**BAB II**

**JAVABEAN & GIT**

**2.1 Tujuan**

* + - 1. Praktikan mengetahui dan memahami tentang apa itu Java Bean
      2. Praktikan mengetahui fungsi dari Java Bean
      3. Praktikan dapat mengimplementasikan komponen dari Java Bean ke dalam sebuah aplikasi
      4. Praktikan dapat membuat sebuah program dengan menggunakan komponen dan beberapa method menggunakan connection mode
      5. Praktikan dapat memahami sebuah program yang telah terbentuk menggunakan berbagai gabungan komponen dan method connection mode

**2.2 Alat dan Bahan**

* + 1. Laptop

Pada praktikum ini digunakan sebuah laptop bersistem operasikan windows.



Gambar 2.1 Laptop

* + 1. Netbeans IDE 8.1

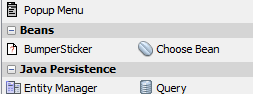
Selain laptop digunakan sebuah program atau IDE yang digunakan untuk membuat sebuah program berbahasa pemrogaman java yaitu IDE Netbeans



Gambar 2.2 IDE Netbeans

* + 1. BumperSticker.jar

Merupakan sebuah komponen yang digunakan untuk membuat program dengan komponen beranimasi



Gambar 2.3 BumperSticker

* + 1. Git

Git adalah version control system yang digunakan para developer untuk mengembangkan software secara bersama-bersama.



Gmabar 2.4 Logo Git

**2.3 Dasar Teori**

**2.3.1 Java Bean**

Java Bean merupakan salah satu model komponen Java yang mendukung prinsip reusability pada proses pengembangan perangkat lunak. Apa itu model komponen? Model komponen dapat dijalankan secara independen atau dapat dijalankan bersama dengan komponen-komponen lain secara interaktif untuk melakukan suatu pekerjaan yang kompleks.

Java Beans dapat dimanipulasi secara visual menggunakan builder tool. Dengan Java Bean memungkinkan pengguna untuk membangun aplikasi secara mudah. Java Beans dapat berupa **Visual Bean** (misalnya : button, text-box) dan **Non-Visual bean** (misalnya : FTP, SMTP, ZipCode Validator). Ciri utama dari Java Bean adalah method-methodnya selalu diawali dengan kata Get atau Set.

Java Bean bersifat portable, sehingga tidak perlu diinstall pada sistem operasi. Arsitekturnya dibangun melalui kolaborasi antar industri dan mengijinkan developer untuk menulis ulang komponen ke dalam bahasa pemrograman Java. Java Beans mendukung beberapa fungsionalitas sebagai berikut :

1. Event

Event merupakan suatu *message*  dari satu objek ke objek yang lain, dimana Event akan memberitahukan ke penerima bahwa telah terjadi ‘sesuatu’, atau sederhananya Event digunakan untuk komunikasi antar bean. Event dibedakan menjadi 3, yaitu *event source, event listener, dan event object.*

1. Property

Mendefinisikan krakteristik dari suatu bean atau method untuk get/set property values. Property sendiri dibedakan atas 4 jenis, *simple property, indexed property, bound property, dan constrained property*.

1. Persistence

Memungkinkan beans untuk menyimpan dan merestore, atau sebagai development tool menyimpan Java Bean di dalam hardisk dan dapat diload pada suatu waktu. Selain itu, memelihara nilai property tanpa tergantung apakah Java Beans dapat menyimpan memori atau hardisk.

1. Object Serialization

Menyimpan semua content dari suatu object pada data stream dan menggenerate kembali object ketika membuat dari data stream.

1. Introspection

Untuk mengetahui property, events dan method. Misalnya dibuat class XXXBeanInfo, khusus untuk menjelaskan class XXX secara detail. BeanInfo ini akan menjelaskan informasi bean seperti icon, property, method dan informasi lainnya.

1. Java Reflection API

Yakni Java API yang berfungsi untuk menemukan metode, field field, construktor, superclasses pada RUNTIME. API ini juga digunakan untuk menulis development tools yang lain yakni debuggers, class browsers, GUI builders.

**2.3.2 Swing/AWT**

AWT (Abstract Window Toolkit) adalah sekumpulan dari aplikasi program antarmuka atau Application Program Interfaces(API) yang digunakan oleh para permragaman java untuk membuat objek Graphical User Interface (GUI) seperti Button, Scroll bars dan Windows. Sedangkan SWING merupakan aplikasi program antarmuka seperti halnya AWT. Swing menggunakan metode Painted yaitu objek GUI digambar dalam frame kosong.

