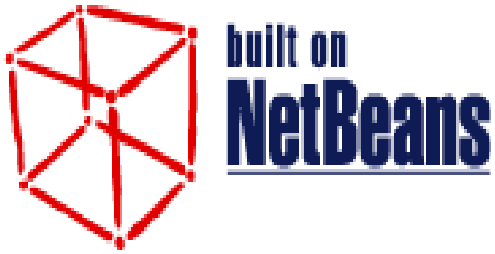
Pemrograman Berbasis Komponen

Java Beans + Swing di Netbeans



Java Beans

- Merupakan **component** pada Java
- Merupakan komponen perangkat lunak yang **reusable** dapat dimanipulasi secara visual menggunakan **builder tool**
- The JavaBeans API provides a *framework for defining reusable, embeddable, modular software components.*
- Memungkinkan pengguna untuk membangun aplikasi secara mudah

Java Beans

- **Visual bean**

  (mis. button, text-box)
- **Non-visual bean**

  (mis. FTP, SMTP, ZipCode validator)

A JavaBean is a "**Plain Old Java Object**" (POJO) that is serializable, has a no-argument constructor, and allows access to properties using getter and setter methods

Chilkat

Chilkat Class Libraries for Java

* Note: If the JDK is not specified, it is JDK6.

Windows Downloads

32-bit (win32) • v9.3.0 • 21-Jan-2012 • chilkatJava.zip (Windows Install Instructions)

64-bit (x64) • v9.3.0 • 21-Jan-2012 • chilkatJava64.zip (Windows Install Instructions)

Linux Downloads

32-bit • 21-Jan-2012 • chilkatJava-9.3.0-jdk6-x86-linux.tar.gz (Linux Install Instructions)

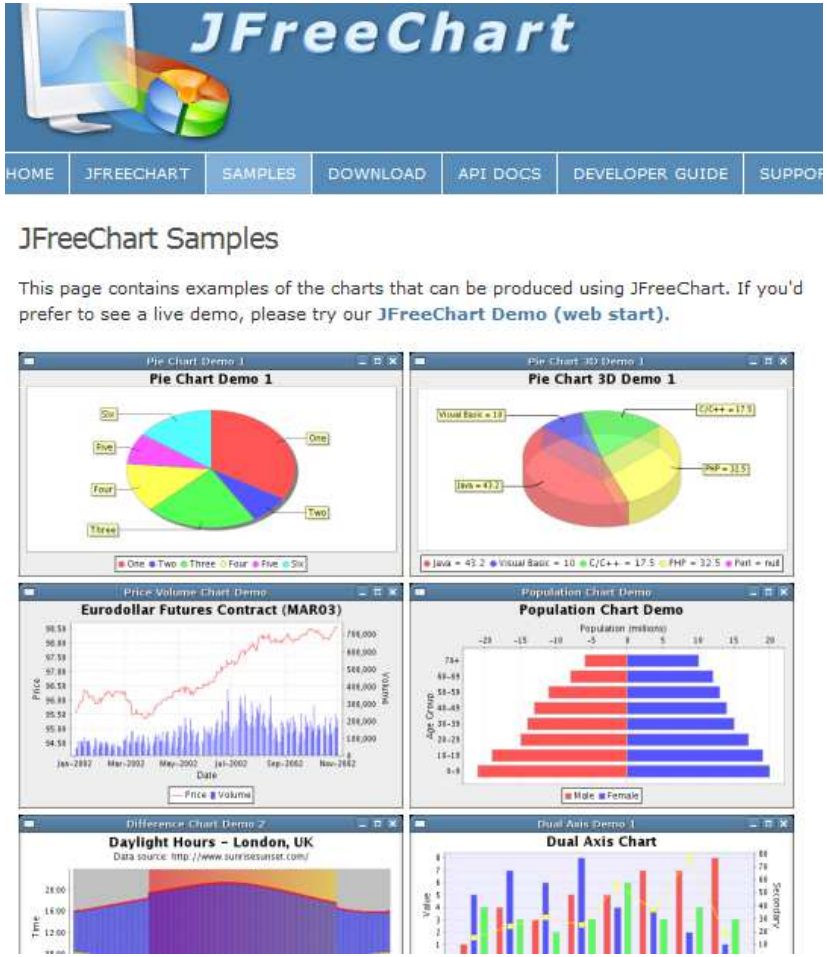
64-bit • 21-Jan-2012 • chilkatJava-9.3.0-jdk6-x86_64-linux.tar.gz (Linux Install Instructions)

MAC OS X Downloads

Standard (32/64-bit Universal, v10.6 Base SDK) • 21-Jan-2012
chilkatJava-9.3.0-macosx.tar.gz (MAC OS X Install Instructions)

Standard (32/64-bit Universal, v10.5 Base SDK) • 21-Jan-2012
chilkatJava-9.3.0-macosx-10.5.tar.gz (MAC OS X Install Instructions)

JFreeChart


JFreeChart
HOME
JFREECHART
SAMPLES
DOWNLOAD
API DOCS
DEVELOPER GUIDE
JFreeChart Samples
This page contains examples of the charts that can be produced using JFreeChart. If you'd prefer to see a live demo, please try our JFreeChart Demo (web start).
Pie Chart Demo 1
Pie Chart 3D Demo 1
Price Volume Chart Demo
Eurodollar Futures Contract (MAR03)
Population Chart Demo
Difference Chart Demo 2
Daylight Hours - London, UK
Dual Axis Demo 1
Dual Axis Chart

JustFormsPDF

JustFormsPDF
PDF Forms on the fly

home download examples docs faq

5 Stars Award
SoftsLand

LISTED WITH PROGRAMMERS HEAVEN.COM

DOWNLOAD FROM TOPSHAREWARE

Examples

Here are some examples, of using JustFormsPDF library in various situations, which can serve as kick starters.

Java Program

Source Code
Input/Template PDF
Output PDF

Servlet

Source Code
Input/Template PDF
Output PDF

Multiline Text

Source Code
Input/Template PDF
Output PDF

JavaBean Component Model

- Stateful
- Visual oriented
- Introspection
- Event driven
- Method naming
- Event naming
- BeanInfo class

JavaBeans

Lima fungsionalitas penting yang didukung oleh JavaBeans

- Event
- Property
- Persistence
  - Enable a bean to save and restore its state.
- Introspection: sifat-sifat suatu class
- Customization
  - Exposed properties could be modified at design time by a property editor or bean customizers.

