = new FormUtama();
                hangMan.setVisible(true);
                int start = JOptionPane.showOptionDialog(hangMan, "Mulai permainan?", "Mulai",
                        JOptionPane.YES_NO_OPTION, JOptionPane.QUESTION_MESSAGE, null, null, null);
                if (start == JOptionPane.YES_OPTION) {
                    hangMan.play();
                } else {
                    System.exit(0);
                }
            }
        });
    }
```

10. Klik kanan txtletter → Events → Action → actionPerformed, kemudian ketikkan kode program berikut ini :

```
132    private void txtLetterActionPerformed(java.awt.event.ActionEvent evt) {
133        guessWord();
134    }
```

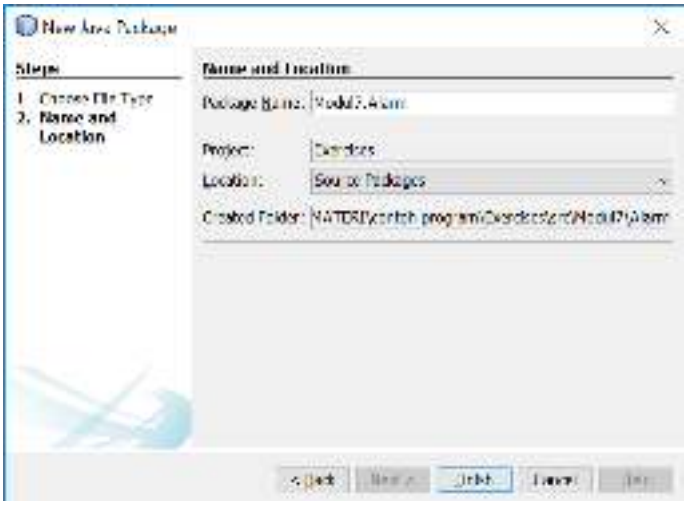
11. Tambahkan library JUnit pada Project Libraries.



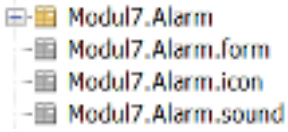

12. Compile dan jalankan dari menu Run → Run File atau tekan Shift + F6.

## 8.6 Aplikasi Alarm

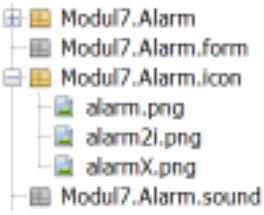
1. Tambahkan paket (*Java Package*) pada projek 'Exercises'. Beri nama paket dengan 'Modul7.Alarm'.



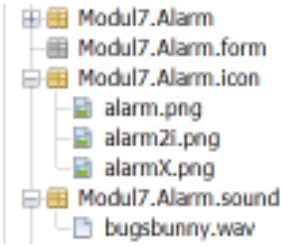

2. Tambahkan 3 folder baru pada paket 'Modul7. Alarm' dan beri nama masing-masing adalah 'form', 'icon', 'sound'.




3. Masukkan file citra yang dibutuhkan kedalam folder icon (Modul7.Alarm.icon) dengan cara copy (ctrl+c) seluruh file citra kemudian paste (ctrl+v). Seluruh file citra dapat diunduh pada link berikut ini :
https://drive.google.com/drive/mobile/folders/1wkx0OrrELfcB5SEiJfc7-y3Hq5XxQsi2/1_7pHs_IkhBtCAKknFfPUFebvlzSGiOCh?usp=sharing&sort=13&direction=a