**2.3.3 GIT**

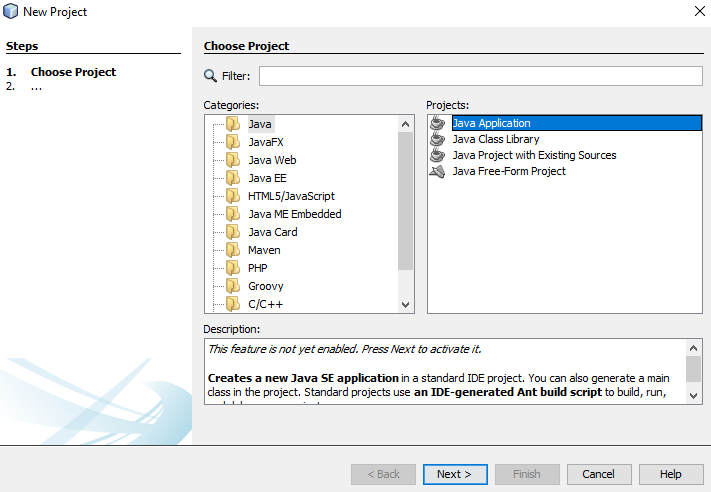
Git adalah *version control system* yang digunakan para developer untuk mengembangkan software secara bersama-bersama. Fungsi utama git yaitu mengatur versi dari source code program anda dengan mengasih tanda baris dan code mana yang ditambah atau diganti.

Berikut perintah-perintah dasar dari Git:

* Git init : untuk membuat *repository* pada file lokal yang nantinya ada folder .git
* Git status : untuk mengetahui status dari *repository* lokal
* Git add : menambahkan file baru pada *repository* yang dipilih
* Git commit : untuk menyimpan perubahan yang dilakukan, tetapi tidak ada perubahan pada *remote repository.*
* Git push : untuk mengirimkan perubahan file setelah di commit ke *remote repository.*
* Git branch : melihat seluruh *branch*yang ada pada repository
* Git checkout : menukar *branch*yang aktif dengan *branch*yang dipilih
* GIt merge : untuk menggabungkan *branch*yang aktif dan *branch*yang dipilih
* Git clone : membuat Salinan *repository*lokal