Event

Apa itu **event**?

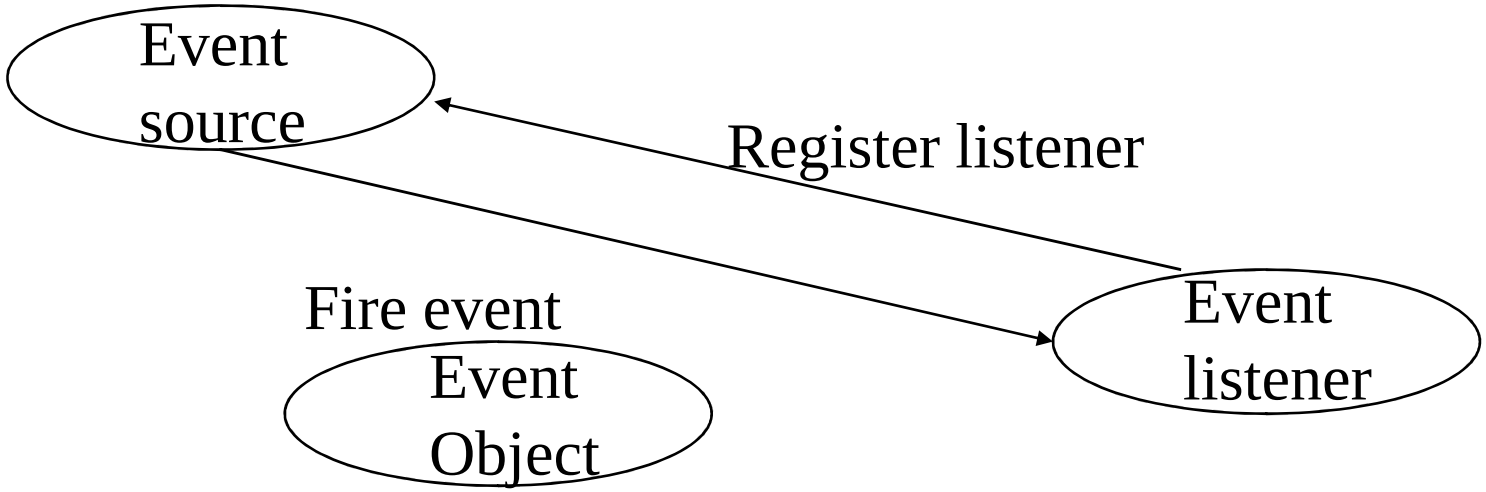
*A **source bean fires** an event, while a **listener bean receives** the event and responds to the event.*

- suatu message yang dikirim dari satu objek ke objek yang lain
- Pemberitahuan kepada penerima (recipient) bahwa telah terjadi ‘sesuatu’

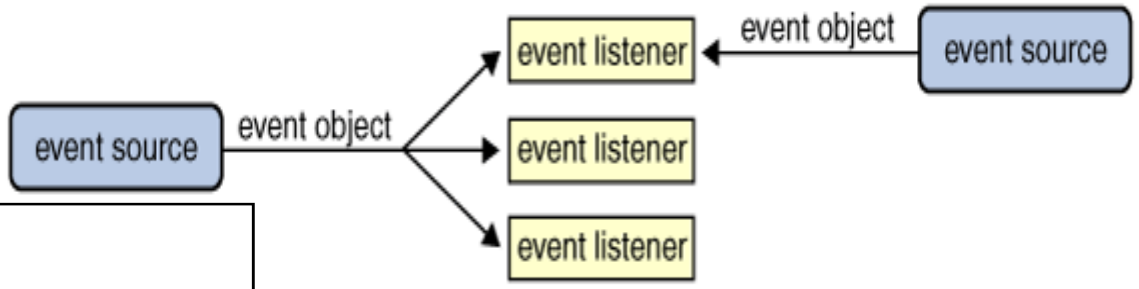
- Ada dua tipe event:
  - “Source” objects.
  - “Listener” objects.
- Berdasarkan **registrasi** -> observer

Event

- **Message** sent from one object to another.
- Sender *fires* event, recipient (listener) *handles* the event
- *There may be many listeners.*

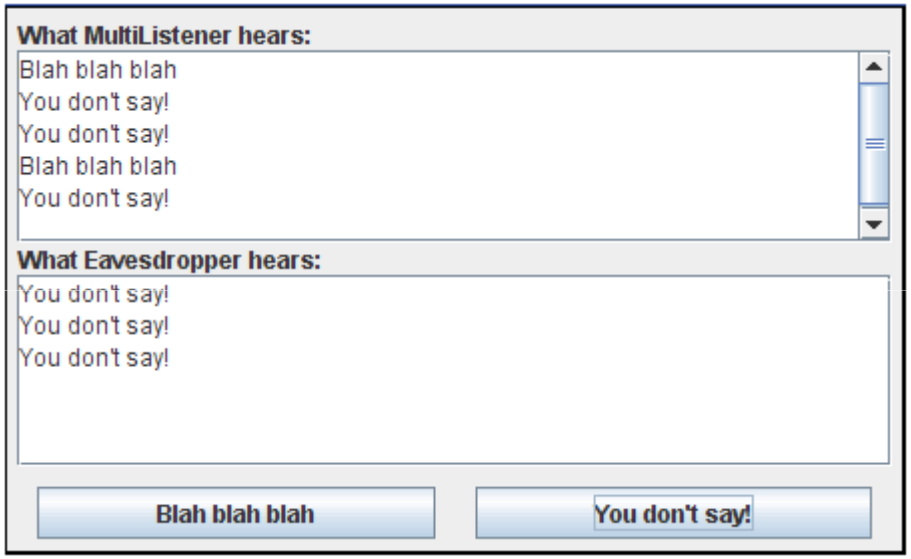


Multiple Listener




```
button1.addActionListener(this);
button2.addActionListener(this);

button2.addActionListener(new Eavesdropper(bottomTextArea));
```

```
class Eavesdropper implements ActionListener {
    JTextArea myTextArea;
    public Eavesdropper(JTextArea ta) {
        myTextArea = ta;
    }

    public void actionPerformed(ActionEvent e) {
        myTextArea.append(e.getActionCommand()
                          + MultiListener.newline);
        myTextArea.setCaretPosition(myTextArea.getDocument().getLength());
    }
}
```



Contoh Kode Listener

```
this.addKeyListener(new java.awt.event.KeyAdapter() {
    @Override
    public void keyTyped(java.awt.event.KeyEvent evt) {
        keyPress(evt);
    }
});
this.addFocusListener(new java.awt.event.FocusAdapter() {
    @Override
    public void focusLost(java.awt.event.FocusEvent evt) {
        lost(evt);
    }
});
```

Property

Apa itu **property** ?