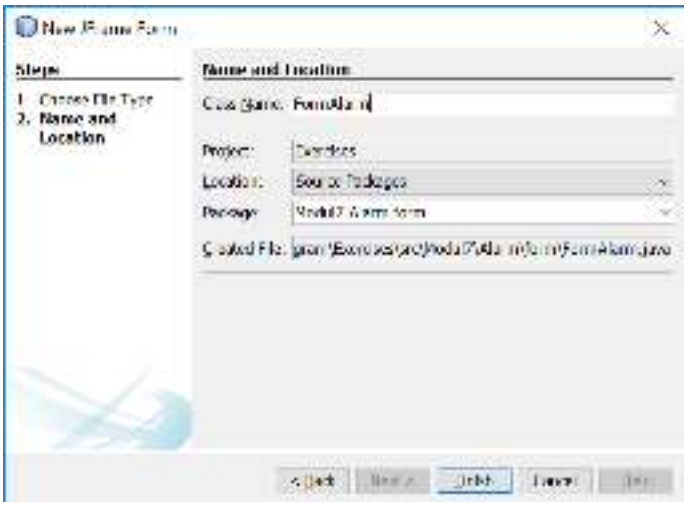



4. Masukkan file suara (*.wav) yang dibutuhkan kedalam folder sound (Modul7.Alarm.sound) dengan cara copy (ctrl+c) file suara kemudian paste (ctrl+v). File suara dapat diunduh pada link di atas.

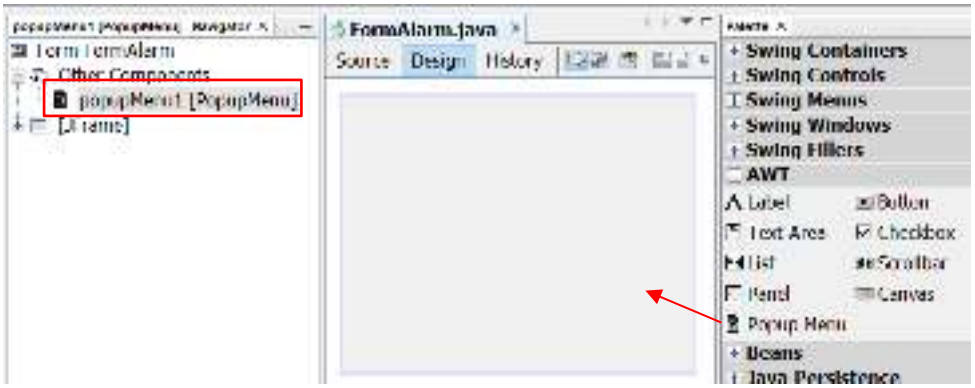



5. Pastikan library JUnit telah ditambahkan kedalam Project Libraries.

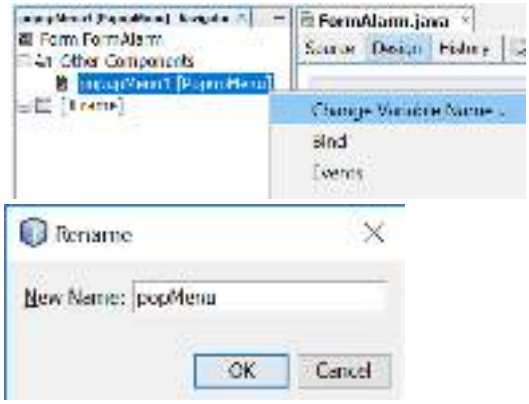
6. Tambahkan JFrame Form pada folder 'form' ('Modul7.Alarm.form') dan beri nama dengan "FormAlarm".



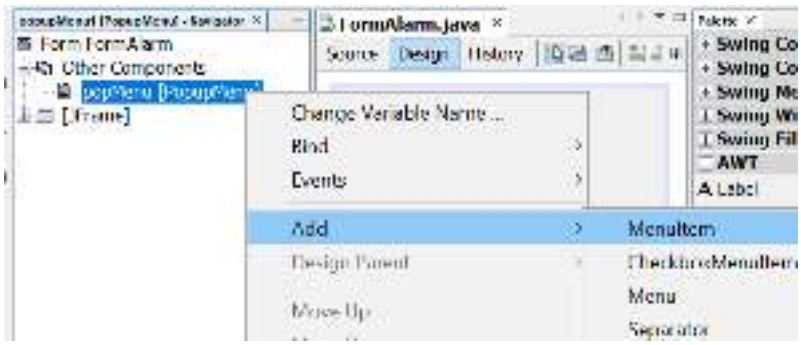

7. Masukkan komponen 'Popup Menu' pada panel palette AWT kedalam form. Bentuk tidak terlihat pada form namun akan muncul pada navigator.



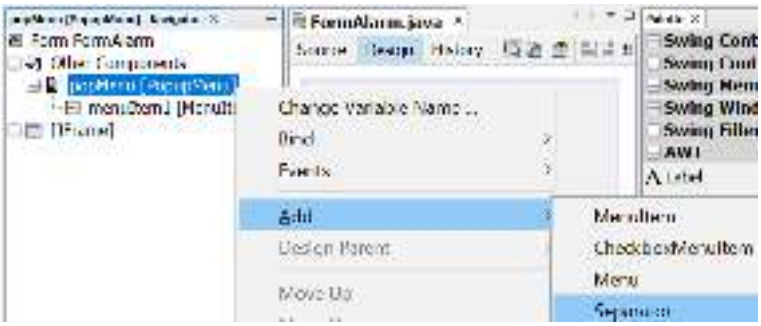
8. Ubah nama variabel komponen Popup Menu menjadi ‘popMenu’.



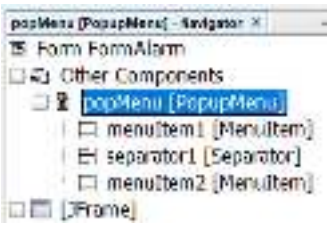

9. Tambahkan MenuItem pada komponen Popup Menu tersebut. Klik kanan komponen Popup Menu pada panel navigator → Add → MenuItem.



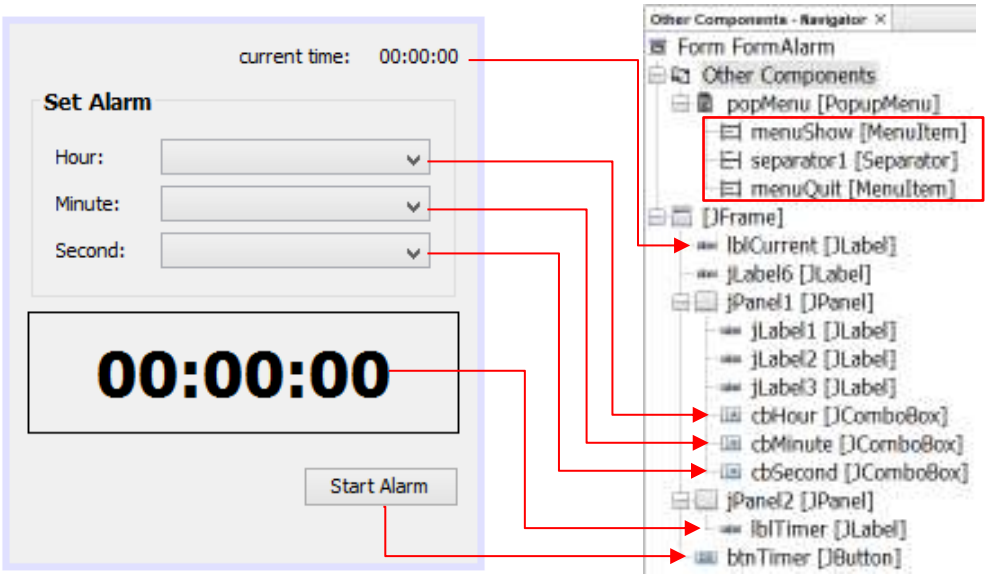

kemudian tambahkan Separator. Klik kanan komponen Popup Menu pada panel navigator → Add → Separator.




dan tambahkan lagi MenuItem pada komponen Popup Menu tersebut.




10. Tambahkan komponen-komponen lainnya. Susun posisi masing-masing komponen, ubah nilai properti setiap komponen dan ubah nama variabel beberapa komponen seperti pada gambar di bawah ini.




Ubah nilai properti 'label' pada menuShow menjadi 'Show' dan nilai properti 'label' pada menuQuit menjadi 'Quit'

11. Pada tab 'Source', tambahkan import dan variable-variable pada kode program seperti pada gambar berikut ini.