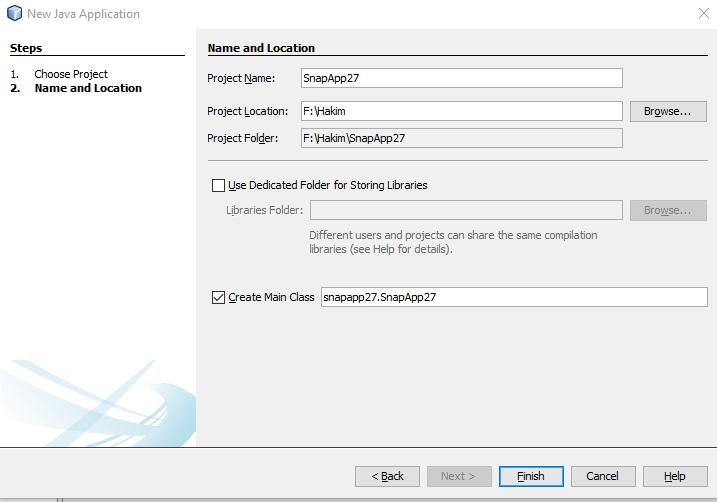
**2.4 Langkah Kerja**

1. **NETBEAN**
2. Buat file baru : File 🡪 New Project 🡪 Java 🡪 Java application

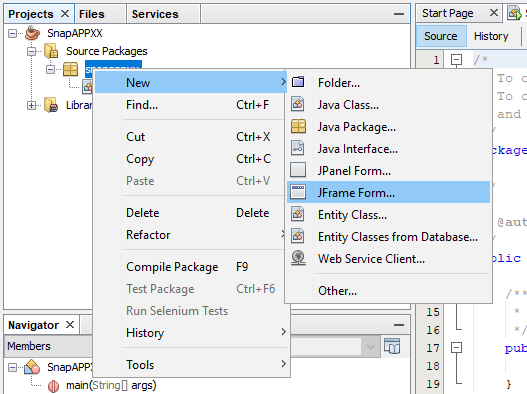


Gambar 2.4 Membuat project baru

1. Beri nama SnapApp27

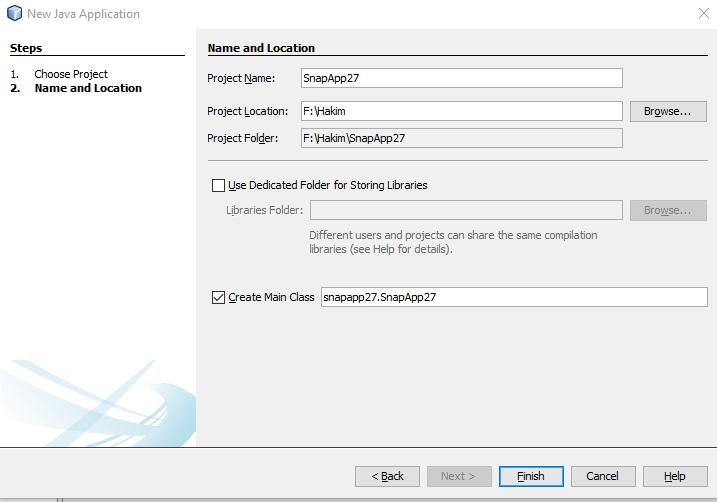


Gambar 2.5 Memberi nama project

1. Klik kanan pada package SnapApp 🡪 pilih New 🡪 pilih JFrame Form

Gambar 2.6 Membuat JFrame Form

1. Beri nama SnapFrame



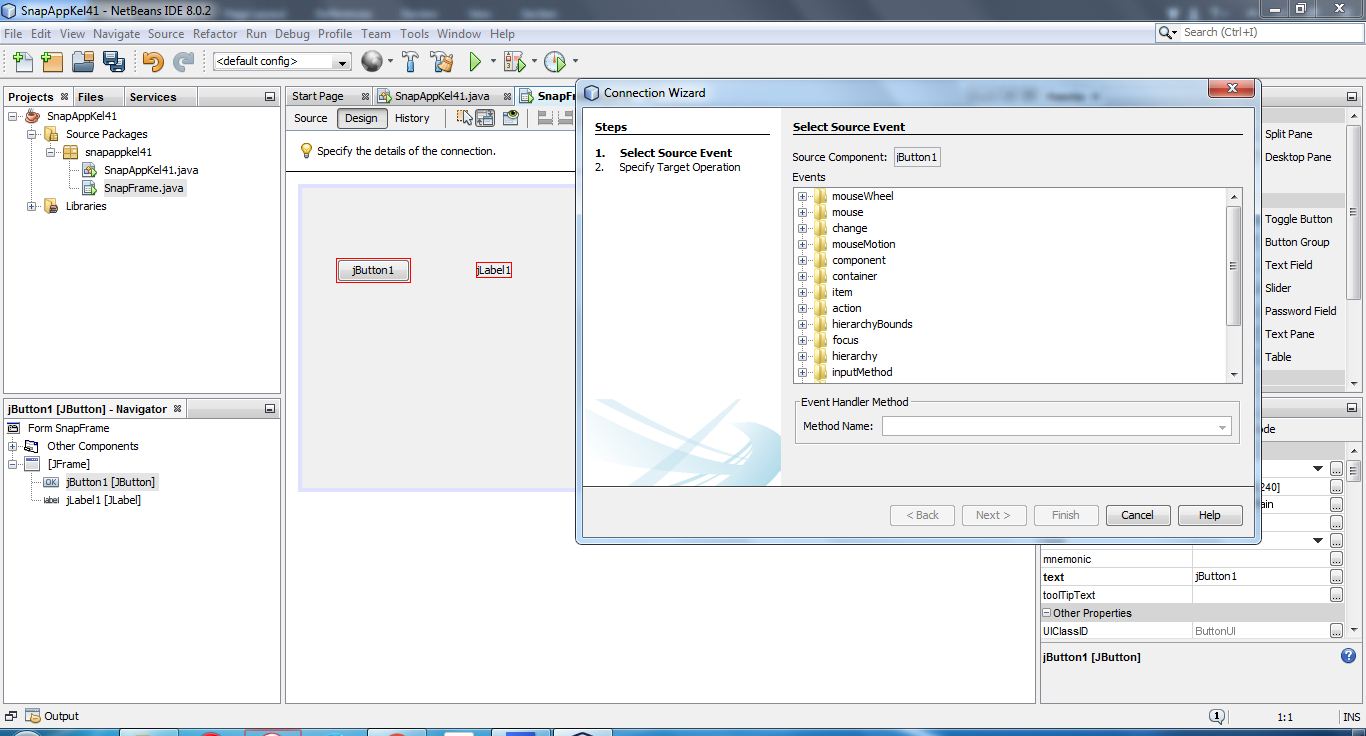
Gambar 2.7 Memberi nama JFrame Form

1. Buat tampilan seperti berikut ini



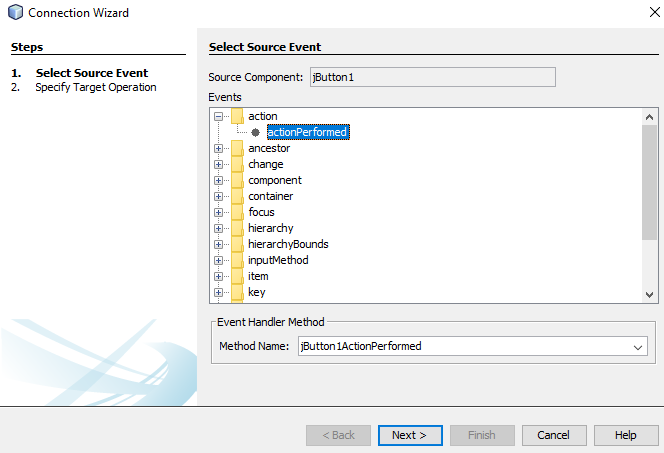
Gambar 2.8 Tampilan pada snapframe

1. Klik Connection Mode, kemudian klik pada tombol, kemudian klik pada label. Connection Mode digunakan untuk memberikan nilai pada label ketika button diberi action.