Public attributes of a bean that affects its appearance or behavior

– Mendefinisikan karakteristik dari suatu bean

Contoh: bean NumericTextBox

- possible properties : maxValue, minValue, text

– dapat di : read/write, read-only atau write-only

Property

- Empat jenis property
  - **Simple Property**
    - Yang paling sederhana (tipe data **single** value)
  - **Indexed Property**
    - Suatu property tunggal dapat menyimpan **array** of values
  - **Bound property**
    - Memberi tahu *listeners* jika salah satu properti dari properti mengalami perubahan
  - **Constrained property**
    - Memungkinkan *listeners* untuk memilih constrain jika akan mengubah property

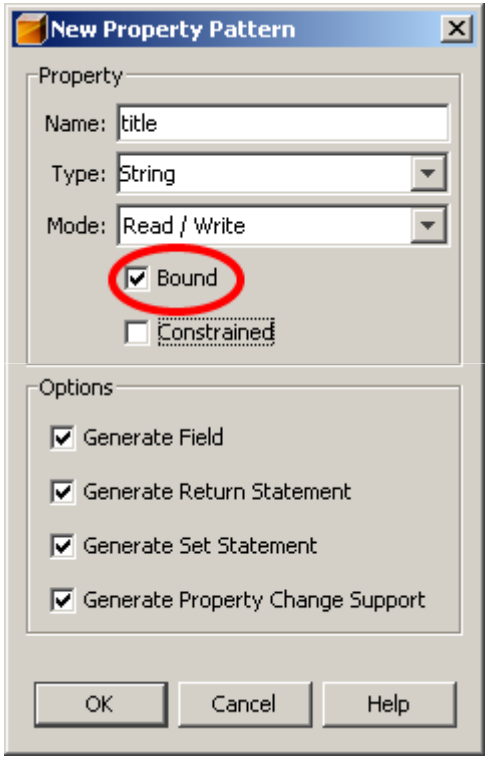
Contoh property

Constructors	A bean has a no argument constructor
Simple Properties	`public T getN()` `public void setN ( T value)`

Boolean Properties	`public boolean isN()`
	`public boolean getN()`
	`public void setN(boolean value)`

Indexed Properties	`public T getN(int index)`
	`public T[] getN()`
	`public void setN(int index, T value)`
	`public void setN(T[] values)`

Bound Property




```
private final PropertyChangeSupport pcs = new PropertyChangeSupport( this );

public String getTitle()
{
    return this.title;
}

public void setTitle( String title )
{
    String old = this.title;
    this.title = title;
    this.pcs.firePropertyChange( "title", old, title );
}
```

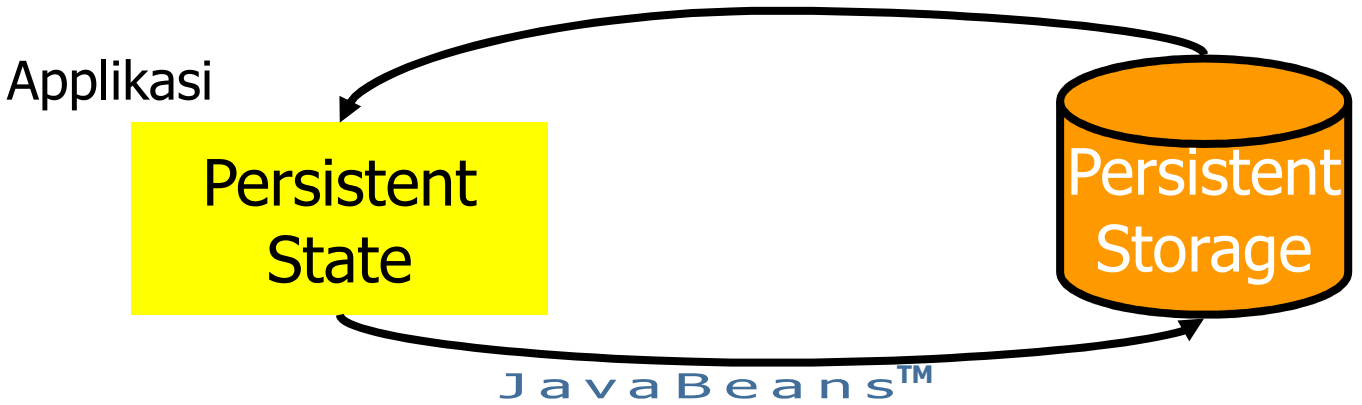
Persistence

Apa itu **persistence** ?

*Enable a bean to **save** and restore its state*

– development tool menyimpan JavaBean di dalam hard disk, dan dapat di-load suatu waktu

– memelihara nilai property tanpa tergantung apakah JavaBeans terdapat di memory atau hard disk



