**BAB II**

**JAVABEAN**

**2.1 Tujuan**

1. Praktikan dapat mengetahui dasar Java Bean
2. Praktikan dapat mengetahui fungsi Java Bean
3. Praktikan dapat membuat Java Bean secara mandiri
4. Praktikan dapat mengetahui penggunaan command Git dan Github

**2.2 Alat dan Bahan**

1. Laptop



Gambar 2. Laptop

Laptop digunakan untuk membuat program.

1. NetBeans



Gambar 2. Netbeans

Netbeans adalah IDE yang digunakan untuk membuat Java Visual Application

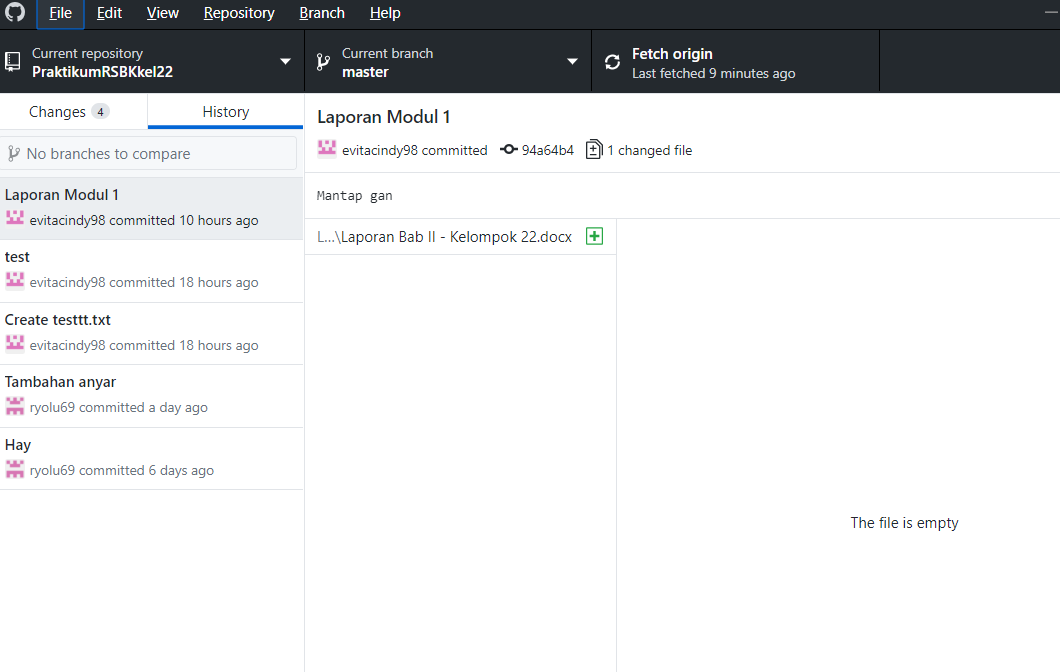
1. BumperSticker.Jar



Gambar 2. BumperSticker.jar

Sebuah Komponen Bumper Sticker yang akan digunakan pada Praktikum

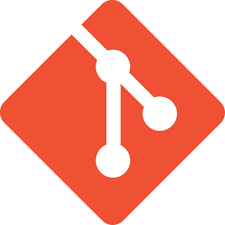
1. Akun Github yang sudah terverifikasi



Gambar 2. Akun GIthub yang sudah Terverifikasi

Akun Github yang terverifikasi digunakan untuk keperluan mengupload tugas dan mendownload modul pada praktikum ini.

1. Git



Gambar 2. Git

Git adalah sebuah software untuk membagikan project dan *sourcecode* dan dapat memanipulasinya secara *real-time* dan ter-tracked.

**2.3 Dasar Teori**

**2.3.1 Java Bean**

Java Bean merupakan format standar pada class java yang dapat digunakan untuk membangun program. Digunakan juga untuk perancangan desktop dan pemrograman web. Java bean bersifat Logic. Pada MVC, Java Bean menduduki tingkatan Model, yang memiliki hak akses ke database.

Java Bean memiliki atribut yaitu : id, scope, class, BeanName, Type.

Java Bean memiliki ciri-ciri yaitu :

1. Memiliki constructor yang bersifat default (constructor kosong)
2. Memiliki variabel dengan akses bukan public (bisa berupa protected, default, private) sebagai penyimpanan data
3. Memiliki method bersifat Get dan Set (memberi dan mendapatkan)

Tag Standar Pada Java Bean :

• <jsp:usebean>

Digunakan untuk meng-instantiate (pembutan objek baru) Java Bean agar dapat digunakan pada halaman JSP. Penggunaan bean ini merupakan salah satu cara untuk memisahkan antara Logic dan Presentation pada JSP.

Contoh : <jsp:useBean id = "oPenilaian" scope = "session" class = “Nilai.Penilaian"/>

• <jsp:setproperty>

Berfungsi untuk men-set nilai properties yang terdapat pada bean. Digunakan bersama dengan action tag <jsp:useBean>

Contoh : <jsp:setProperty name="oPenilaian" property="tglMul" param="tglMul"/>

• <jsp:getproperty>

Berfungsi untuk mengambil nilai property pada Java Bean, dan kebalikan dari action tag <jsp:setProperty>,

Contoh :

<jsp:getProperty name=“beanName” property=“propertyName/>

<jsp:setProperty name=“oPenilaian” property=“tglMul”/>

*(Sumber: http://rentagultom.blogspot.com/2015/06/pertemuan-11-java-bean\_56.html)*

**2.3.2 Swing/AWT**

Abstract Windowing Toolkit (AWT), atau disebut juga “Another Windowing Toolkit”, adalah pustaka windowing bertujuan umum dan multiplatform serta menyediakan sejumlah kelas untuk membuat GUI di Java. Dengan AWT, dapat membuat window, menggambar, manipulasi gambar, dan komponen seperti Button, Scrollbar, Checkbox, TextField, dan menu pull-down.

Swing merupakan perbaikan kelemahan di AWT. Banyak kelas swing menyediakan komponen alternatif terhadap AWT. Contohnya kelas JButton swing menyediakan fungsionalitas lebih banyak dibanding kelas Button. Selain itu komponen swing umumnya diawali dengan huruf “J”, misalnya JButton, JTextField, JFrame, JLabel, JTextArea, JPanel, dan sebagainya. Teknologi swing menggunakan dan memperluas gagasan-gagasan AWT. Sementara, penggunaan komponen Swing ditandai dengan adanya instruksi : import javax.swing.

*(Sumber: http://renamuslimahmihardjo.blogspot.com/2012/10/belajar-awt-dan-swing.html)*

**2.3.3 Prinsip Rekayasa Komponen (Reuse dan Compose)**

 Component-based Software Engineering (CBSE) mulai muncul pada akhir tahun 1990 sebagai pendekatan dalam pengembangan sistem perangkat lunak berdasarkan penggunaan ulang komponen perangkat lunak. Hal ini dibuat diawali dengan frustasi para desainer pengembangan berbasis objek tidak menunjang penggunaan ulang seperti ekspetasi.

Komponen adalah abstrak dengan tingkat lebih tinggi dari objek dan dibedakan melalui tampilannya. Komponen biasanya lebih besar dari satuan objek.

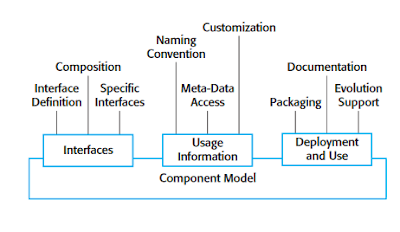
Bagian utama dari CBSE adalah:

* Komponen independen yang keseluruhan dispesifikan oleh tampilannya.
* Komponen standar yang memfasilitasi integrasi dari komponen.
* Middleware yang menyediakan perangkat lunak pendukung untuk intrgrasi komponen
* Proses pengembangan yang disiapkan untuk CBSE

Terdapat beberapa macam karakteristik komponen dalam CBSE, yaitu :

* Standarized 🡪 Standarisasi komponen berarti komponen proses CBSE harus berpatok pada model standar.
* Independent 🡪 Komponen harus bisa independen dan bisa membuat dan menjalankan komponen tanpa bergantung pada komponen lainnya.
* Composable 🡪 Agar komponen dapat dengan mudah dibuat, semua interaksi eksternal harus terjadi di interface publik.
* Deployable 🡪 Agar dapat di deploy, komponen harus bisa mengontrol bagian komponen sendiri.
* Documented 🡪 Komponen harus di dokumentasi menyeluruh agar calon pengguna bisa memilih apakah iya atau tidak komponen dapat memenuhi kebutuhan mereka.

Model komponen adalah definisi standar untuk implementasi komponen, dokumentasi, dan penyebaran. Berikut ini adalah ilustrasi sederhana model komponen:

[](https://4.bp.blogspot.com/-qcZCLi8LEiI/WAfcZIy5azI/AAAAAAAAAKo/phrF0IZ8TpgFOD0HqOgUJ0tH38TtYqk5ACEw/s1600/component%2Bmodel.PNG)

Gambar 2. Ilustrasi model komponen

Servis yang disediakan oleh komponen implementasi model, terdapat dua kategori:

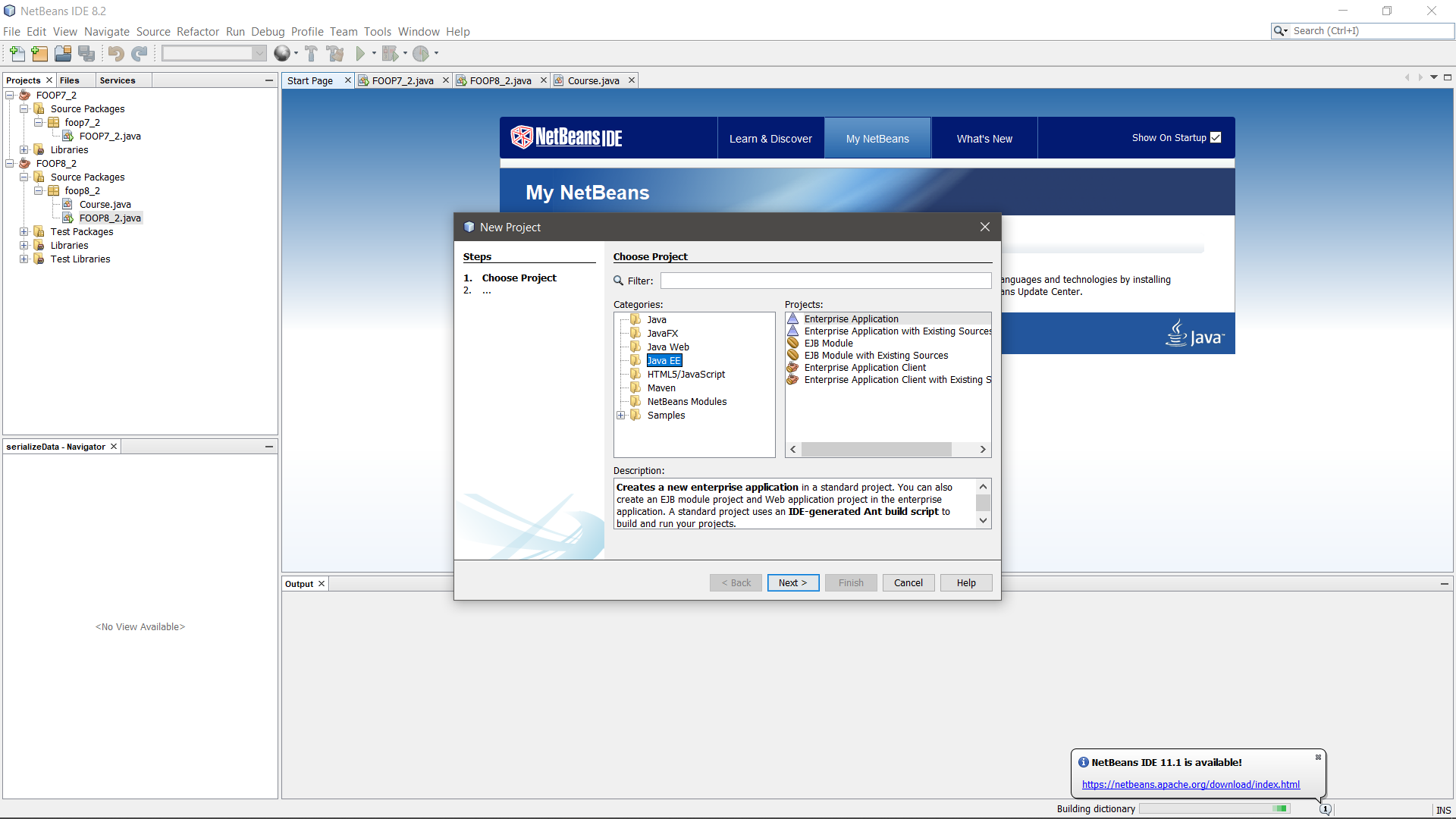
* Platfrom Service, yaitu untuk mengaktifkan komponen untuk  berkomunikasi .
* Support Service adalah layanan mainstream yang mungkin saja memerlukan berbagai macam komponen.