```
import java.awt.Image;
import java.awt.SystemTray;
import java.awt.TrayIcon;
import java.io.IOException;
import java.util.Calendar;
import java.util.Timer;
import java.util.TimerTask;
import javax.sound.sampled.AudioInputStream;
import javax.sound.sampled.AudioSystem;
import javax.sound.sampled.Clip;
import javax.sound.sampled.DataLine;
import javax.sound.sampled.LineUnavailableException;
import javax.sound.sampled.UnsupportedAudioFileException;
import javax.swing.Icon;
import javax.swing.ImageIcon;
```

```
26
27
28  public final class FormAlarm extends javax.swing.JFrame implements Runnable {
29
30      Timer timer;
31      int delay = 0;
32      int period = 1000;
33      int status = 0;
34      Icon icon = new ImageIcon("icon/alarm.png");
35      Icon iconX = new ImageIcon("icon/alarmX.png");
36      Thread th;
37
38      /**
39       * Creates new form FormAlarm
40       */
41      public FormAlarm() {
42          initComponents();
43          setComboBox();
44          startClock();
45          addTrayIcon();
46      }
```

Variable Universal

12. Edit sintaks "*public class FormAlarm extends javax.swing.JFrame*" menjadi "*public final class FormAlarm extends javax.swing.JFrame implements Runnable*".

13. Ketikkan method-method berikut ini pada tab 'Source' di FormAlarm.

```
218     public void setComboBox() {
219         for (int i = 0; i < 24; i++) {
220             cbHour.addItem(i);
221         }
222         for (int i = 0; i < 60; i++) {
223             cbMinute.addItem(i);
224         }
225         for (int i = 0; i < 60; i++) {
226             cbSecond.addItem(i);
227         }
228     }
```

```
229
230     public void startClock() {
231         timer = new Timer();
232         timer.schedule(new TimerTask() {
233
234             public void run() {
235                 while (true) {
236                     Calendar now = Calendar.getInstance();
237                     int h = now.get(Calendar.HOUR_OF_DAY);
238                     int m = now.get(Calendar.MINUTE);
239                     int s = now.get(Calendar.SECOND);
240                     lblCurrent.setText(h + ":" + m + ":" + s);
241                 }
242             }
243         }, delay, period);
244     }
```

```
297
298     public void addTrayIcon() {
299         ImageIcon icon = new ImageIcon("icon/alarm2i.png");
300         Image img = icon.getImage();
301         TrayIcon tray = new TrayIcon(img, "Menu Search", popMenu);
302         SystemTray sysTray = SystemTray.getSystemTray();
303         try {
304             sysTray.add(tray);
305         } catch (Exception e) {
306         }
307     }
```

***Copy-paste method ini ke bagian 'Source'***

```
public void run() {
      int hr, min, sec;
      try {
         while (true) {
            Thread.sleep(1000);
            Calendar now = Calendar.getInstance();
            int h = now.get(Calendar.HOUR_OF_DAY);
            int m = now.get(Calendar.MINUTE);
            int s = now.get(Calendar.SECOND);

            hr = Integer.valueOf(cbHour.getSelectedItem().toString());
            min = Integer.valueOf(cbMinute.getSelectedItem().toString());
            sec = Integer.valueOf(cbSecond.getSelectedItem().toString());

            if (sec < s) {
               sec = sec + 60;
               min = min - 1;
            }
            if (min < m) {
               min = min + 60;
               hr = hr - 1;
            }
            if (hr < h) {
               hr = hr + 24;
            }
            hr = hr - h;
            min = min - m;
            sec = sec - s;

            lblTimer.setText(hr + ":" + min + ":" + sec);
            if ((hr == 0) && (min == 0) && (sec == 0)) {
               AudioInputStream source =
                 AudioSystem.getAudioInputStream(getClass().getResource("../sound/bugsbunny.wav"));
               Clip newClip = (Clip) AudioSystem.getLine(new DataLine.Info(Clip.class, source.getFormat()));
               newClip.open(source);
               newClip.start();
               status = 0;
               btnTimer.setIcon(icon);
               btnTimer.setText("Start Alarm");
               cbHour.setEnabled(true);
               cbMinute.setEnabled(true);
               cbSecond.setEnabled(true);
               th.stop();
            }
         }
      } catch (InterruptedException ex) {
      } catch (NumberFormatException ex) {
      } catch (UnsupportedAudioFileException ex) {
      } catch (IOException ex) {
      } catch (LineUnavailableException ex) {
      }
   }
```

14. Tambahkan ActionPerformed pada btnTimer. Klik kanan btnTimer pada panel navigator → Events → Actions → actionPerformed. Ketik kode program berikut ini.