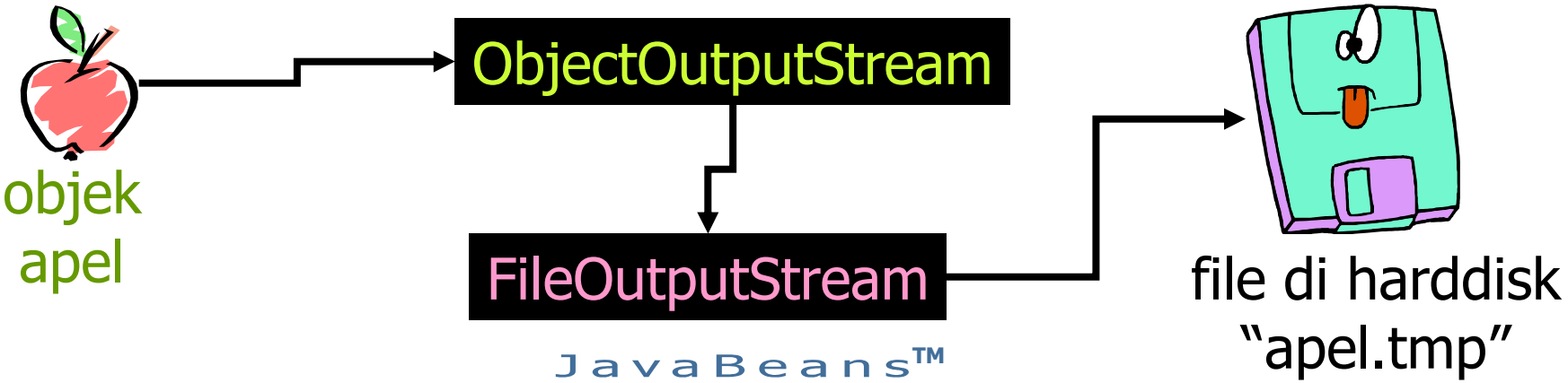
Object serialization

- Persistence dapat diperoleh dengan ***object serialization (***implements serializable***)***
  - Save semua content dari suatu object pada data stream
  - Generate kembali objek ketika membaca data dari data stream
- Selected property fields can bypass the serialization using keywords ***transient***

Object serialization

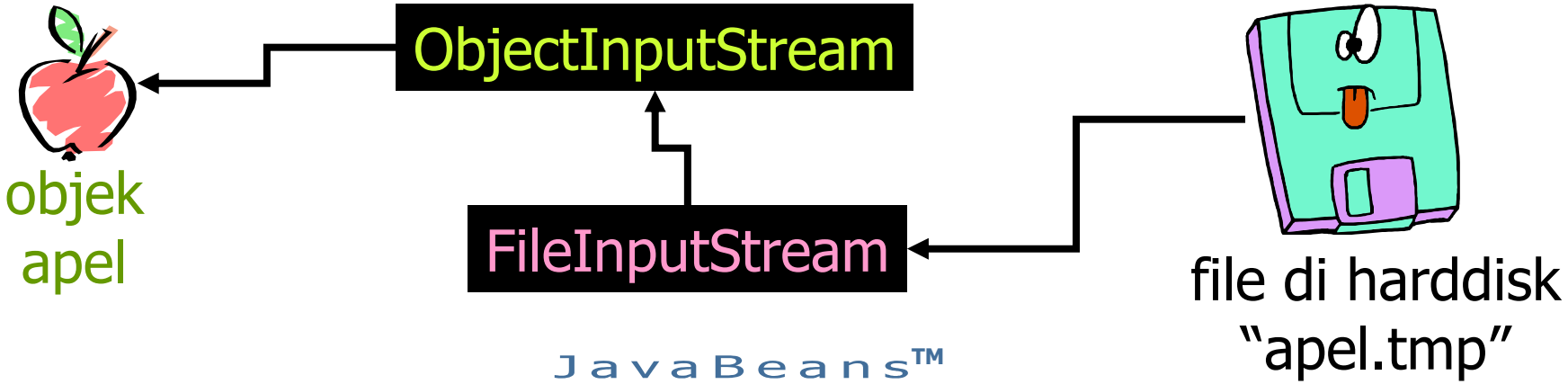
Contoh

```
Apel apel = new Apel(Color.green);
FileOutputStream f = new FileOutputStream("apel.tmp");
ObjectOutputStream s = new ObjectOutputStream(f);
s.writeObject(apel);
s.flush();
f.close();
```



Object serialization

```
FileInputStream f = new FileInputputStream("apel.tmp");
ObjectInputStream s = new ObjectInputStream(f);
Apel apel = (Apel)s.readObject();

f.close();
```



Contoh Lain: Pegawai

```
import java.io.*;

public class Pegawai implements Serializable {
	private String nama;
	private int umur;
	private int gaji;

	public Pegawai(String nama, int umur, int gaji) {
		this.nama = nama;
		this.umur = umur;
		this.gaji = gaji;
	}

	public void print() {
		System.out.println("Data untuk " + this.nama);
		System.out.println("nama " + this.nama);
		System.out.println("umur " + this.umur);
		System.out.println("gaji " + this.gaji);
	}
}
```

Contoh SimpanPegawai

```
import java.io.*;

public class SimpanPegawai {
  public static void main(String[] args) {
    Pegawai aaa = new Pegawai("aaa", 28, 100);
    Pegawai bbb = new Pegawai("bbb", 30, 150);

    try {
      FileOutputStream fos = new FileOutputStream("db");
      ObjectOutputStream oos = new ObjectOutputStream(fos);
      oos.writeObject(aaa);
      oos.writeObject(bbb);
      oos.flush();
    }
    catch (IOException e) {
      System.out.print("error " + e);
      System.exit(1);
    }
  }
}
```

Contoh: BacaPegawai

```
import java.io.*;

public class BacaPegawai {
   public static void main(String[] args) {
      try {
        FileInputStream fis = new FileInputStream("db");
        ObjectInputStream ois = new ObjectInputStream(fis);

        Pegawai aaa = (Pegawai) ois.readObject();
        Pegawai bbb = (Pegawai) ois.readObject();

        aaa.print();
        bbb.print();
      }
      catch (IOException e) {
        System.exit(1);
      }
      catch (Exception e) {
        System.exit(1);
      }
   }
}
```

```
Data untuk aaa
nama aaa
umur 28
gaji 100
Data untuk bbb
nama bbb
umur 30
gaji 150
```

Introspection

- Bagaimana development tools mengetahui method/event/property yang terdapat pada suatu beans?
  - **Introspection**
  - Java Reflection API (*java.lang.reflect*)
- Java Reflection API digunakan jika bean tidak mendukung introspection

Introspection

Apa itu **introspection**?

Information about the properties, events, and methods supported by a target Java Bean.

- Buat class XXXBeanInfo khusus untuk menjelaskan class XXX secara detil
- BeanInfo mendefinisikan informasi bean berikut:
  - Icon (displayed in development tool)
  - Property
  - Method
  - Other information

Java Reflection API

Apa itu **Java Reflection API** ?

