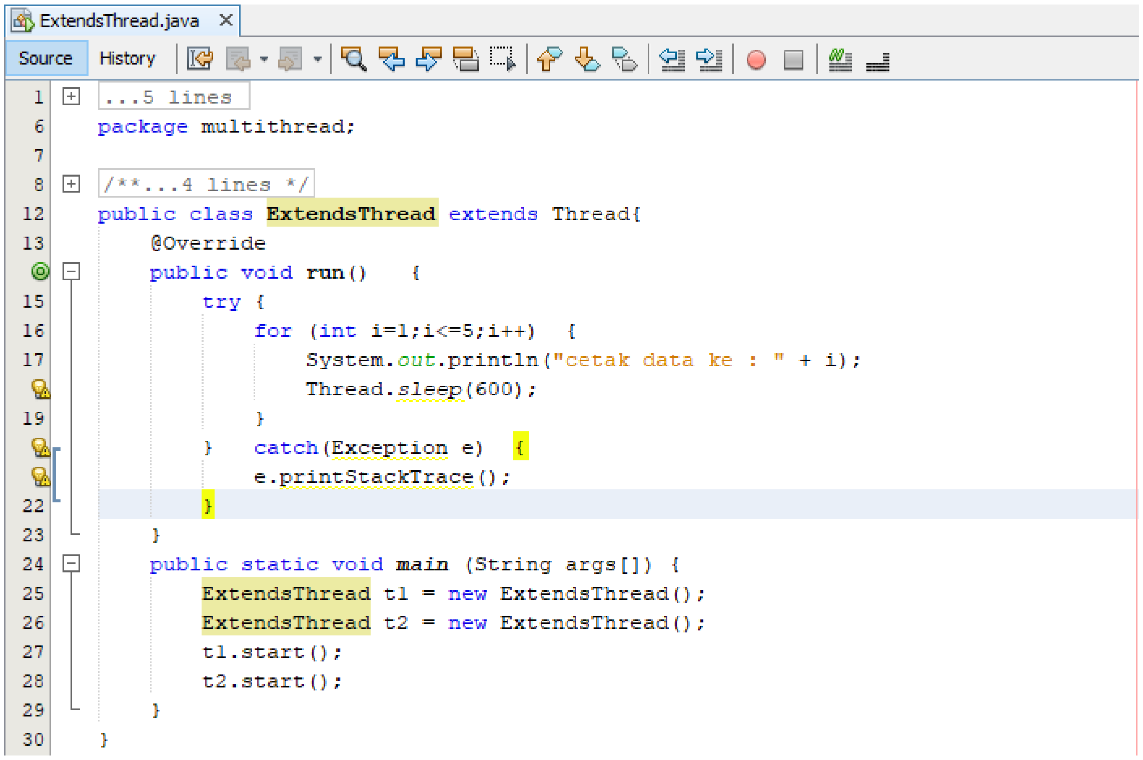
Nama : Alya Aiman Salsabila Arif

NPM : 1817101379

Kelas : III Rekayasa Perangkat Lunak Kripto

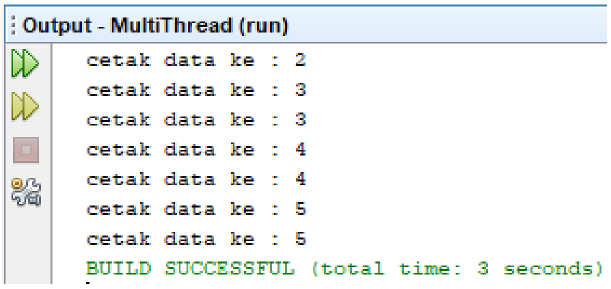
Tugas 10

1. Implementasi *Thread* dengan Kelas Thread



ExtendsThread.java

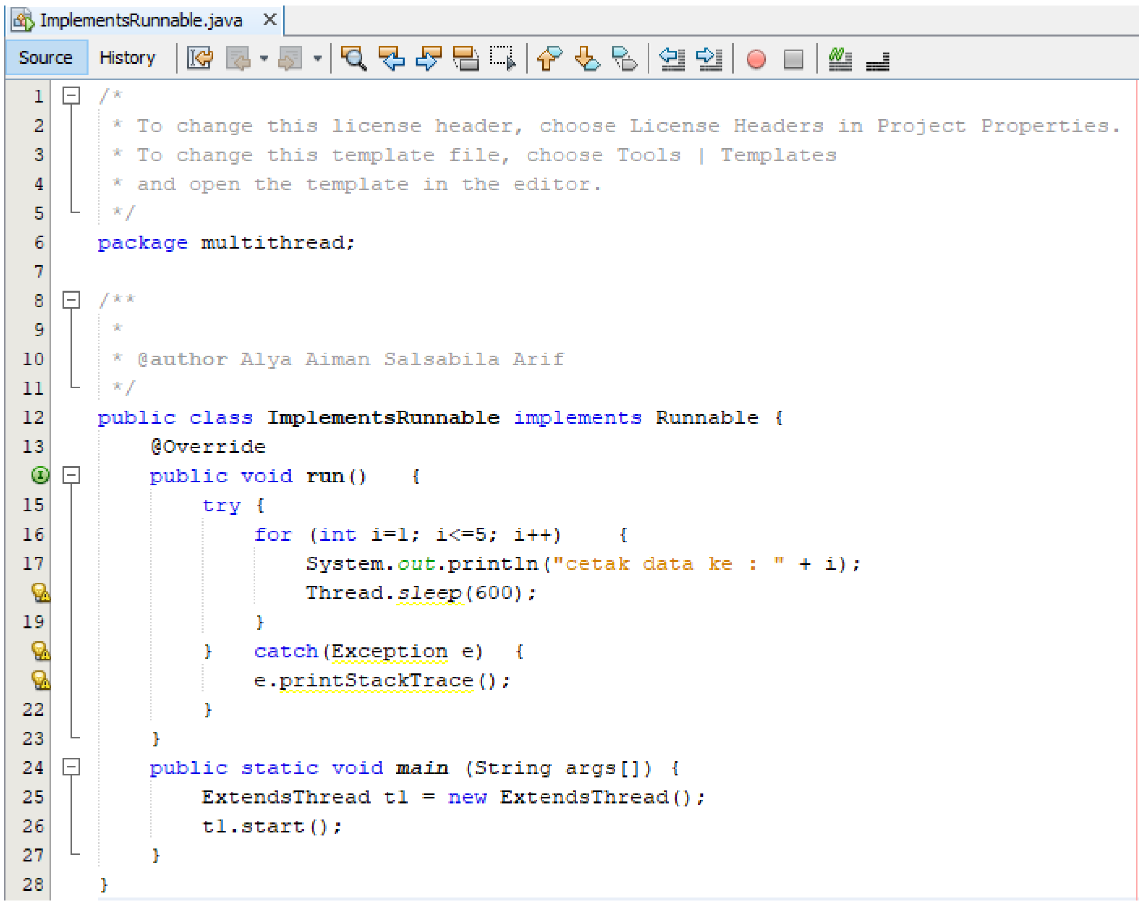
```
...5 lines
package multithread;

/**...4 lines */
public class ExtendsThread extends Thread{
    @Override
    public void run()   {
        try {
            for (int i=1;i<=5;i++)  {
                System.out.println("cetak data ke : " + i);
                Thread.sleep(600);
            }
        }   catch(Exception e)  {
            e.printStackTrace();
        }
    }
    public static void main (String args[]) {
        ExtendsThread t1 = new ExtendsThread();
        ExtendsThread t2 = new ExtendsThread();
        t1.start();
        t2.start();
    }
}
```



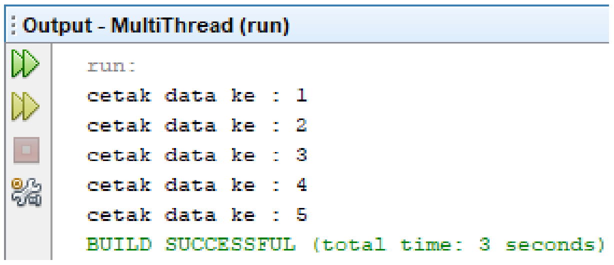
Output - MultiThread (run)

```
cetak data ke : 2
cetak data ke : 3
cetak data ke : 3
cetak data ke : 4
cetak data ke : 4
cetak data ke : 5
cetak data ke : 5
BUILD SUCCESSFUL (total time: 3 seconds)
```

