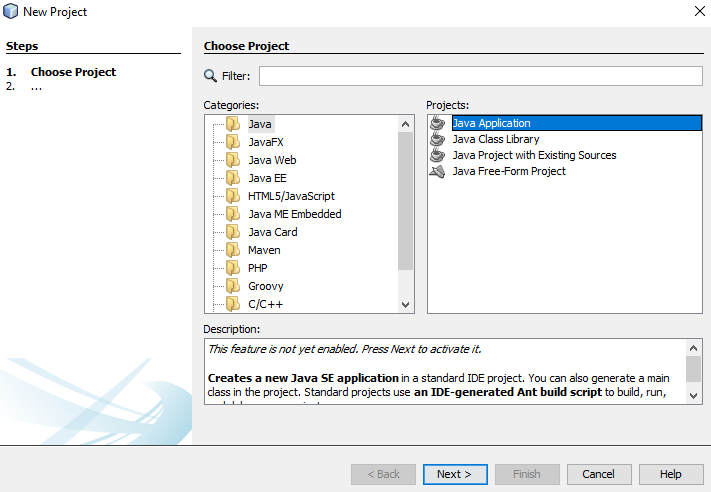
**BAB II**

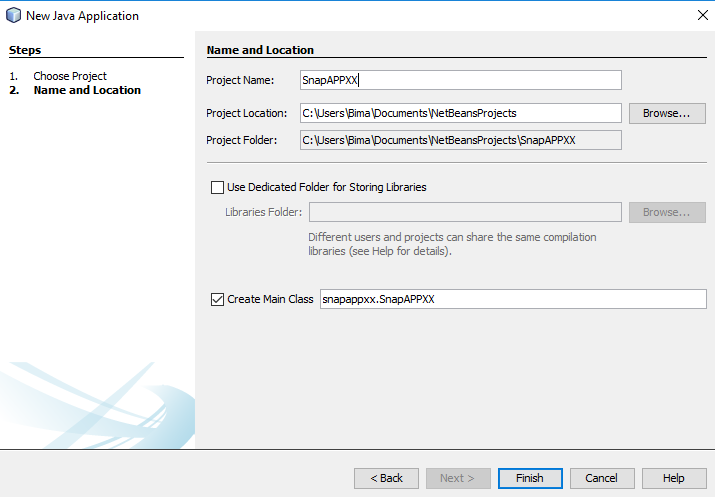
**JAVA BEAN**

* 1. **TUJUAN**

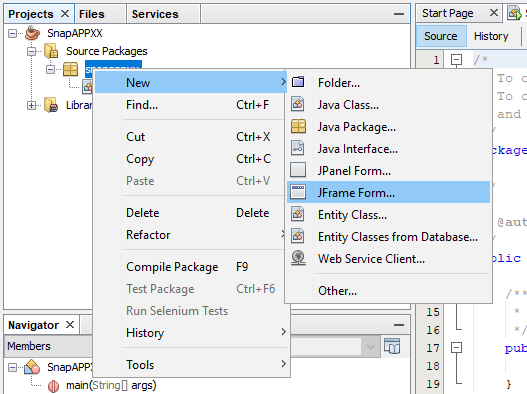
1. Praktikan dapat mengetahui dasar Java Bean
2. Praktikan dapat mengetahui fungsi Java Bean
3. Praktikan dapat pengetahui penggunaan command Git dan Github
   1. **ALAT DAN BAHAN**
4. Laptop
5. NetBeans
6. BumperSticker.Jar
7. Akun Github yang sudah terverifikasi
8. Git
   1. **LANGKAH KERJA**
9. **NETBEAN**
10. Buat file baru : File 🡪 New Project 🡪 Java 🡪 Java application