*(Sumber: http://proyeksoftware.blogspot.com/2016/10/component-ased-software-engineering.html)*

**2.4 Langkah Kerja**

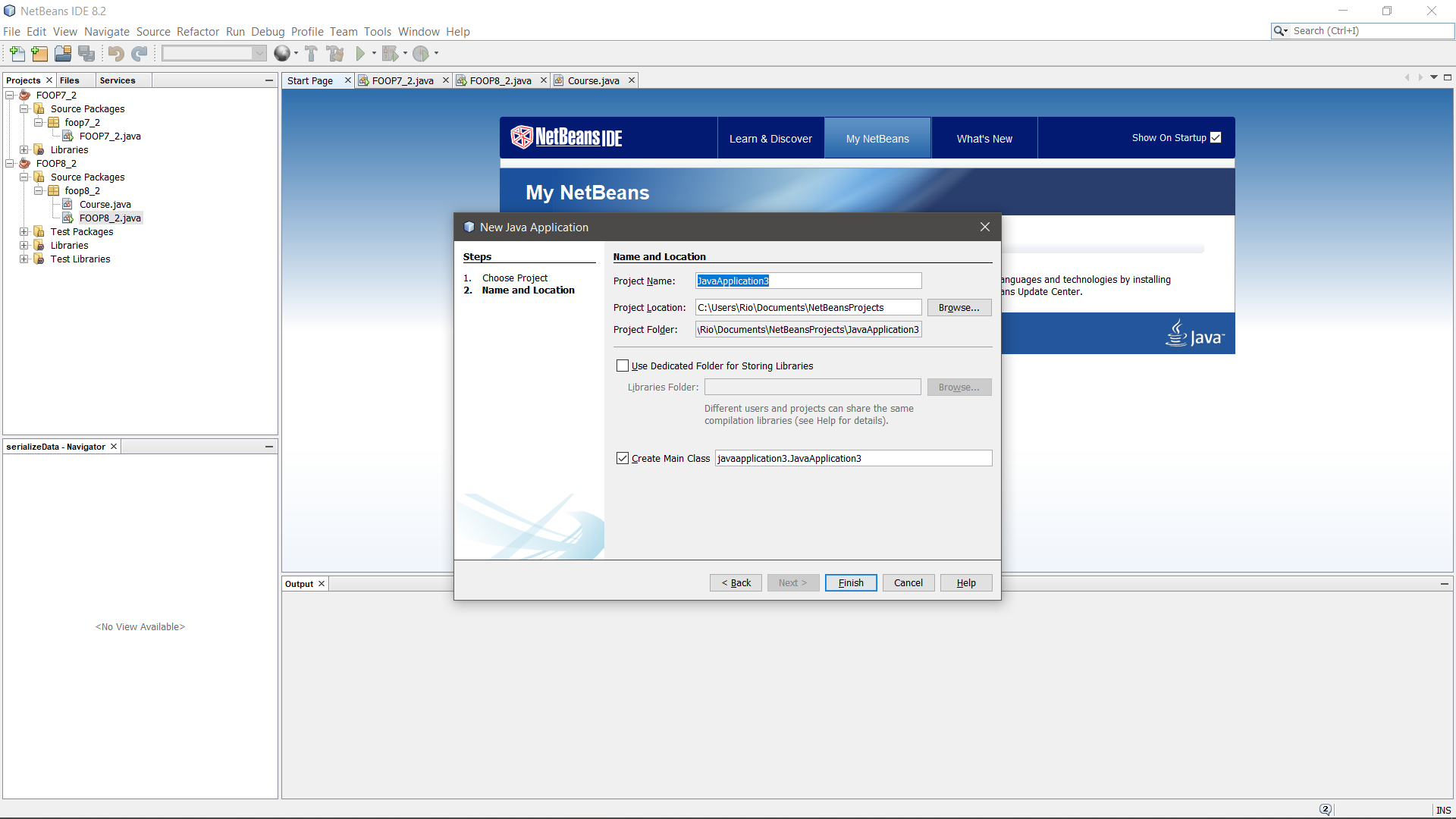
**2.4.1. Netbeans**

1. Buatlah project baru : File 🡪 New Project 🡪 Java 🡪 Java application



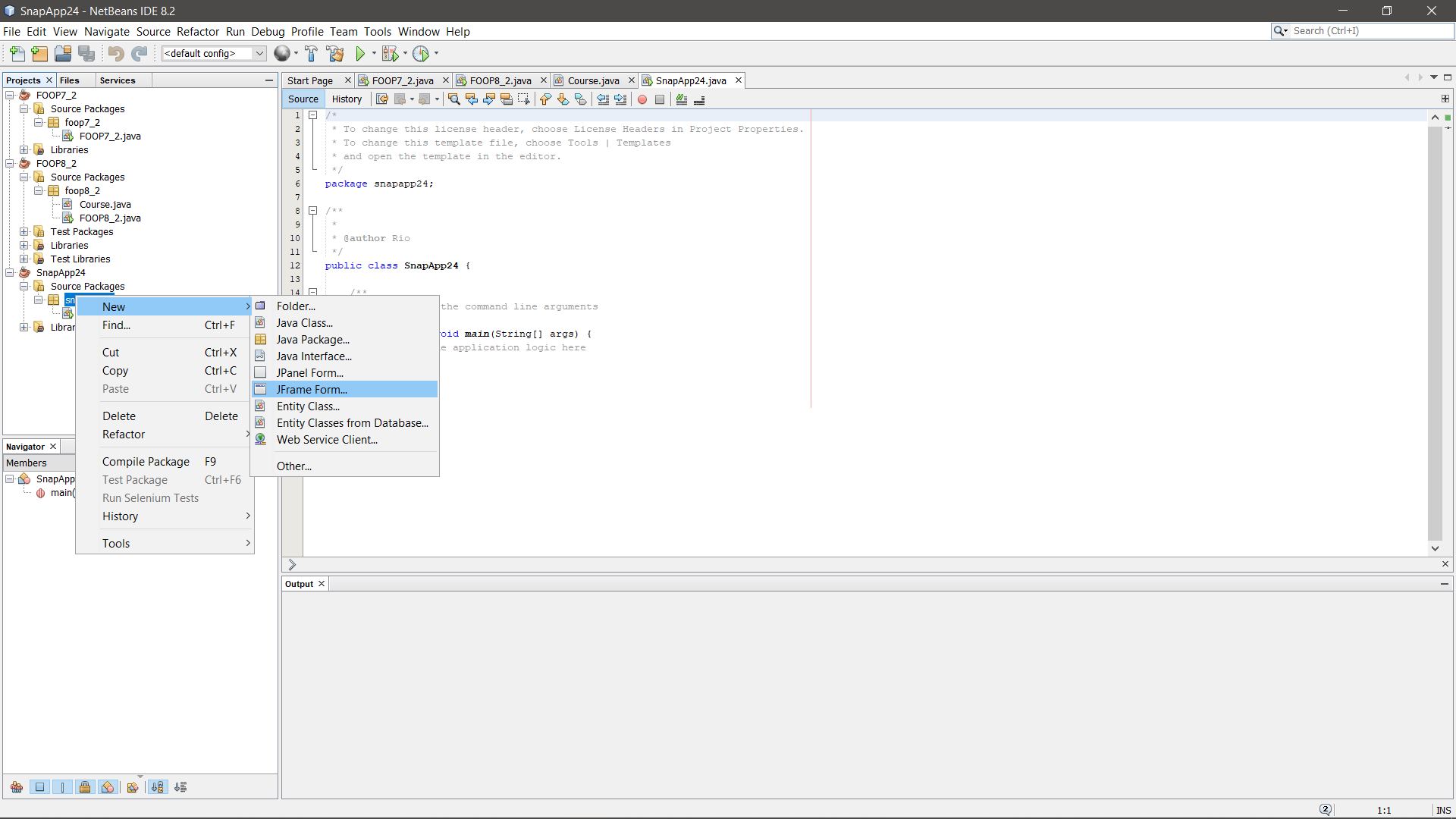
Gambar 2. Buka Project Baru

1. Beri nama SnapApp22



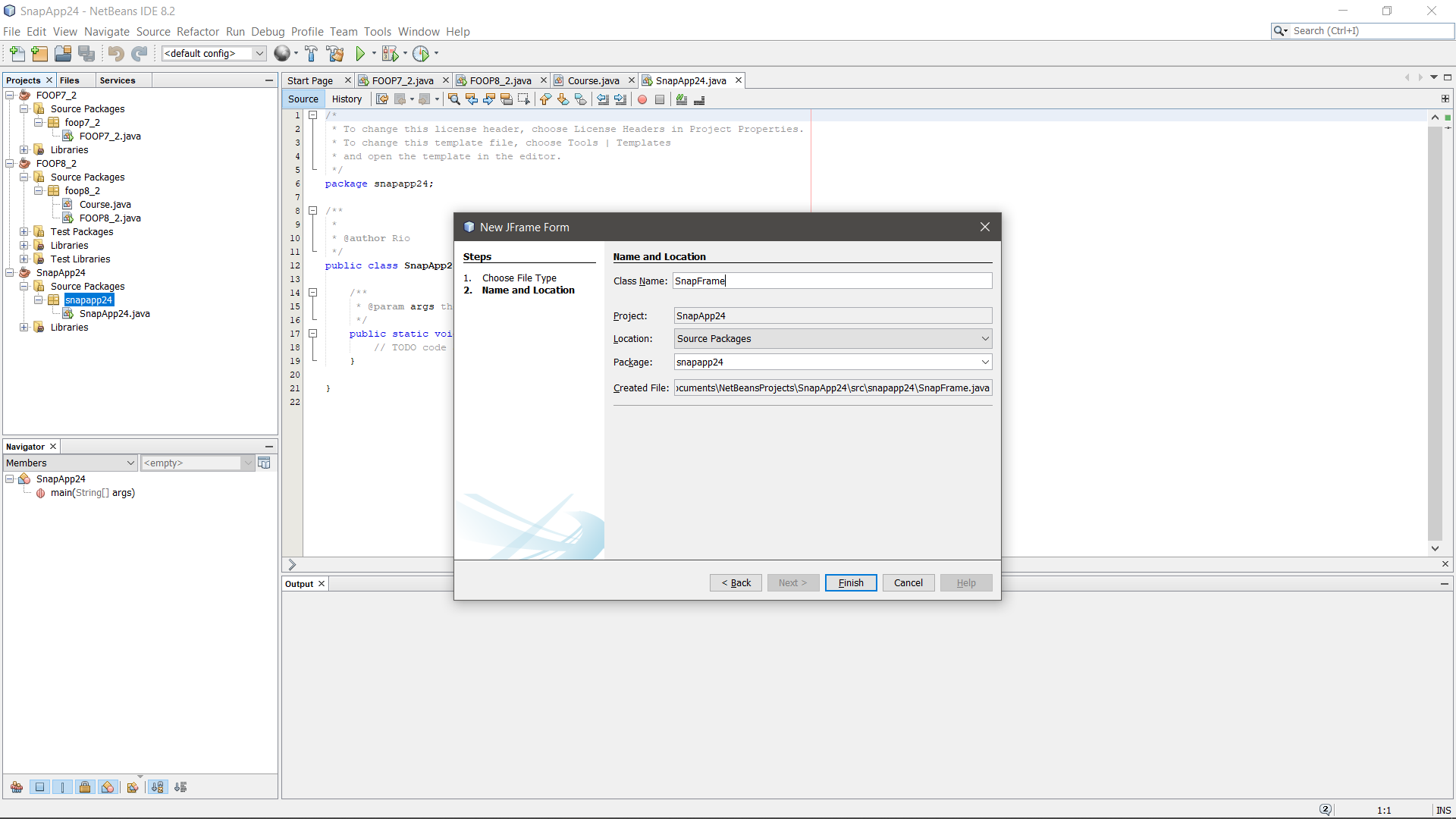
Gambar 2. Beri nama

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



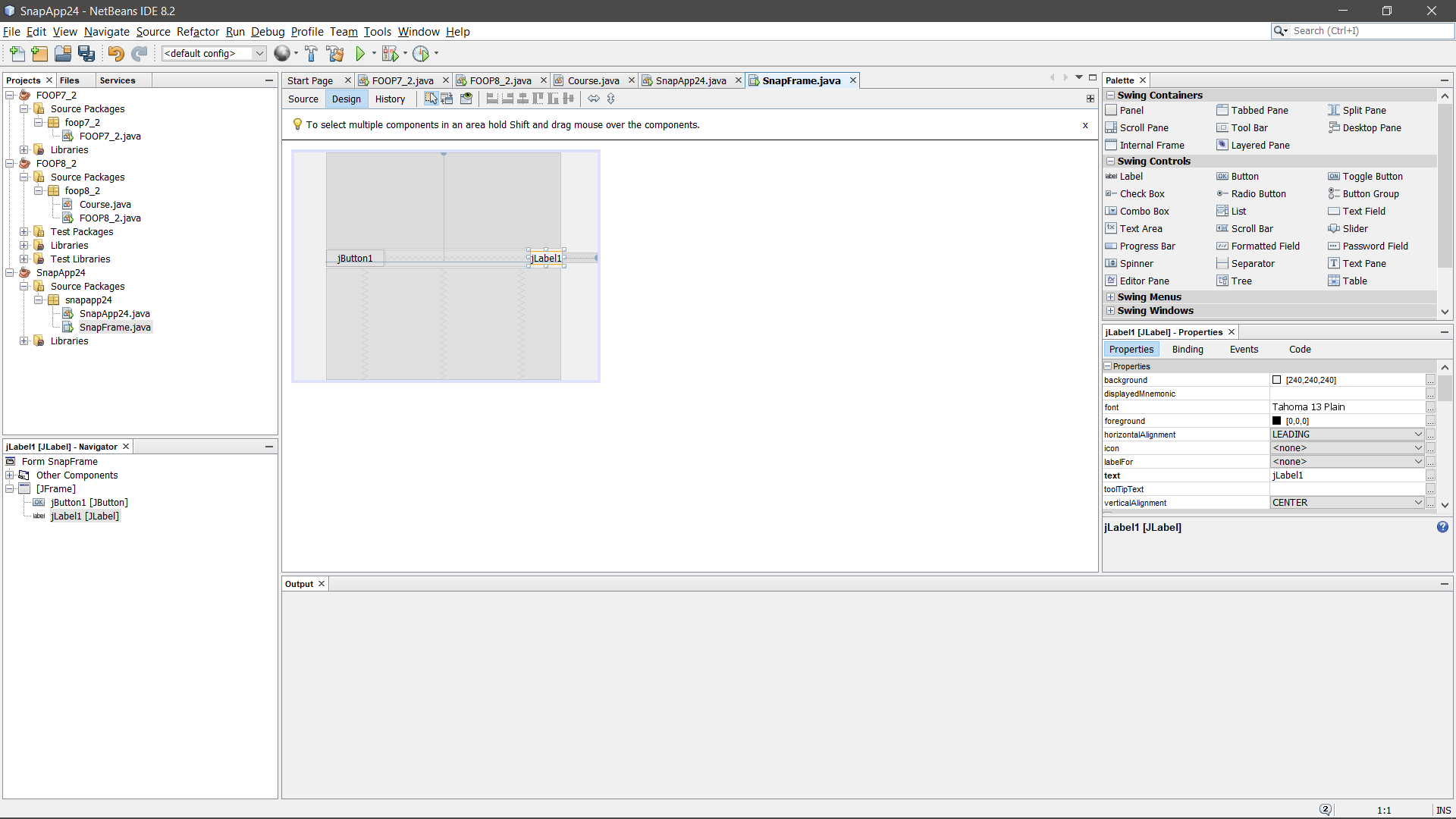
Gambar 2. Pilih JFrameForm