```
196    private void btnTimerActionPerformed(java.awt.event.Act
197        if (status == 0) {
198            status = 1;
199            cbHour.setEnabled(false);
200            cbMinute.setEnabled(false);
201            cbSecond.setEnabled(false);
202            btnTimer.setIcon(iconX);
203            btnTimer.setText("Stop Alarm");
204            th = new Thread (this);
205            th.start();
206        } else {
207            status = 0;
208            btnTimer.setIcon(icon);
209            btnTimer.setText("Start Alarm");
210            cbHour.setEnabled(true);
211            cbMinute.setEnabled(true);
212            cbSecond.setEnabled(true);
213            th.stop();
214            lblTimer.setText("00:00:00");
215        }
216    }
```

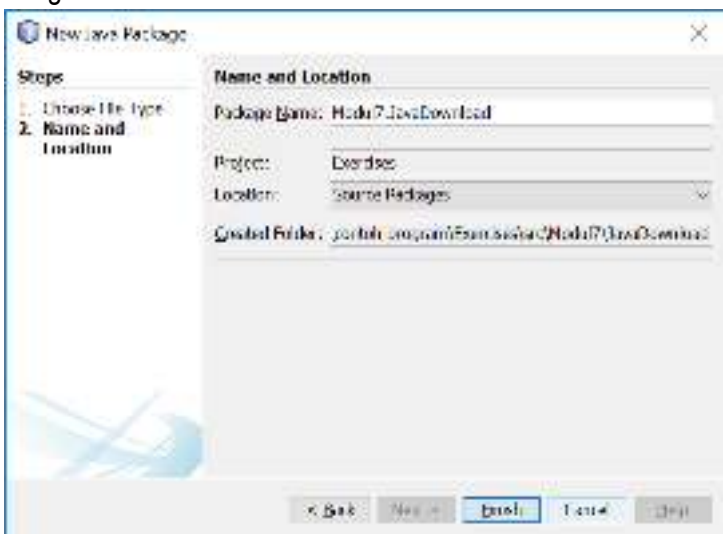
15. Tambahkan ActionPerformed pada menuShow. Klik kanan menuShow pada panel navigator → Events → Actions → actionPerformed. Kemudian ketikkan kode program : ***this.show();***

16. Tambahkan ActionPerformed pada menuQuit. Klik kanan menuQuit pada panel navigator → Events → Actions → actionPerformed. Kemudian ketikkan kode program : ***System.exit (0);***

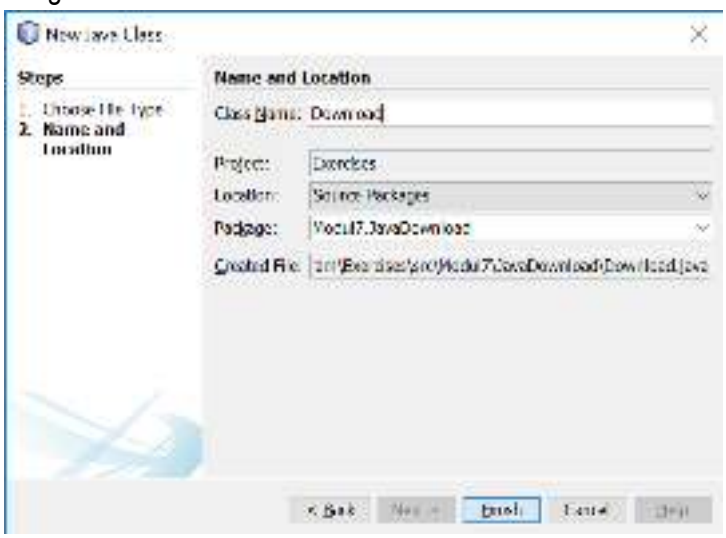
17. Running Program, atur waktu alarm kemudian start alarm .

### 8.7 Aplikasi Download Manager

1. Tambahkan paket (*Java Package*) pada projek 'Exercises'. Beri nama paket dengan 'Modul7.JavaDownload'.




2. Tambahkan Java Class pada paket 'Modul7.JavaDownload' dan beri nama dengan "Download".




Ketikkan import yang dibutuhkan seperti berikut ini pada class Download.

```
6   package Modul7.JavaDownload;
7
8   import java.io.File;
9   import java.io.InputStream;
10  import java.io.RandomAccessFile;
11  import java.net.HttpURLConnection;
12  import java.net.URL;
13  import java.util.Observable;
```

Ubah public class Download

```
21    public class Download {
22
23    }
```

menjadi

```
21    public class Download extends Observable implements Runnable {
22
23    }
```

selanjutnya ketikkan kode program berikut ini didalam `public class Download`

***copy-paste kode program berikut ini***

```
static String action[] = {"Downloading", "Paused", "Complete", "Error"};
int DOWNLOADING = 0;
int PAUSED = 1;
int COMPLETE = 2;
int ERROR = 3;
URL url;
int size, downloaded, status;
File directory;

public Download(URL url, File dest) {
    this.url = url;
    String strURL = url.toString();
    this.directory = new File(dest.getAbsolutePath() + System.getProperty("file.separator") +
strURL.substring(strURL.lastIndexOf('/') + 1));
    size = 0;
    downloaded = 0;
    status = DOWNLOADING;
    Thread thread = new Thread(this);
    thread.start();
}

public void pauseDownload() {
    status = PAUSED;
    update();
}

public void resumeDownload() {
    status = DOWNLOADING;
    update();
    Thread thread = new Thread(this);
    thread.start();
}

private void error() {
    status = ERROR;
    update();
}
```

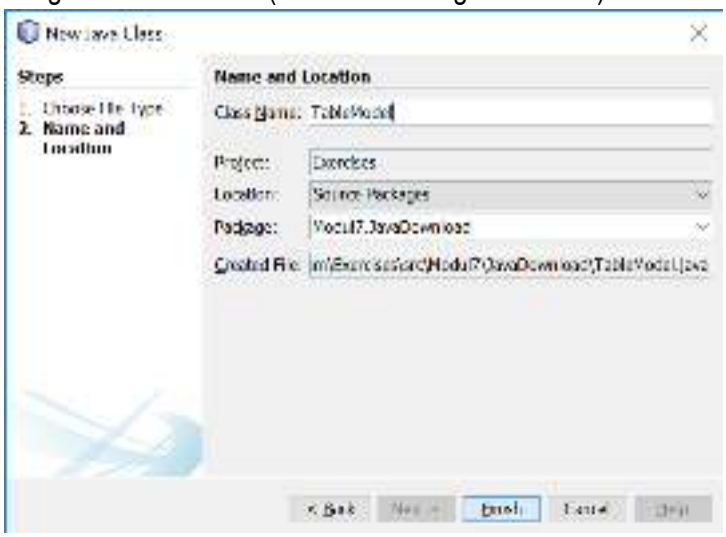
***copy-paste kode program berikut ini (lanjutan sebelumnya)***