Gambar 2.1 Membuat project baru

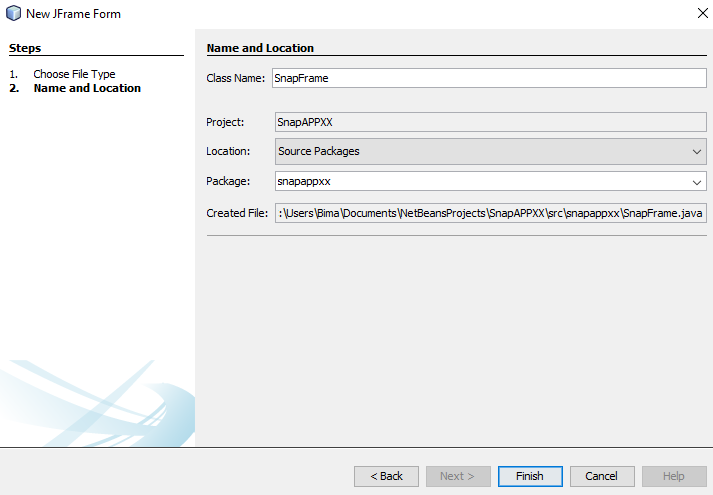
1. Beri nama SnapAppXX (XX adalah kelompok kalian)