2. Implementasi *Thread* dengan *Interface Runnable*

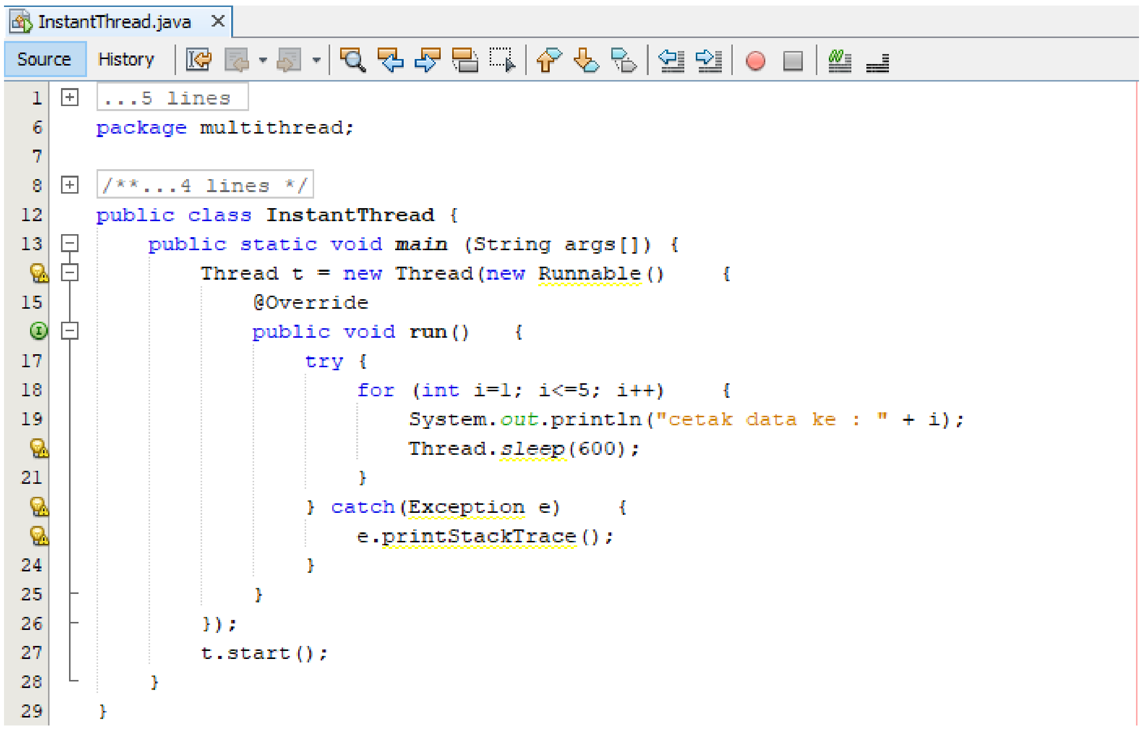


```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package multithread;

/**
 *
 * @author Alya Aiman Salsabila Arif
 */
public class ImplementsRunnable implements Runnable {
    @Override
    public void run()   {
        try {
            for (int i=1; i<=5; i++)    {
                System.out.println("cetak data ke : " + i);
                Thread.sleep(600);
            }
        }   catch(Exception e)  {
            e.printStackTrace();
        }
    }
    public static void main (String args[]) {
        ExtendsThread t1 = new ExtendsThread();
        t1.start();
    }
}
```

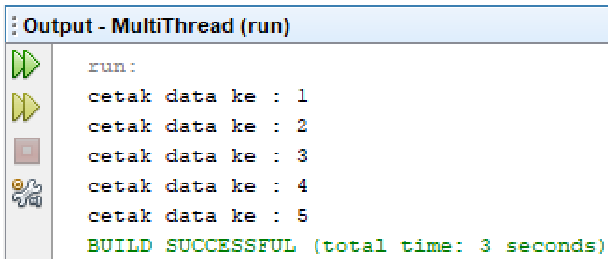


```
run:
cetak data ke : 1
cetak data ke : 2
cetak data ke : 3
cetak data ke : 4
cetak data ke : 5
BUILD SUCCESSFUL (total time: 3 seconds)
```

