



Pemrograman Berorientasi Objek

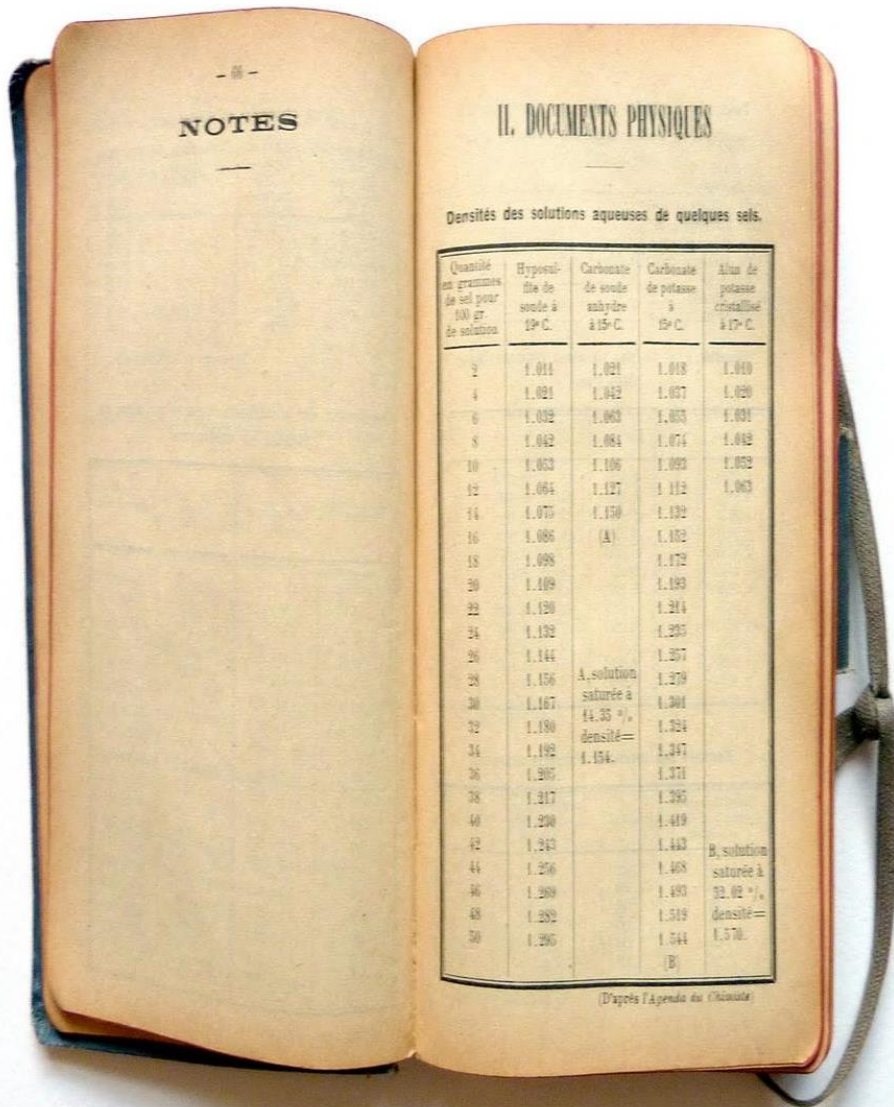


Pertemuan 12

Aplikasi GUI dengan JAVA (1)

Pemateri : Chrismikha Hardyanto S.Kom., M.Kom.

KONTEN PERKULIAHAN



- **Pengenalan konsep GUI**
- **Membuat Aplikasi GUI dengan JAVA**
- **Konsep Dasar Container & Component pada GUI**
- **Menata Component didalam Container**
- **Menambahkan Aksi Pada Component**

Setelah sebelumnya kita belajar **Konsep Dasar** dari **Paradigma PBO**, selanjutnya kita akan mencoba memahami bagaimana **membuat aplikasi JAVA sederhana**

Konsep Dasar Pembuatan Program

Ketika membuat sebuah program, umumnya seorang programmer perlu memikirkan **3 hal dasar**, yaitu :



[JAVA Application]

- ☐ **Antarmuka Pengguna
[Tampilan Program]**
- ☐ **Logika Program
[Perilaku Program]**
- ☐ **Kebutuhan DATA
[Aliran Data Didalam Program]**

Pengantar Antarmuka Pengguna

Dalam pembuatan program ada 2 metode tampilan antarmuka yang umum digunakan yaitu berbasis **CLI** dan **GUI**

- ❑ **CLI (Command Line Interface)** merupakan suatu konsep antarmuka pengguna dengan cara **mengetikkan suatu perintah** berbentuk **text** pada sebuah console untuk **menjalankan suatu tugas** tertentu didalam aplikasi.
- ❑ **GUI (Graphical User Interface)** merupakan suatu konsep antarmuka pengguna di dunia IT yang memanfaatkan **objek – objek grafis** sebagai media **interaksi** dan **komunikasi** antara sistem dengan penggunanya



Kenapa Pakai GUI ?


Untuk melakukan aksi tertentu pada program, pengguna cukup **berinteraksi** dengan salah satu **object grafis** didalam program, maka program sudah dapat **memahami aksi apa yang diminta pengguna**


Sehingga tampilan program yang **berbasis GUI** lebih mudah untuk **digunakan** dan **dipahami** oleh **orang awam** yang baru menggunakan program tersebut


Contoh Aplikasi GUI seperti apa ??





Contoh Aplikasi GUI


**SISTEM INFORMASI
AKADEMIK - UNIKOM**





**Chrismikha Hardyanto** ?


 Profile


 Perwalian <

 Aktivitas Dosen v


 **Perkuliahan**

 Penelitian

 PKM & Penunjang


 Dokumen Perkuliahan


 Monitoring BAP









Jadwal & Absensi Perkuliahan

Melihat jadwal perkuliahan dan melakukan pencatatan kehadiran mahasiswa

 Semester Aktif : **20212 (2021/2022 Genap)**

 **Download Lembar Nilai**

| No | Hari & Jam | Ruangan | Kode & Nama Mata Kuliah | SKS Penjadwalan | Ruangan | Nama Kelas | Jenis Perkuliahan | |
|----|-----------------------|---------|--|--------------------|---------|-------------------------------|----------------------|--|
| 1. | Belum diseting | | 01057 - Kerja Praktek (2 SKS) Lihat RPS Mata Kuliah | | | KP_Ulang_Chrysmikha 316096 | Bimbingan |  Pengumuman |
| 2. | Senin, 07.00-09.30 | LAB-5 | 01070 - Pemrograman Berorientasi Objek (3 SKS) Lihat RPS Mata Kuliah | 3 | LAB-5 | IF-5/S1/II 314188 | Tatap Muka |  Lihat Absensi  Absensi Dari LMS  Pengumuman |
| 3. | Senin, 09.30-11.10 | LAB-5 | 01126 - Pemrograman Visual / GUI (2 SKS) Lihat RPS Mata Kuliah | 2 | LAB-5 | GUI UL1 314314 | Tatap Muka |  Lihat Absensi  Absensi Dari LMS |

Platform Program Visual/GUI

Program berbasis visual / GUI dapat dibuat di berbagai platform, contohnya :



Program Berbasis
Desktop



Program Berbasis
Web



Program Berbasis
Mobile

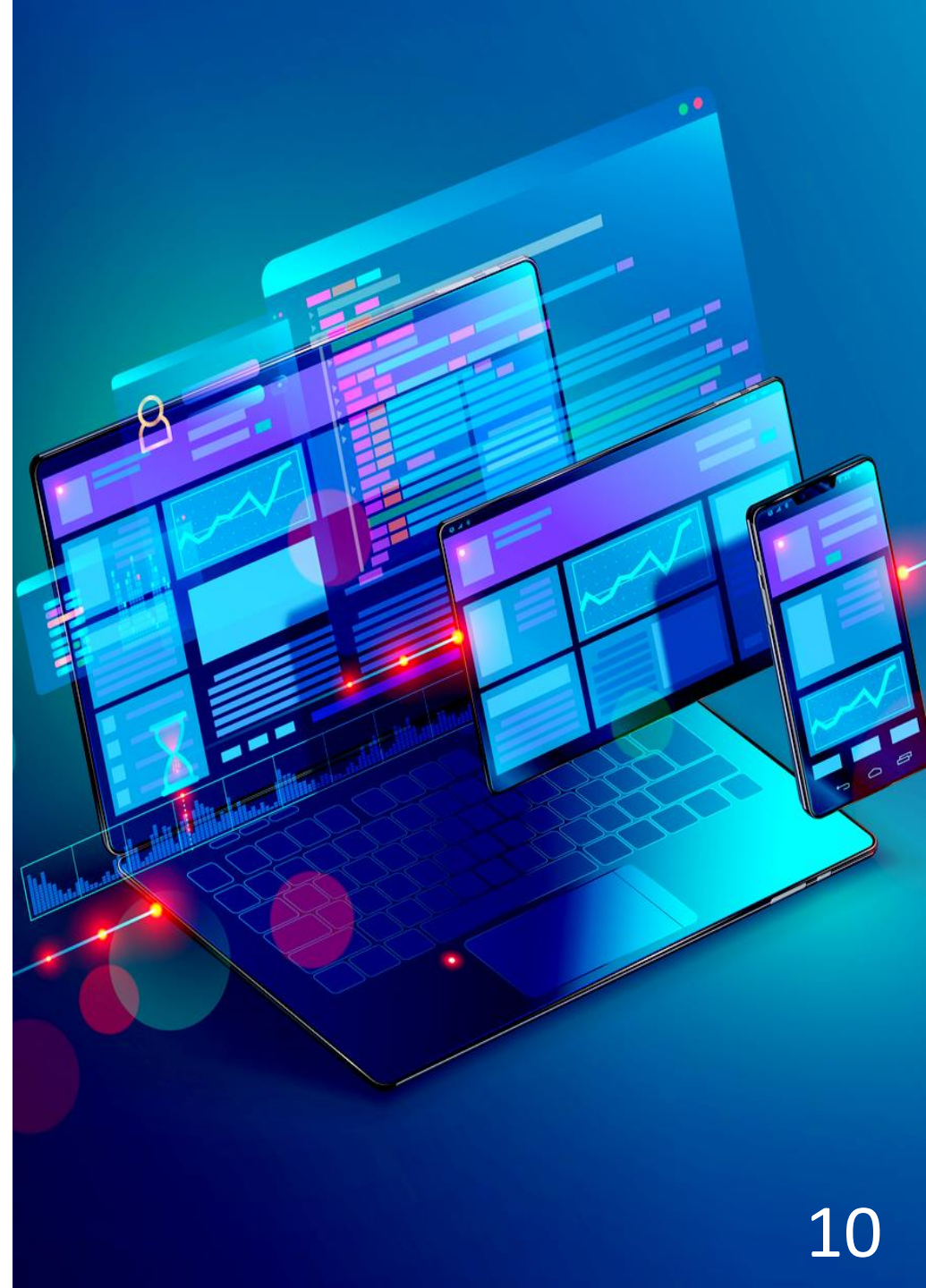
Bagaimana Membuat Aplikasi Berbasis GUI di JAVA ?