A Java API for finding out the methods, fields, constructors, superclasses at ***RUNTIME***

- API ini juga digunakan untuk menulis development tools yang lain:
  - debuggers
  - class browsers
  - GUI builders

Java Reflection API

Contoh:

– Bagaimana mencari nama class suatu objek?

```
Button b        = new Button();
Class c         = b.getClass();
String s        = c.getName();
System.out.println(s);
```

– Bagaimana mencari superclass suatu class?

```
Button b        = new Button();
Class c         = b.getClass();
Class sc        = c.getSuperclass();
String s        = sc.getName();
System.out.println(s);
```

Java Reflection API

Contoh :

– Bagaimana mengetahui fields suatu objek?

```
Button b      = new Button();
Class c       = b.getClass();
Field[] f     = c.getFields();
System.out.println(f[0].getName() + f[0].getType());
```

– Bagaimana mengetahui method suatu objek?

```
Button b      = new Button();
Class c       = b.getClass();
Method[] m = c.getMethods();
System.out.println(m[0].getName() + m[0].getReturnType());
```

Java API yang dipakai pada JavaBean

- **The Java event model**: java.util.EventObject, java.awt.event
- **Object serialization**: java.io.Serializable, java.io.Object
- **Reflection**: java.lang.reflect

Contoh PersonBean

```java
public class PersonBean implements java.io.Serializable {

    private String name;

    private boolean deceased;

    /** No-arg constructor (takes no arguments). */
    public PersonBean() {
    }

    /**
     * Property <code>name</code> (note capitalization) readable/writable.
     */
    public String getName() {
        return this.name;
    }

    /**
     * Setter for property <code>name</code>.
     * @param name
     */
    public void setName(final String name) {
        this.name = name;
    }

    /**
     * Getter for property "deceased"
     * Different syntax for a boolean field (is vs. get)
     */
    public boolean isDeceased() {
        return this.deceased;
    }

    /**
     * Setter for property <code>deceased</code>.
     * @param deceased
     */
    public void setDeceased(final boolean deceased) {
        this.deceased = deceased;
    }
}
```

TestPersonBean

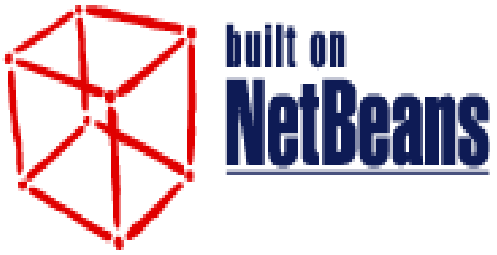
```java
/**
 * Class <code>TestPersonBean</code>.
 */
public class TestPersonBean {
    /**
     * Tester method <code>main</code> for class <code>PersonBean</code>.
     * @param args
     */
    public static void main(String[] args) {
        PersonBean person = new PersonBean();
        person.setName("Bob");
        person.setDeceased(false);

        // Output: "Bob [alive]"
        System.out.print(person.getName());
        System.out.println(person.isDeceased() ? " [deceased]" : " [alive]");
    }
}
```

Beberapa Hal yang diperhatikan

- All beans should **implement** the **Serializable** interface so that their state can be **saved** and later restored.
- Methods that are to be **exposed** to the builder tool and to other beans must bemade **public**.
- All exposed methods should be made **thread-safe** (possibly **synchronized**) to prevent more than one thread from calling a method at any given time.
- Properties are exposed through the use of public “set”and“get” methods.
  - Properties with no “set” method are **read-only**.
  - Properties with no “get” method are **write-only**.
- The “get” side of a boolean property may be exposed either through the use of a public “is” method or an ordinary “get” method.
- Events that the bean can multicast are exposed through public “add”and“remove” methods.