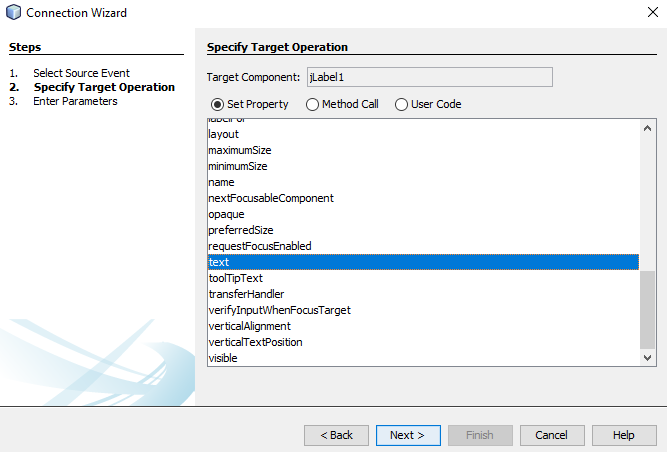
Gambar 2.9 Connection mode

1. Setelah itu akan muncul connection Wizard, pilih action 🡪 actionPerformed. Langkah ini akan membuat button ketika diberi action akan menjalankan events actionPerformed.



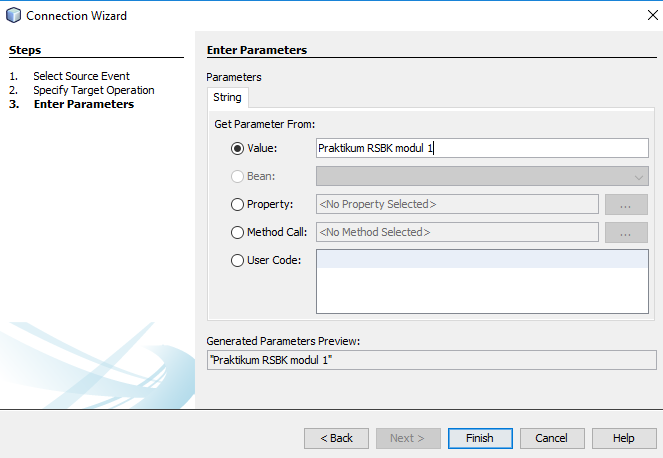
Gambar 2.10 Memilih action

1. Sekarang pengaturan pada labelnya, pilih set property 🡪 Text. Hal ini bertujuan untuk memilih action yang akan terjadi pada label.



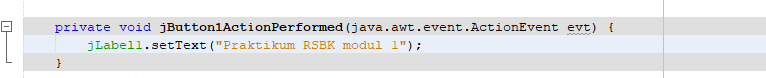
Gambar 2.11 Memilih ouput pada target component

1. Isi Value “praktikum rsbk modul 1”, kemudian finish. Ini akan menghasilkan output “praktikum rsbk modul 1” pada label ketika button diberi action.



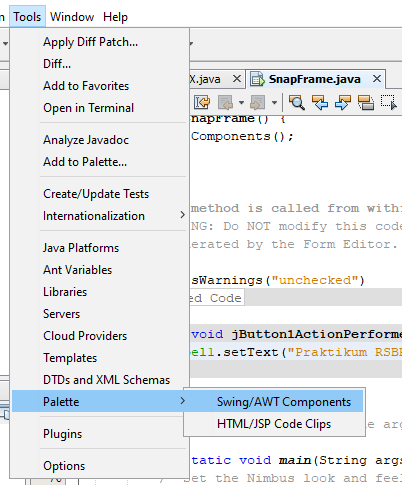
Gambar 2.12 Isi value label

1. Secara otomatis value akan muncul seperti source code berikut



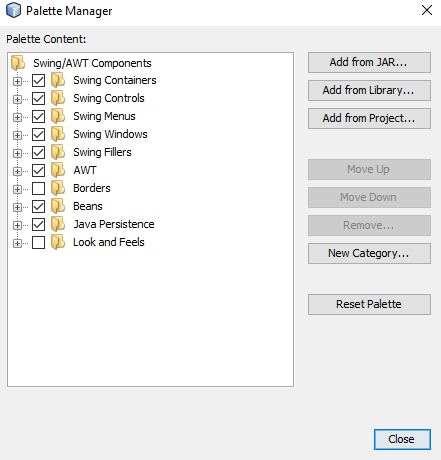
Gambar 2.13 Source code jButtonActionPerformed