Kita dapat **memanfaatkan API** yang disediakan oleh JAVA untuk mendvelop tampilan program berbasis GUI , salah satunya adalah **JAVA Swing**



Pengenalan JAVA Swing

- ❑ Swing merupakan bagian dari **Java Foundation Class (JFC)**, yaitu **kumpulan class-class Java** yang digunakan untuk mengembangkan perangkat lunak **berbasis GUI (Graphical User Interface)**.
- ❑ Package yang digunakan adalah : **javax.swing.***
Java 6.0 menyertakan setidaknya **tujuh belas (17)** buah package yang berisi **class-class swing** yang siap digunakan.
- ❑ GUI dibangun menggunakan berbagai **component** yang disediakan oleh **Swing (dalam bentuk Class)** dimana Component tersebut dapat digunakan untuk berinteraksi dengan user

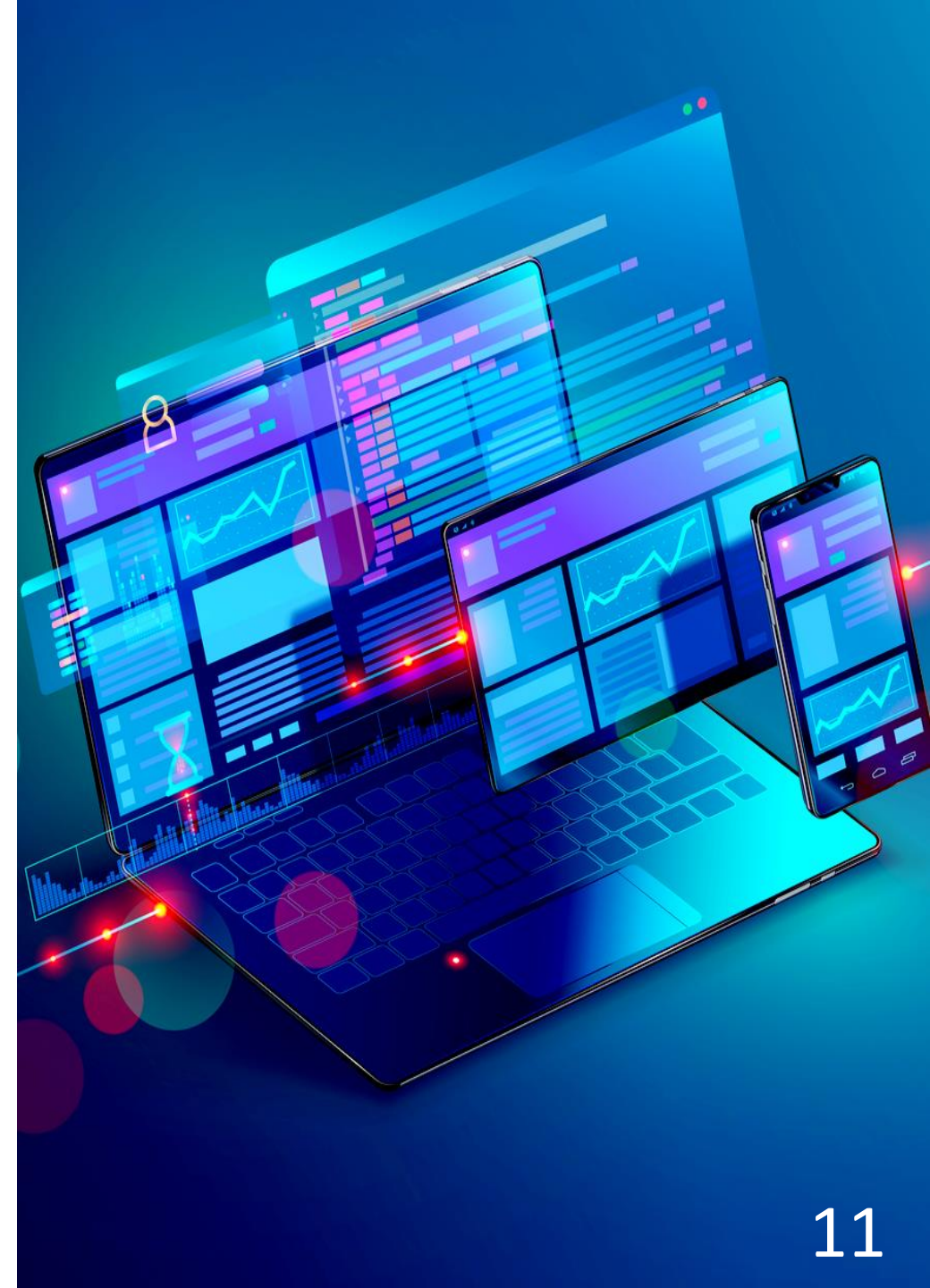


Pengenalan JAVA Swing

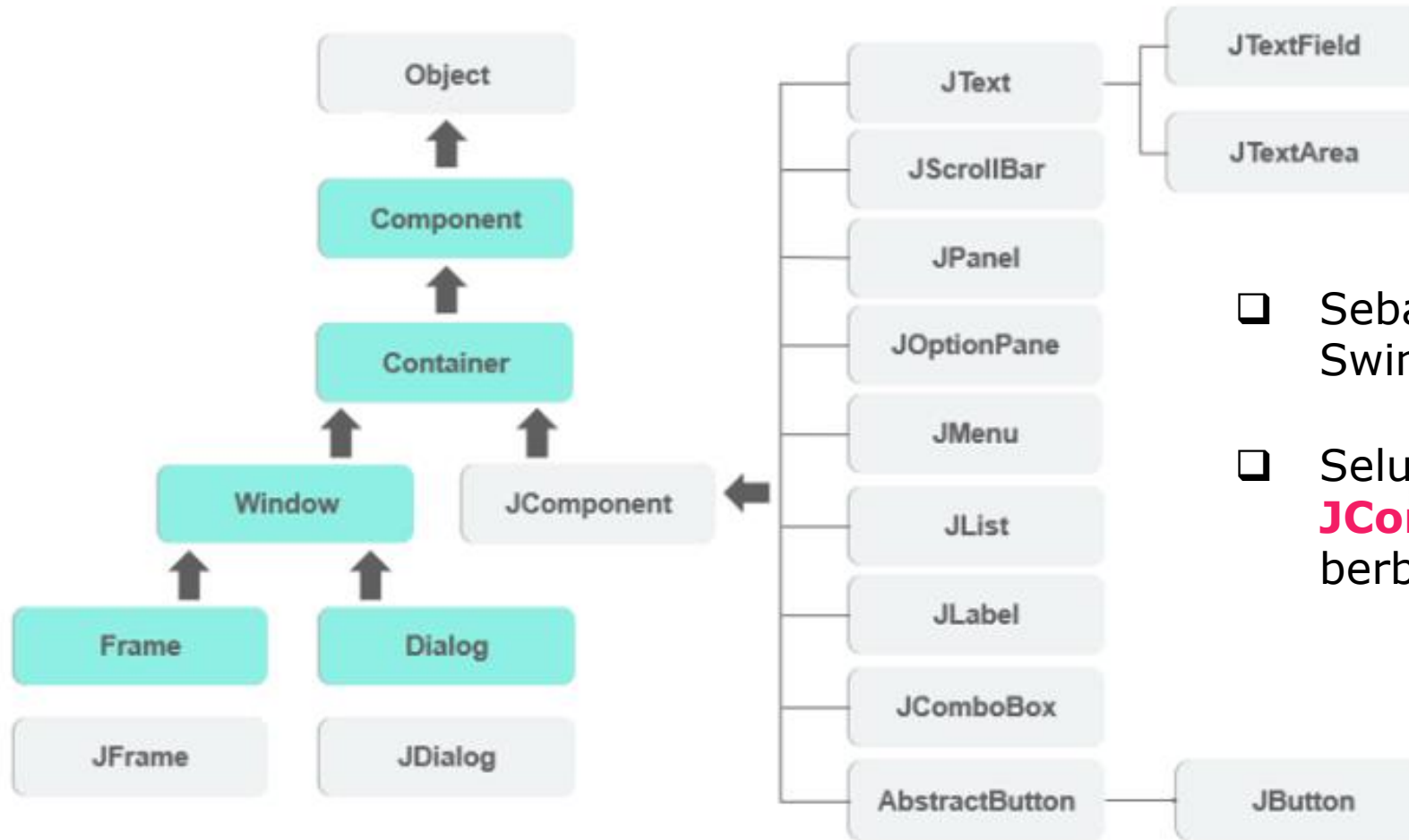
- ❑ Contoh package Swing pada JAVA

| | | |
|--------------------------|------------------------|-----------------------|
| javax.accessibility | javax.swing.plaf | javax.swing.text |
| javax.swing | javax.swing.plaf.basic | javax.swing.text.html |
| javax.swing.border | javax.swing.plaf.metal | javax.swing.text.rtf |
| javax.swing.colorchooser | javax.swing.plaf.multi | javax.swing.table |
| javax.swing.event | javax.swing.plaf.synth | javax.swing.tree |
| javax.swing.filechooser | | javax.swing.undo |

- ❑ kita tidak akan menggunakan semua class-class dalam package swing didalam pembuatan program, hanya sebagian kecil saja dari class-class tersebut yang nantinya akan benar-benar kita gunakan.
- ❑ Penjelasan detail tiap class dpt dibaca di Javadoc:
<https://docs.oracle.com/javase/8/docs/api/index.html?javax/swing/package-summary.html>



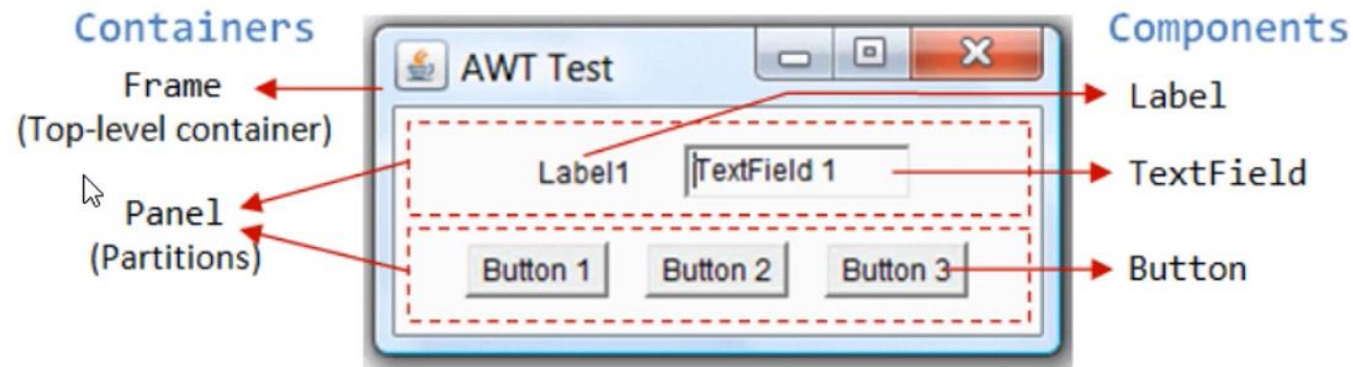
Hirarki Component Swing



- ❑ Sebagian besar nama class dalam Swing diawali dengan '**J**'
- ❑ Seluruhnya merupakan **subclass** dari **JComponent**, yang menyediakan berbagai **method turunan**.

Penempatan Component Swing

Untuk mendesain aplikasi kita wajib mengetahui 2 jenis elemen GUI **Container** dan **Component**



Component: Components adalah elemen dasar dari entitas GUI seperti Button, Label, dan TextField.

Container: Containers, seperti Frame dan Panel, digunakan untuk meletakkan komponen di dalam sebuah layout (seperti FlowLayout, GridLayout atau yang lainnya). Sebuah container bisa juga menampung sebuah sub-containers (konsep container di dalam container lain).

Mari kita coba praktek...

Bagaimana cara membuat **tampilan GUI** pada JAVA ??

**Mari kita coba membuat FORM Sederhana
pada JAVA yang menerapkan GUI**



Langkah – Langkah Membuat Tampilan GUI

Berikut adalah **guideline** dari langkah yang akan kita lalui dalam praktikum membuat tampilan aplikasi berbasis GUI di JAVA

1. **Membuat project & package Baru**
2. **Membuat Container Dasar (Class JFrame)**
3. **Membuat Container Menengah (Class JPanel)**
4. **Menambahkan Component Untuk interaksi Pengguna**
5. **Menambahkan Aksi pada Component**
6. **Testing Program**



Ada 2 cara untuk membuat Tampilan GUI dengan Swing pada JAVA :

A) Hard Coding (Seluruh Coding Manual)

B) Soft Coding (Memanfaatkan GUI Builder Netbeans)

Langkah 1A:

Membuat Project & Package Baru

Praktikum Java Swing (1A)

- ❑ Buatlah project baru pada netbeans. dengan cara : **File -> New Project-> Java Application -> Tulis nama Project -> unchecklist main class - > finish**

New Java Application

Steps

1. Choose Project
2. **Name and Location**

Name and Location

Project Name:

Project Location:

Project Folder:

☐ Use Dedicated Folder for Storing Libraries

Libraries Folder:

Different users and projects can share the same compilation libraries (see Help for details).