3. Implementasi *Thread* dengan Membuat *Instance* dari Kelas Thread



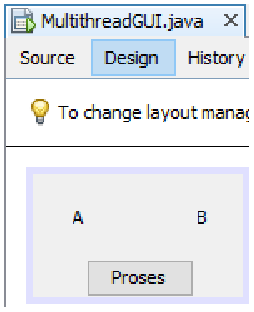
```
...5 lines
package multithread;

/**...4 lines */
public class InstantThread {
    public static void main (String args[]) {
        Thread t = new Thread(new Runnable()    {
            @Override
            public void run()   {
                try {
                    for (int i=1; i<=5; i++)    {
                        System.out.println("cetak data ke : " + i);
                        Thread.sleep(600);
                    }
                } catch(Exception e)    {
                    e.printStackTrace();
                }
            }
        });
        t.start();
    }
}
```



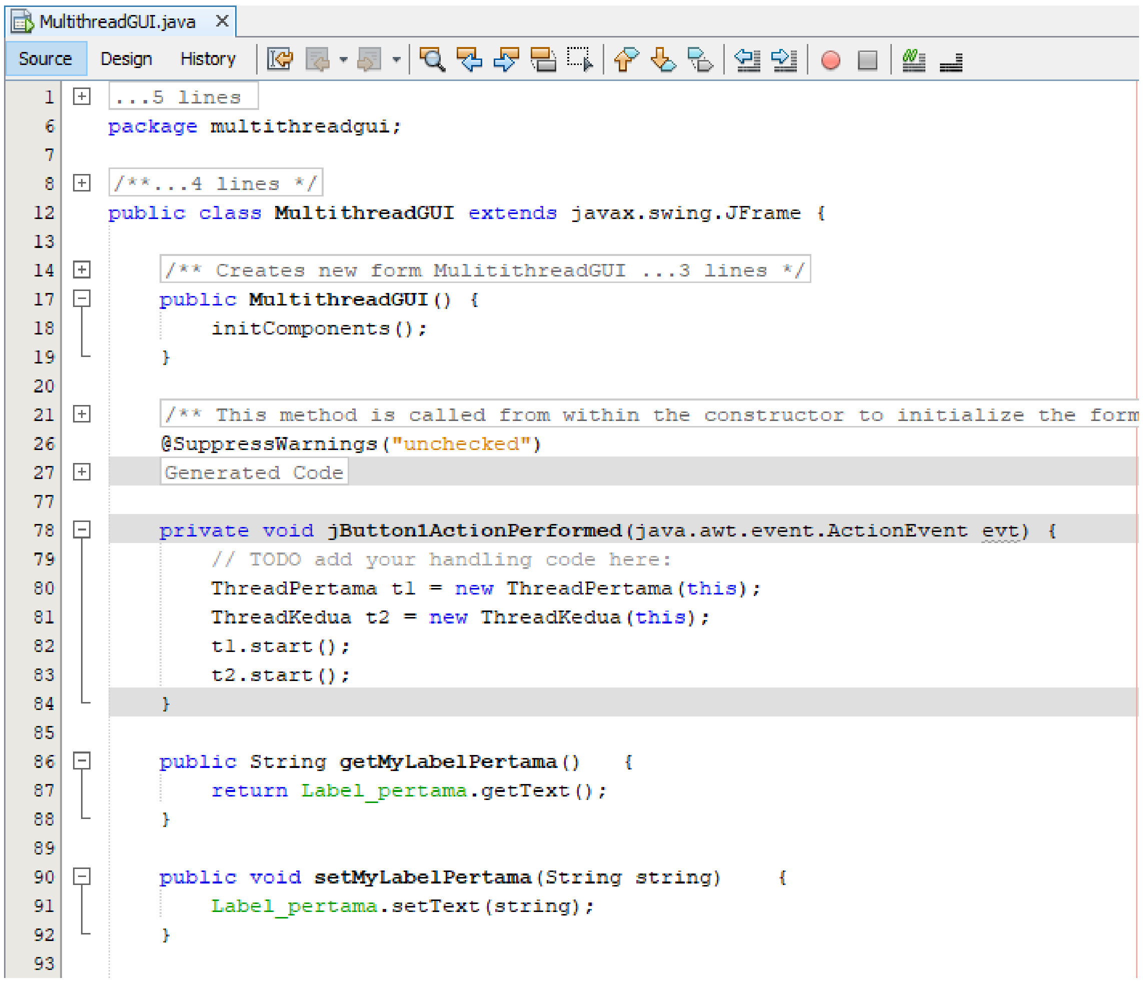
```
run:
cetak data ke : 1
cetak data ke : 2
cetak data ke : 3
cetak data ke : 4
cetak data ke : 5
BUILD SUCCESSFUL (total time: 3 seconds)
```