1. Klik Tool 🡪 Pallete 🡪 Swing / AWT Components untuk mengimport component baru.



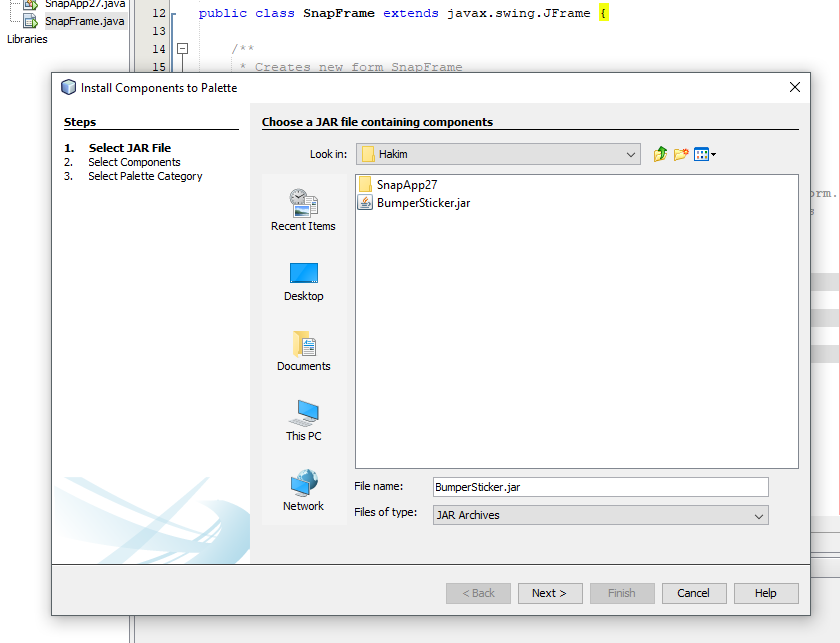
Gambar 2.14 Import beans

1. Klik Add From Jar untuk mulai memilih package component.



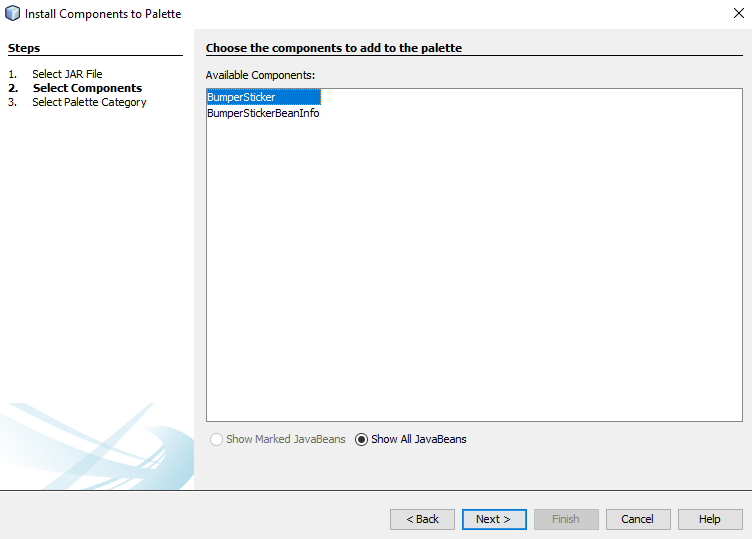
Gambar 2.15 Menambahkan component dari .jar yang sudah ada

1. Kemudian cari lokasi File BumperSticker, pilih filenya BumperSticker.jar



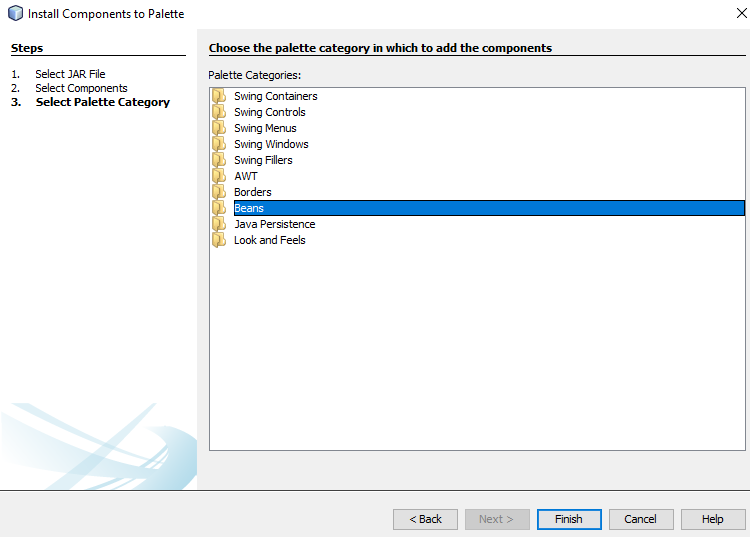
Gambar 2.16 Mencari lokasi bumperSticker.jar