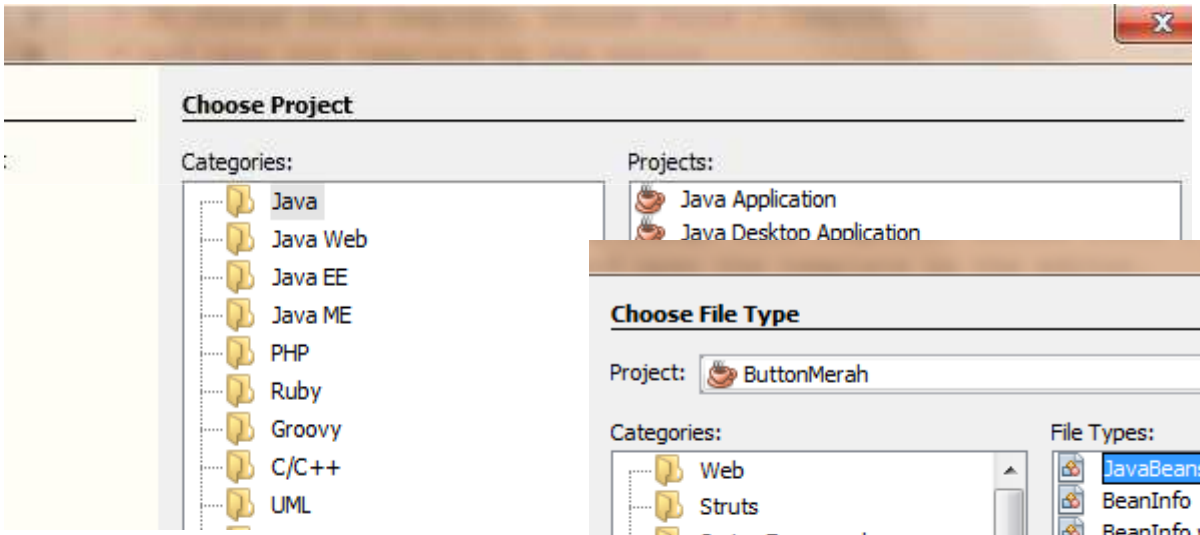
Komponen Bean pada Netbeans

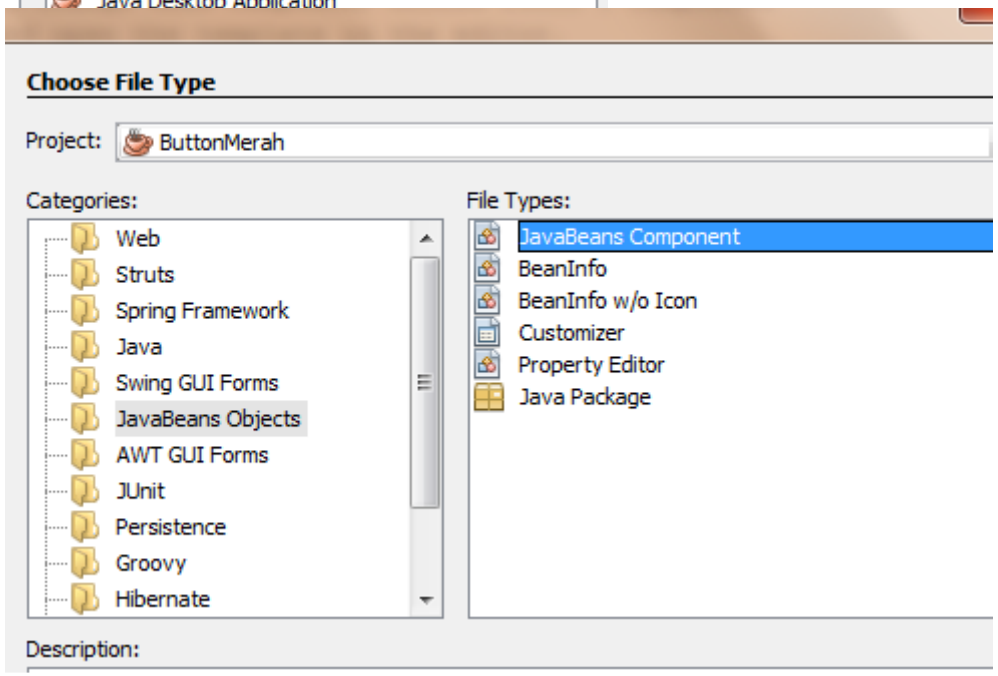
- Contoh kasus: TombolMerah
- Contoh kasus: NumericTextBox
- Contoh kasus: AdvancedEdit



TombolMerah

- Create New Project from Netbeans
  - Java Class Library
  - Beri nama ButtonMerah





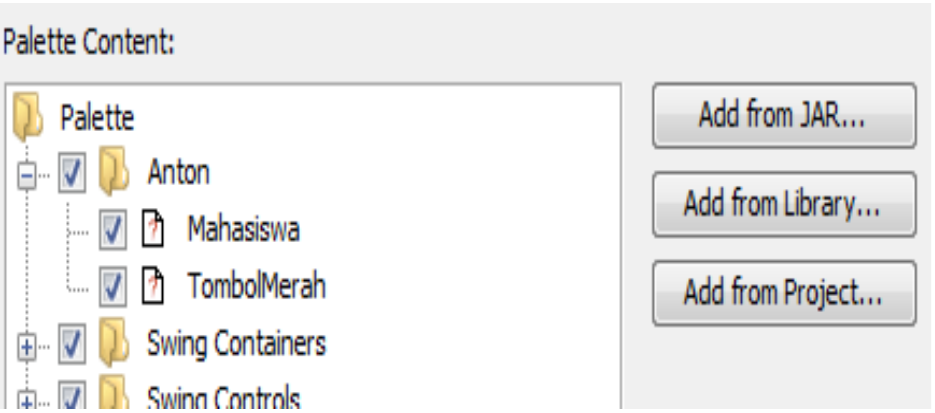
  - Tambahkan JavaBean

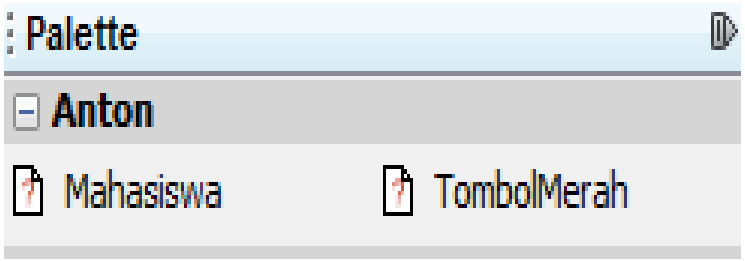
Source Code

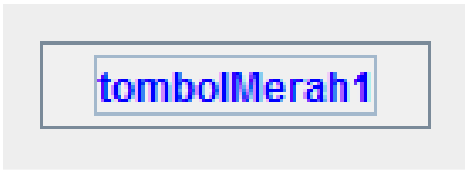
```
import java.awt.event.MouseEvent;
import java.io.Serializable;
import javax.swing.JButton;

public class TombolMerah extends JButton implements Serializable {
    public TombolMerah() {
        this.setForeground(Color.BLUE);
        this.addMouseListener(new java.awt.event.MouseAdapter() {
            @Override
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                MouseClicked(evt);
            }
        });
    }
    public void MouseClicked(MouseEvent evt){
        if(this.getForeground() == Color.BLUE){
            this.setForeground(Color.WHITE);
            this.setBackground(Color.RED);
        } else {
            this.setForeground(Color.BLUE);
            this.setBackground(new Color(240,240,240));
        }
    }
}
```

Tambahkan ke Pallete

- Dari Pallete > klik kanan, pilih Pallete Manager
- Buat New Category > misal Anton
  - From JAR, pilih JAR TombolMerah dari folder dist
  - Pilih komponen
  - Pilih kategori Anton



