

BAB IX

BEKERJA DENGAN *IMAGE ICON*

A. CAPAIAN PEMBELAJARAN

1. Mahasiswa dapat mengerti dan dapat menggunakan *IMAGE Icon* dan *JLabel*
2. Mahasiswa dapat menganalisa kapan menggunakan *IMAGE Icon*
3. Mahasiswa dapat membuat program menggunakan *IMAGE Icon*

B. MATERI

1. Image Icon

Penanganan Image Icon dalam aplikasi adalah masalah umum bagi banyak programmer Java pemula. Cara standar untuk mengakses gambar dalam aplikasi Java adalah dengan menggunakan metode `+getResource()+`. Pertemuan ini menunjukkan cara menggunakan GUI Builder IDE untuk menghasilkan code untuk menyertakan gambar (dan sumber daya lainnya) dalam aplikasi Kita. Selain itu, Kita akan

mempelajari cara menyesuaikan cara IDE menghasilkan kode penanganan Image .

Untuk bekerja dengan Image kita membutuhkan JLabels sebagai media yang akan menampilkan file gambar.

Pada label kita bisa set Properti icon untuk menentukan file grafik yang akan ditampilkan. Terdapat Tiga jenis file gambar yang dapat ditampilkan

file grafik tersebut dapat dilihat seperti dibawah ini :

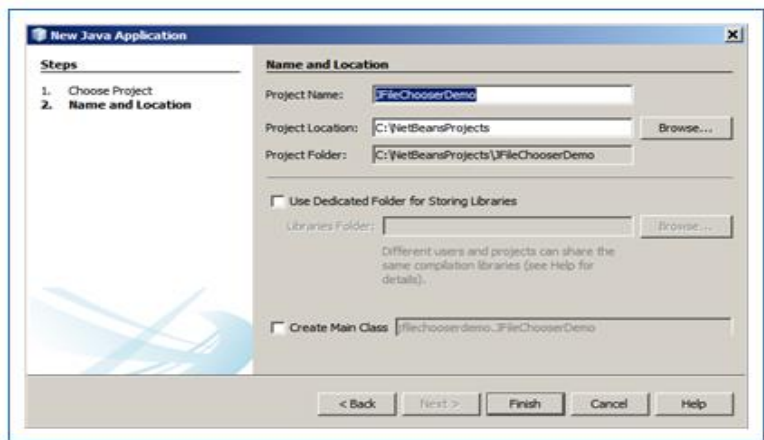
- a. JPEG JPEG (Joint Photographic Experts Group) adalah format bitmap terkompresi yang mendukung warna 8 dan 24 bit. Ini populer di Internet dan adalah format umum untuk kamera digital. Nama file JPEG memiliki ekstensi .jpg.
- b. GIF GIF (Graphic Interchange Format) adalah format bitmap terkompresi awalnya dikembangkan oleh CompuServe. Mendukung hingga 256 warna dan juga populer di Internet. Nama file GIF memiliki .gif perpanjangan.
- c. Format PNG PNG (Portable Network Graphics) adalah format populer untuk mentransfer file grafik di antara platform yang berbeda. PNG nama file memiliki ekstensi .png.

2. Membuat Aplikasi Image menggunakan Forms

a. Membuat Project baru

- 1) Pilih Menu **File > New Project.**

- 2) Pada New Project wizard, pilih **Java** > Java Application lalu click **Next**.
- 3) Ketik nama proejct Project misalnya ImageDisplayApp.
- 4) Hlangkan centang **Create Main Class** pada checkbox.