```
public String getFileName() {
    return url.toString().substring(url.toString().lastIndexOf('/') + 1);
 }

 public int getSize() {
    return size;
 }

 public float getProgress() {
    return ((float) downloaded / size) * 100;
 }

 public int getStatus() {
    return status;
 }

 private void update() {
    setChanged();
    notifyObservers();
 }

 public void run() {
    RandomAccessFile file = null;
    InputStream stream = null;
    try {
       HttpURLConnection connection = (HttpURLConnection) url.openConnection();
       connection.connect();
       if (connection.getResponseCode() != 200) {
          error();
       }
       int contentLength = connection.getContentLength();
       if (contentLength < 1) {
          error();
       }
       if (size == 0) {
          size = contentLength;
          update();
       }
       file = new RandomAccessFile(directory, "rw");
       file.seek(downloaded);
       stream = connection.getInputStream();
```

***copy-paste kode program berikut ini (lanjutan sebelumnya)***

```
        while (status == DOWNLOADING) {
            byte buffer[];
            buffer = new byte[size - downloaded];
            int read = stream.read(buffer);
            if (read == -1) {
                break;
            }
            file.write(buffer, 0, read);
            downloaded += read;
            update();
        }
        if (status == DOWNLOADING) {
            status = COMPLETE;
            update();
        }
    } catch (Exception e) {
        error();
    }
}
```

3. Tambahkan Java Class pada paket 'Modul7.JavaDownload' dan beri nama dengan "TableModel". (Download listing di edmodo)




Ketikkan import yang dibutuhkan seperti berikut ini pada class TableModel.

```
6   package Modul7.JavaDownload;
7
8   import java.util.ArrayList;
9   import java.util.Observable;
10  import java.util.Observer;
11  import javax.swing.table.AbstractTableModel;
```

Ubah public class TableModel

```
15    public class TableModel {
16
17    }
```

menjadi

```
15    public class TableModel extends AbstractTableModel implements Observer {
16
17    }
```

selanjutnya ketikkan kode program berikut ini didalam `public class TableModel`

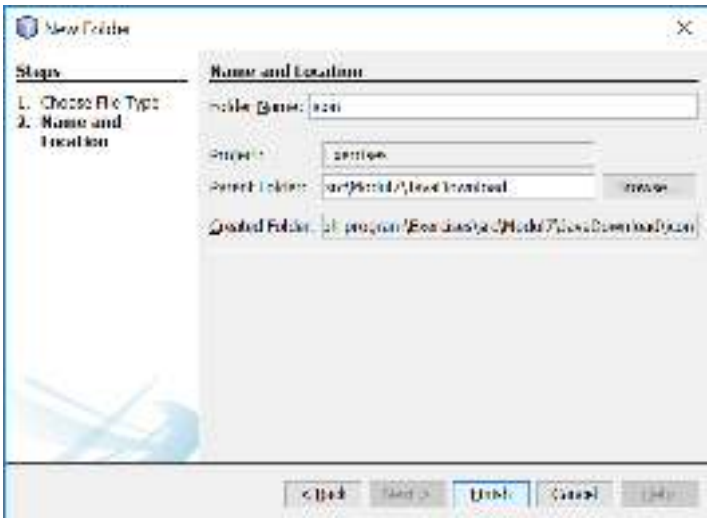
***copy-paste kode program berikut ini***

```
String[] columnNames = {"File Name", "Size", "Progress", "Status"};
ArrayList downloadList = new ArrayList();

@Override
public String getColumnName(int col) {
    return columnNames[col];
}

public int getRowCount() {
    return downloadList.size();
}

public int getColumnCount() {
    return columnNames.length;
}

public void update(Observable o, Object arg) {
    int index = downloadList.indexOf(o);
    fireTableRowsUpdated(index, index);
}

public Object getValueAt(int rowIndex, int colmIndex) {
    Download download = (Download) downloadList.get(rowIndex);
    if (colmIndex == 0) {
        return download.getFileName();
    } else if (colmIndex == 1) {
        return download.getSize();
    } else if (colmIndex == 2) {
        return String.valueOf((int) download.getProgress()) + " %";
    } else if (colmIndex == 3) {
        return Download.action[download.getStatus()];
    } else {
        return "";
    }
}
```

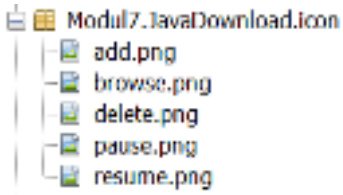
***copy-paste kode program berikut ini (lanjutan sebelumnya)***

```
public void addDownload(Download download) {
    download.addObserver(this);
    downloadList.add(download);
    fireTableRowsInserted(getRowCount() - 1, getRowCount() - 1);
}

public Download getDownload(int row) {
    return (Download) downloadList.get(row);
}