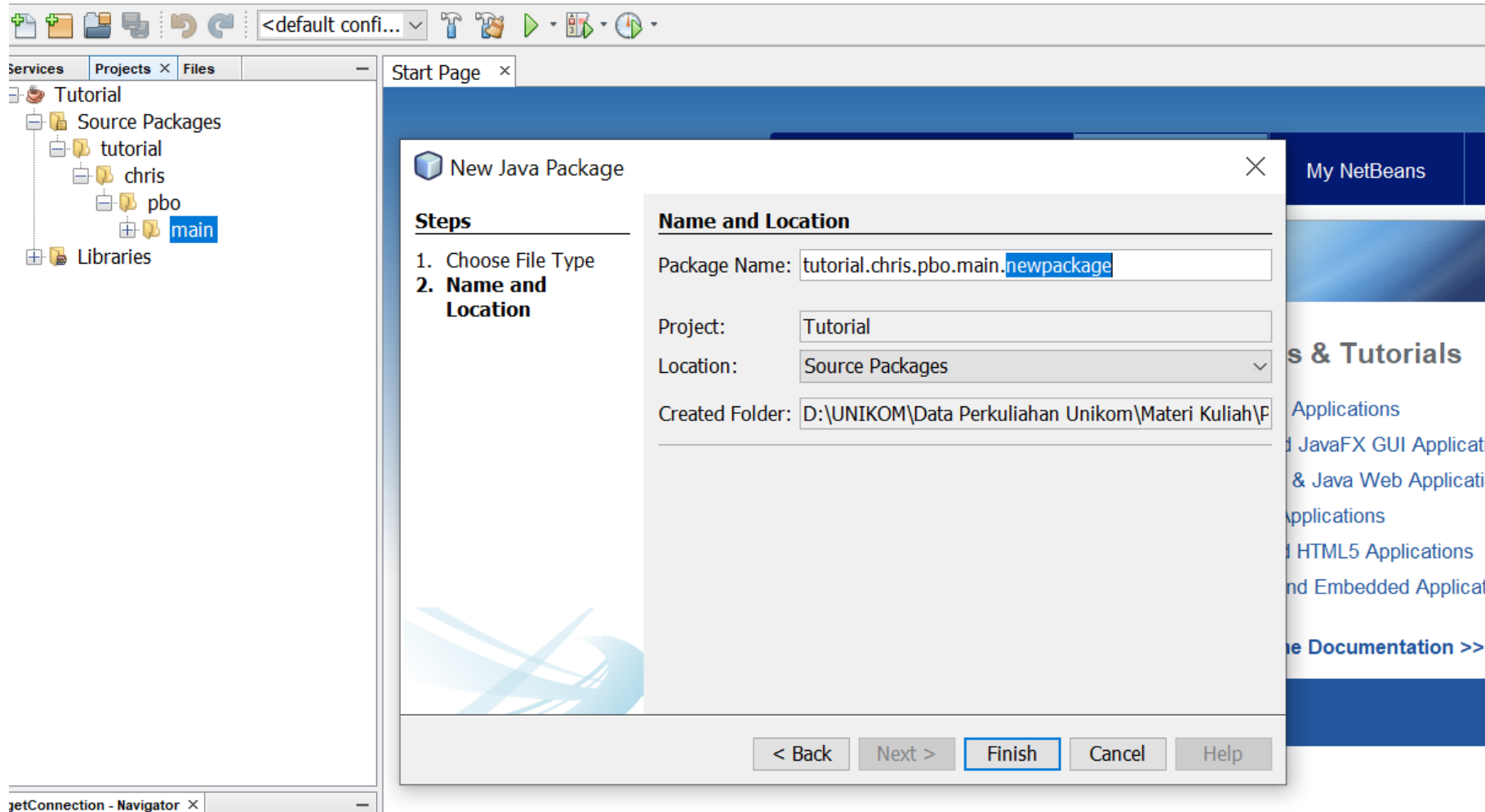
☐ **Create Main Class**

< Back Next > **Finish** Cancel Help

***Nama Project Sesuaikan Kebutuhan Praktek**

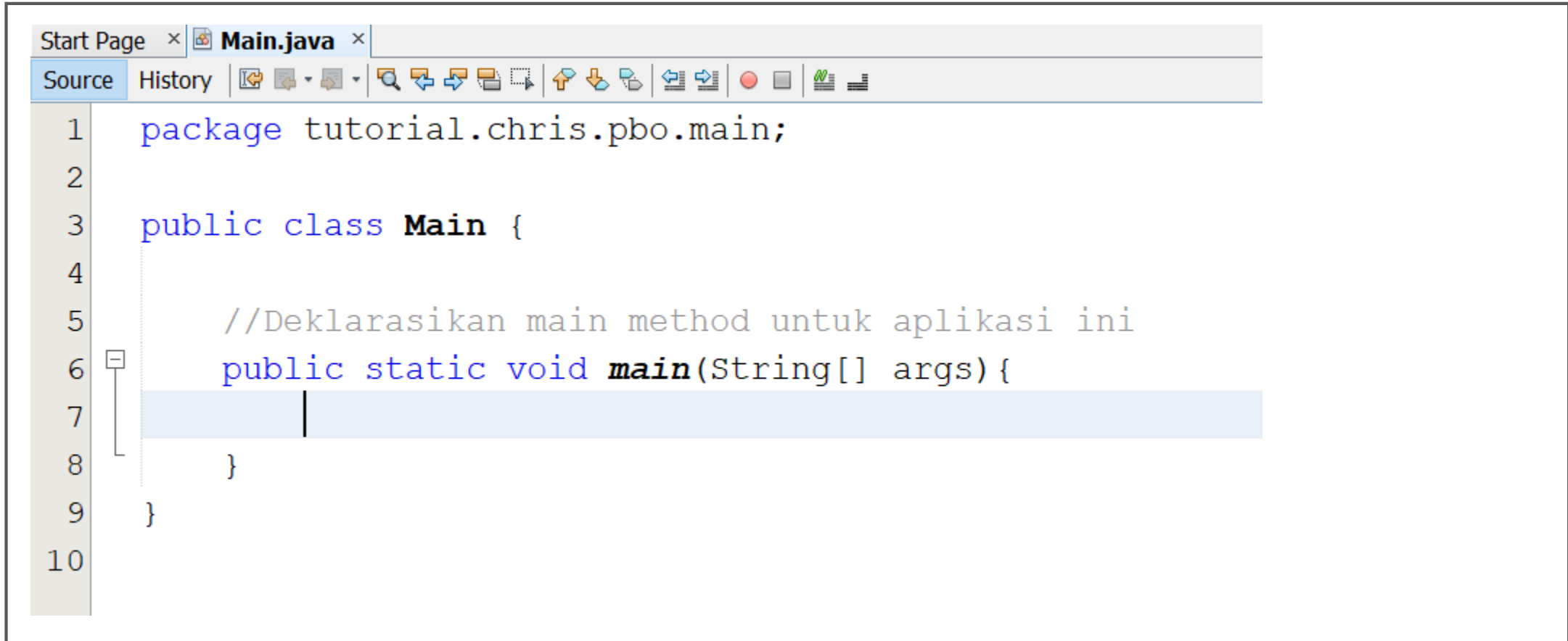
Praktikum Java Swing (1A)

- ❑ Buatlah **sebuah package main** pada project tersebut dengan format nama : nama_project.nama_Anda.pbo.nama_package. (contoh: praktikum.chrismikha.pbo.main)



Praktikum Java Swing (A)

- ❑ Tambahkan sebuah java class dengan nama Main didalam package main. Didalam class Main tambahkan juga sebuah **method main()** dengan **parameter (String[] args)**

A screenshot of an IDE window titled 'Main.java'. The window has a menu bar with 'Source' and 'History' tabs. Below the menu bar is a toolbar with various icons for editing and navigation. The code is displayed in a text area with line numbers on the left. The code defines a package 'tutorial.chris.pbo.main' and a public class 'Main'. Inside the class, there is a comment '//Deklarasikan main method untuk aplikasi ini' and a public static void method 'main' that takes a String array 'args' as a parameter. The method body is currently empty, with a cursor on line 7.

```
1 package tutorial.chris.pbo.main;
2
3 public class Main {
4
5     //Deklarasikan main method untuk aplikasi ini
6     public static void main(String[] args) {
7
8     }
9 }
10
```

Langkah 2A:

Membuat Container Dasar

Class JFrame

❑ Untuk membuat **container** dasar pada aplikasi GUI, kita perlu **membuat object frame dari Class JFrame**. Adapun beberapa hal yang perlu kita perhatikan dalam menggunakan JFrame :

- Memiliki 2 constructor, yaitu :
 - `JFrame()`
 - `JFrame(String)` → frame's title
- Method yang ada dalam class ini adalah :
 - `setTitle(String)` → frame's title
 - `setSize(int, int)` → width & height (pixel)
- Default frame ketika dibuat adalah invisible, maka setelah selesai diset, selanjutnya frame didisplaykan dengan method :
 - `show()` atau
 - `setVisible(true)`

Praktikum Java Swing (2A)

- ❑ Pada Class Main, buatlah sebuah object dari class JFrame (import class nya terlebih dahulu). Lalu tambahkan beberapa aksi berikut dengan memanfaatkan method dari class JFrame.
 - Set title dengan Aplikasi Java Swing
 - Set ukuran frame dengan 900 x 760 px
 - Set close dengan aksi stop program
 - Set frame selalu muncul (visible)

```
1  package tutorial.chris.pbo.main;
2  import javax.swing.JFrame;
3
4  public class Main {
5
6      //Deklarasikan main method untuk aplikasi ini
7      public static void main(String[] args){
8
9          JFrame frame = new JFrame("Aplikasi Java Swing"); //Instansiasi Object JFrame
10         frame.setSize(900,800); //Method untuk manipulasi ukuran
11         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); //Set button close
12         frame.setVisible(true); //Menampilkan Frame
13
14     }
15 }
```


Praktikum Java Swing (2A)

- ❑ Berikut ini adalah tampilan dari program ketika kode tersebut dijalankan.



- ❑ Setelah object dari Frame terbentuk (seperti contoh), maka langkah selanjutnya adalah **mengatur jenis layout** dari container

Layout Manager

- Layout komponen dalam sebuah Container (misalnya Frame, atau Panel) dapat diatur dengan menggunakan method `setLayout()`. Parameternya adalah objek Layout yang diantaranya adalah :
 - `null` : Untuk layout absolute, dimana posisi dan ukuran ditentukan secara manual oleh programmer.
 - `FlowLayout`
 - `BorderLayout`
 - `GridLayout`

Untuk contoh layout manager lainnya yang dapat digunakan, dokumentasinya dapat dilihat pada link berikut :

<https://docs.oracle.com/javase/tutorial/uiswing/layout/visual.html>

Absolut Layout / Null

- Absolut (null) adalah pengaturan layout secara absolute (programmer menentukan lokasi dan ukurannya secara manual).
- Untuk menentukan lokasi dan ukuran suatu komponen, gunakan method `setBounds` milik setiap komponen. Parameter `setBounds` adalah
 - `x` : lokasi koordinat x (bagian kiri dari komponen)
 - `y` : lokasi koordinat y (bagian atas dari komponen)
 - `width` : lebar komponen
 - `height` : tinggi komponen
- Semua satuannya adalah pixel.
- Contoh : `setBounds(40,50,100,20)` : Koordinat kiri atas adalah 40,50 dengan lebar 100 dan tinggi 20.