1. Pilih bumperSticker



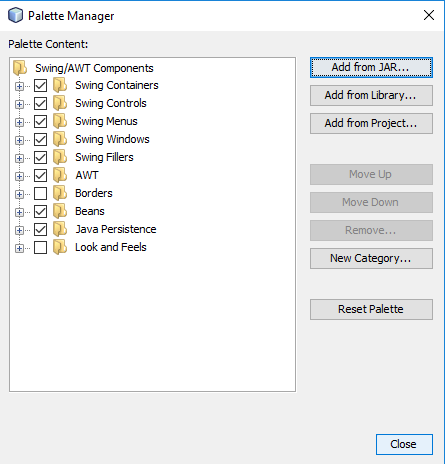
Gambar 2.17 Memilih bumperSticker

1. Pilih Beans yang berarti meletakkan component baru tersebut ke folder beans.



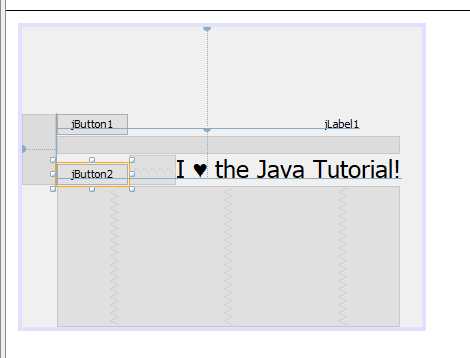
Gambar 2.18 Memilih folder tujuan

1. Klik close setelah selesai mengimport component.



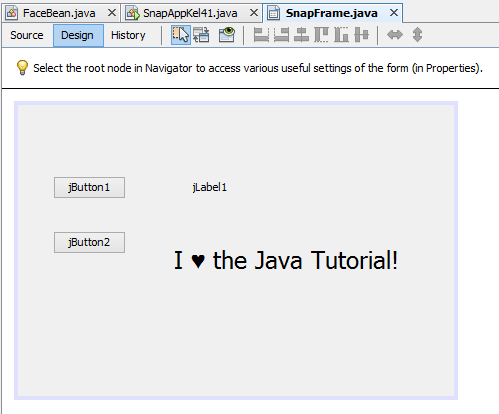
Gambar 2.19 Selesai mengimport component

1. Drag Beans 🡪 BumperSticker ke Form



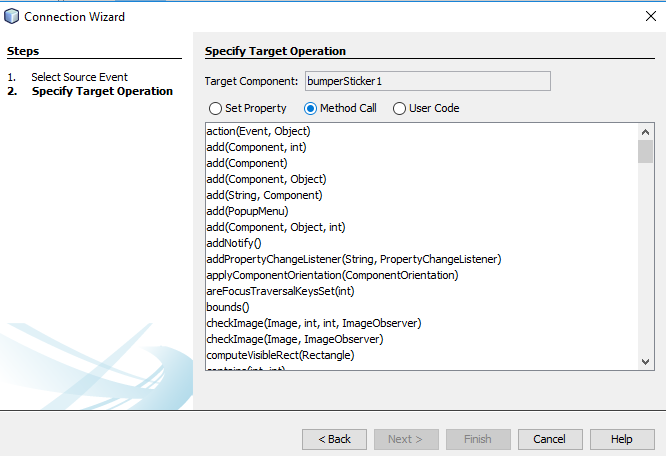
Gambar 2.20 Menambahkan bumpersticker ke form

Tambahkan satu button lagi untuk menjalankan animasinya,

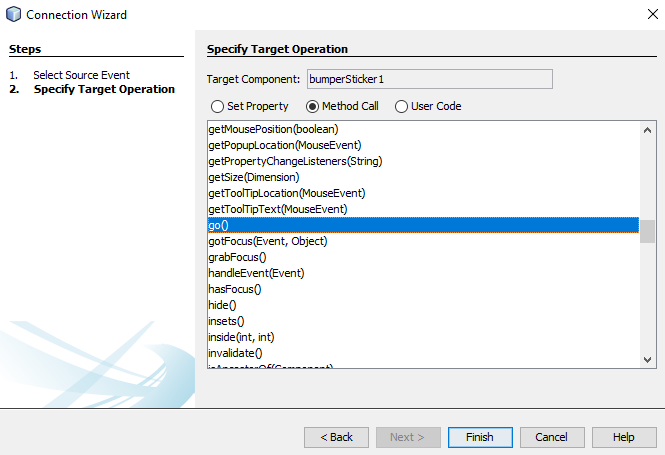


Gambar 2.21 Menambahkan button

1. Koneksikan Button 2 dengan Bean BumperSticker, gunakan action performed, pilih MethodCall, pilih method go(). Method go() berguna untuk menjalankan atau memainkan animasi yang ada terhubung terhadap button ke komponen java yang telah di import.