public void clearDownload(int row) {
    downloadList.remove(row);
    fireTableRowsDeleted(row, row);
}
```

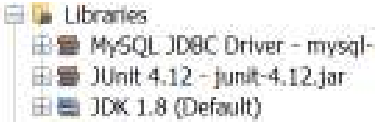
4. Tambahkan folder baru pada paket 'Modul7.JavaDownload' dan beri nama 'icon'.



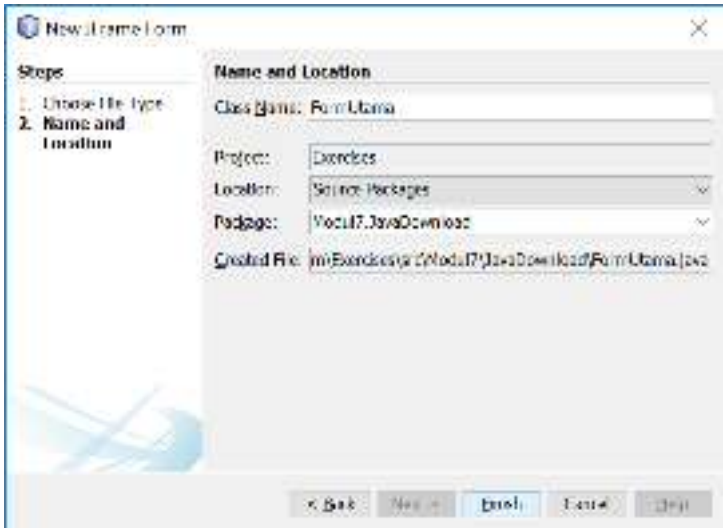

5. Masukkan file citra yang dibutuhkan kedalam folder icon (Modul7.JavaDownload.icon) dengan cara copy (ctrl+c) seluruh file citra kemudian paste (ctrl+v). Seluruh file citra dapat diunduh pada link berikut ini : https://drive.google.com/drive/mobile/folders/1wkx0OrrELfcB5SEiJfc7-y3Hq5XxQsi2/1_7pHs_lkhBtCAKknFfPUFebvlzSGiOCh?usp=sharing&sort=13&direction=a



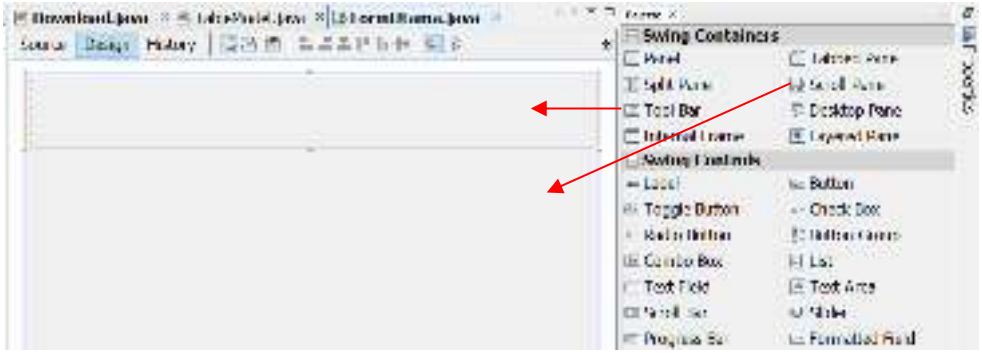

6. Pastikan library JUnit telah ditambahkan kedalam Project Libraries.



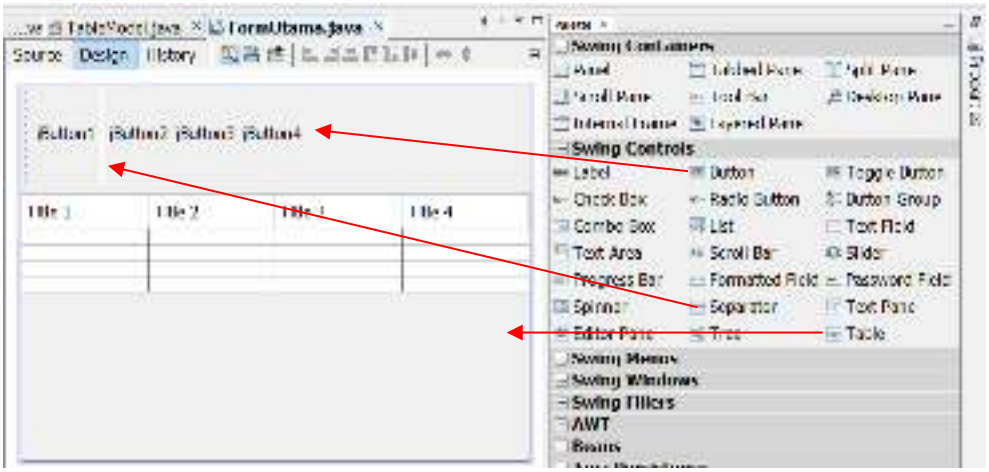

7. Tambahkan JFrame Form pada paket 'Modul7.JavaDownload' dan beri nama dengan "FormUtama".




8. Masukkan komponen 'Tool Bar' dan 'Scroll Pane' kedalam form.



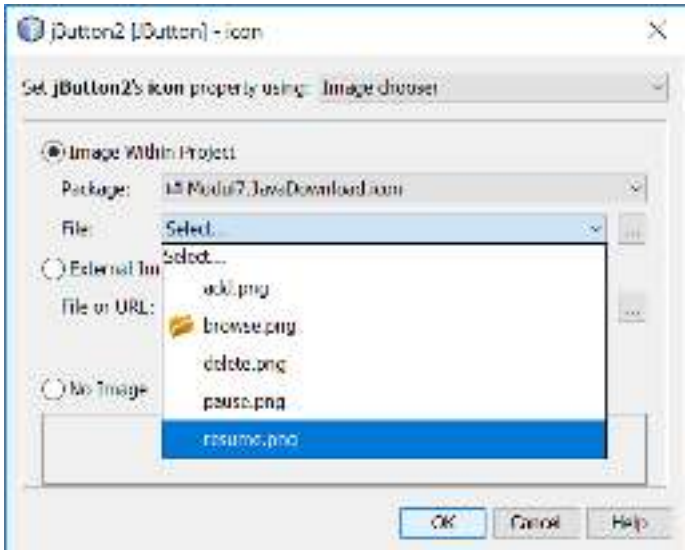

kemudian masukkan komponen 'Table' di dalam komponen 'Scroll Pane', komponen 'Button' dan 'Separator' di dalam komponen 'Tool Bar'.




9. Ubah nama variabel dan nilai properti beberapa komponen seperti pada tabel di bawah ini.

| Komponen | Nama Variabel | Properties |
|---|---|---|
| jButton1 | btnAdd | text = add |
| | | icon = add.png |
| jButton2 | btnResume | text = resume |
| | | icon = resume.png |
| | | enabled = false |
| jButton3 | btnPause | text = pause |
| | | icon = pause.png |
| | | enabled = false |
| jButton4 | btnStop | text = stop |
| | | icon = delete.png |
| | | enabled = false |
| jTable1 | tblDownload | - |

10. Untuk memasukkan file citra (icon) kedalam komponen 'Button', klik kanan komponen 'Button' → Properties → Icon → klik tombol . Pilih Package: 'Modul7. JavaDownload.icon' kemudian pilih File: yang diinginkan.




11. Pada tab 'Source', ketikkan kode program berikut ini

```
17  public class FormUtama extends javax.swing.JFrame {
18
19      Download selectedDownload;
20      static TableModel model = new TableModel();
21
22      /** Creates new form FormUtama */
23      public FormUtama() {
24          initComponents();
25          tblDownload.setModel(model);
26      }
```

12. Tambahkan ActionPerformed pada btnAdd. Klik kanan btnAdd pada panel navigator → Events → Actions → actionPerformed. Ketik kode program berikut ini.