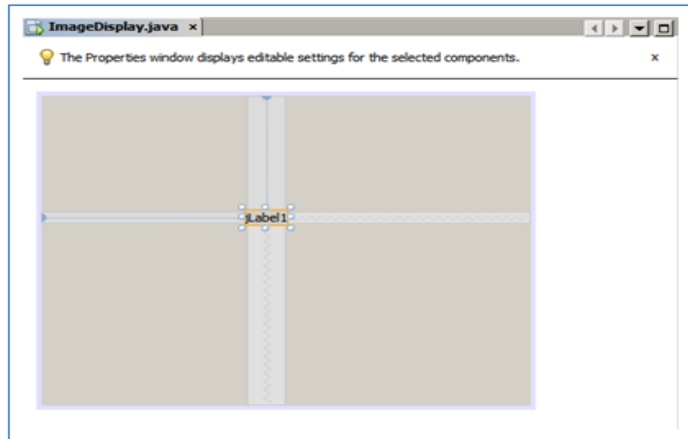
Gambar 9. 1 Project Baru Image Icon

- b. Menyiapkan Form baru pada project
 - 1) Pada Jendela Projects, expand dengan mengklik the ImageDisplayApp
 - 2) Klik Kanan Source Packages and pilih New > JFrame Form.
 - 3) Untuk penamaan Class form, ketik ImageDisplay.
 - 4) Untuk nama Package, ketik org.me.myimageapp.
 - 5) Klik Finish.

c. Menambahkan JLabel kedalam Form :

Pada jendela Palette, pilih komponen Label dan seret ke JFrame.

Untuk saat ini, Form akan terlihat seperti gambar berikut:



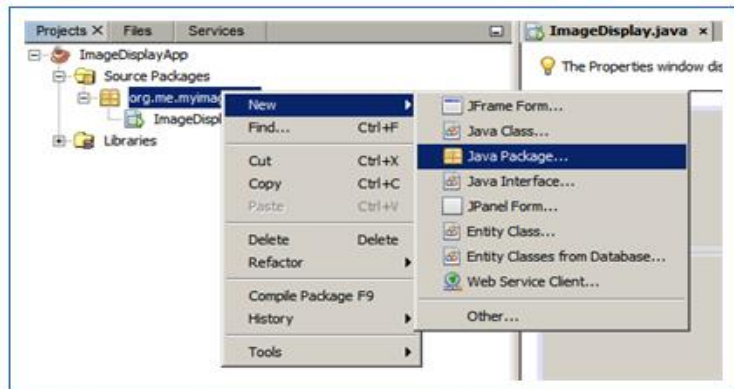
Gambar 9. 2 Desain Label kedalam Form

d. Menambahkan Package untuk menampung Image

Saat kita menggunakan Image atau sumber daya lain dalam aplikasi, biasanya kita membuat Package Java terpisah untuk sumber daya. Pada sistem file lokal Anda, sebuah *Package* sesuai dengan folder.

Langkah membuat Package untuk gambar:

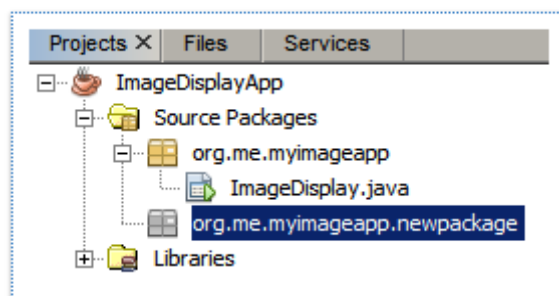
- a) Pada jendela Projects, klik kanan node `org.me.myimageapp` dan pilih `New > Java Package`.



Gambar 9. 3 Menambahkan Package menampung Image

- b) Klik Finish

Sekarang kita melihat *package* yang baru muncul di dalam folder *Source package Folder*.