Gambar 2.2 Memberi nama project

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

Gambar 2.3 Membuat JFrame Form

1. Beri nama SnapFrame



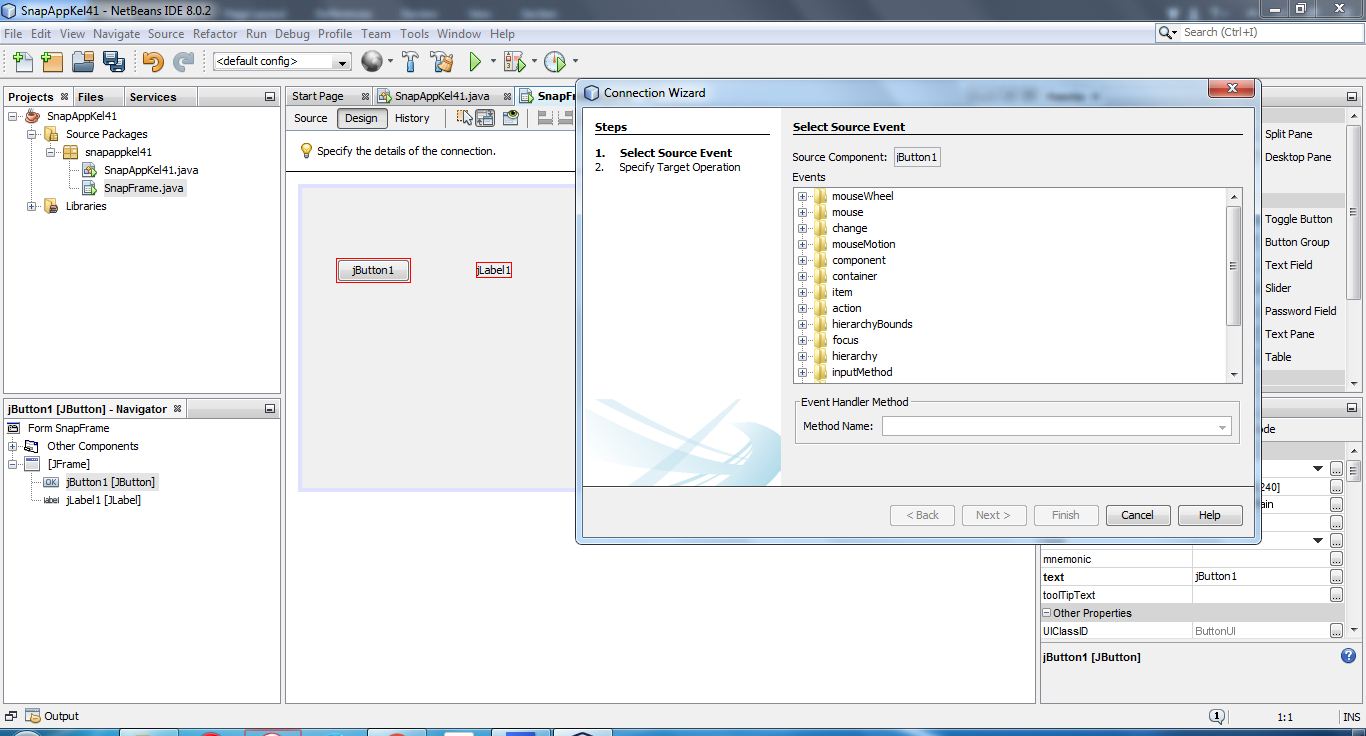
Gambar 2.4 Memberi nama JFrame Form

1. Buat tampilan seperti berikut ini



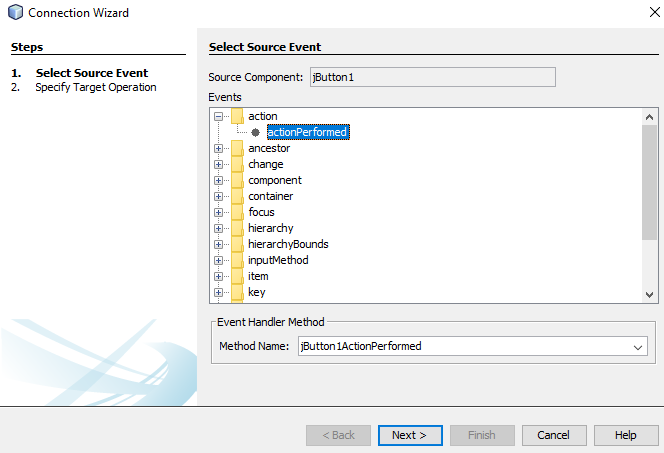
Gambar 2.5 Tampilan pada snapframe

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



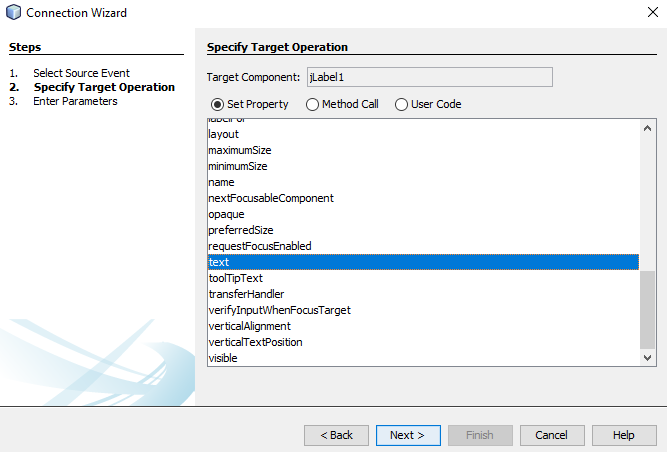
Gambar 2.6 Connection mode

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



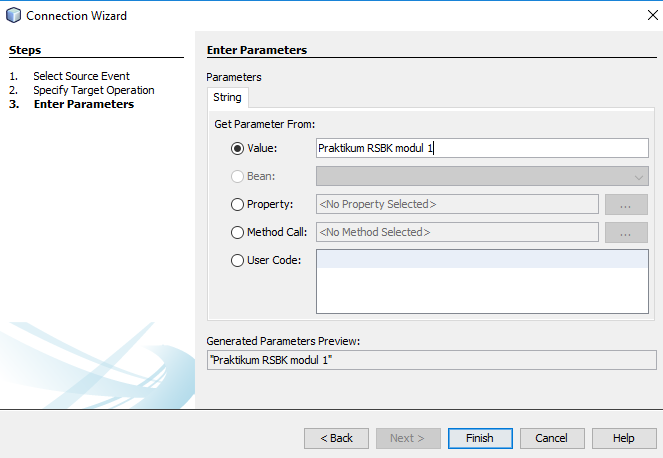
Gambar 2.7 Memilih action

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



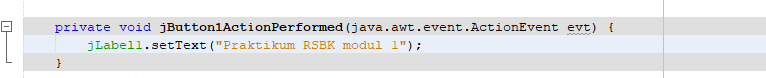
Gambar 2.8 Memilih ouput pada target component