Praktikum Java Swing (2A)

- ❑ Tambahkan kode berikut didalam Class Main untuk **menset layout manager** dari object JFrame.
 - Set Layout manager dengan Absolut layout = **objFrame.setLayout(objLayout)**

```
1  package tutorial.chris.pbo.main;
2  ❑ import javax.swing.JFrame;
3
4  public class Main {
5
6      //Deklarasikan main method untuk aplikasi ini
7      ❑ public static void main(String[] args){
8
9          JFrame frame = new JFrame("Aplikasi Java Swing"); //Instansiasi Object JFrame
10         frame.setSize(900,760); //Method untuk manipulasi ukuran
11         frame.setLayout(null); //Menambahkan layout manager = absolut layout
12         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); //Set button close
13         frame.setVisible(true); //Menampilkan Frame
14
15     }
16 }
```

Langkah 3A:

Menambahkan Panel

Praktikum Java Swing (3A)

JPanel

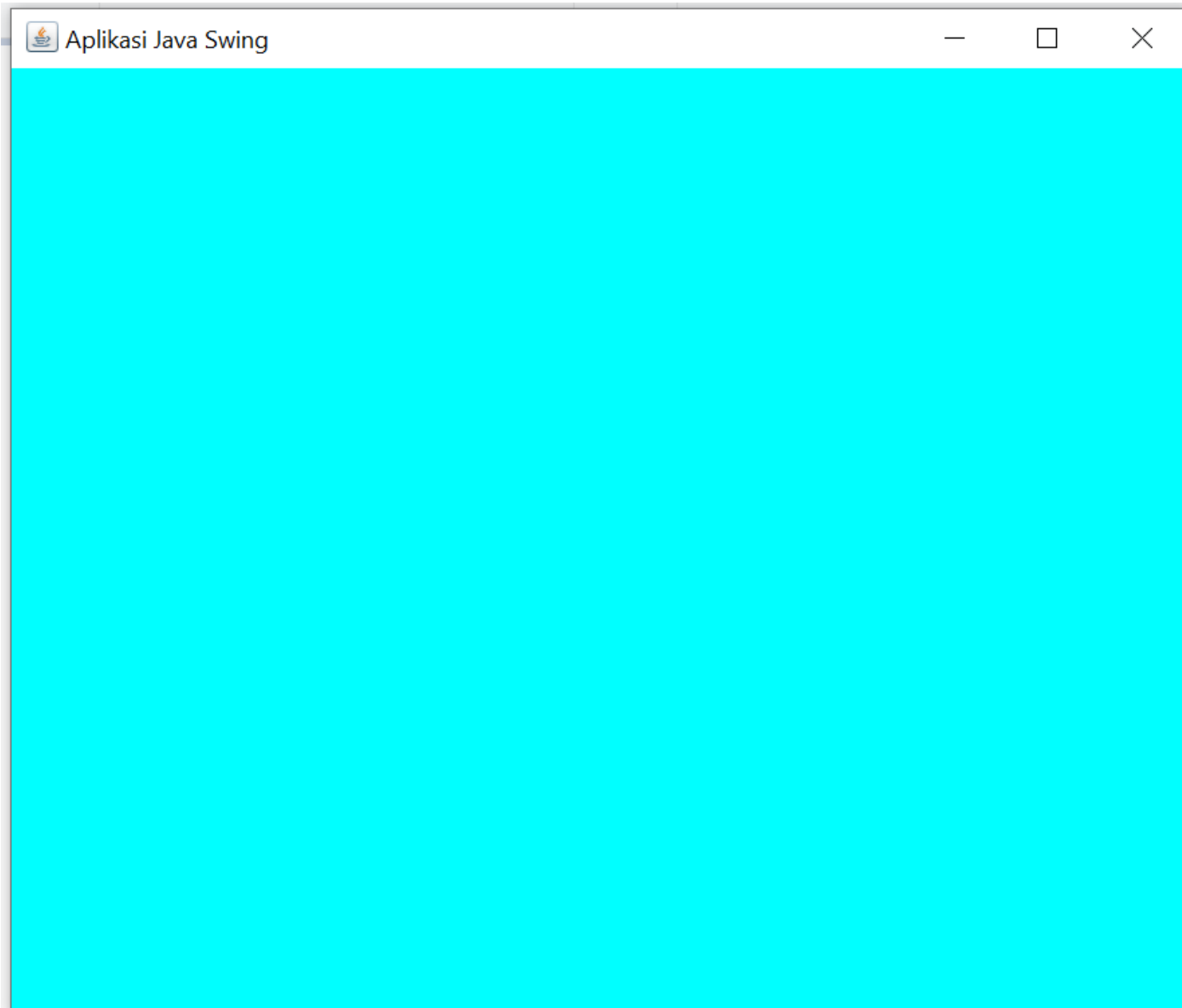
Merupakan komponen yang digunakan **untuk menyusun/ mengelompokkan komponen lain** untuk diletakan didalam kontainer agar menjadi lebih rapi. Panel berfungsi sebagai intermediete container didalam hirarki pembuatan GUI.

Sintak

```
//Membuat dan Menambahkan JPanel kedalam Frame
JPanel mainPanel = new JPanel(); //Buat obj panel dari clas JPanel
mainPanel.setBounds(0, 0, 900, 720); //Set ukuran dan posisi panel
mainPanel.setBackground(Color.cyan); //set warna background
mainPanel.setLayout(null); //set layout didalam panel

frame.add(mainPanel); //menambahkan object panel kedalam frame
```

Praktikum Java Swing (3A)



- ❑ Tampilan frame dengan ditambahkan sebuah object panel didalamnya

Langkah 4A:

Menambahkan Component (aksi)

Component GUI pada SWING

Swing menyediakan beberapa Class untuk membuat **komponen kontrol (atomic)** GUI yang dapat dimanfaatkan didalam pembuatan aplikasi visual. Antara lain :

A light gray hexagon with a thin black border, containing the text 'JLabel' in green.

JLabel

Merupakan komponen gui yang digunakan untuk **membuat & menampilkan tulisan atau gambar** pada container sebagai suatu bentuk informasi untuk pengguna program. Untuk menggunakan JLabel, maka sebelumnya perlu dibuat suatu **obyek menggunakan class JLabel**.

A light gray hexagon with a thin black border, containing the text 'JTextField' in green.

JTextField

Merupakan komponen GUI yang digunakan untuk memasukkan **1 baris text (String)** yang selanjutnya dapat digunakan sebagai nilai masukan (input) bagi proses selanjutnya. Pembuatan JTextField Dilakukan dengan **membuat obyek menggunakan class JTextField**.

Component GUI pada SWING

JButton

Merupakan komponen gui berbentuk tombol. Komponen ini banyak digunakan sebagai media interaksi pengguna untuk **melakukan eksekusi terhadap suatu proses/perintah** tertentu pada program. Pembuatan JButton Dilakukan dengan **membuat obyek menggunakan class JButton**.

JRadioButton

Merupakan komponen gui yang digunakan untuk **memberikan beberapa opsi pilihan masukan program** kepada pengguna dimana pengguna perlu **memilih satu** diantara beberapa pilihan yang tersedia. Pembuatan JRadioButton Dilakukan dengan **membuat obyek menggunakan class JRadioButton**.

JCheckBox Box

Merupakan komponen gui yang digunakan untuk memberikan **beberapa opsi pilihan masukan** kepada pengguna dimana opsi yang disediakan **dapat dipilih lebih dari satu**. Pembuatan JCheckBox Dilakukan dengan **membuat obyek menggunakan class JCheckBox**.

Component GUI pada SWING

JCombo Box

Merupakan komponen gui yang digunakan **memberikan beberapa opsi pilihan masukan program** kepada pengguna yang berbentuk semacam TextField dan ada panah ke bawah (drop down). Pembuatan JComboBox Dilakukan dengan **membuat obyek menggunakan class JComboBox**.

JTextArea

merupakan komponen dengan fungsi yang mirip dengan komponen JTextField tetapi dapat **menampung masukan teks yang lebih dari satu baris (paragraf)**. Pembuatan JTextArea Dilakukan dengan **membuat obyek menggunakan class JTextArea**

JPanel

Merupakan komponen yang digunakan **untuk menyusun/ mengelompokkan komponen lain** untuk diletakan didalam kontainer agar menjadi lebih rapi. Panel berfungsi sebagai intermediete container didalam hirarki pembuatan GUI.

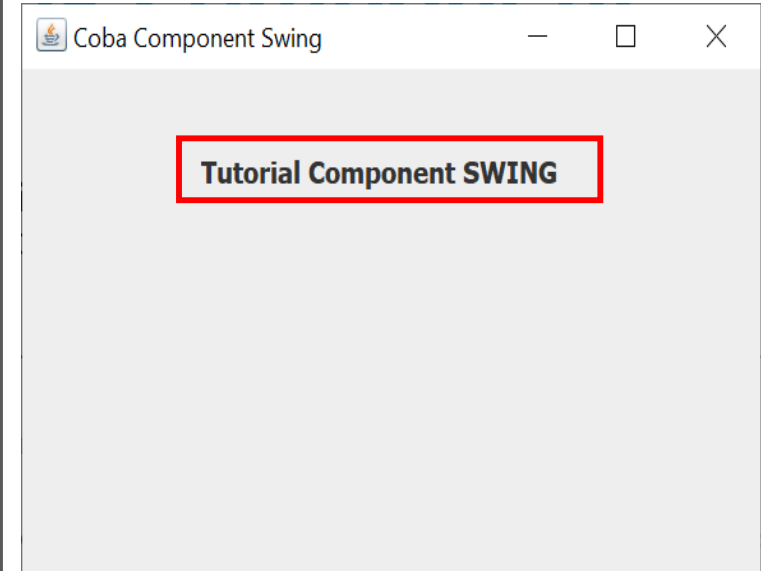
Sintak Component GUI (JLabel)

JLabel

Contoh sintaks dasar untuk membuat Objek JLabel dan menambahkannya kedalam container (Objek dari Class JFrame) :

```
/*--Pembuatan component JLabel untuk ditambahkan kedalam container--*/
JLabel label1;
label1 = new JLabel("Tutorial Component SWING"); //Instansiasi Object
label1.setBounds(140, 50, 500, 40); //Set Posisi & Ukuran Label (pixel)
label1.setFont(new Font("tahoma",1,20)); // Set Jenis & ukuran font

/*--Menambahkan Object JLabel kedalam container--*/
form.add(label1); //Menambahkan objek JLabel
```



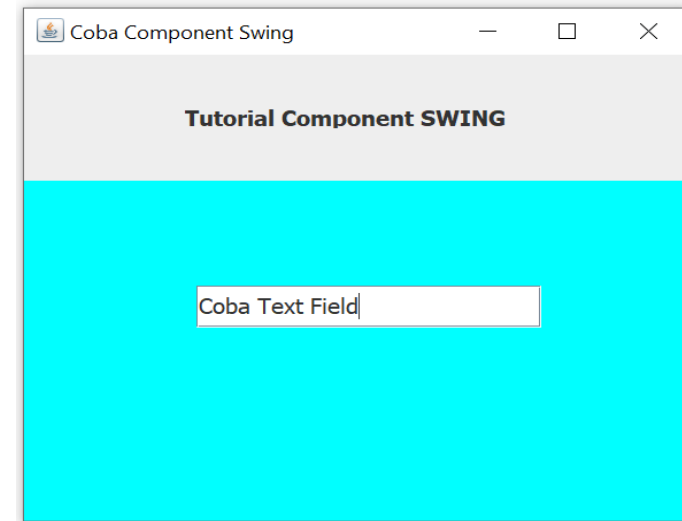
Sintak Component GUI (JTextField)

JTextField

Contoh sintaks dasar untuk membuat Objek JTextField dan menambahkannya kedalam kontainer (Objek dari Class JPanel) :

```
/*--Pembuatan Component JTextField untuk ditambahkan kedalam container--*/
JTextField textField1;
textField1 = new JTextField(); //Instansiasi Object textField1
textField1.setBounds(150, 100, 300, 40); //Set Posisi & Ukuran object (pixel)
textField1.setFont(new Font("tahoma",0,20)); // Set Jenis & ukuran font

/*--Menambahkan Object JButton kedalam container --*/
panel.add(textField1); //Menambahkan objek textField1 pada panel
```

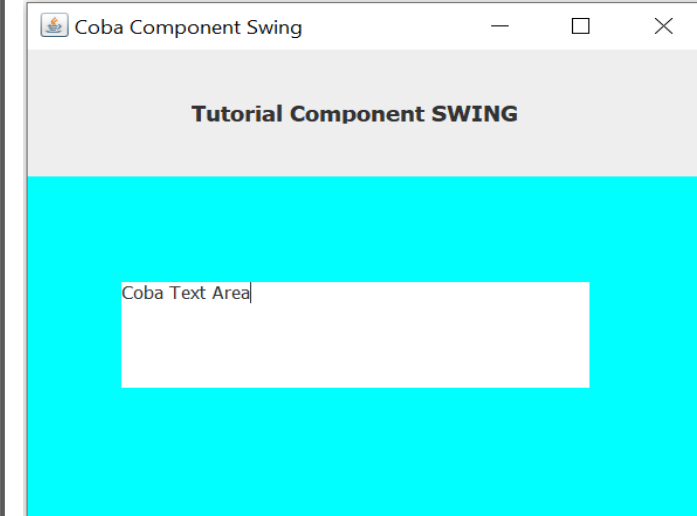


Sintak Component GUI (JTextArea)