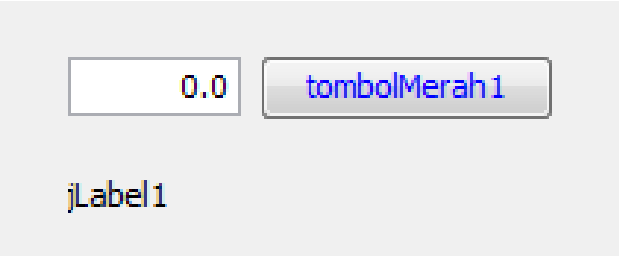




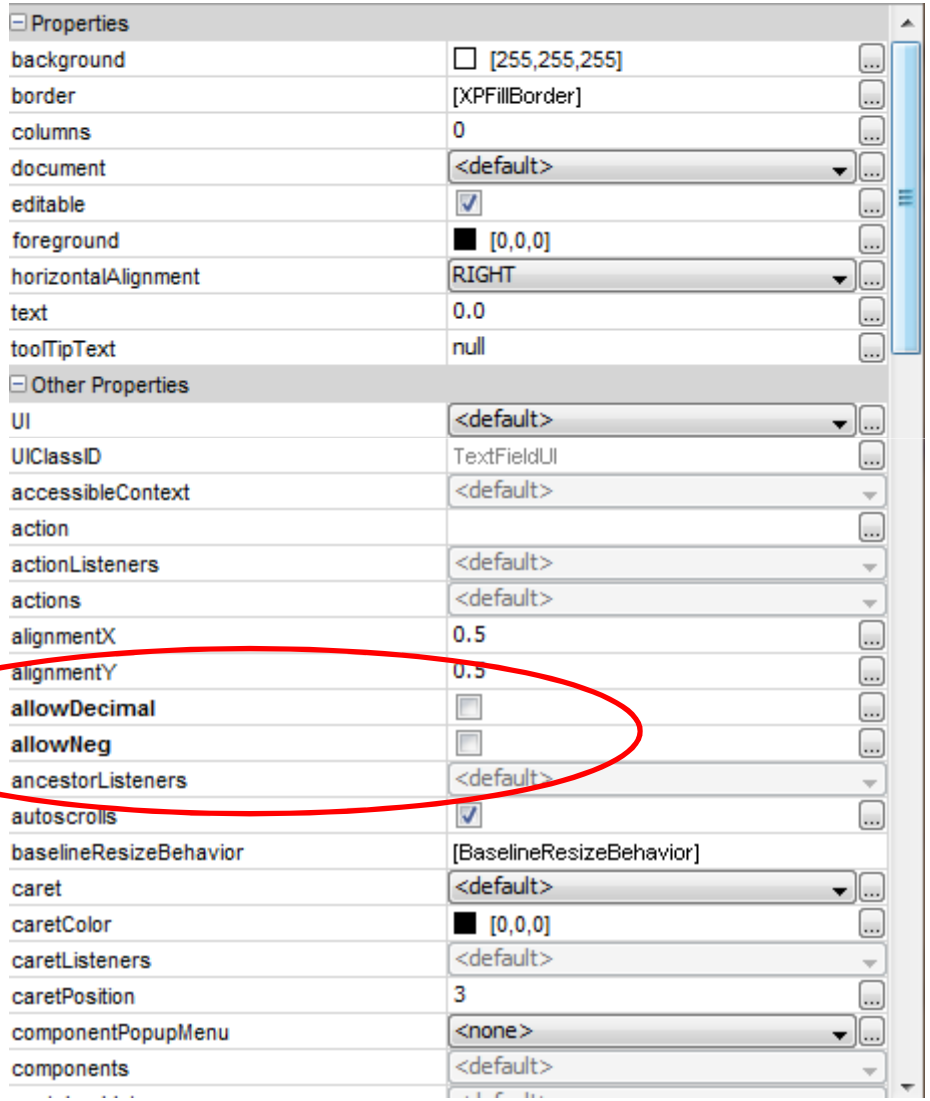
Numeric TextBox

- Extends dari **JTextField**
- Mampu menerima input berupa data angka (numeric) saja, plus minus, dan decimal

Numeric TextBox




```
import java.awt.Dimension;
import java.awt.event.KeyEvent;
import java.io.Serializable;
import javax.swing.JTextField;
```



Konstruktor

```
private boolean _AllowNeg = true;
private boolean _AllowDecimal = true;
private double _MaxValue;
private double _MinValue;
private int _intnilai = 0;
private double _dnilai = 0.0;

public NumericTextBox(){
    this.setText("0");
    setMinValue(Double.MIN_VALUE);
    setMaxValue(Double.MAX_VALUE);
    setIntnilai(0);
    setDnilai(0);
    this.setHorizontalAlignment(JTextField.RIGHT);
    this.addKeyListener(new java.awt.event.KeyAdapter() {
        @Override
        public void keyTyped(java.awt.event.KeyEvent evt) {
            XkeyTyped(evt);
        }
    });
    this.addFocusListener(new java.awt.event.FocusAdapter() {
        @Override
        public void focusLost(java.awt.event.FocusEvent evt) {
            lost(evt);
        }
    });
    this.setPreferredSize(new Dimension(59,20));
}
```

Event Keypress

```
private void XkeyTyped(KeyEvent evt) {
    if(Character.isDigit(evt.getKeyChar())){

    } else
    if(evt.getKeyChar() == '.' || evt.getKeyChar() == ','){
        if(isAllowDecimal() == true && this.getText().indexOf(DecimalSeparator()) == -1){
            evt.setKeyChar(DecimalSeparator());
        } else {
            evt.setKeyChar('\u0000');
        }
    } else {
        switch(evt.getKeyChar()){
            case '-':
                int p;
                if(isAllowNeg()==true){
                    p = this.getSelectionStart();
                    System.out.println(p);
                    setDnilai(-1*getDnilai());
                    if(getDnilai() > 0)
                        this.setSelectionStart(p-1);
                    else
                        this.setSelectionStart(p+1);
                    evt.setKeyChar('\u0000');
                } else {
                    evt.setKeyChar('\u0000');
                }
                break;
            case '\u0008':
                break;
            default: evt.setKeyChar('\u0000');
        }
    }
}
```

AdvancedEdit

- Sama dengan VB.NET, sudah memiliki fitur aligment, yaitu diset dari property **horizontalAligment = left, right, center**
- Extends dari JTextField
- Variable
  - private Color FOldBackColor;
  - private Color FColorOnEnter;
  - private boolean FTabOnEnter;
- Imports:
  - import java.awt.Color;
  - import java.awt.Dimension;
  - import java.awt.event.KeyEvent;
  - import javax.swing.*;

Class AdvancedEdit

```
import java.awt.Color;
import java.awt.Dimension;
import java.awt.event.KeyEvent;
import javax.swing.*;

public class AdvancedEdit extends JTextField {
    private Color FOldBackColor;
    private Color FColorOnEnter;
    private boolean FTabOnEnter;

    public AdvancedEdit(){
        this.setFColorOnEnter(this.getBackground());
        this.setPreferredSize(new Dimension(59,20));
        this.addKeyListener(new java.awt.event.KeyAdapter() {
            @Override
            public void keyPressed(java.awt.event.KeyEvent evt) {
                OnKeyPress(evt);
            }
        });
        this.addFocusListener(new java.awt.event.FocusAdapter() {
            @Override
            public void focusLost(java.awt.event.FocusEvent evt) {
                OnLostFocus(evt);
            }
            @Override
            public void focusGained(java.awt.event.FocusEvent evt){
                OnFocus(evt);
            }
        });
    }
```