1. Beri nama SnapFrame



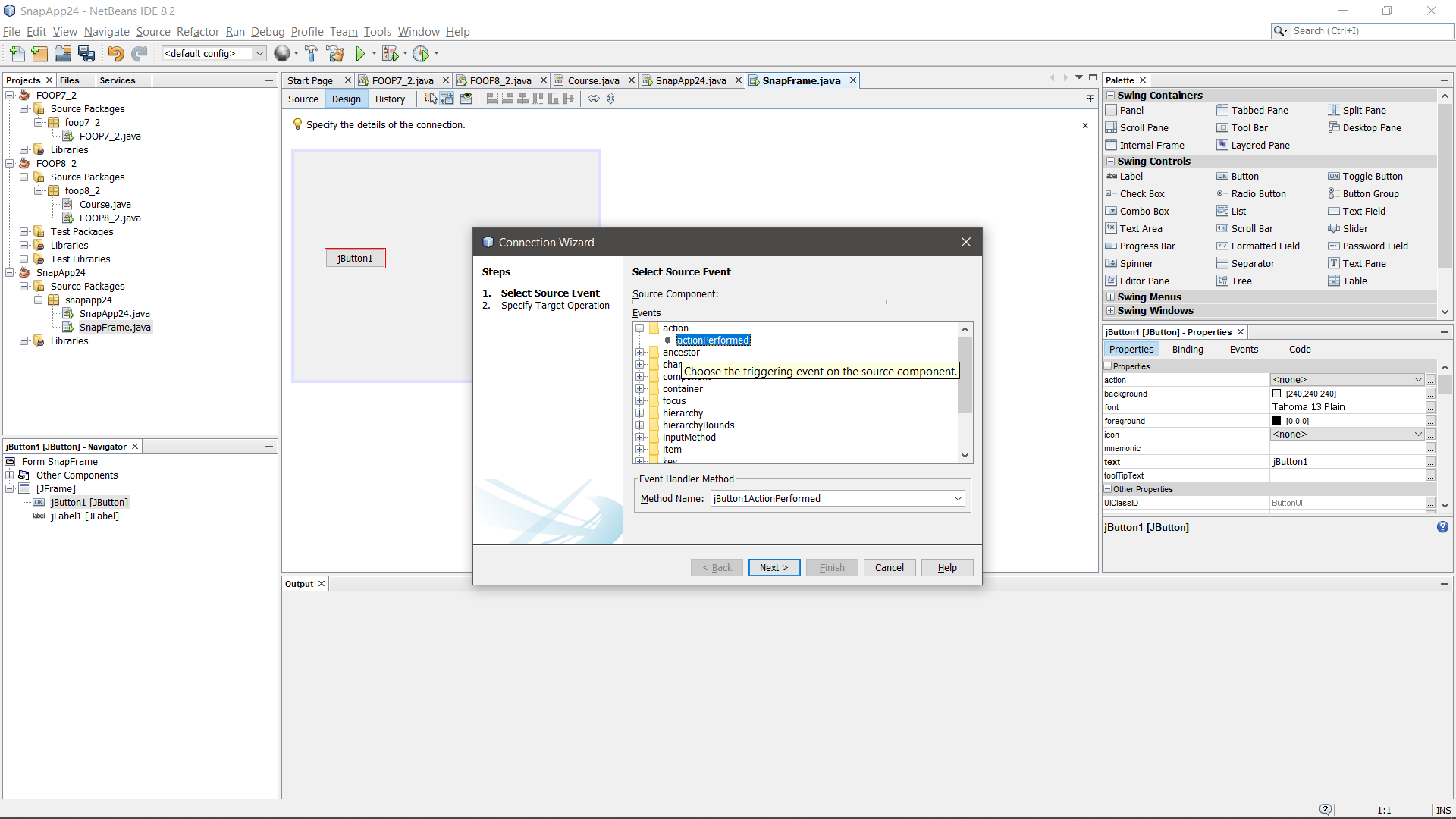
Gambar 2. Beri nama

1. Buat tampilan seperti berikut pada JFrame Form



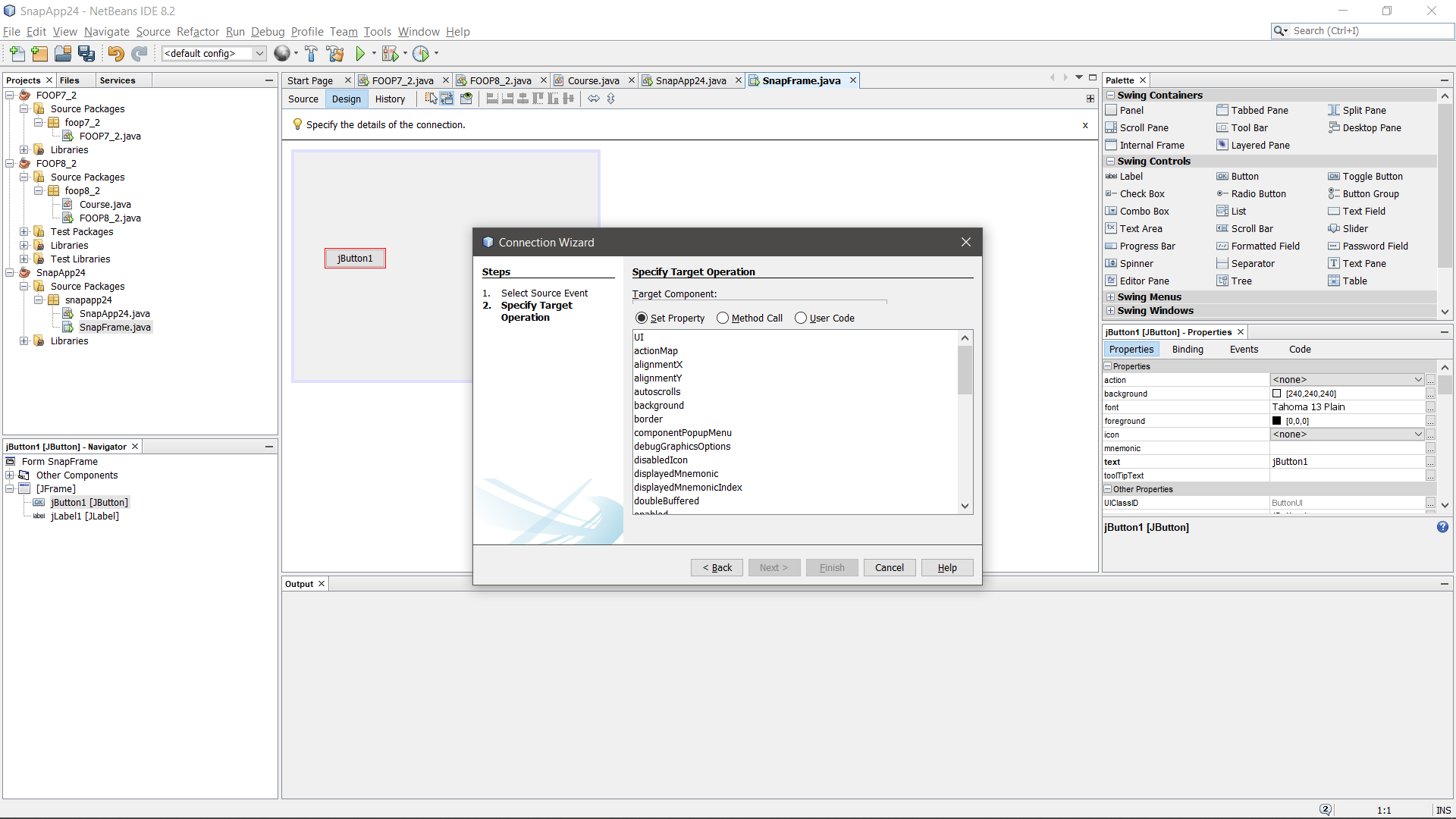
Gambar 2. Tampilan JFrame

1. Klik Connection Mode, kemudian klik pada tombol, kemudian klik pada label. Connection Mode digunakan untuk memberikan nilai pada label ketika button diberi action.
2. Setelah itu akan muncul connection Wizard, pilih action 🡪 actionPerformed. Langkah ini akan membuat button ketika diberi action akan menjalankan events actionPerformed.