JText Area

Contoh sintaks dasar untuk membuat Objek JTextArea dan menambahkannya kedalam kontainer (Objek dari Class JPanel) :

```
/*--Pembuatan Component JTextArea untuk ditambahkan kedalam container--*/  
JTextArea textAreal;  
textAreal = new JTextArea(); //Instansiasi Object textAreal  
textAreal.setBounds(80, 100, 400, 100); //Set Posisi & Ukuran object(pixel)  
textAreal.setFont(new Font("tahoma",0,16)); // Set Jenis & ukuran font  
  
/*--Menambahkan Object JButton kedalam container --*/  
panel.add(textAreal); //Menambahkan objek textAreal pada panel
```



Sintak Component GUI (JRadioButton)

JRadio Button

Contoh sintaks dasar untuk membuat Objek JRadioButton dan menambahkannya kedalam kontainer (Objek dari Class JPanel) :

```
/*--Pembuatan Component JRadioButton untuk ditambahkan kedalam container--*/
JRadioButton radiol;
radiol = new JRadioButton(); //Instansiasi Object radiol

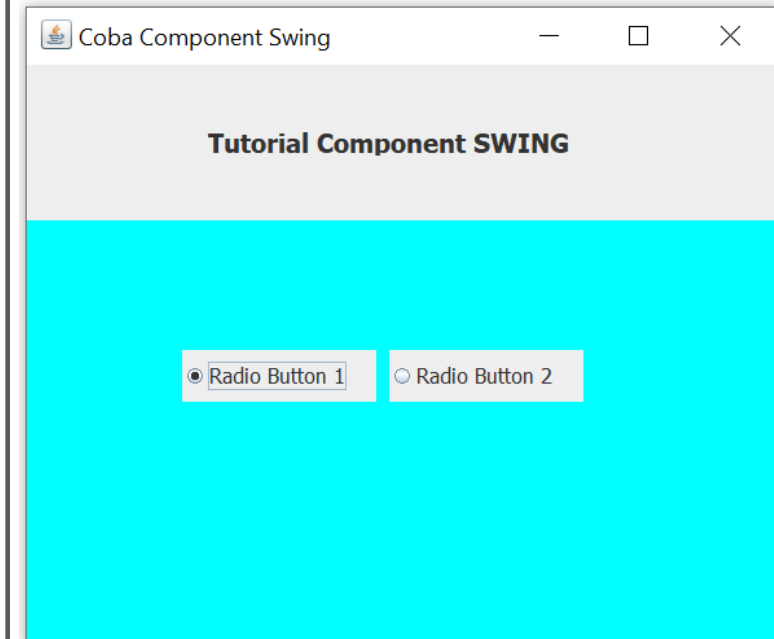
JRadioButton radio2;
radio2 = new JRadioButton(); //Instansiasi Object radio2

radiol.setText("Radio Button 1");
radiol.setBounds(120, 100, 150, 40); //Set Posisi & Ukuran object(pixel)
radiol.setFont(new Font("tahoma",0,16)); // Set Jenis & ukuran font

radio2.setText("Radio Button 2");
radio2.setBounds(280, 100, 150, 40); //Set Posisi & Ukuran object(pixel)
radio2.setFont(new Font("tahoma",0,16));

/*--Pembuatan Button Grup
   agar hanya bisa memilih 1 radio button saja di grup yg sama*/
ButtonGroup radioGroup = new ButtonGroup();
radioGroup.add(radiol); radioGroup.add(radio2); //Add R Button ke group

/*--Menambahkan Object JRadioButton kedalam container --*/
panel.add(radiol); /
panel.add(radio2);
```



Sintak Component GUI (JCheckBox)

JCheck Box

Contoh sintaks dasar untuk membuat Objek JCheckBox dan menambahkannya kedalam kontainer (Objek dari Class JPanel) :

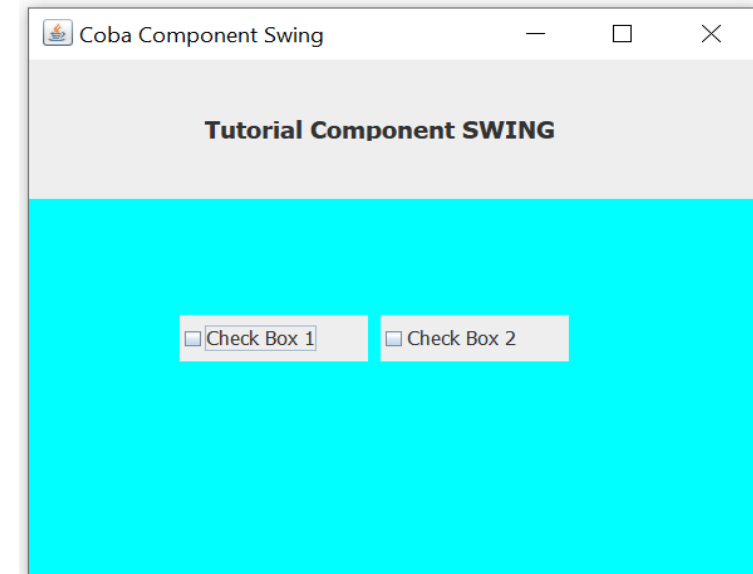
```
/*--Pembuatan Component JCheckBox untuk ditambahkan kedalam container--*/
JCheckBox box1;
box1 = new JCheckBox();

JCheckBox box2;
box2 = new JCheckBox();

box1.setText("Check Box 1");
box1.setBounds(120, 100, 150, 40); //Set Posisi & Ukuran object(pixel)
box1.setFont(new Font("tahoma",0,16)); // Set Jenis & ukuran font

box2.setText("Check Box 2");
box2.setBounds(280, 100, 150, 40); //Set Posisi & Ukuran object(pixel)
box2.setFont(new Font("tahoma",0,16));

/*--Menambahkan Object JRadioButton kedalam container --*/
panel.add(box1);
panel.add(box2);
```



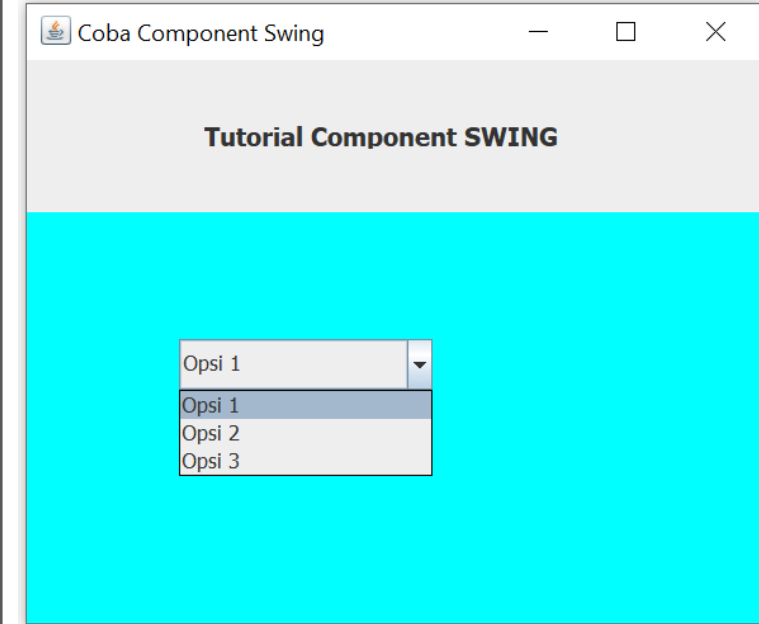
Sintak Component GUI (JComboBox)

JCombo Box

Contoh sintaks dasar untuk membuat Objek JComboBox dan menambahkannya kedalam kontainer (Objek dari Class JPanel) :

```
/*--Pembuatan Component JComboBox untuk ditambahkan kedalam container--*/
JComboBox comboBox;
String model[]={"Opsi 1","Opsi 2", "Opsi 3"}; // Buat Array untuk Isi Combo Box
comboBox = new JComboBox(model); // Panggil Array didalam Konstruktor
comboBox.setBounds(120, 100, 200, 40); //Set Posisi & Ukuran object(pixel)
comboBox.setFont(new Font("tahoma",0,16)); // Set Jenis & ukuran font

/*--Menambahkan Object JComboBox kedalam container --*/
panel.add(comboBox); //Menambahkan objek comboBox pada panel
```

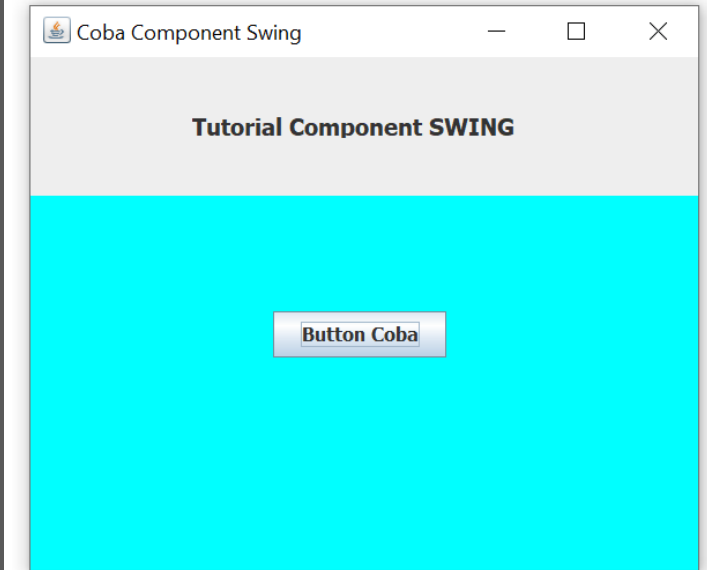


Sintak Component GUI (JButton)

JButton

Contoh sintaks untuk membuat Objek JButton dan menambahkannya kedalam kontainer (Objek dari Class JPanel) :

```
/*--Pembuatan component JButton untuk ditambahkan kedalam container--*/  
JButton button1;  
button1 = new JButton();  
button1.setText("Button Coba");  
button1.setBounds(210, 100, 150, 40);  
button1.setFont(new Font("tahoma",1,16));  
  
/*--Menambahkan Object JButton kedalam container--*/  
panel.add(button1); //Menambahkan objek button1
```