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



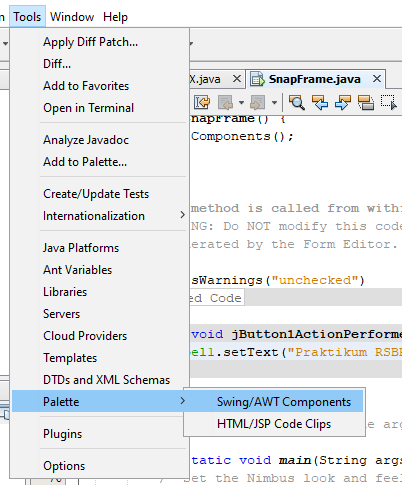
Gambar 2.9 Isi value label

1. Secara otomatis value akan muncul seperti source code berikut



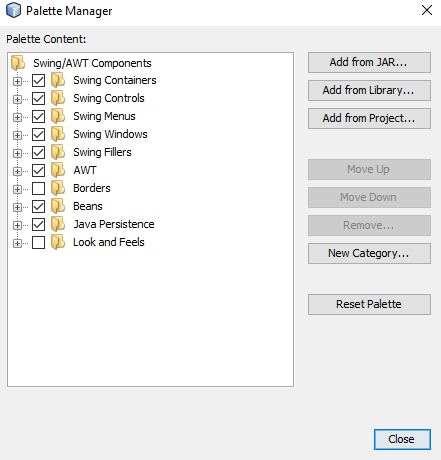
Gambar 2.10 Source code jButtonActionPerformed

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



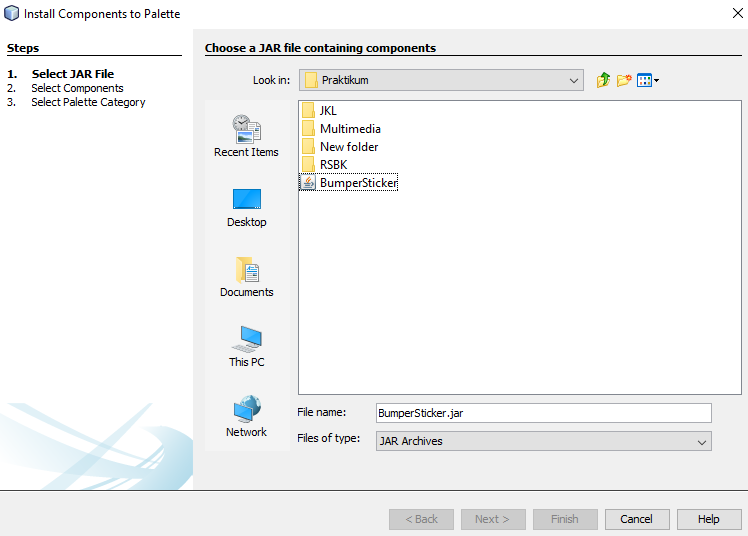
Gambar 2.11 Import beans

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



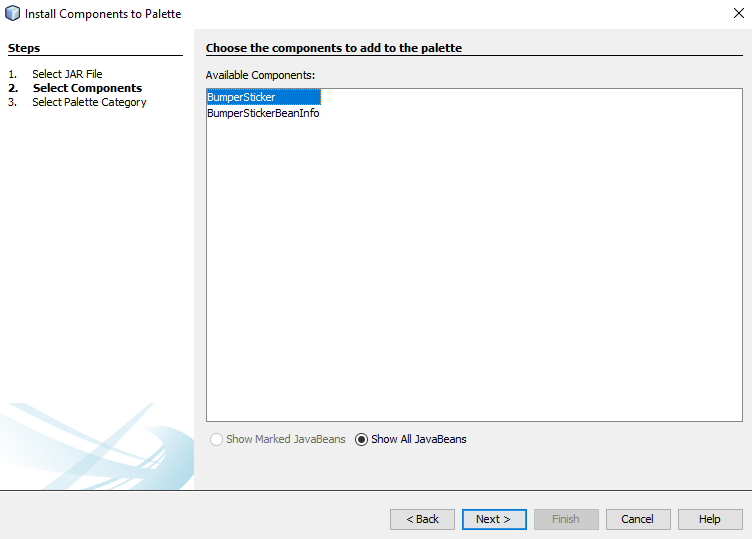
Gambar 2.12 Menambahkan component dari .jar yang sudah ada