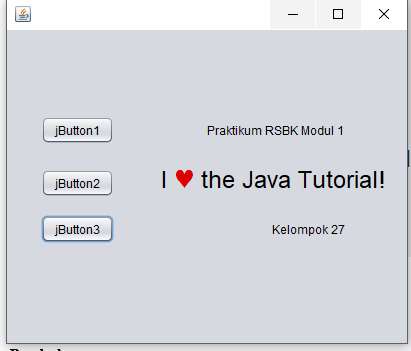


Gambar 2.22 Connection mode antara bumpersticker dan button



Gambar 2.23 Connection mode antara bumpersticker dan button

1. Setelah itu run projectnya. Jika jButton1 ditekan maka label akan menampilkan teks “praktikum rsbk modul 1” dan jika jButton2 ditekan maka gambar love akan berkedip merah hitam.

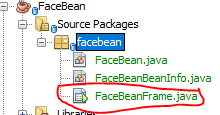


Gambar 2.24 Hasil run program

**2.5 Tugas dan Pembahasan**

Pada tugas modul 1 praktikum RSBK 2018 kali ini, praktikan diminta untuk menggunakan component facebean yang telah diberikan. Dengan menggunakan fungsi tiga tombol button dan sebuah label.

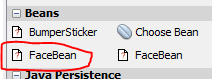
Setelah project di download, kemudian project tersebut di clean & build. Setelah itu pada package facebean buat JFrame Form baru yang akan digunakan untuk menggunakan komponen yang akan digunakan.



Gambar 2.25 Hasil Jframe Form yang ditambahkan

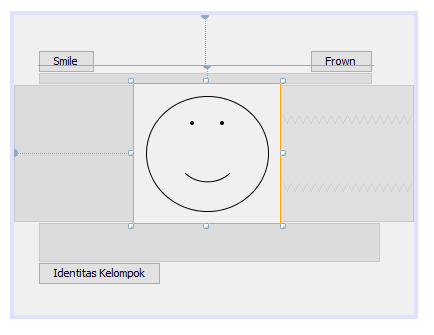
Kemudian lakukan penginputan terhadap komponen facebean melalui tools => palette => AWT/Swing => Add from Jar. Kemudian cari file dist dan setelahnya input file facebean.jar yang ada didalam file dist.

Setelah berhasil maka component facebean akan muncul di pallete. Seperti gambar 2.26.



Gambar 2.26 Hasil Facebean yang muncul di pallete

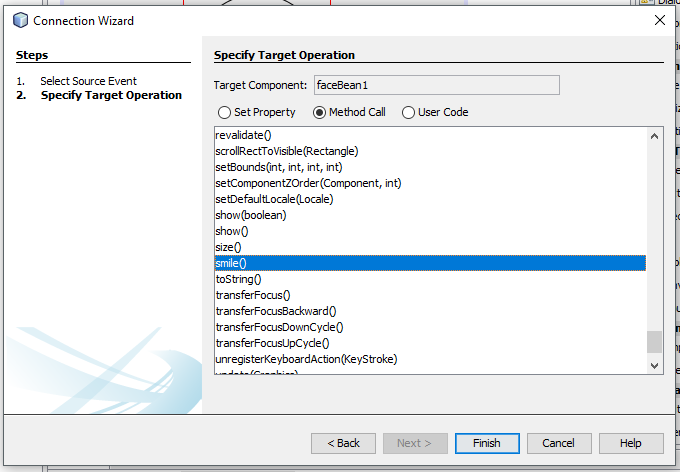
Kemudian drag and drop komponen facebean, beserta tiga button dan sebuah label pada jframe form. Susun sehingga menjadi seperti gambar 2.27.



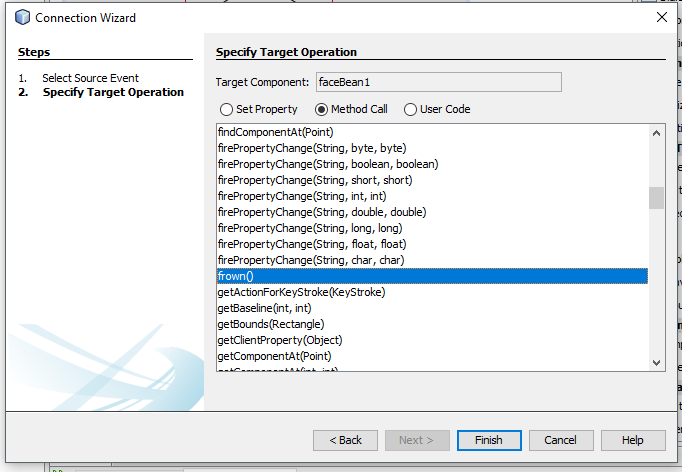
Gambar 2.27 Susunan komponen

Kemudian koneksikan antara button smile dan frown dengan komponen facebean, button Identitas kelompok dengan label.