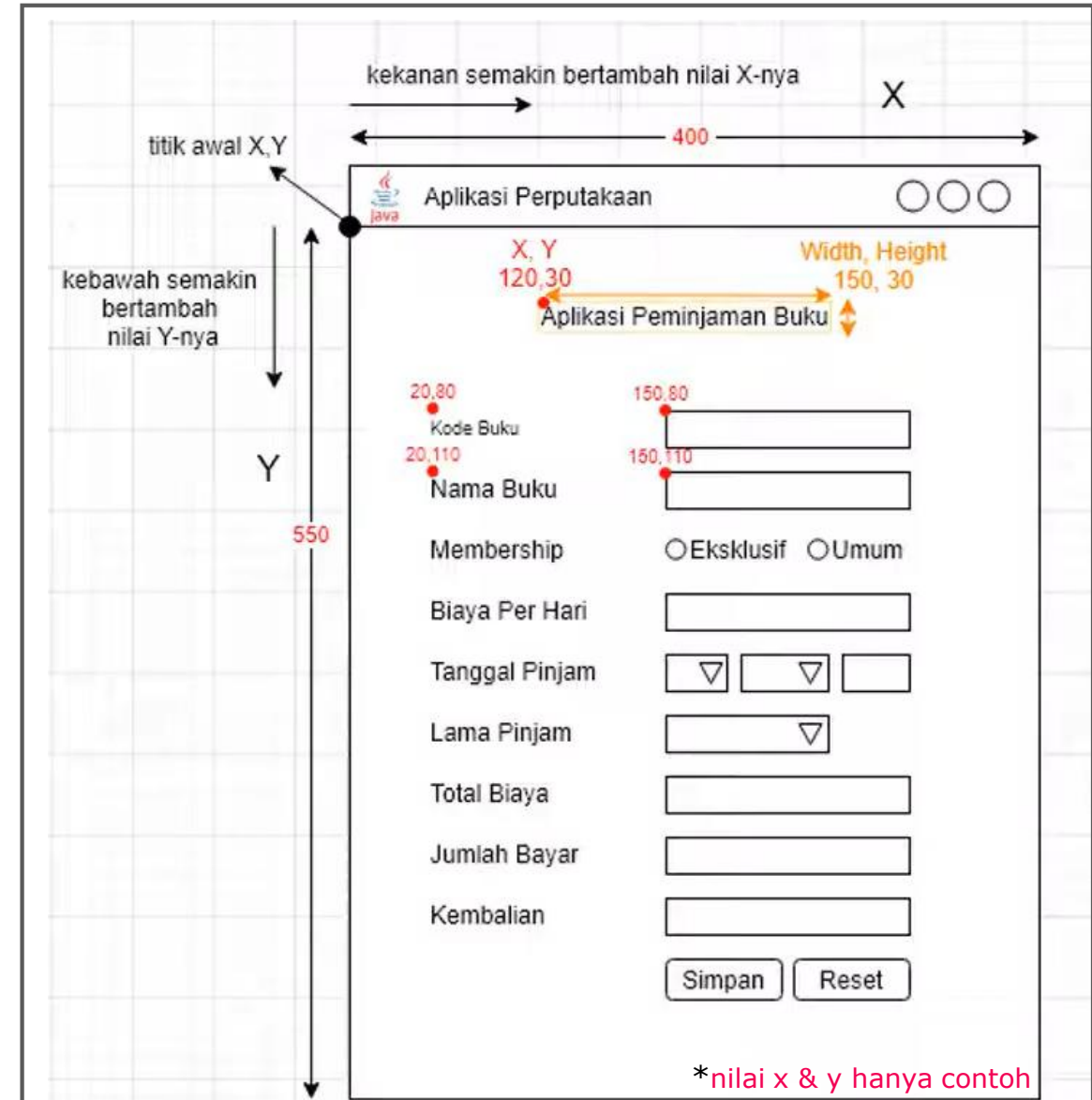


Konsep Tata Letak Component

Dalam membuat program berbasis GUI, **Menata component** merupakan pekerjaan pertama yang perlu kita lakukan. Tujuannya agar tampilan program **dapat tersusub dengan rapi**.

Beberapa hal yang harus diperhatikan :

- ❑ Kontainer memiliki **2 parameter** x & y (panjang & lebar) dengan satuan pixel
- ❑ Ruang (space) yang dapat dimanfaatkan untuk meletakan component bergantung **ukuran dari container**
- ❑ Component memiliki **4 parameter untuk posisi yaitu x,y,height,width** dengan satuan pixel. Merepresentasikan **posisi & dimensi** component didalam container (**absolut layout**)
- ❑ Titik awal / posisi awal container adalah **(0,0)** , perhatikan contoh disamping
- ❑ Untuk dapat menata component dengan baik, pahami bagaimana meletakan object di bidang datar.



Praktikum Java Swing (4A)

Object 1

Tambahkan Sebuah Object JLabel kedalam program. Aturlah object tersebut dengan aturan berikut :

- Set text dengan "Belajar GUI dengan JAVA Swing"
- Set ukuran object Label = 600 x 50 px. Set posisi label (x = 185, y = 75)
- Set font dengan Tahoma, bold, 32
- Letakan Label didalam object Frame

Sintak

```
23  
24      //Membuat dan Menambahkan object Label kedalam Frame  
25      JLabel label1 = new JLabel(); //Buat object Label  
26      label1.setText("Belajar GUI dengan JAVA Swing"); //Set Text object Label  
27      label1.setBounds(185,75,600,50); //Set posisi & ukuran object label  
28      label1.setFont(new Font("Tahoma",1,32)); //Set font object label  
29  
30      frame.add(label1); //tambahkan object label kedalam object frame
```

Praktikum Java Swing (4A)

Object 2

Tambahkan Sebuah Object JTextField kedalam program. Aturlah object tersebut dengan aturan berikut :

- Set ukuran object Label = 600 x 50 px. Set posisi label (x = 150, y = 100)
- Set font dengan Tahoma, bold, 18
- Letakan TextField didalam object Panel

Sintak

```
34 //Membuat dan Menambahkan object Text Field kedalam Frame
35 JTextField txtField1 = new JTextField(); // buat object text field
36 txtField1.setBounds(150,100,600,50); //set posisi & ukuran object
37 txtField1.setFont(new Font("tahoma", 1, 18)); //se font
38
39 mainPanel.add(txtField1); //tambahkan object txtField kedalam panel
40
```

Praktikum Java Swing (4A)

Object 3

Tambahkan Sebuah Object JButton kedalam program. Aturlah object tersebut dengan aturan berikut :

- Set Text button = "TAMPIL"
- Set ukuran object Label = 200 x 50 px. Set posisi label (x = 350, y = 180)
- Set font dengan arial, bold, 20
- Letakan TextField didalam object Panel

Sintak

```
34 //Membuat dan Menambahkan object Text Field kedalam Frame
35 JTextField txtField1 = new JTextField(); // buat object text field
36 txtField1.setBounds(150,100,600,50); //set posisi & ukuran object
37 txtField1.setFont(new Font("tahoma", 1, 18)); //se font
38
39 mainPanel.add(txtField1); //tambahkan object txtField kedalam panel
40
```

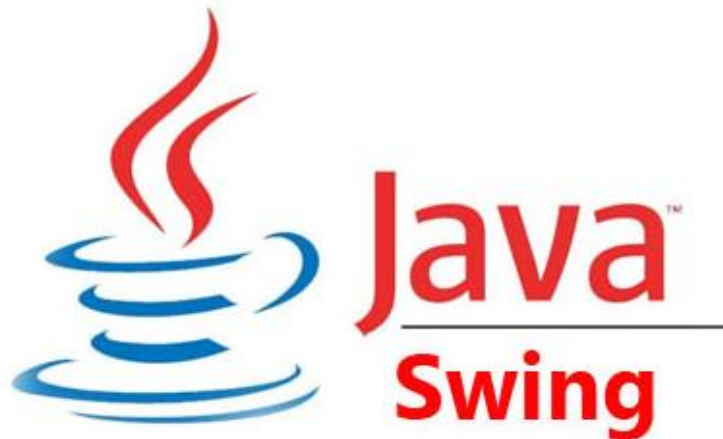
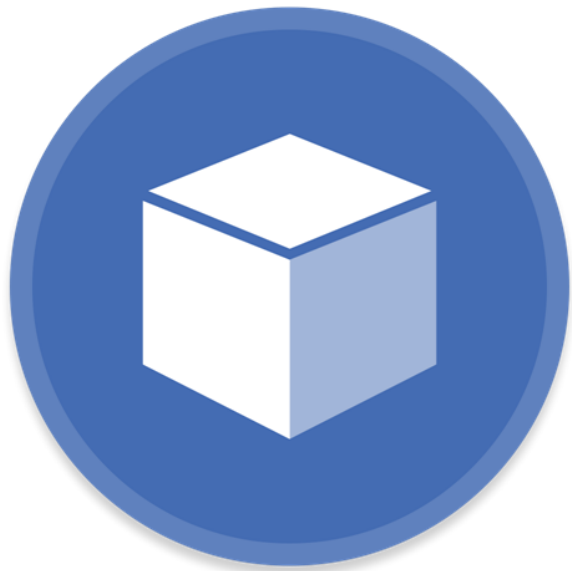
Praktikum Java Swing (4A)

Tampilan akhir dari aplikasi adalah sebagai berikut :



Bagaimana Dengan Cara ke 2 atau SoftCoding ??

Disini kita akan **memanfaatkan GUI Builder** yang disediakan oleh IDE Netbeans untuk mempermudah mendvelop tampilan program berbasis GUI.



Langkah 1B:

Membuat Project & Package Baru

Praktikum Java Swing (1B)

- ❑ Buatlah project baru pada netbeans. dengan cara : **File -> New Project-> Java Application -> Tulis nama Project -> unchecklist main class - > finish**

New Java Application

Steps

1. Choose Project
2. **Name and Location**

Name and Location

Project Name:

Project Location:

Project Folder:

☐ Use Dedicated Folder for Storing Libraries

Libraries Folder:

Different users and projects can share the same compilation libraries (see Help for details).