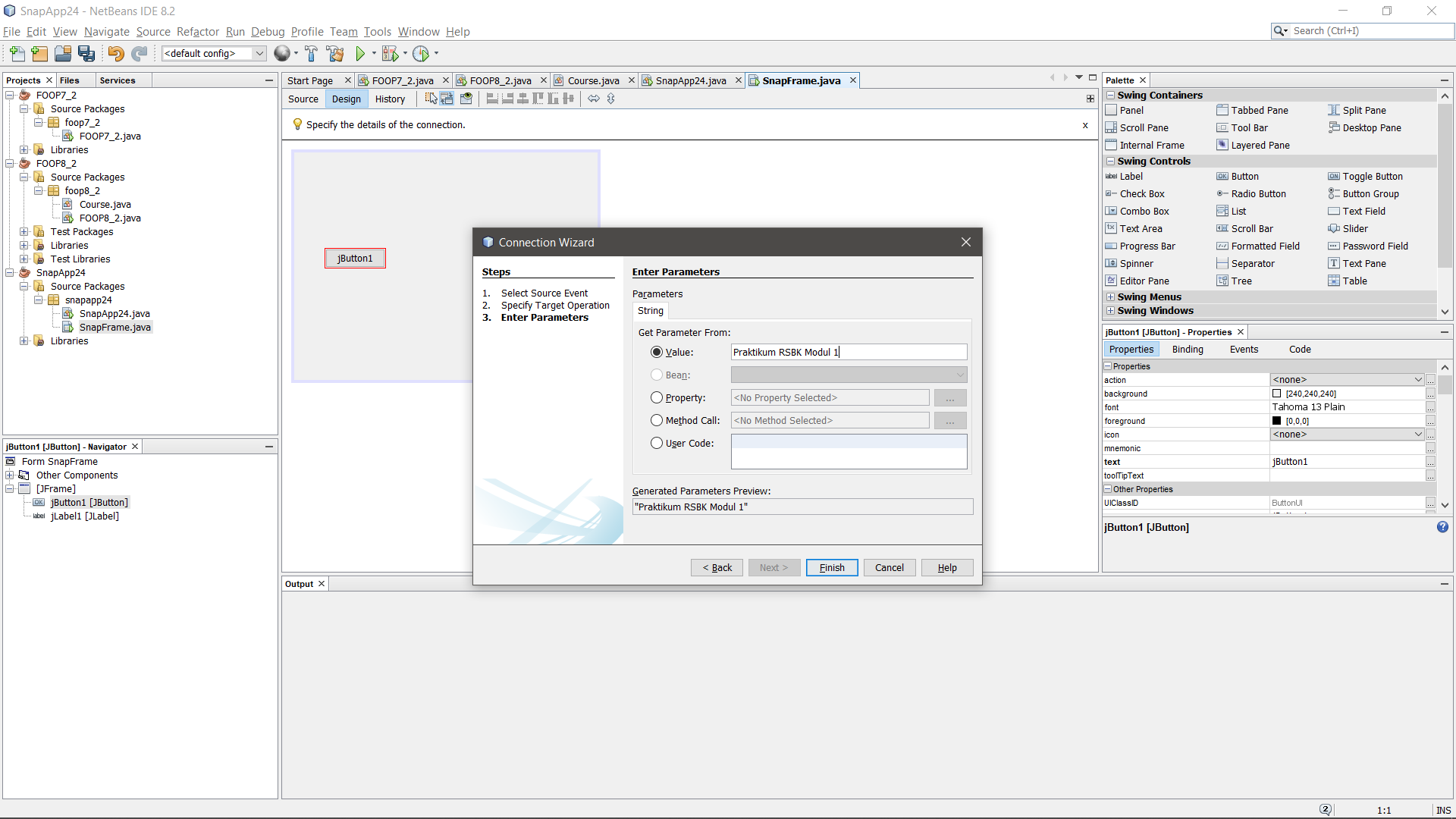
Gambar 2. Pilih *actionPerformed*

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



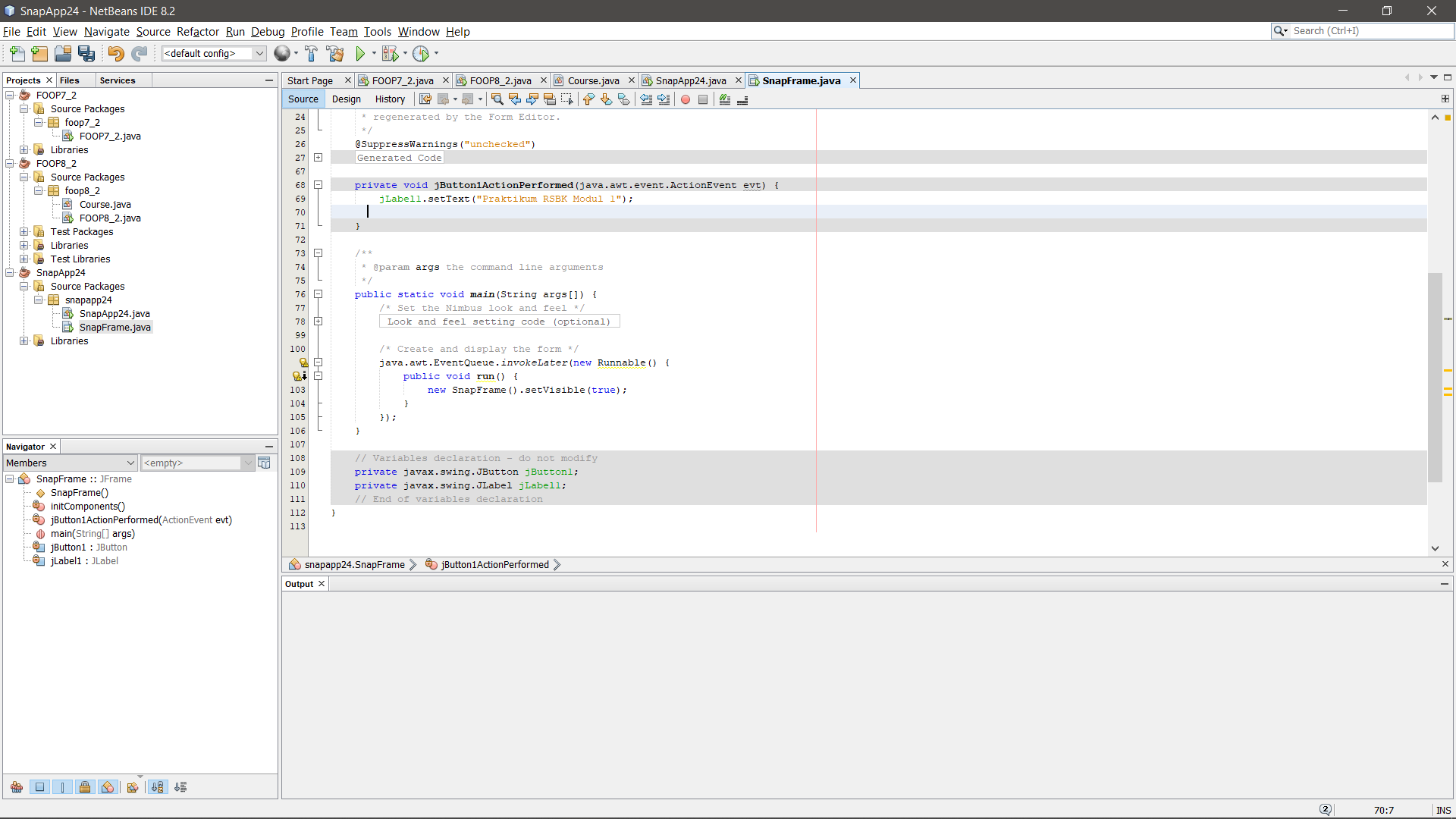
Gambar 2. Set Property

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



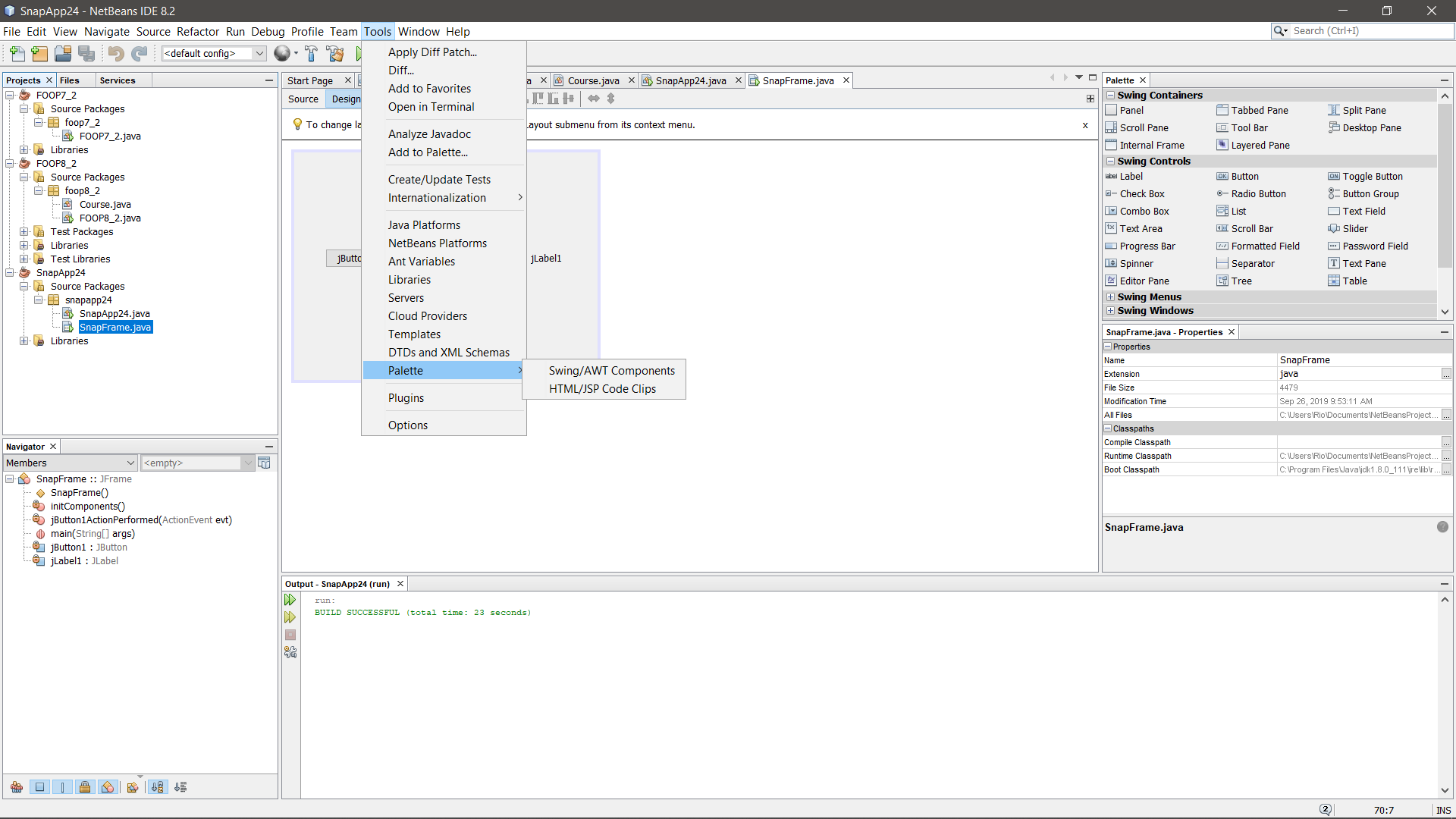
Gambar 2. Isi *Value*

1. Secara otomatis value akan muncul seperti source code berikut



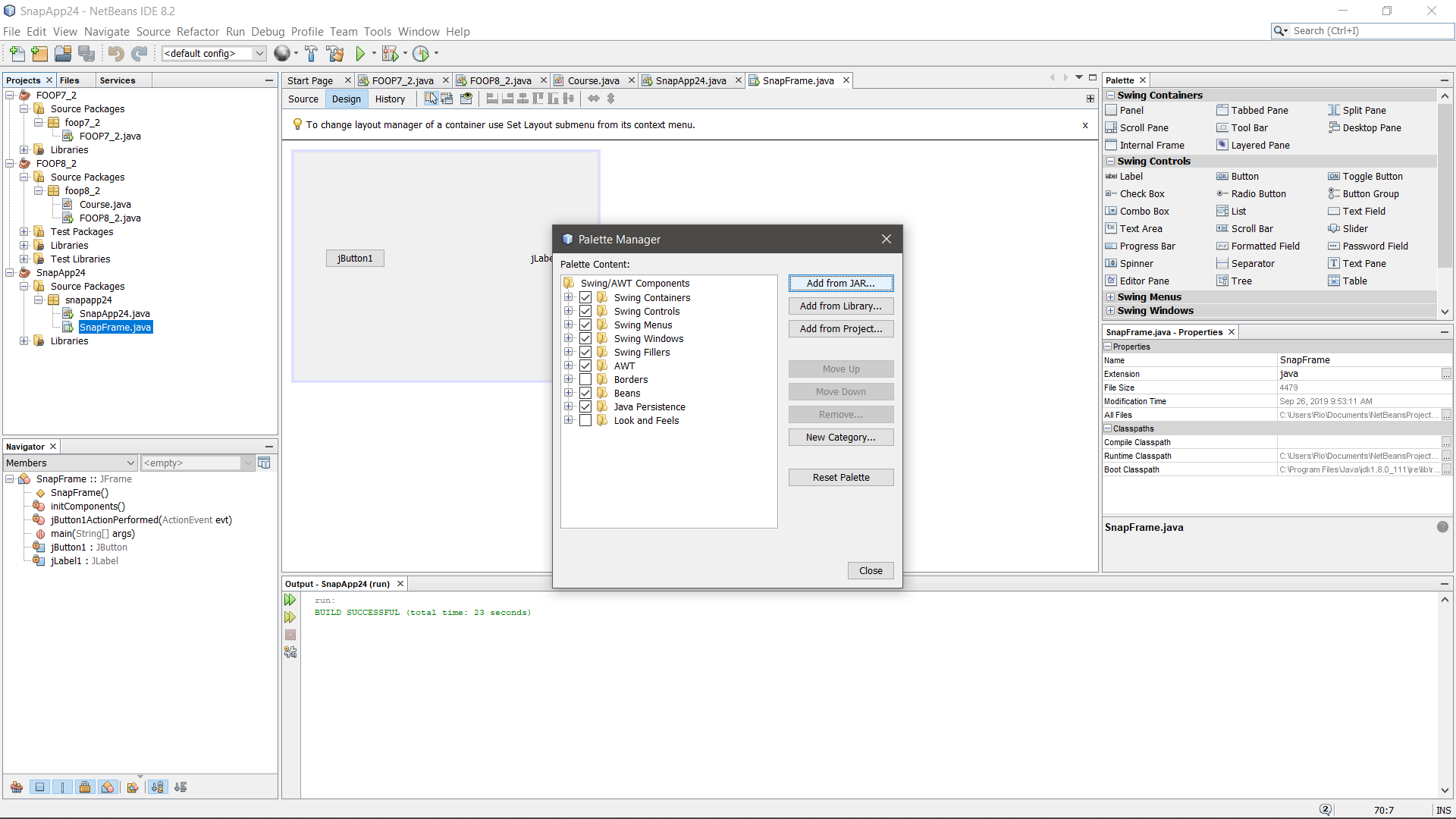
Gambar 2. Tampilan *Sourcecode*

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



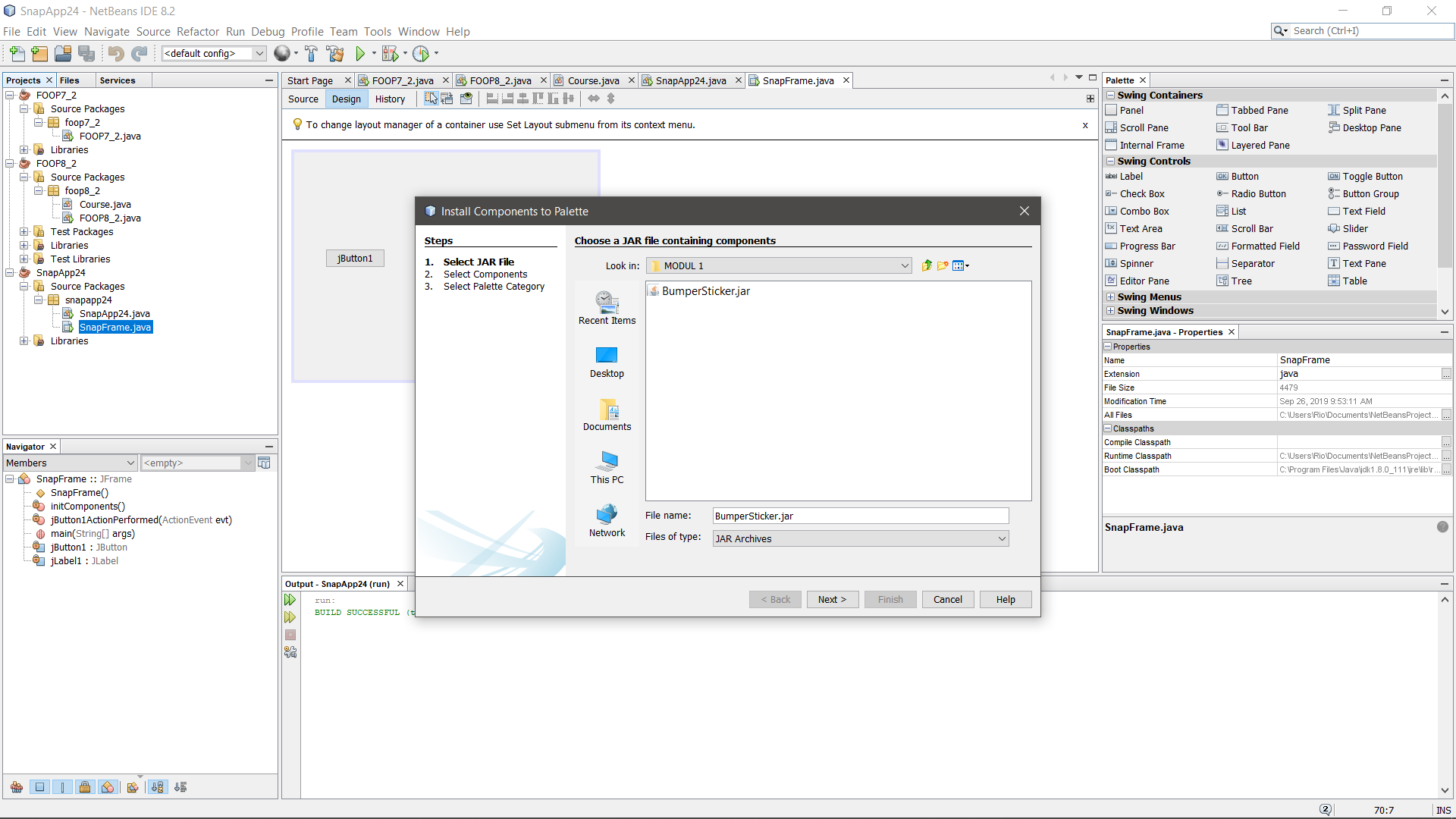