```
139     private void btnAddActionPerformed(java.awt.event.ActionEvent evt) {
140         new InputDownload(this, true).setVisible(true);
141     }
```

13. Tambahkan ActionPerformed pada btnPause. Klik kanan btnPause pada panel navigator → Events → Actions → actionPerformed. Ketik kode program berikut ini.

```
143     private void btnPauseActionPerformed(java.awt.event.ActionEvent evt) {
144         selectedDownload = model.getDownload(tblDownload.getSelectedRow());
145         selectedDownload.pauseDownload();
146     }
```

14. Tambahkan ActionPerformed pada btnResume. Klik kanan btnResume pada panel navigator → Events → Actions → actionPerformed. Ketik kode program berikut ini.

```
148     private void btnResumeActionPerformed(java.awt.event.ActionEvent evt) {
149         selectedDownload = model.getDownload(tblDownload.getSelectedRow());
150         selectedDownload.resumeDownload();
151     }
```

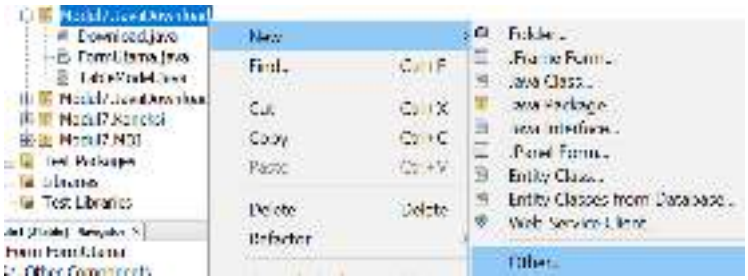
15. Tambahkan ActionPerformed pada btnStop. Klik kanan btnStop pada panel navigator → Events → Actions → actionPerformed. Ketik kode program berikut ini.

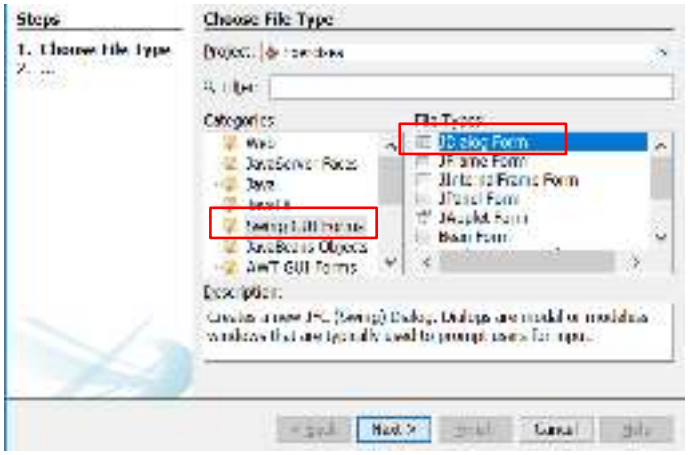
```
153     private void btnStopActionPerformed(java.awt.event.ActionEvent evt) {
154         selectedDownload = model.getDownload(tblDownload.getSelectedRow());
155         selectedDownload.pauseDownload();
156         model.clearDownload(tblDownload.getSelectedRow());
157     }
```

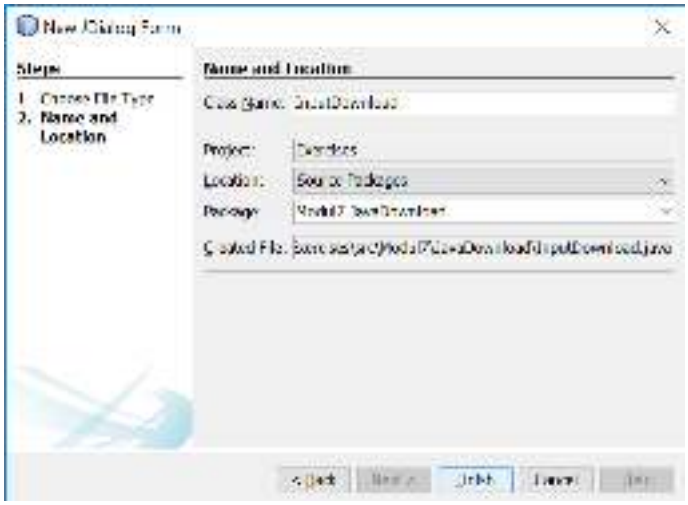
16. Tambahkan Event MouseClicked pada tblDownload. Klik kanan tblDownload pada panel navigator → Events → Mouse → mouseClicked. Ketik kode program berikut ini.

```
159     private void tblDownloadMouseClicked(java.awt.event.MouseEvent evt) {
160         btnPause.setEnabled(true);
161         btnStop.setEnabled(true);
162         btnResume.setEnabled(true);
163     }
```

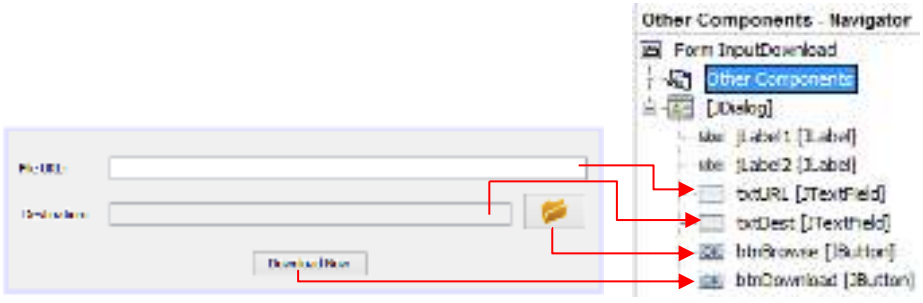
17. Tambahkan JDialog Form pada paket 'Modul7.JavaDownload' dan beri nama dengan "InputDownload".






18. Masukkan komponen ‘Label’, ‘Text Field’, dan ‘Button’ kedalam form “InputDownload”. Susun posisi masing-masing komponen, ubah nilai properti, dan ubah nama variabel beberapa komponen sesuai dengan gambar di bawah ini.




19. Pada tab ‘Source’, ketikkan kode program seperti pada gambar berikut ini.