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



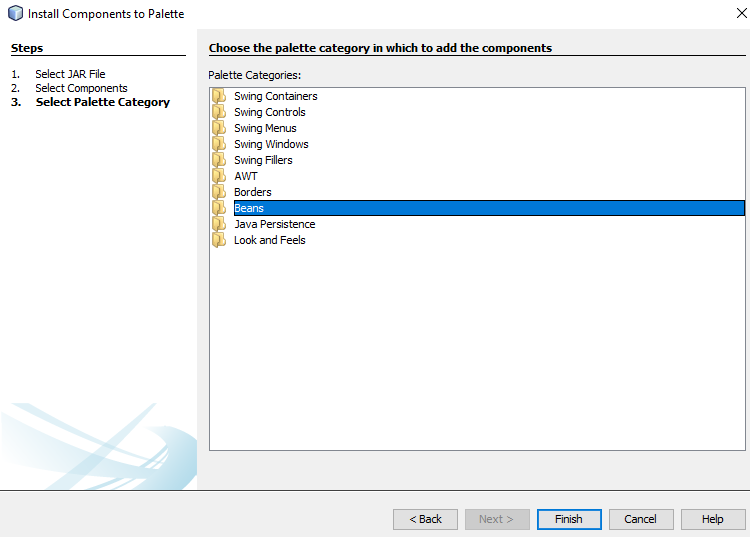
Gambar 2.13 Mencari lokasi bumperSticker.jar