Gambar 2. Import Beans baru

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



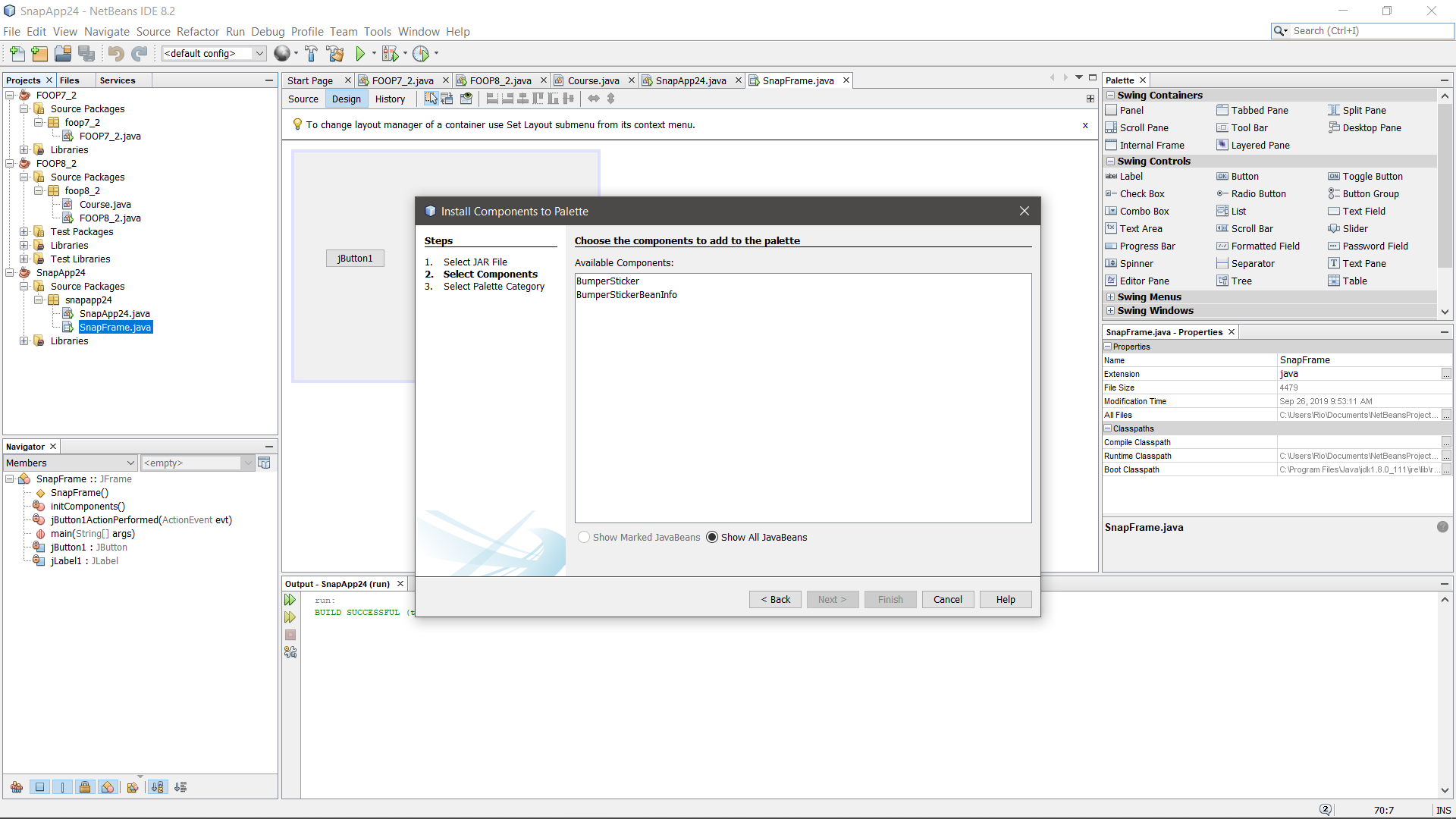
Gambar 2. Menu *Add From Jar*

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



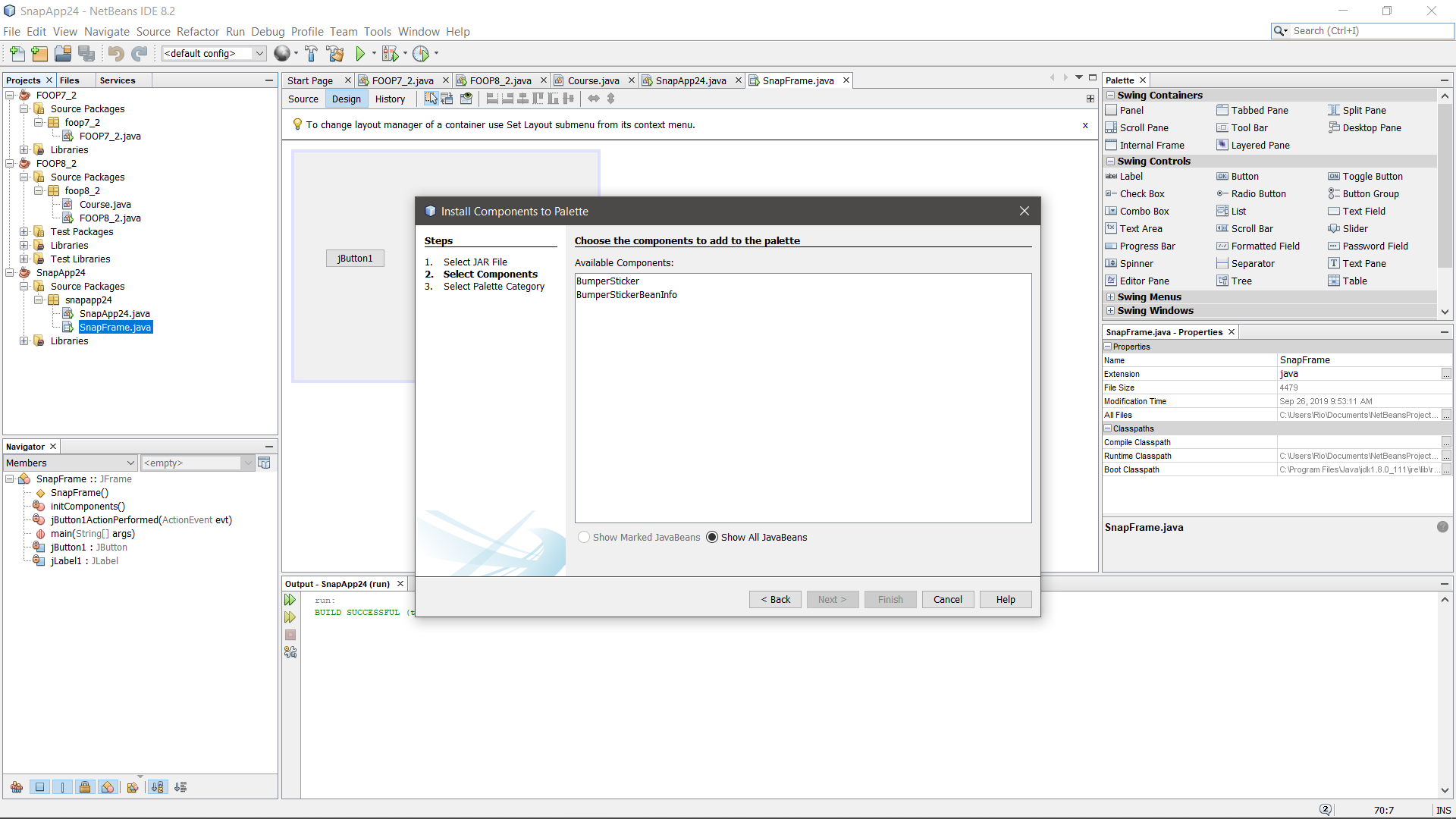
Gambar 2. Cari File BumperSticker

1. Pilih bumperSticker.



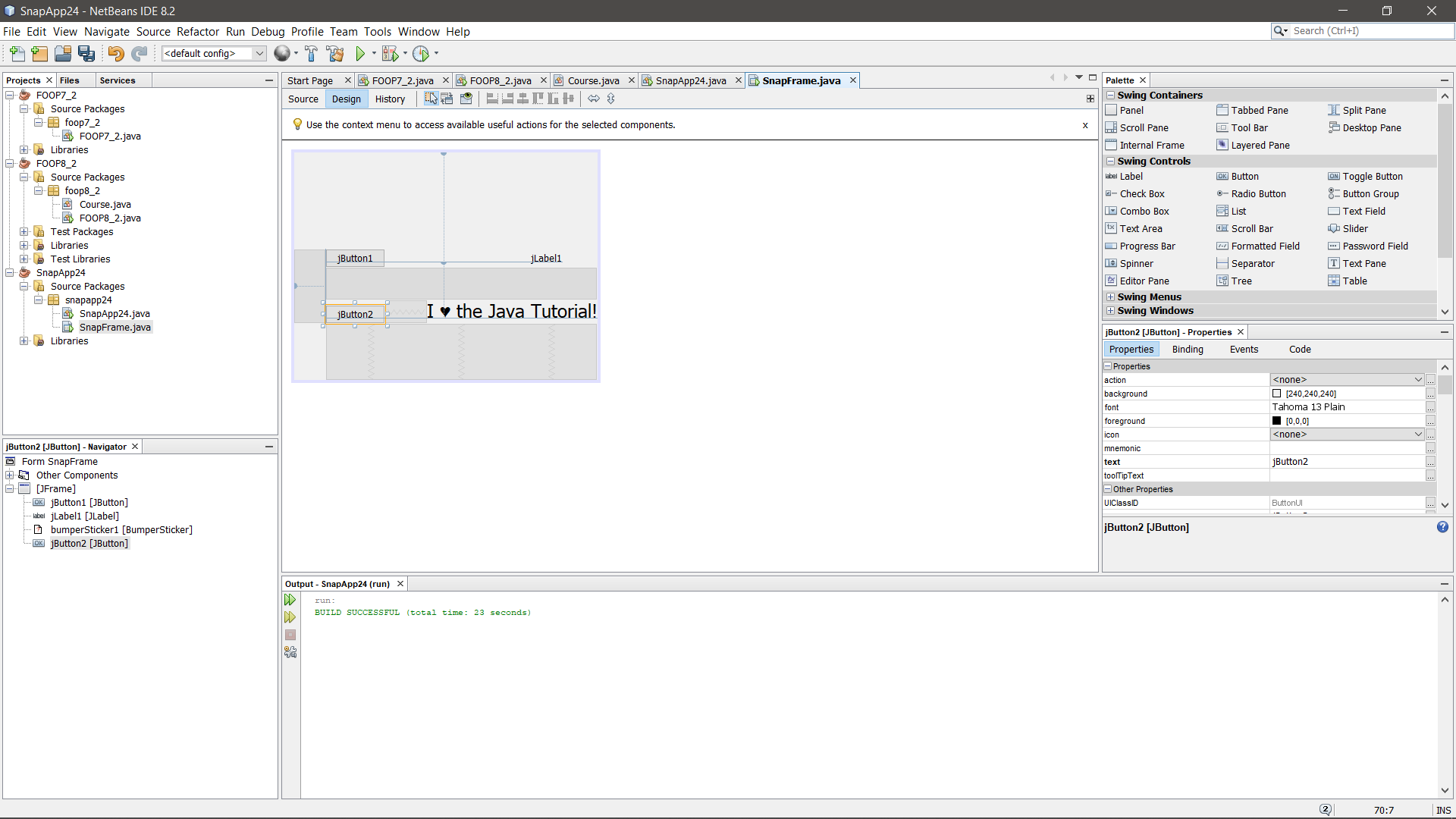
Gambar 2. Pilih BumperSticker

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



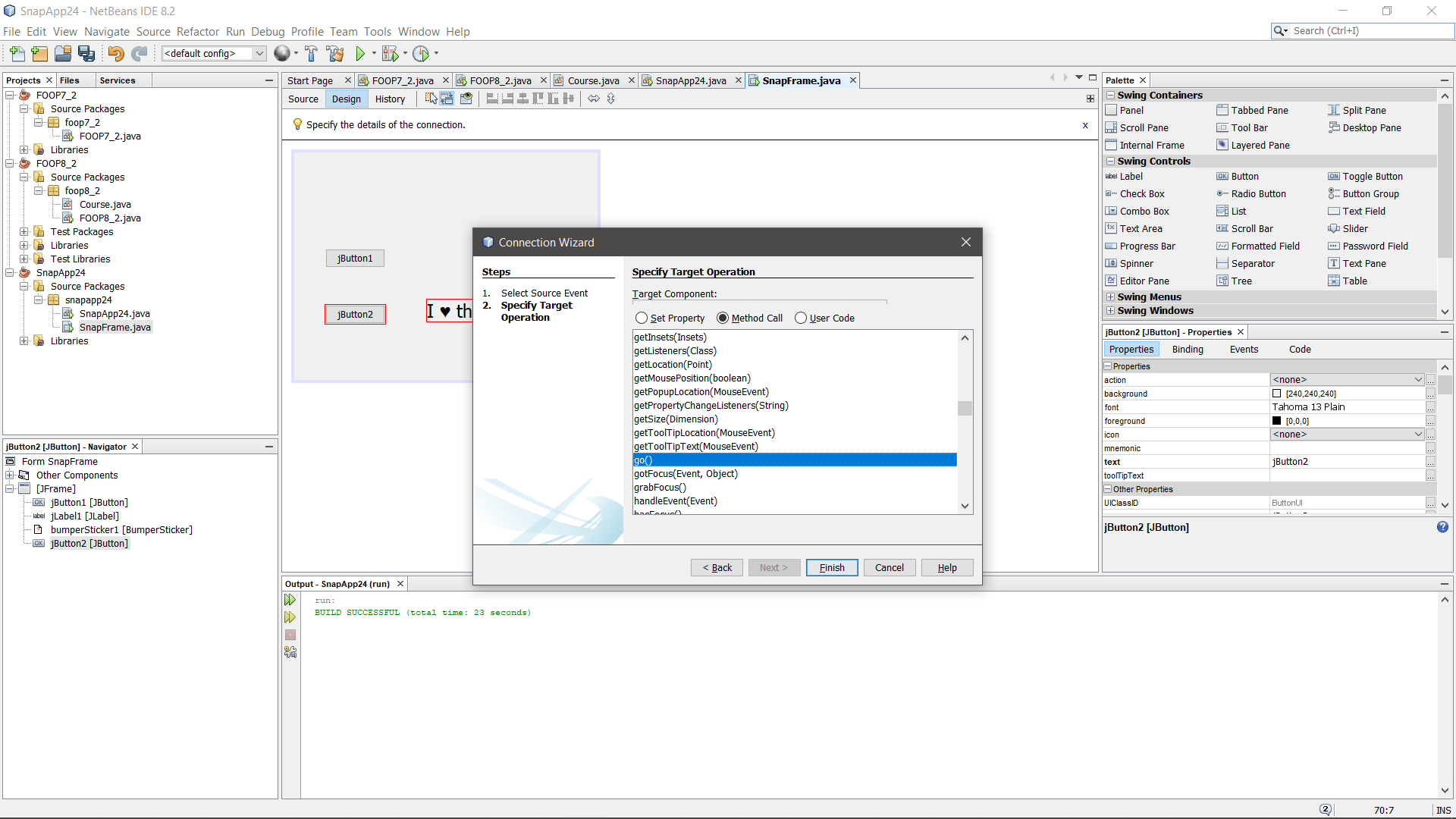
Gambar 2. Letakkan di Folder Beans

1. Klik close setelah selesai mengimport component.