4. Praktikum

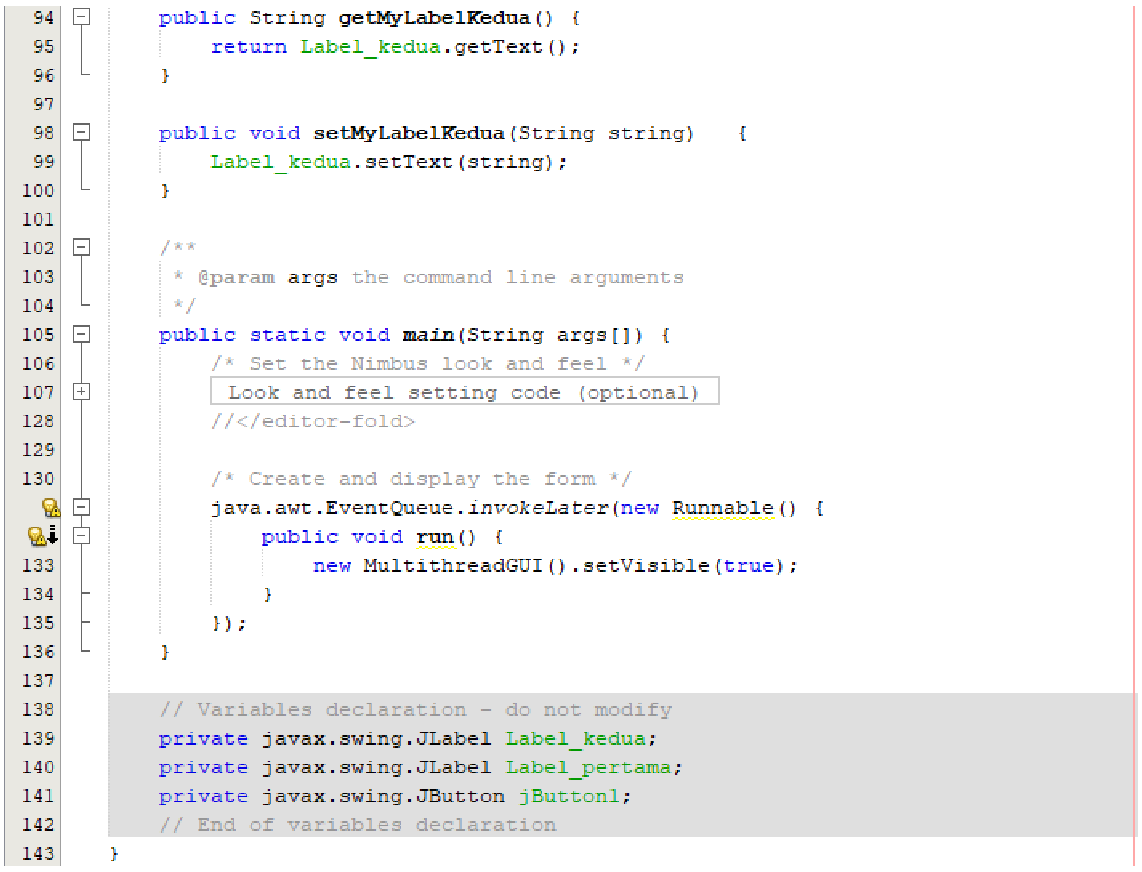
Buatlah projek baru dengan nama MultithreadGUI. Buatlah kelas MultithreadGUI dengan JFrame Form. Buatlah desain kelas MultithreadGUI