1. Pilih bumperSticker



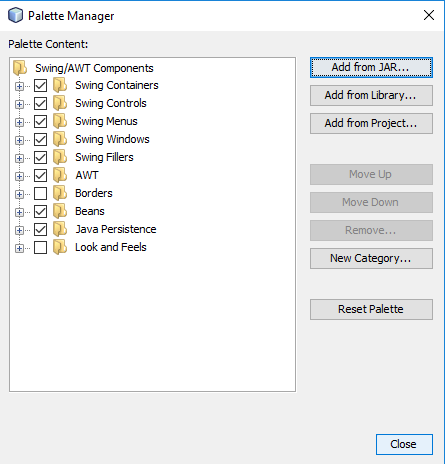
Gambar 2.14 Memilih bumperSticker

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



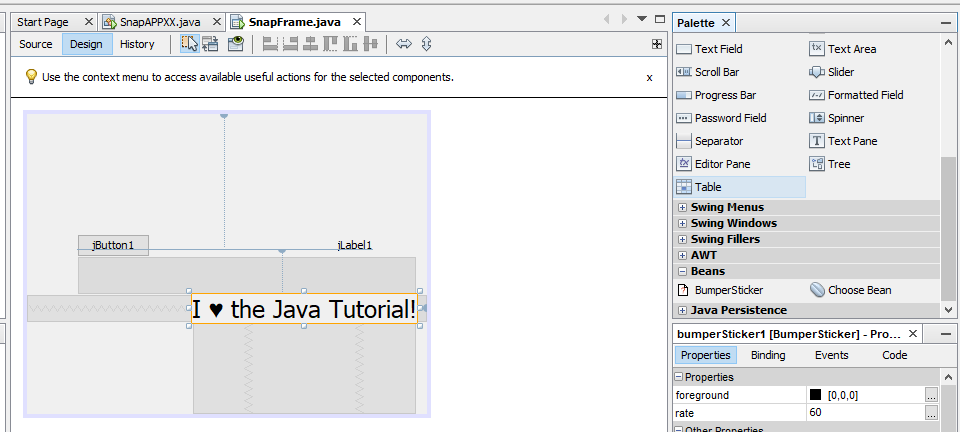
Gambar 2.15 Memilih folder tujuan

1. Klik close setelah selesai mengimport component.



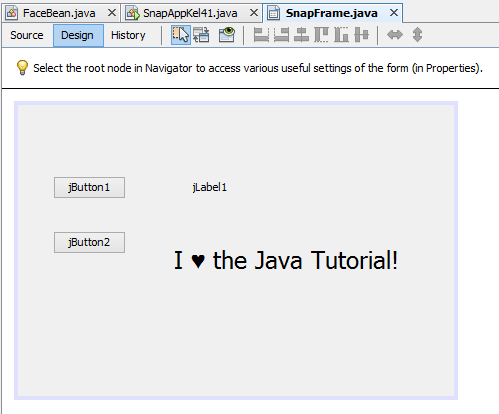
Gambar 2.16 Selesai mengimport component

1. Drag Beans 🡪 BumperSticker ke Form



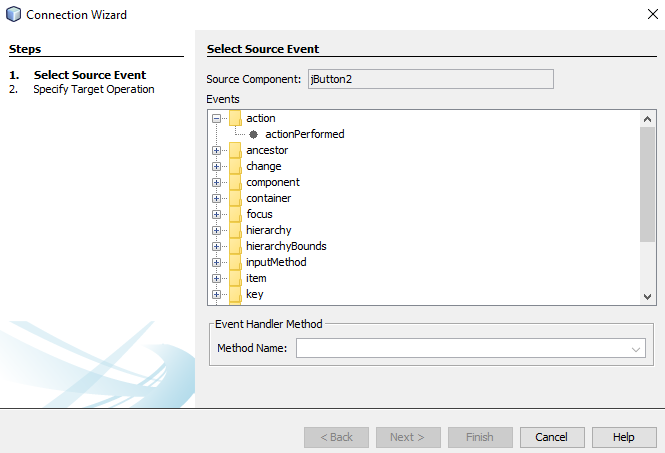
Gambar 2.17 Menambahkan bumpersticker ke form

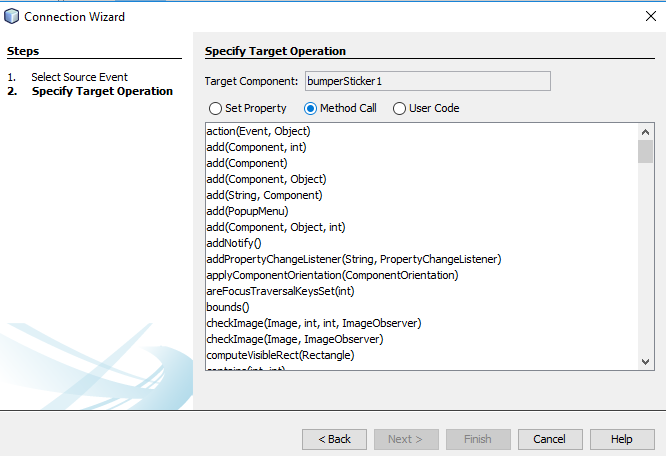
Tambahkan satu button lagi untuk menjalankan animasinya,