2. Drag Beans 🡪 BumperSticker ke Form. Kemudian tambahkan satu button lagi untuk menjalankan animasinya.



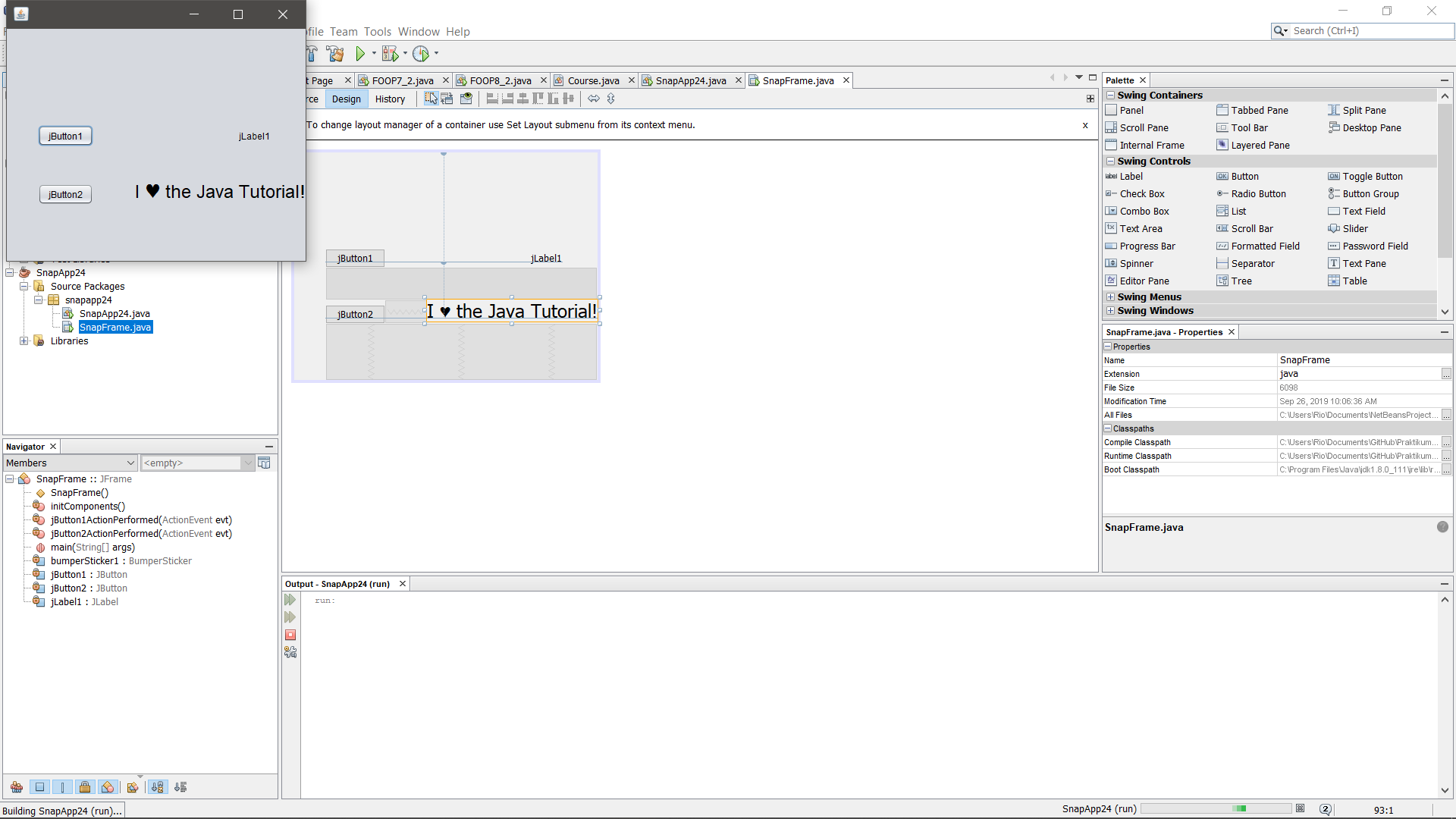
Gambar 2. Drag Beans dan tambah animasi

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

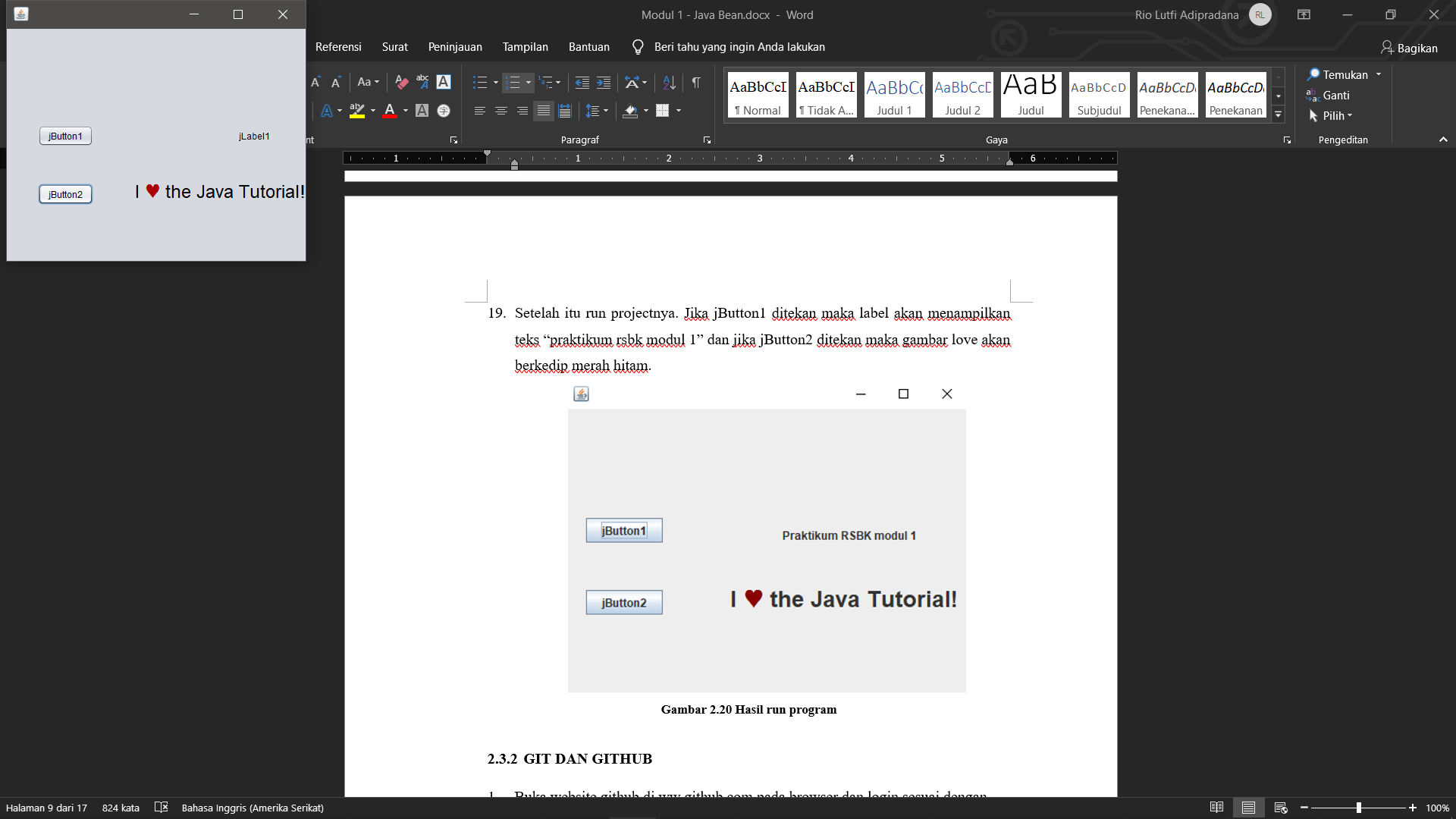


Gambar 2. Pilih Method Go()

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. Sebelum di klik



Gambar 2. Setelah di Klik

**2.4.2. GIT dan GitHub**

1. Buka website github di ww.github.com pada browser dan login sesuai dengan akun masing masing
2. Pilih tombol new yang terletak di sebelah kiri atas di dashboard masing masing akun untuk membuat repository pada akun github masing masing.
3. Beri nama repository tersebut dengan nama PraktikumRSBKkel22, kemudian klik create repository
4. 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.
5. Buka folder project java bean yang sudah dibuat tadi, kemudian klik kanan dan pilih opsi git bash untuk membuka console git.
6. 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

1. 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
3. 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.

Git add .