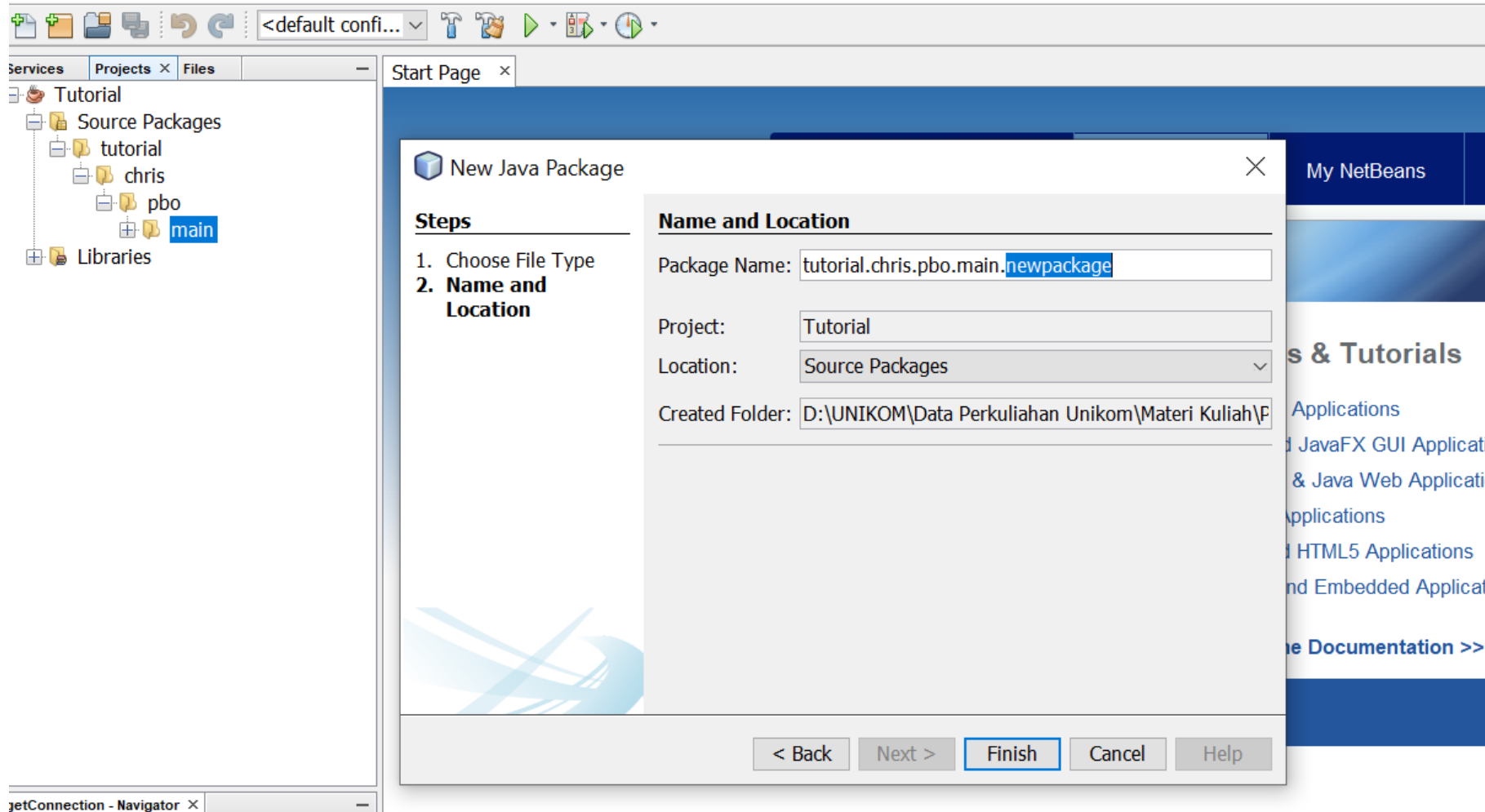
☐ **Create Main Class**

< Back Next > **Finish** Cancel Help

***Nama Project Sesuaikan Kebutuhan Praktek**

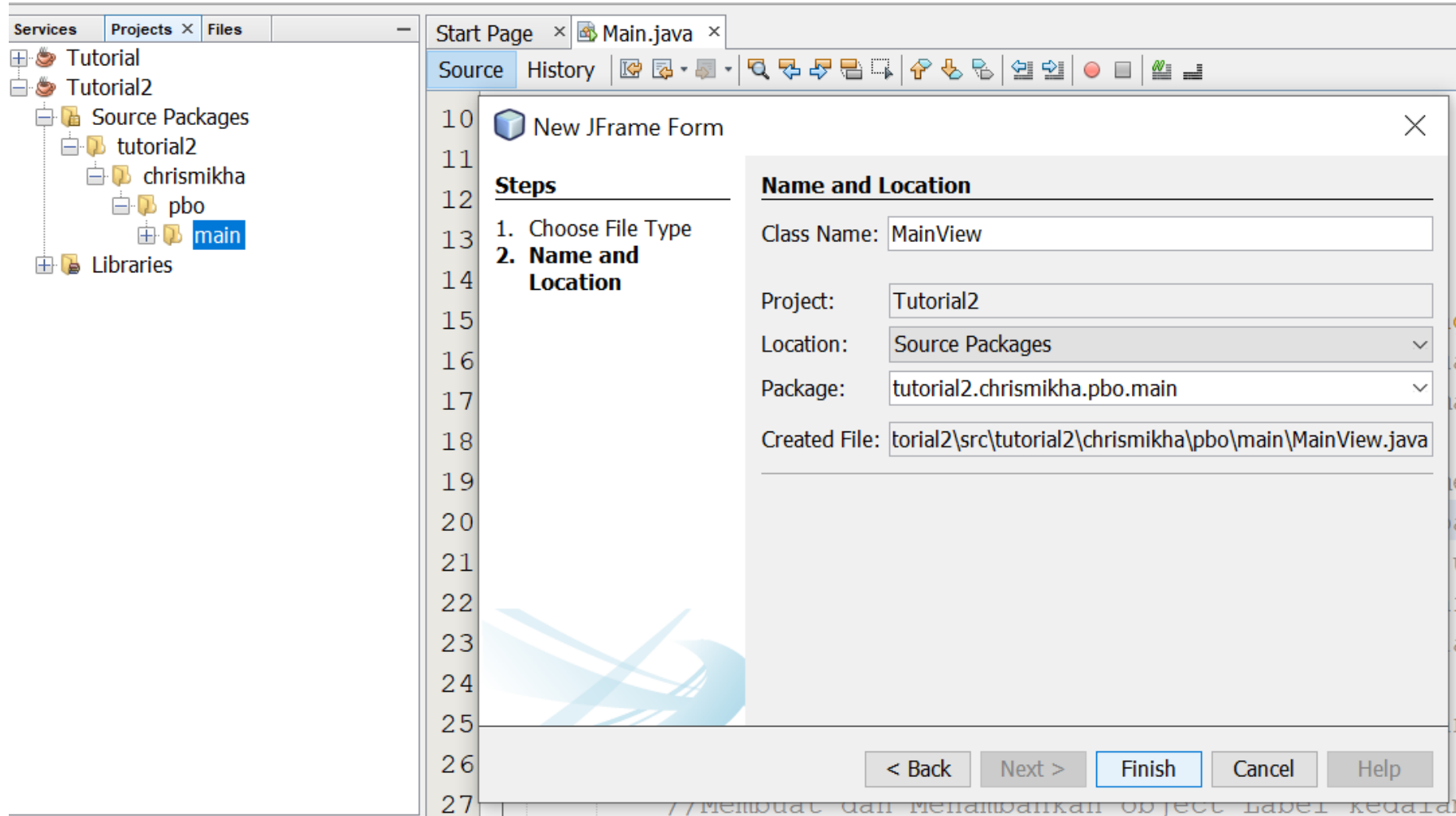
Praktikum Java Swing (1B)

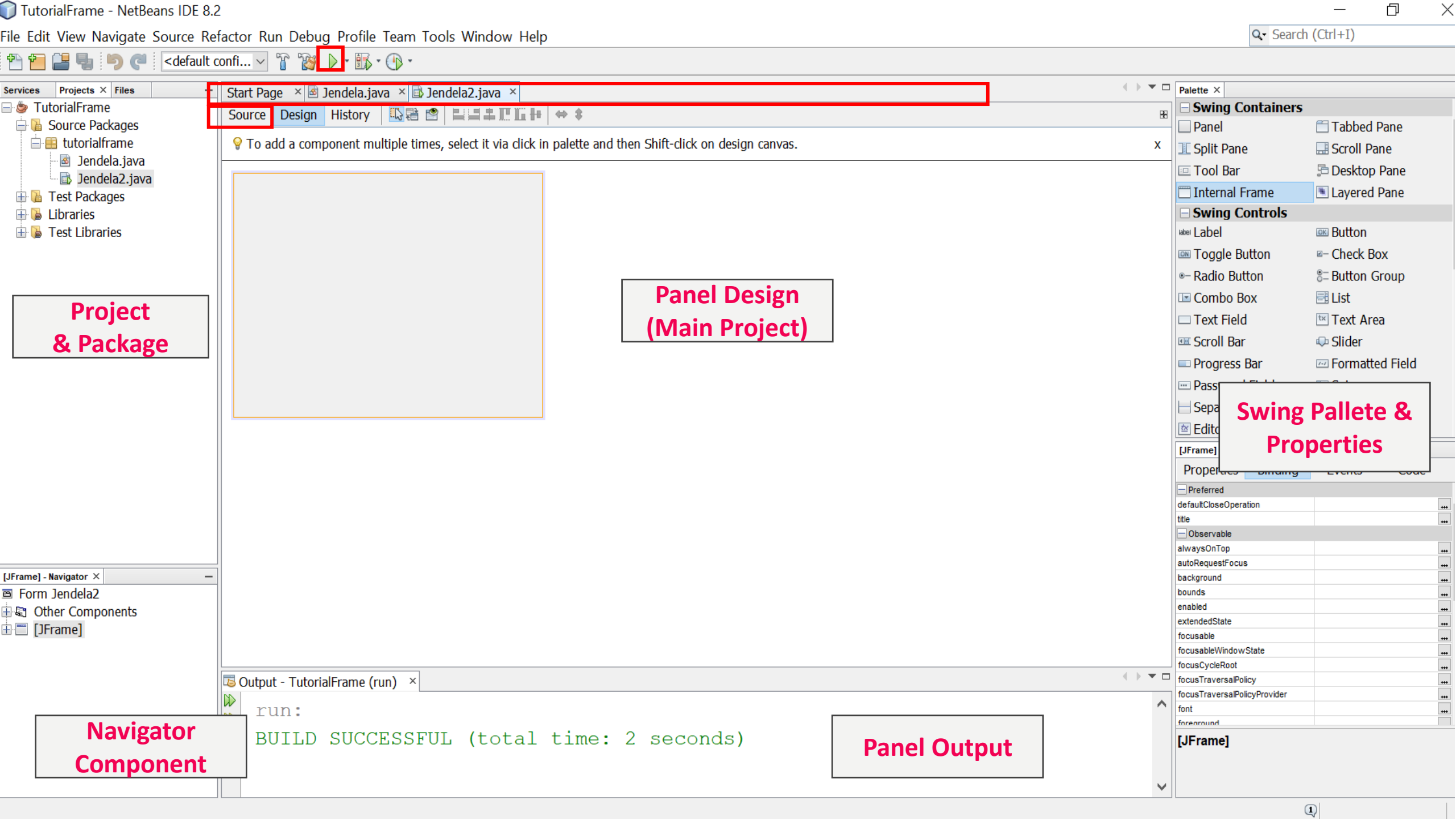
- ❑ Buatlah **sebuah package main** pada project tersebut dengan format nama : nama_project.nama_Anda.pbo.nama_package. (contoh: praktikum.chrismikha.pbo.main)



Praktikum Java Swing (1B)

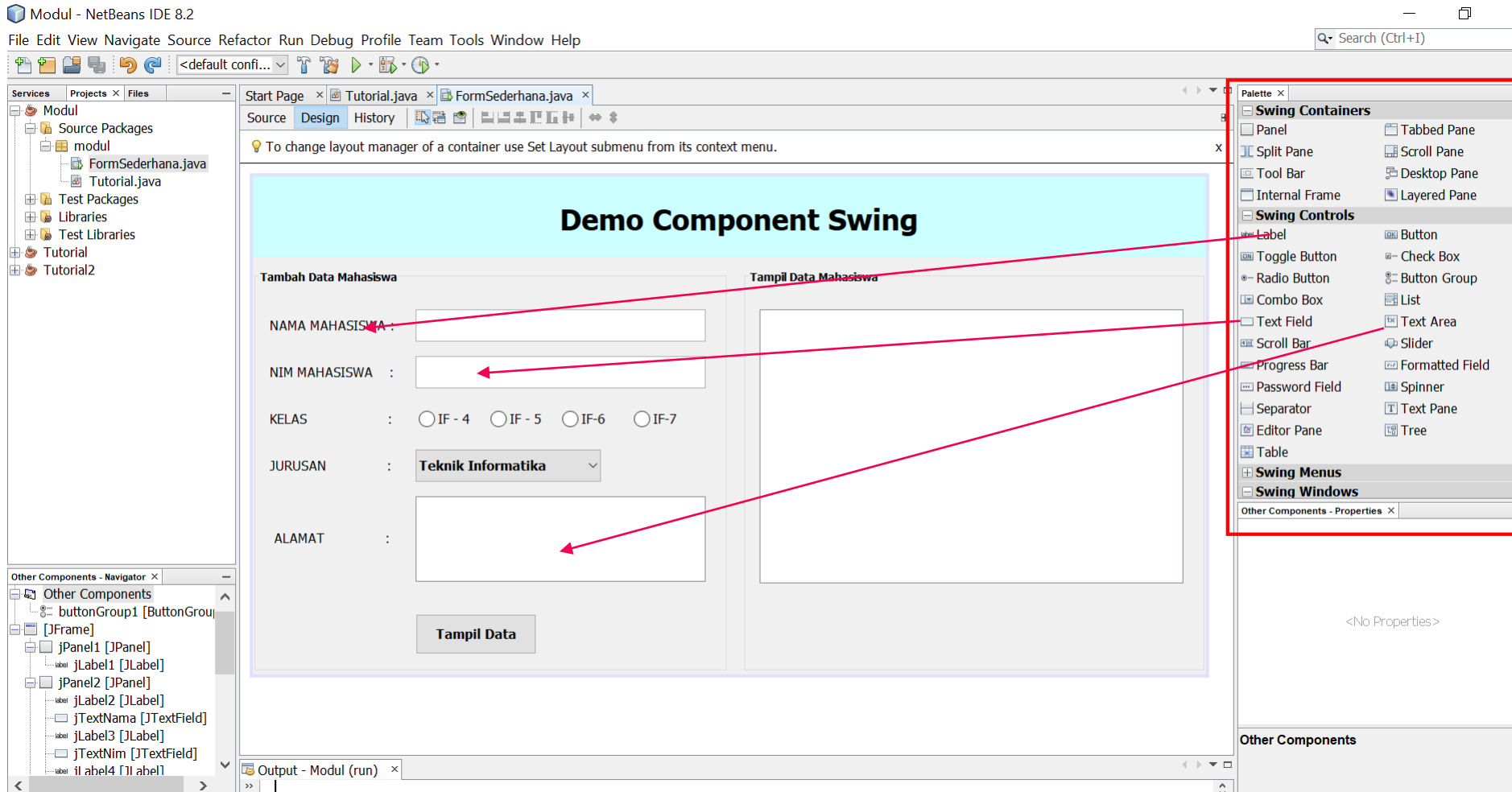
- ❑ Tambahkan sebuah Class JFrame Form dan berilah nama class tersebut dengan MainView. Untuk menambahkan JFrame Form : pada package click kanan -> new -> JFrame Form -> isi nama Class.