Masukkan *source code* berikut pada kelas MultithreadGUI

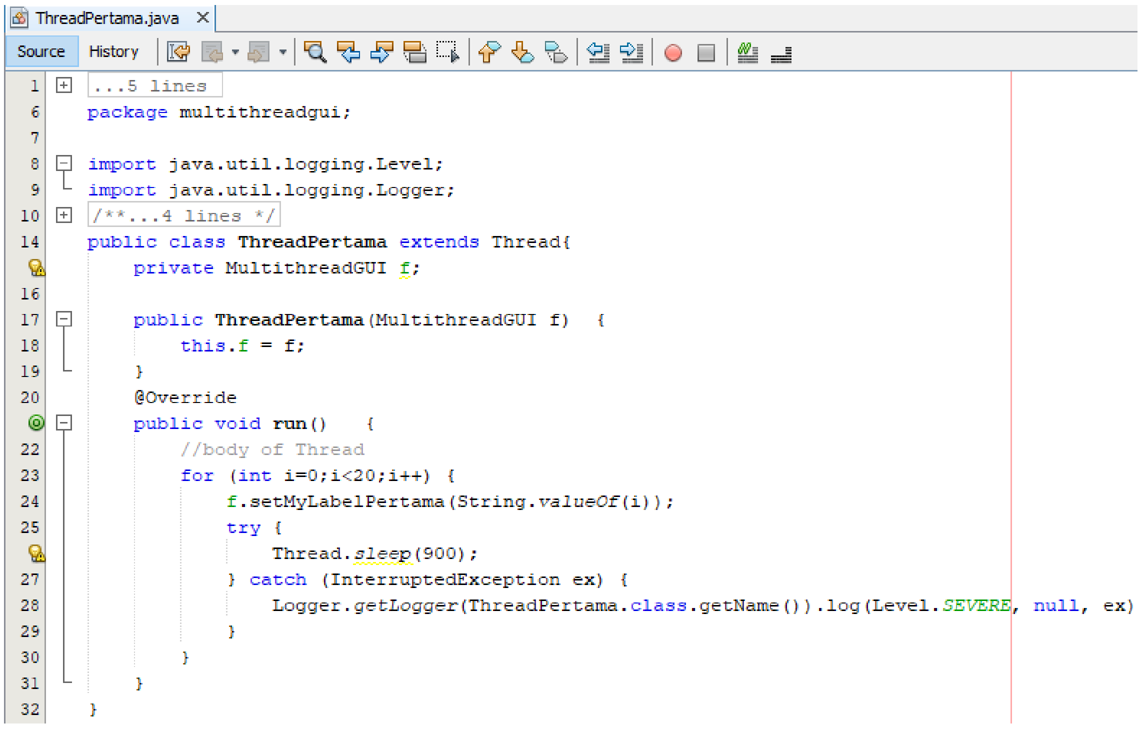


```
...5 lines
package multithreadgui;

/**...4 lines */
public class MultithreadGUI extends javax.swing.JFrame {

    /** Creates new form MulitithreadGUI ...3 lines */
    public MultithreadGUI() {
        initComponents();
    }

    /** This method is called from within the constructor to initialize the form
    @SuppressWarnings("unchecked")
    Generated Code

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        ThreadPertama t1 = new ThreadPertama(this);
        ThreadKedua t2 = new ThreadKedua(this);
        t1.start();
        t2.start();
    }

    public String getMyLabelPertama()   {
        return Label_pertama.getText();
    }

    public void setMyLabelPertama(String string)    {
        Label_pertama.setText(string);
    }

```



```
    public String getMyLabelKedua() {
        return Label_kedua.getText();
    }

    public void setMyLabelKedua(String string)    {
        Label_kedua.setText(string);
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        Look and feel setting code (optional)
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new MultithreadGUI().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JLabel Label_kedua;
    private javax.swing.JLabel Label_pertama;
    private javax.swing.JButton jButton1;
    // End of variables declaration
}
```