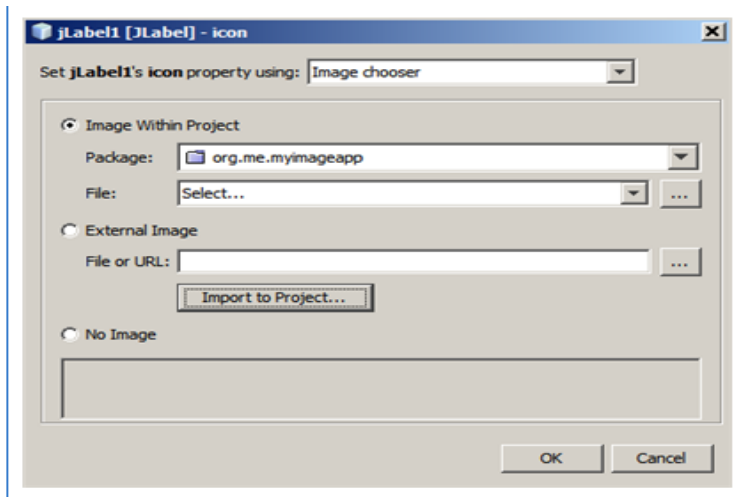
Gambar 9. 4 Source Package

e. Menampilkan Gambar pada Label

Pada aplikasi ini, Image akan disematkan di dalam komponen JLabel.

Untuk menambahkan gambar ke label:

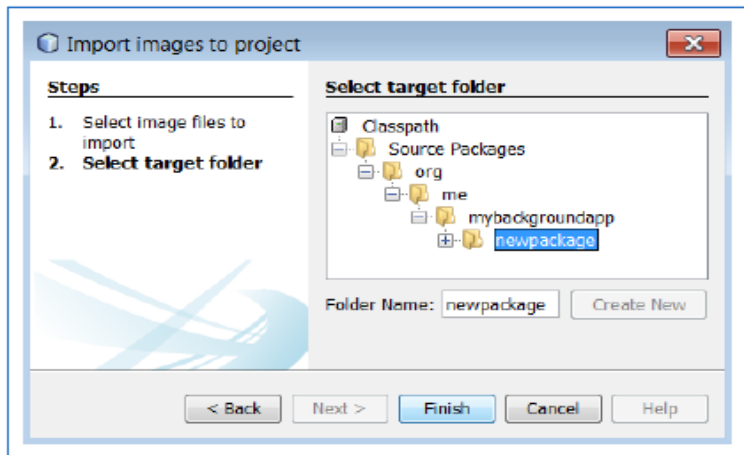
- 1) Pada GUI Design, pilih label yang telah Anda tambahkan ke formulir Anda.
- 2) Pada jendela Properties, klik kategori *Properties* dan gulir ke properti *Icon*
- 3) Klik tombol elipsis (...). Editor properti ikon yang ditampilkan.



Gambar 9. 5 Image Chooser

- a) Di kotak dialog properti *Icon*, klik *Import to Project*.
- b) Di *file chooser*, navigasikan ke gambar apa pun yang ada di sistem Anda yang ingin Anda gunakan. Kemudian klik Next.

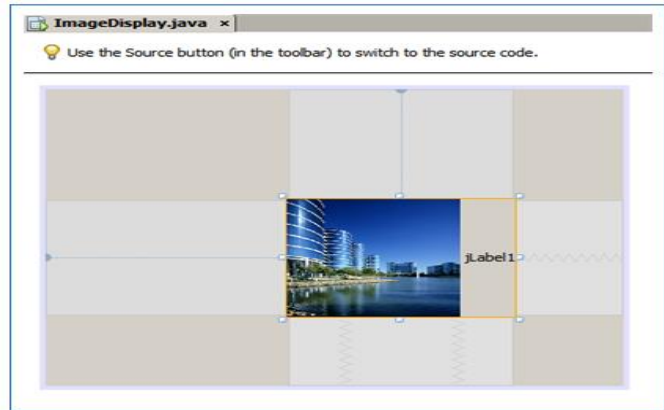
- c) Di halaman *Select target folder* dari wizard, pilih folder *newpackage* dan klik *Finish*.
- d) Klik *Ok* untuk menutup kotak dialog properties Icon



Gambar 9. 6 Import Image

Setelah Anda mengklik OK, IDE melakukan hal-hal berikut:

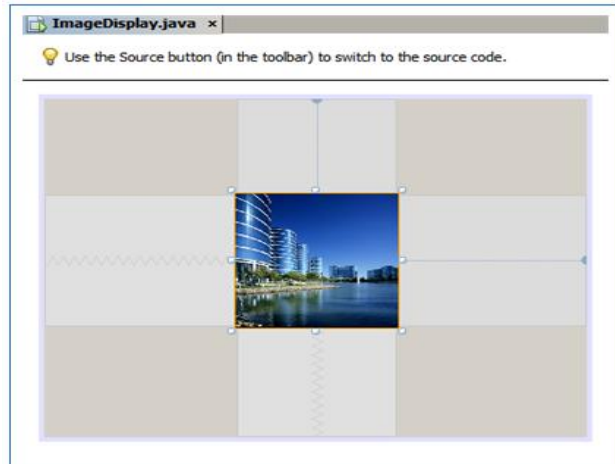
- Salin gambar ke proyek Anda. Oleh karena itu, saat Anda membuat dan mendistribusikan aplikasi, gambar disertakan dalam file JAR yang dapat didistribusikan.
- Menghasilkan kode di Class ImageDisplay untuk mengakses Image
- Menampilkan Image Anda pada label pada tampilan Desain Form Anda.



Gambar 9. 7 Image disisipkan ke Label

Pada titik ini, kita dapat melakukan beberapa hal sederhana untuk meningkatkan tampilan formulir, seperti:

- Di jendela Properties, pilih Text properti dan hapus `jLabel1`. Nilai tersebut dihasilkan oleh GUI Builder sebagai teks tampilan untuk label. Namun, Anda menggunakan label untuk menampilkan gambar di teks, sehingga teks tidak diperlukan.
- Seret `jLabel1` ke tengah formulir.



Gambar 9. 8 Hasil Image pada Label

Untuk melihat kode yang dihasilkan:

- Pada GUI Designer, klik tombol source. (Pilih View > Source Editor Toolbar dari main menu jika tombol Source disembunyikan.)
- Gulir ke bawah ke baris yang mengatakan Kode yang di generate.
- Klik tanda plus (+) di sebelah kiri baris Kode yang Dihasilkan untuk menampilkan kode yang telah dibuat oleh GUI Designer.

Baris kuncinya adalah sebagai berikut:

```
jLabel1.setIcon(new  
javax.swing.ImageIcon(getClass().getResource("/org/  
me/myimageapp/newpackage/image.png"))); //  
NOI18N
```

Karena Anda telah menggunakan editor properti untuk properti `Icon jLabel1`, IDE telah menghasilkan metode `setIcon`. Parameter metode tersebut berisi panggilan ke metode `+getResource()+` pada kelas dalam `ImageIcon` anonim. Perhatikan bahwa jalur yang dihasilkan untuk gambar sesuai dengan lokasinya dalam struktur paket aplikasi.

Catatan:

Jika kita menggunakan opsi `Image Eksternal` di editor properti `Icon`, IDE akan menghasilkan jalur absolut ke gambar alih-alih menyalin gambar ke proyek Anda. Oleh karena itu, gambar akan muncul saat kita menjalankan aplikasi di sistem kita, tetapi mungkin tidak akan muncul saat menjalankan aplikasi di sistem lain.

Metode `getResource` juga berguna untuk mengakses jenis sumber daya lainnya, seperti file teks yang berisi data yang mungkin perlu digunakan oleh aplikasi kita .



You can get the mouse coordinates (for example, the location of a mouse click) in the event handler using the `event.getPoint()`, `event.getX()`, or `event.getY()` methods. See [Class MouseEvent](#) for details.