Menata Component Dengan GUI Builder

Dengan Memanfaatkan **GUI Builder** pada Netbeans, Anda akan lebih mudah untuk mengaplikasikan **konsep tata letak component swing** kedalam container (fitur drag and drop)



Cari **component GUI** yang ingin digunakan pada **palette**.

Drag & drop **component** **kedalam container** Program Anda (cth pada Object JFrame atau Object Jpanel)

Menata Component Dengan GUI Builder

Untuk mengatur **atribut nilai** dari tiap component (contoh : mengatur setText, setSize, setBounds, dst) juga dapat dilakukan melalui GUI Builder

Modul - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Pilih component yang ingin anda kelola.

Carilah pengaturan pada tab **properties** yang sesuai dengan apa yang ingin anda kerjakan. (sederhannya ini seperti anda **menset nilai** **kedalam method**)

Langkah 2B:

Membuat Container Dasar (Frame)

Praktikum Java Swing (3B)

- ❑ Ketika kita membuat Class sebagai JFrameForm, maka secara otomatis object Frame sudah dibuatkan oleh Netbeans GUI generator. Kita hanya perlu memodifikasi bentuk frams sesuai kebutuhan.



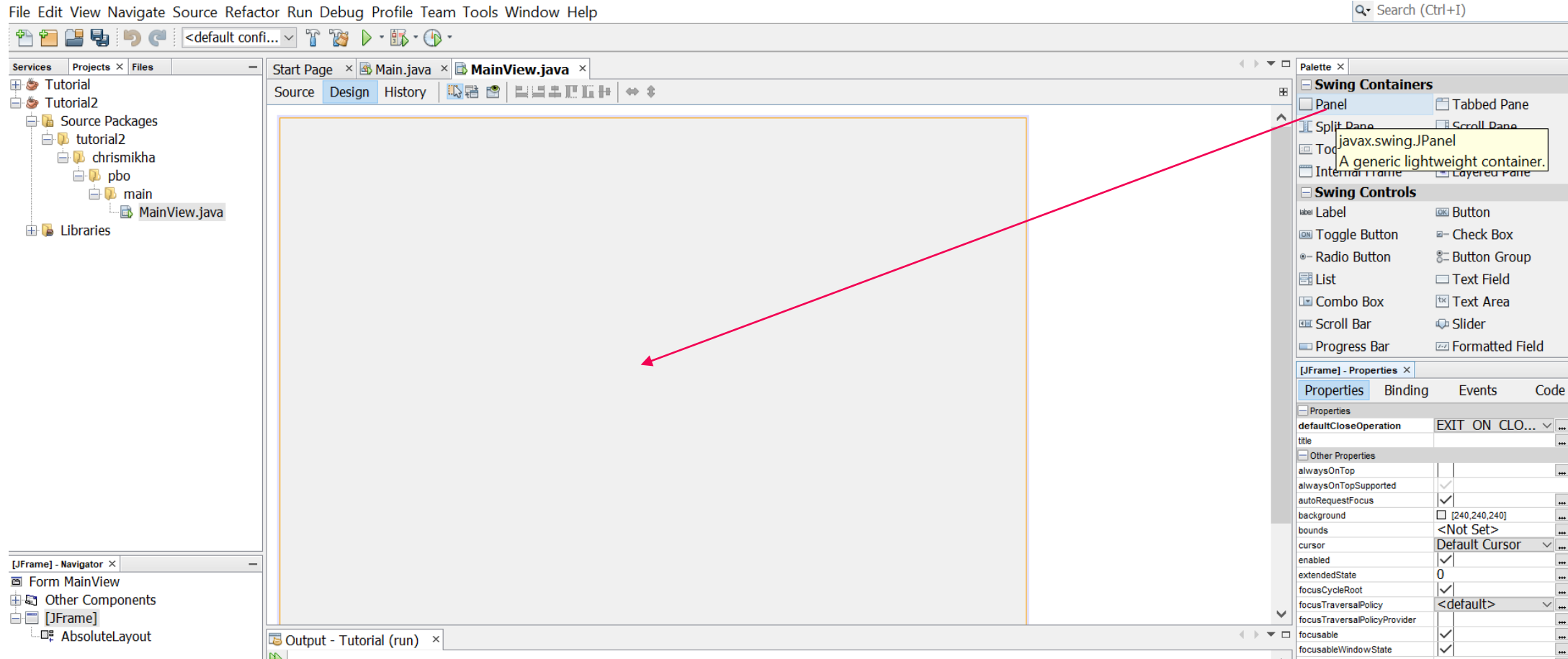
- Set title dengan Aplikasi Java Swing
- Set ukuran frame dengan 900 x 760 px
- Set layout dengan absolut layout
- Set frame selalu muncul ditengah (center)

Langkah 3B:

Menambahkan Panel

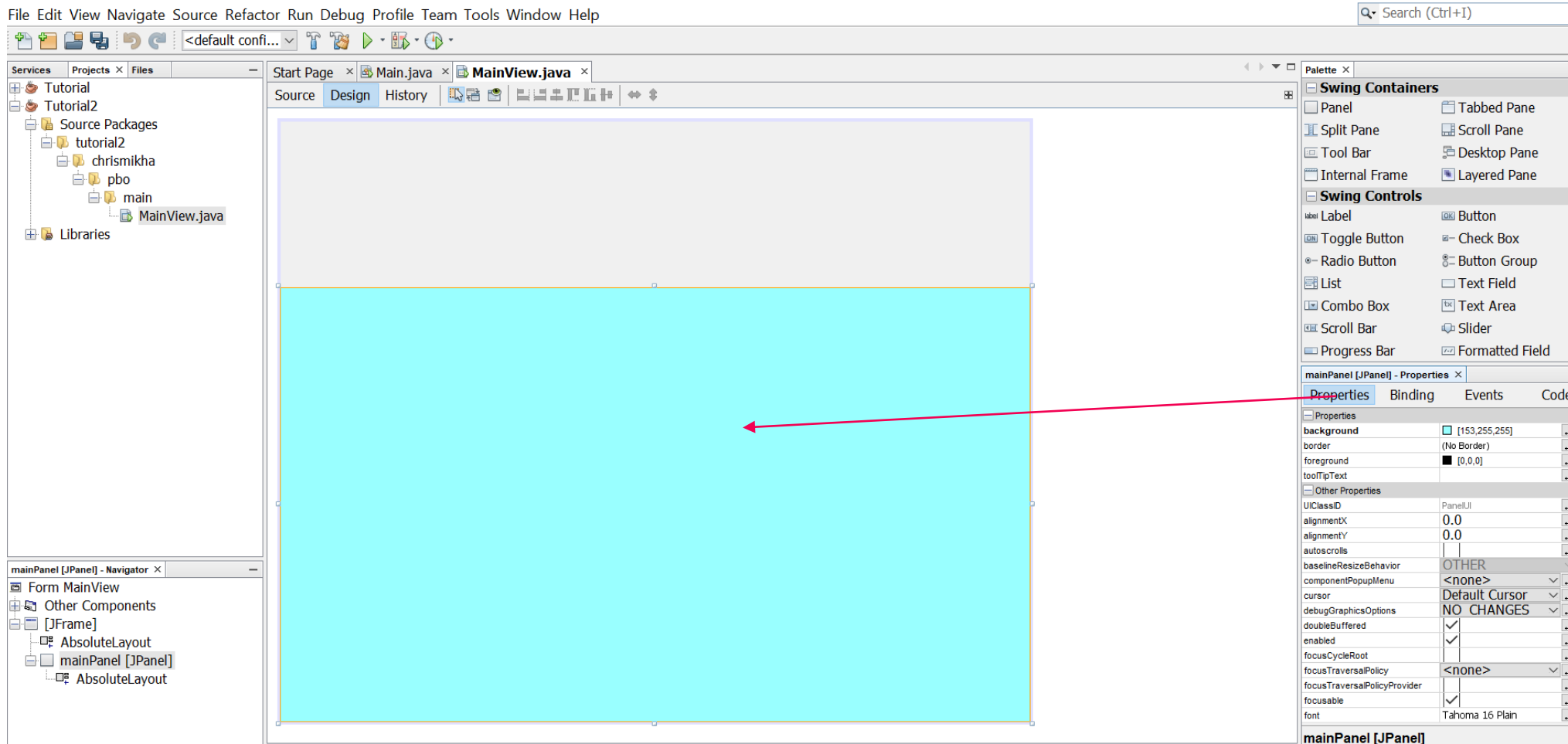
Praktikum Java Swing (4B)

- ❑ **Drag & Drop** Component **JPanel** pada pallette dan letakan didalam Frame pada builder



Praktikum Java Swing (4B)

- ❑ Aturlah kebutuhan panel melalui properties . Set ukuran, layout dst berdasarkan kasus sebelumnya (bagian A)



Langkah 4B:

Menambahkan Component (aksi)

Praktikum Java Swing (4B)

- ❑ Tambahkan component JLabel, JButton, & JTextField dengan men **drag & drop** dari pallete. Untuk bentuk object sesuaikan dengan kebutuhan.

Start Page x Main.java x MainView.java x

Source Design History

Belajar GUI dengan JAVA Swing

TAMPIL

Palette x

Swing Containers

- Panel
- Split Pane
- Tool Bar
- Internal Frame
- Tabbed Pane
- Scroll Pane
- Desktop Pane
- Layered Pane

Swing Controls

- Label
- Toggle Button
- Radio Button
- List
- Combo Box
- Scroll Bar
- Progress Bar
- Button
- Check Box
- Button Group
- Text Field
- Text Area
- Slider
- Formatted Field

JButton1 [JButton] - Properties x

Properties Binding Events Code

Properties

| | |
|-------------|---------------|
| action | <none> |
| background | [240,240,240] |
| font | Arial 22 Bold |
| foreground | [0,0,0] |
| icon | <none> |
| mnemonic | |
| text | TAMPIL |
| toolTipText | |

Other Properties

| | |
|------------------------|-----------------|
| UIClassID | ButtonUI |
| actionCommand | TAMPIL |
| alignmentX | 0.0 |
| alignmentY | 0.5 |
| autoscrolls | |
| baselineResizeBehavior | CENTER_OFFSET |
| border | [XPEmptyBorder] |
| borderPainted | ✓ |
| buttonGroup | <none> |
| componentPopupMenu | <none> |
| contentAreaFilled | ✓ |

Praktikum Java Swing (4B)

- ❑ Coba compile & jalankan program dan lihat hasilnya. Untuk contoh source codenya (yang tergenerate oleh gui builder) dapat anda lihat dibawah ini :



```
28 private void initComponents() {
29
30     mainPanel = new javax.swing.JPanel();
31     txtField1 = new javax.swing.JTextField();
32     jButton1 = new javax.swing.JButton();
33     jLabel1 = new javax.swing.JLabel();
34
35     setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
36     setTitle("APLIKASI JAVA SWING");
37     setPreferredSize(new java.awt.Dimension(900, 760));
38     getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());
39
40     mainPanel.setBackground(new java.awt.Color(153, 255, 255));
41     mainPanel.setPreferredSize(new java.awt.Dimension(900, 520));
42     mainPanel.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());
43
44     txtField1.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
45     txtField1.setPreferredSize(new java.awt.Dimension(500, 50));
46     mainPanel.add(txtField1, new org.netbeans.lib.awtextra.AbsoluteConstraints(200, 150, -1, -1));
47
48     jButton1.setFont(new java.awt.Font("Arial", 1, 22)); // NOI18N
49     jButton1.setText("TAMPIL");
50     jButton1.setPreferredSize(new java.awt.Dimension(200, 50));
51     mainPanel.add(jButton1, new org.netbeans.lib.awtextra.AbsoluteConstraints(350, 230, -1, -1));
52
53     getContentPane().add(mainPanel, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 200, 900, -1));
54
55     jLabel1.setFont(new java.awt.Font("Tahoma", 1, 32)); // NOI18N
56     jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
57     jLabel1.setText("Belajar GUI dengan JAVA Swing");
58     jLabel1.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
59     jLabel1.setPreferredSize(new java.awt.Dimension(600, 50));
60     getContentPane().add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 75, 900, -1));
61
62     pack();
63     setLocationRelativeTo(null);
64 }
```


Terima Kasih