Git commit –m “comment”

Git push origin master

1. 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. Namun diawali dengan langkah add file. Langkahnya sebagai berikut :
2. Perintah Pull Digunakan untuk mengupdate repository local sehingga sama dengan repository yang ada pada github. Caranya cukup mudah yaitu dengan cara :

Git pull

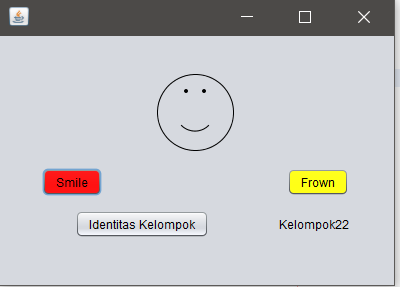
1. Terahir jika ingin mendownload / menggunakan repository orang lain yang ada di github cuku dengan copy link dari repository yang diinginkan bisa menggunakan Clone.

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]**

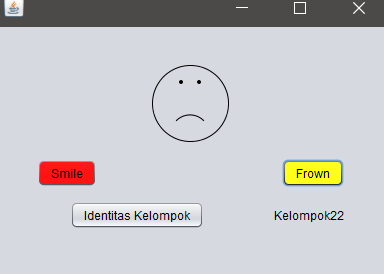
Directory dapat diubah tergantung dari dimana kalian mengklik kanan dan membuka bash, misal membuka gitbash di folder D:\contoh. Maka repository akan berada di D:\contoh\[nama repository]

**2.5 Tugas dan Pembahasan**

**2.5.1 Tugas 1**



Gambar 2. 25 Gambar Smile



Gambar 2. 26 Gambar Frown

Pada tugas ini, Praktikan diharuskan mengunduh sebuah *project* latihan dari netbeans sendiri, yang berjudul FaceBean. Didalam project tersebut terdapat komponen bernama SmileFace.

Yang pertama dilakukan adalah *clean and build* project FaceBean, sehingga akan muncul FaceBean.jar di dalam folder dist.

Setelah itu, import jar tersebut. Lalu, buat sebuah Frame baru, yang nantinya akan diisi tiga buah Button, dan beans yang baru saja di import.

Lalu, buatlah *Connection* Mode pada Button *Smile* dan *FaceBean*, lalu masukan *method* berjudul smile(). Lalu, lakkan hal yang sama pada Button *Frown*, dan masukan *method* berjudul frown().

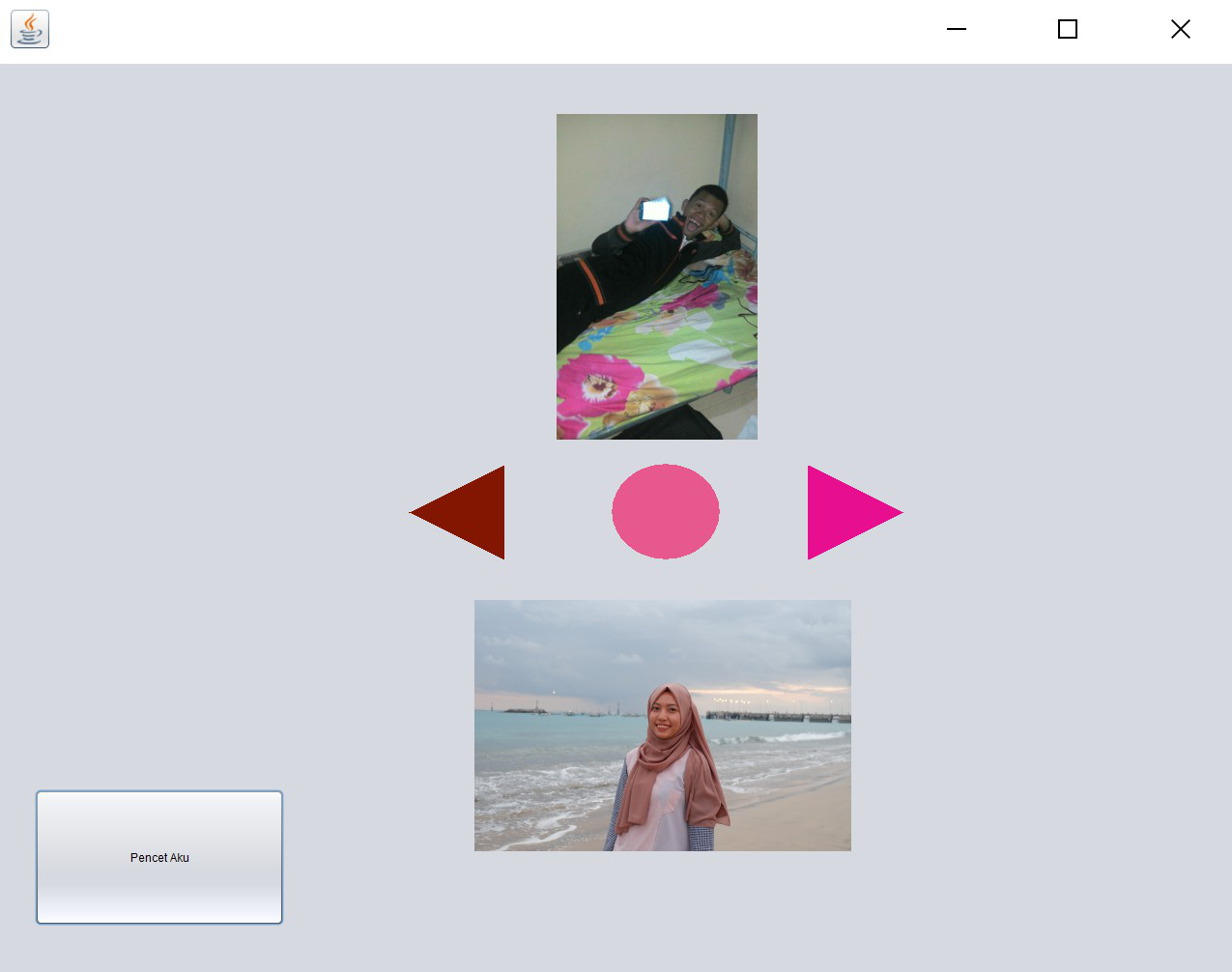
**2.5.2 Tugas 2**

Pada tugas 2, praktikan membuat sebuah komponen Lingkaran, Segitigia menghadap kiri, dan segitiga menghadap ke kanan. Komponen-komponen tersebut nantinya akan digunakan untuk membuat suatu program untuk memindahkan foto praktikan 1 dan praktikan 2.

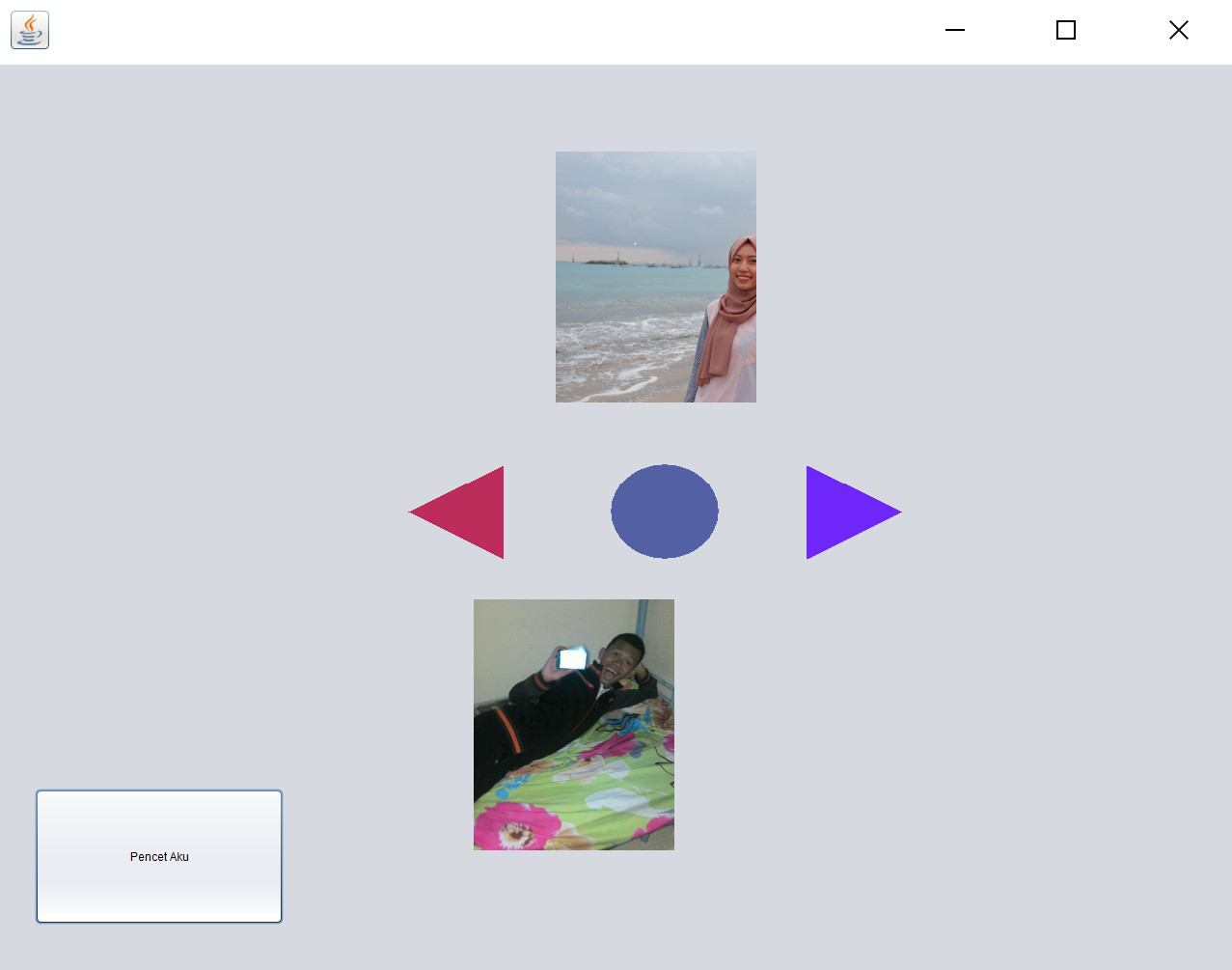
Lalu, file foto yang ingin di masukan kedalam program kita masukan ke directory dari program tersebut, sehingga nantinya dapat di masukan ke *coding*.

Selanjutnya, kita membuat suatu Button, yang nantinya akan mengubah letak foto praktikan 1 dan foto praktikan 2 itu, dan mengubah warna pada ketiga komponen yang telah dibuat, dengan method change().

Foto praktikan 1 dan 2 akan diletakkan kesebuah label, sehingga kita hanya perlu mengubah posisi label yang berisi foto tersebut, untuk mengubah letak foto.



Gambar 2. 27 Tampilan form saat awal



Gambar 2. 28 Tampilan form setelah button di-klik

**2.6 Kesimpulan**

1. Dalam membuat Java Bean, penting untuk mengecek apakah *library* yang akan digunakan sudah sesuai atau tidak, karena jika tidak maka komponen akan *error* dan tidak dapat digunakan.
2. Untuk menghubungkan antara komponen satu dan lainnya perlu menggunakan *Connection Mode*.
3. File Java Bean akan memiliki ekstensi .jar
4. *Clone repository* digunakan untuk mengunduh *project* yang ada pada GitHub ke *local storage*, sehingga nantinya dapat diedit sesuai dengan keinginan sendiri.
5. *Fork* digunakan untuk pemilik *repository* melakukan kolaborasi dengan pengguna GitHub lainnya.

Link Repo: <https://github.com/ryolu69/PraktikumRSBKkel22>