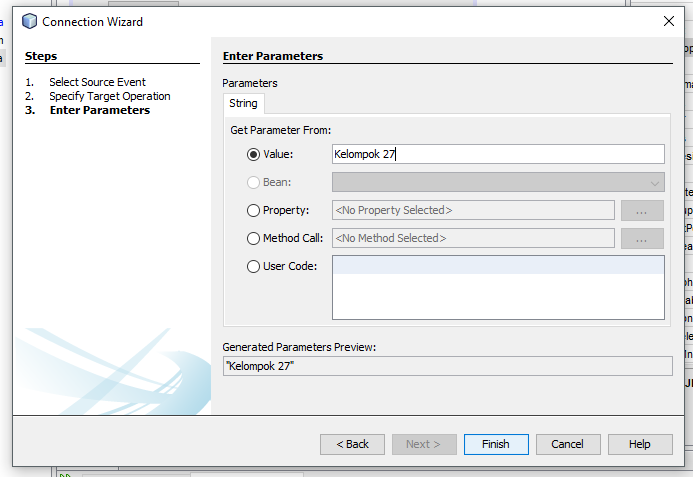
Pada button Smile dan Frown action yang dilakukan adalah dengan menggunakan method smile() dan frown() seperti gambar 2.28 dan gambar 2.29. Kemudian hubungkan button Identitas dengan label untuk memunculkan text seperti gambar 2.30.



Gambar 2.28 Menggunakan Method Smile

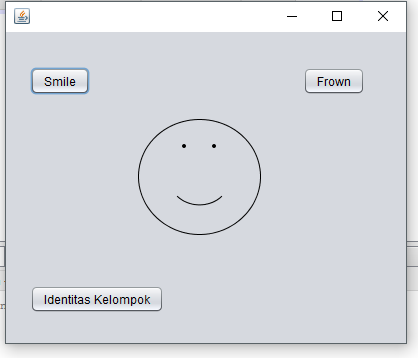


Gambar 2.29 Menggunakan Method Frown

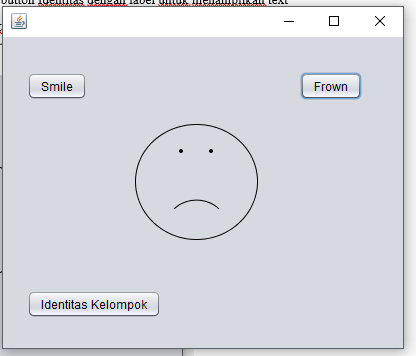


Gambar 2.30 Menghubungkan button Identitas dengan label untuk menampilkan text

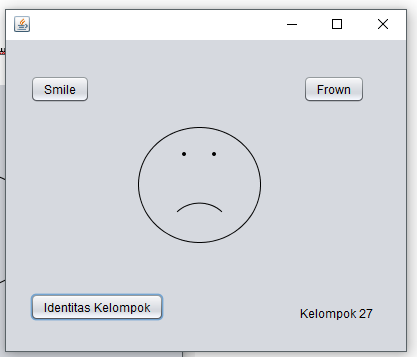
Sehingga hasil dari project tersebut dapat berjalan seperti gambar 2.31 berikut :



Gambar 2.31 Kondisi Ketika ditekan tombol button Smile



Gambar 2.32 Kondisi Ketika ditekan tombol button Frown



Gambar 2.33 Kondisi Ketika ditekan tombol button Identitas

**2.7 Kesimpulan**

1. Dalam menyambungkan berbagai komponen, digunakan *connection mode* dan pengaturan secara automatis tanpa harus melakukan penulisan *source code*.
2. Method digunakan untuk melakukan sebuah perintah pada komponen.
3. Komponen java beans merupakan suatu komponen yang dapat digunakan untuk mengubah tampilan beranimasi.
4. State id variable yang terdapat pada komponen merupakan variable private yang tidak bisa diganti
5. Pada perubahan label dihubungkan melalui connection mode yang nantinya akan diubah menjadi suatu method
6. File .jar pada suatu komponen program akan muncul pada file dist setelah program di lakukan clean and build
7. Ketika ingin memunculkan program atau project yang dibuat diperlukan untuk menambahkan fungsi visible
8. Dengan membagi fungsi-fungsi program menjadi sebuah komponen dapat mempermudah pengembangannya ataupun mencari kesalahan yang terjadi.

**Ketentuan Asistensi:**

Subjek “Laporan\_RSBK\_BAB II\_KelompokXX”

Shift 1 : ACC ke Adi ([madinugroho@student.ce.undip.ac.id](mailto:madinugroho@student.ce.undip.ac.id)) Line : @adinugi / 081212583838

Shift 2 & 3 : ACC ke Bima ([bimafajarsetiawan@gmail.com](mailto:bimafajarsetiawan@gmail.com)) Line : @bima\_iero / 089610687965

Deadline ACC 6 November 2018