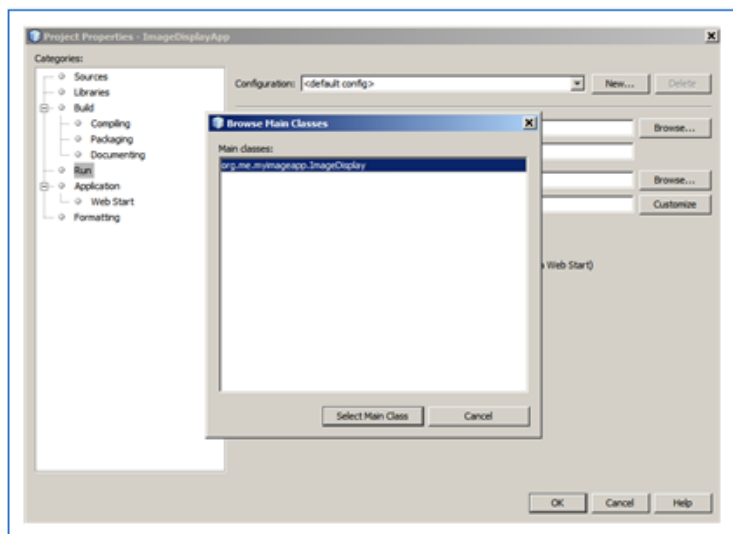
f. Membangun dan Menjalankan Aplikasi

Sekarang setelah kita membuat kode untuk mengakses dan menampilkan gambar, Anda dapat membangun dan menjalankan aplikasi untuk memastikan bahwa gambar tersebut diakses.

Pertama, kita perlu mengatur *main class* Project. Saat kita mengatur *main project*, IDE mengetahui class mana yang akan dijalankan saat kita menjalankan *project*. Selain itu, ini memastikan bahwa elemen Kelas Utama dalam file JAR aplikasi dihasilkan saat kita membangun aplikasi.

Untuk mengatur **Main Class** proyek:

- Klik kanan node Project ImageDisplayApp dan pilih Properties.
- Di kotak dialog Properti Project, pilih kategori *Run*.
- Klik tombol Browse yang ada di sebelah bidang Main Class. Kemudian pilih kelas `org.me.myimageapp.ImageDisplay`.



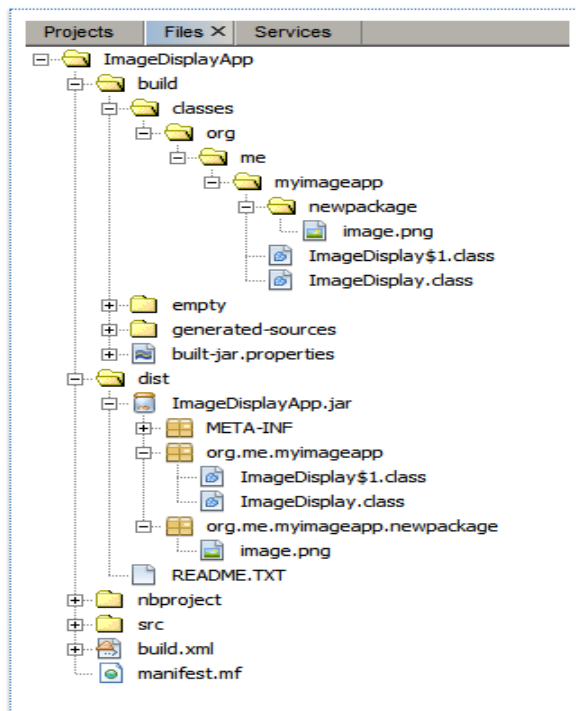
Gambar 9. 9 Main Class Image

- Klik tombol Pilih main class.
- Klik OK untuk menutup kotak dialog project Properties.

Untuk membangun proyek:

Pilih Run > Clean & Build Project (project_name) dari toolbar utama.

Anda dapat melihat produk build aplikasi di jendela Files. Folder build berisi class yang dikompilasi. Folder dist berisi file JAR yang dapat dijalankan yang berisi kelas yang dikompilasi dan Image.