```
2  import java.io.File;
3  import java.net.URL;
4  import javax.swing.JFileChooser;
5  import javax.swing.JOptionPane;
6  import javax.swing.filechooser.FileSystemView;
```

```
20  * @author Indie
21  */
22  public class InputDownload extends javax.swing.JDialog {
23
24      /** Creates new form InputDownload */
25      public InputDownload(java.awt.Frame parent, boolean modal) {
26          super(parent, modal);
27          initComponents();
28          txtDest.setText(FileSystemView.getFileSystemView().getDefaultDirectory().getAbsolutePath());
29      }
```

20. Ketikkan method-method dibawah :

```
128     private void selectDestination() {
129         JFileChooser jfc = new JFileChooser(txtDest.getText());
130         jfc.setFileSelectionMode(JFileChooser.DIRECTORIES_ONLY);
131         int valid = jfc.showOpenDialog(this);
132         if (valid == JFileChooser.APPROVE_OPTION) {
133             txtDest.setText(jfc.getSelectedFile().getAbsolutePath());
134         }
135     }
136
137     public void addDownload() {
138         if (txtURL.getText().toLowerCase().startsWith("http://")) {
139             try {
140                 URL verifiedUrl = new URL(txtURL.getText());
141                 File destFile = new File(txtDest.getText() + System.getProperty("file.separator"));
142                 FormUtama.model.addDownload(new Download(verifiedUrl, destFile));
143                 this.dispose();
144             } catch (Exception ex) {
145             }
146         } else {
147             JOptionPane.showMessageDialog(this, "Invalid Download Url", "Error", JOptionPane.ERROR_MESSAGE);
148         }
149     }
```

21. Tambahkan ActionPerformed pada btnBrowse. Klik kanan btnBrowse pada panel navigator → Events → Actions → actionPerformed. Kemudian ketikkan kode program berikut ini.

```
120     private void btnBrowseActionPerformed(java.awt.event.ActionEvent
121         selectDestination();
122     }
```

22. Tambahkan ActionPerformed pada btnDownload. Klik kanan btnDownload pada panel navigator → Events → Actions → actionPerformed. Kemudian ketikkan kode program berikut ini.

```
124    private void btnDownloadActionPerformed(java.awt.event.ActionEvent
125        addDownload();
126    }
```

23. Running Program (Shift + F6), klik Button Add masukkan URL file yang akan di download.

    Contoh URL :
    http://download1974.mediafire.com/w8x8d9n50f1g/kapzfpejq8e4woy/smadav10.exe

# DAFTAR PUSTAKA


[1] David Etheridge, "*Java : Graphical User Interfaces" Bookbon.com*

[2] Justin Simmons, "*Java Programming "* Global Media

[3] James Levenick "*Simply Java An Introduction toJava Programming"* Willamette University

[4] Aloysius Sigit W, "*7Proyek Aplikas Dengan Java,*" Elexmedia Komputindo

[5] Y. Daniel Liang, "*Java Programming Comprehensive Version 10th*", Pearson

[6] R.H. Sianipar, "*Java : Teori, Algoritma, dan Aplikasi*", Andi Publisher

[7] Kishori Sharan, "*Beginning Java 9 Fundamentals: Arrays, Objects, Modules, JShell, and Regular Expressions 2nd Edition*", Apress Publisher

[8] Adam Drozdek, "*Data Structure and Algorithms in C++ 4th edition*", Cengage Learning

## Biodata Penulis Buku
## "Dasar Pemrograman Java"

**Eko Hariyanto, S.Kom., M.Kom**. Lahir di Medan, 23 Juni 1985, menempuh pendidikan SD, SMP, SMA di Kota Medan, Sumatera Utara. Melanjutkan pendidikan ke jenjang D-III Teknik Komputer, Politeknik LP3I Medan dan lulus tahun 2006. Kemudian menyelesaikan pendidikan S1 Teknik Informatika di STMIK TIME tahun 2010 dan pasca sarjana S2 Teknik Informatika di Universitas Sumatera Utara tahun 2012. Mengajar sebagai dosen di Universitas Pembangunan Panca Budi sejak tahun 2012 hingga kini pada mata kuliah Algoritma dan Pemrograman Dasar, Pemrograman Visual, Basis Data, Analisis Perancangan Sistem dan Rekayasa Perangkat Lunak.

**Indri Sulistianingsih S.Kom,. M.Kom**. Lahir di Pematangsiantar 04 Maret 1991. Menamatkan pendidikan di SMK Negeri 9 Medan, Lulus S1 di Program Studi Sistem Komputer Fakultas Teknik Universitas Pembangunan Panca Budi (UNPAB) tahun 2012, lulus S2 di Program Magister Teknik Informatika Program Pascasarjana Universitas AMIKOM Yogyakarta tahun 2017. Saat ini adalah dosen tetap Program Studi Sistem Komputer Fakultas Teknik Universitas Pembangunan Panca Budi Medan. Mengampu mata kuliah Pemrograman Visual (Java NetBeans) dan Algoritma dan Pemrograman Dasar. Aktif dalam organisasi Relawan TIK Indonesia (RTIK) dan Pendonor aktif bagi anak kanker pada organisasi Smiling Kids Foundation.

ISBN 978-602-51956-5-3