Buat kelas ThreadPertama dan masukkan *source code* sebagai berikut

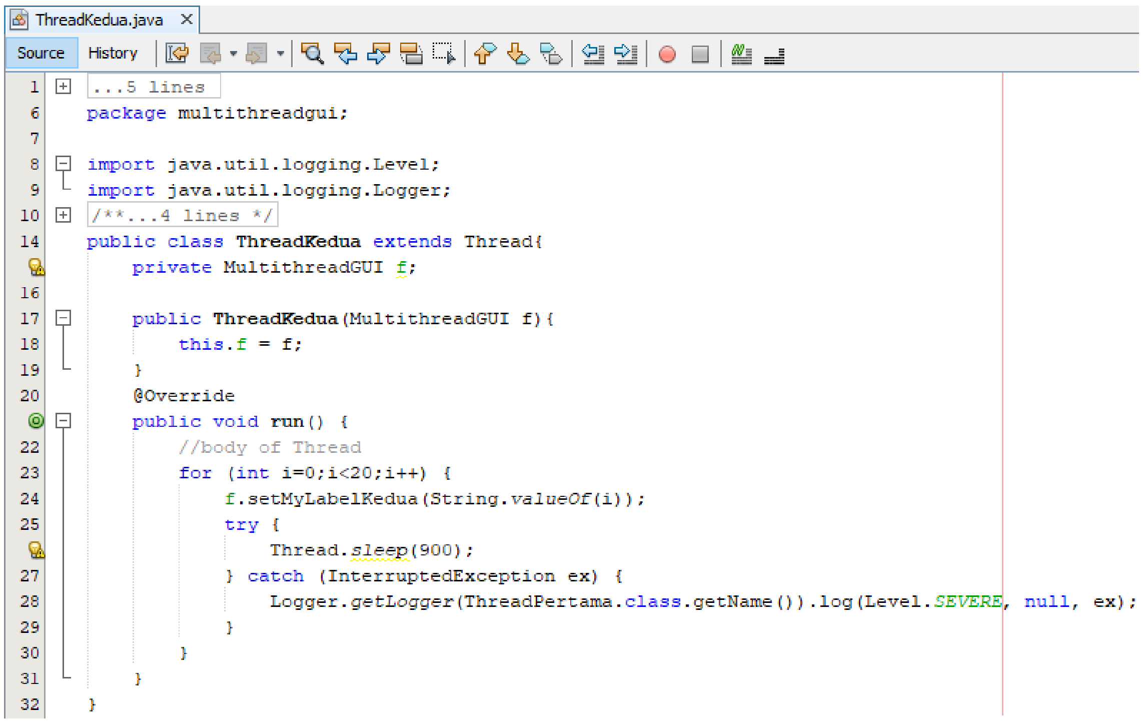


ThreadPertama.java

Source History

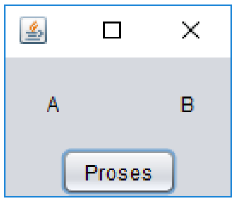
```
...5 lines
package multithreadgui;

import java.util.logging.Level;
import java.util.logging.Logger;
/**...4 lines */
public class ThreadPertama extends Thread{
    private MultithreadGUI f;

    public ThreadPertama(MultithreadGUI f)  {
        this.f = f;
    }
    @Override
    public void run()   {
        //body of Thread
        for (int i=0;i<20;i++) {
            f.setMyLabelPertama(String.valueOf(i));
            try {
                Thread.sleep(900);
            } catch (InterruptedException ex) {
                Logger.getLogger(ThreadPertama.class.getName()).log(Level.SEVERE, null, ex)
            }
        }
    }
}
```

Buat kelas ThreadKedua dan masukkan *source code* sebagai berikut



```
...5 lines
package multithreadgui;

import java.util.logging.Level;
import java.util.logging.Logger;
/**...4 lines */
public class ThreadKedua extends Thread{
    private MultithreadGUI f;

    public ThreadKedua(MultithreadGUI f){
        this.f = f;
    }
    @Override
    public void run() {
        //body of Thread
        for (int i=0;i<20;i++) {
            f.setMyLabelKedua(String.valueOf(i));
            try {
                Thread.sleep(900);
            } catch (InterruptedException ex) {
                Logger.getLogger(ThreadPertama.class.getName()).log(Level.SEVERE, null, ex);
            }
        }
    }
}
```

Jalankan program. Program akan menampilkan sebagai berikut



Tekan tombol *button* Proses. Program akan menampilkan angka perulangan secara bersamaan sesuai perulangan yang dibuat pada kelas ThreadPertama dan kelas ThreadKedua