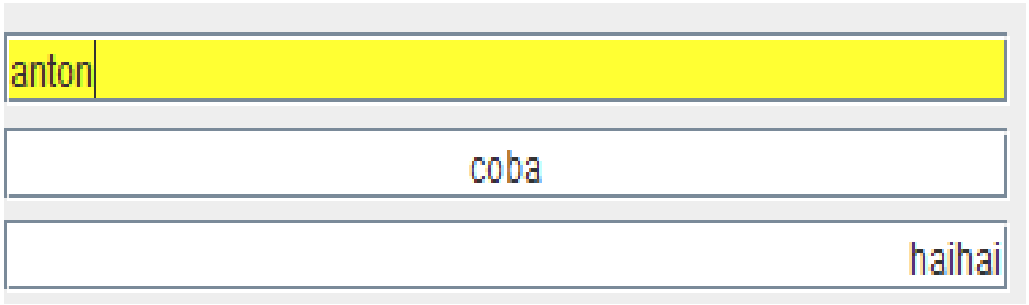
Setter dan getter

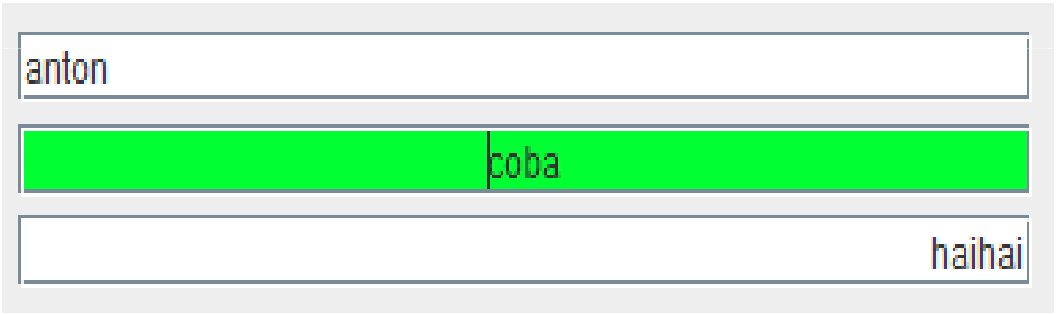
```
public Color getFOldBackColor() {
    return FOldBackColor;
}

public void setFOldBackColor(Color FOldBackColor) {
    this.FOldBackColor = FOldBackColor;
}

public Color getFColorOnEnter() {
    return FColorOnEnter;
}

public void setFColorOnEnter(Color FColorOnEnter) {
    this.FColorOnEnter = FColorOnEnter;
}

public boolean isFTabOnEnter() {
    return FTabOnEnter;
}

public void setFTabOnEnter(boolean FTabOnEnter) {
    this.FTabOnEnter = FTabOnEnter;
}
```

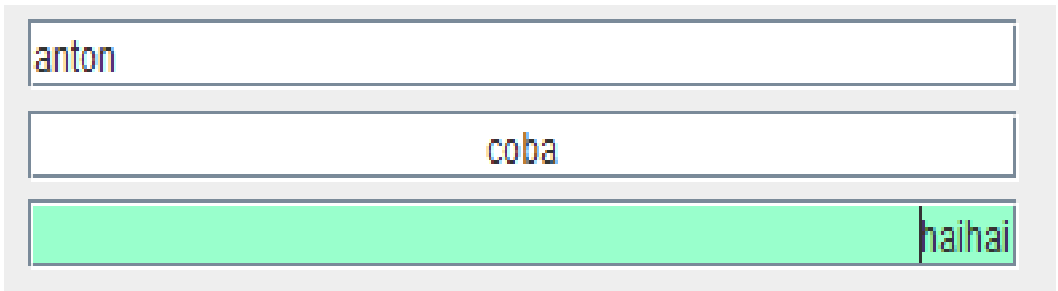
Method inti

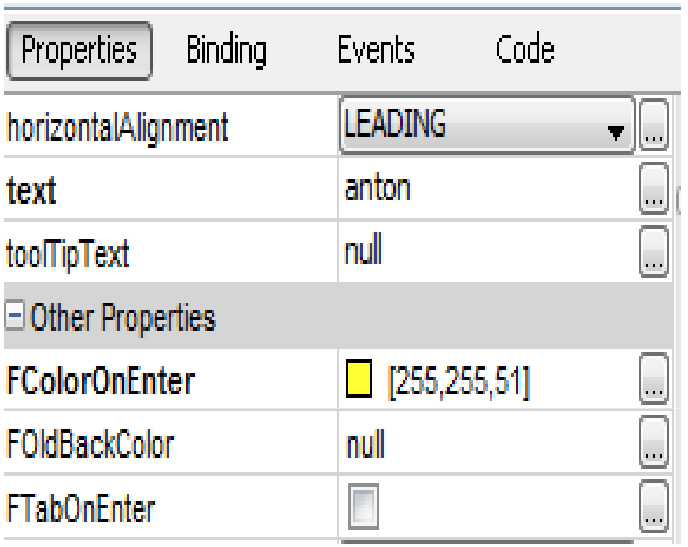
```
public void OnFocus(java.awt.event.FocusEvent evt){
    this.setFOldBackColor(this.getBackground());
    this.setBackground(this.getFColorOnEnter());
}

public void OnLostFocus(java.awt.event.FocusEvent evt){
    this.setBackground(this.getFOldBackColor());
}

public void OnKeyPress(KeyEvent evt){
    int key = evt.getKeyCode();
    if (key == KeyEvent.VK_ENTER)
        if(this.isFTabOnEnter())
            transferFocus();
}
```

Hasil


anton
coba
haihai


anton
coba
haihai


anton
coba
haihai


Properties
Binding
Events
Code
horizontalAlignment
LEADING
text
anton
toolTipText
null
Other Properties
FColorOnEnter
[255,255,51]
FOldBackColor
null
FTabOnEnter

Next

- TTS
- Web Component