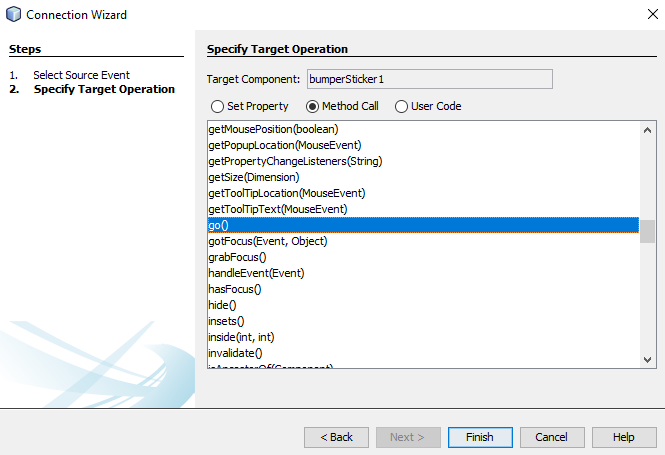


Gambar 2.18 Menambahkan button

1. Koneksikan Button 2 dengan Bean BumperSticker, gunakan action performed, pilih MethodCall, pilih method go()

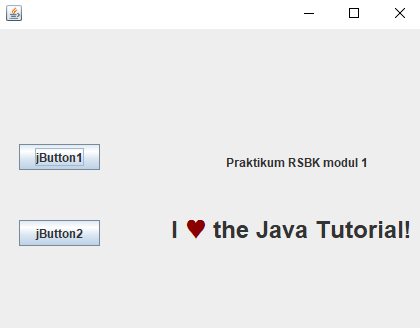






Gambar 2.19 Connection mode antara bumpersticker dan button

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 2.20 Hasil run program

1. **GIT DAN GITHUB**
2. Buka website github di ww.github.com pada browser dan login sesuai dengan akun masing masing
3. Pilih tombol new yang terletak di sebelah kiri atas di dashboard masing masing akun untuk membuat repository pada akun github masing masing.
4. Beri nama repository tersebut dengan nama PraktikumRSBKkelxx (xx = nomer kelompok), kemudia klik create repository
5. Setelah membuat repository maka repository tersebut siap diisi, copy link https yang ada di repository tersebut untuk dijadikan remote link yang nantinya dipakau untuk mengakses dari git.
6. Buka folder project java bean yang sudah dibuat tadi, kemudian klik kanan dan pilih opsi git bash untuk membuka console git.
7. lalu ketikan beberapa perintah CLI, seperti berikut:

Git init

Git add .

Git commit –m “comment”

Git remote add origin link repository yang sudah kalian copy tadi

Git push origin master

Publish / init repository digunakan untuk mengunggah projek yang ada pada penyimpanan local atau computer kita ke web hosting github.com. Hal ini bertujuan untuk men-*track* perubahan atau untuk mem-*backup* file dan lain – lain. Untuk mengunggah repository ke web hosting github klik kanan pada folder project yang ingin diunggah.

Keterangan :

* + - * 1. “comment” dapat diganti dengan keterangan tentang apa yang diganti pada project
        2. Origin merupakan nama dari branch

1. Setelah repository berhasil di upload bisa di lihat di akun github masing masing bahwa repository di github sudah terisi dan terhubung dengan local repository yang sudah di buat tadi.
2. Kemudian untuk melakukan perubahan bisa dilakukan dengan cara update repository yang digunakan untuk mengubah code atau file pada repository kita yang ada di github kita sehingga sama dengan repository local, caranya hampir sama dengan saat kita menginisialisasi repository. Namu diawali dengan langkah add file. Langkahnya sebagai berikut :

Git add .

Git commit –m “comment”

Git push origin master

1. Perintah Pull Digunakan untuk mengupdate repository local sehingga sama dengan repository yang ada pada github. Caranya cukup mudah yaitu dengan cara :

Git pull

1. Terahir jika ingin mendownload / menggunakan repository orang lain yang ada di github cuku dengan copy link dari repository yang diinginkan bisa menggunakan Clone

Clone repository bertujuan untuk mengunduh sebuah projek yang ada pada github ke computer atau penyimpanan local, sehingga nantinya dapat diedit sesuai dengan keinginan sendiri. Untuk meng-*clone* repository git cukup ketikan perintah **git clone [url repository]**

Directory dapat diubah tergantung dari dimana kalian mengklik kanan dan membuka bash, misal membuka gitbash di folder D:\contoh. Maka repository akan berada di D:\contoh\[nama repository]