Gambar 9. 10 Class Image

Untuk run project:

Pilih Run > Run Project (*project_name*) dari the main toolbar.

g. Membuat *Custome Code*

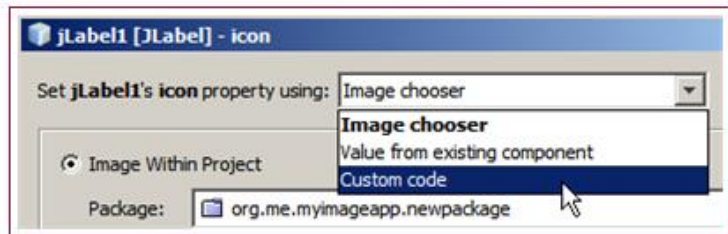
Dalam banyak aplikasi, gambar yang ditampilkan tidak ditentukan secara statis seperti dalam contoh ini. Misalnya Jika gambar yang akan ditampilkan mungkin ditentukan oleh sesuatu yang diklik pengguna.

Jika kita harus dapat memilih gambar untuk ditampilkan secara terprogram, kita dapat menulis kode kustom kita sendiri untuk mengakses dan menampilkan sumber daya. IDE mencegah Anda menulis kode secara langsung di "*guarded blocks*" tampilan Sumber yang berisi kode yang dihasilkan oleh Pembuat GUI. Namun, kita dapat menyisipkan kode di blok yang dijaga melalui editor properti yang dapat Anda akses melalui jendela Properties. Menggunakan editor properti dengan cara ini memastikan bahwa kode kustom Anda tidak hilang saat kita membuat perubahan desain di GUI Builder.

Misalnya, untuk menulis kode khusus untuk properti ikon JLabel:

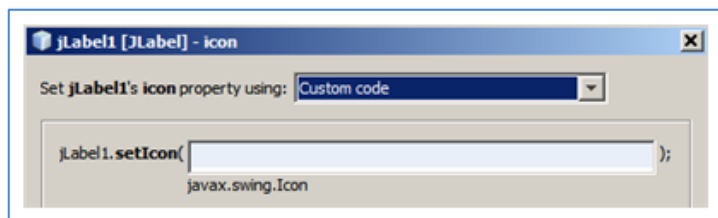
- 1) Pilih JLabel di Design View atau di jendela Navigator.

- 2) Pada jendela Properties, klik tombol elipsis (...) yang ada di sebelah properti Icon.
- 3) Dari daftar dropdown di bagian atas kotak dialog, pilih opsi **Custom Code**.



Gambar 9. 11 Custom Code

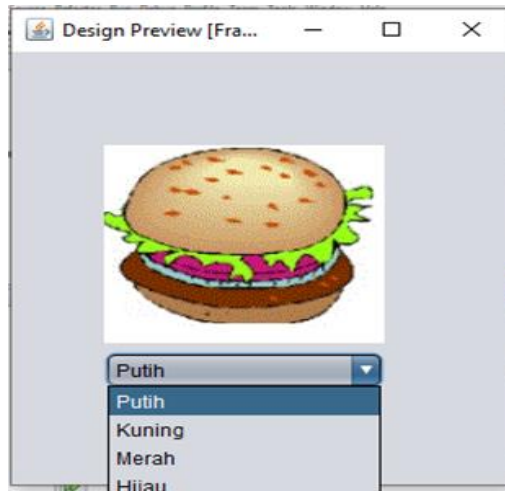
Opsi *Custom Code* di editor properti ini memungkinkan kita mengisi sendiri parameter metode `setIcon`. Kita dapat mengisi parameter ini dengan logika yang diperlukan atau dengan panggilan ke metode terpisah yang telah kita kodekan di tempat lain pada kelas.



Gambar 9. 12 Parameter Custom Code

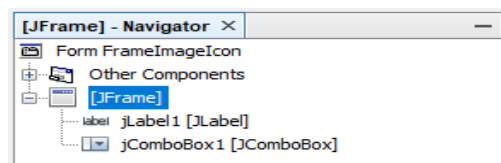
C. LATIHAN

1. Buat project baru namakan project: LatihanImagelcon
2. Siapkan frame baru namakan : Framelmagelcon
3. Desain Frame jika di preview seperti gambar dibawah ini:



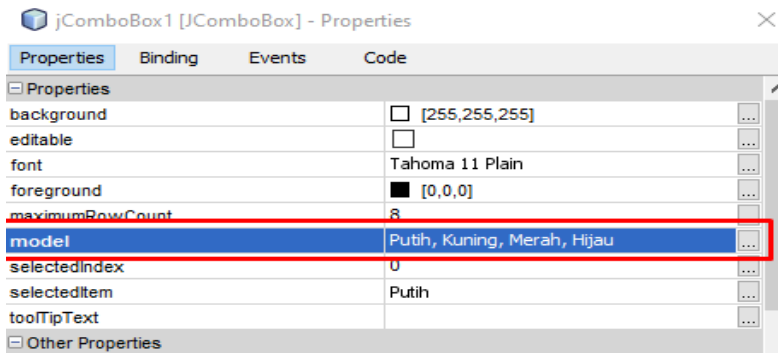
Gambar 9. 13 Latihan Image

4. Komponen yang dibutuhkan pada Frame, JLabel dan JComboBox. jika dilihat dari navigator terdiri dari



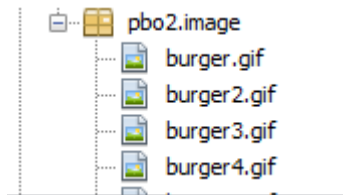
Gambar 9. 14 Navigator Latihan Image

5. Pengisian item pada combo propoerties model seperti dibawah ini



Gambar 9. 15 Properties Model Image

6. Siapkan folder image selanjutnya copy 4 file gif ke target folder image



Gambar 9. 16 Folder Image

7. Hasil yang diharapkan ketika run file, jika pada combo memilih putih akan menampilkan gambar burger dengan latar putih , jika memilih warna lain akan menyesuaikan warna latar sesuai item yang dipilih (Gunakan fungsi if..Else)

8. Berikan kode jComboBox pada events

>actions>actionPerformed

private void

jComboBox1ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:


```

if (jComboBox1.getSelectedItem().equals("Kuning")){
    ImageIcon myImage = new
    ImageIcon("E:/agustav/Unpam/Materi
    Kuliah/Algoritma
    Pemrograman_Java/Latihan/PBO2/src/pbo2/image/
    burger2.gif");
    jLabel1.setIcon(myImage);
}else
if(jComboBox1.getSelectedItem().equals("Merah")){
    ImageIcon myImage = new
    ImageIcon("E:/agustav/Unpam/Materi
    Kuliah/Algoritma
    Pemrograman_Java/Latihan/PBO2/src/pbo2/image/
    burger3.gif");
    jLabel1.setIcon(myImage);
}else
if(jComboBox1.getSelectedItem().equals("Hijau")){
    ImageIcon myImage = new
    ImageIcon("E:/agustav/Unpam/Materi
    Kuliah/Algoritma
    Pemrograman_Java/Latihan/PBO2/src/pbo2/image/
    burger4.gif");
    jLabel1.setIcon(myImage);
} else {
    ImageIcon myImage = new
    ImageIcon("E:/agustav/Unpam/Materi
    Kuliah/Algoritma

```

```

Pemrograman_Java/Latihan/PBO2/src/pbo2/image/
burger.gif");
        jLabel1.setIcon(myImage);
    }
}

```

D. REFERENSI

Philip Conrod & Lou Tylee. Learn Java™ GUI Applications A JFC Swing NetBeans Tutorial 8th Edition , Kidware Software, LLC, 2015

Danny Poo Derek Kiong Swarnalatha Ashok, Object-Oriented Second edition Programming and Java , Springer 2008

<https://www.javatpoint.com/java-bufferedreader-class>, diakses pada tanggal 23 Desember 2021.

https://www.homeandlearn.co.uk/java/java_check_boxes.html, diakses pada tanggal 23 Juni 2022.

https://netbeans.apache.org/kb/docs/java/gui-image_display.html#_creating_the_application_form, pada diakses pada tanggal 23 Juni 2022.

Romi satrio Wahono, Java Gui, 2016