

## Praktikum 12 – SwingWorker

Nama : Cania Nabilatul Adawah

NIM : 2403102

Kelas : D3TI2C

Mata Kuliah : Pemrograman Berbasis Objek

Sajikanlah Laporan Praktikum 12 - SwingWorker dengan format (\*.pdf) dan ketentuan sebagai berikut:

1. Terdapat Ringkasan materi pada Modul 12 – SwingWorker

### 1. Pendahuluan: Masalah Dasar di Aplikasi Swing

Swing berjalan dengan Event Dispatch Thread (EDT) sebagai pusat:

- menggambar UI,
- memproses interaksi user (event),
- memperbarui komponen UI.

EDT bersifat single-thread, sehingga hanya satu task dapat berjalan pada satu waktu. Jika ada operasi berat dijalankan di EDT, maka:

- Frame tidak bisa di-resize,
- Tombol tidak bisa ditekan,
- UI tidak merespons,
- Aplikasi tampak not responding.

### 2. Concurrency (Konkurensi) — Konsep Teoritis

Concurrency adalah kemampuan sistem untuk:

mengerjakan lebih dari satu task dalam periode waktu yang sama.

Tidak berarti task benar-benar paralel secara fisik, tetapi sistem dapat membuat kemajuan pada beberapa task secara simultan. Ini dapat terjadi melalui:

#### A. True Parallelism

- Eksekusi fisik terjadi bersamaan pada core CPU yang berbeda.

- Hanya terjadi pada CPU multi-core.

#### B. Overlapping

- Sebagian waktu task A menunggu (misal I/O),
- CPU menjalankan task B,
- Ada tumpang tindih waktu eksekusi.

#### C. Interleaving

- CPU membagi waktu menjadi slot kecil (time slicing),
- Task A → Task B → Task C → kembali ke A,
- Terlihat seolah paralel.

### 3. Mekanisme Mencapai Concurrency

#### 1. Multithreading

- Satu proses memiliki banyak thread.
- Paling umum di Java.

#### 2. Multiprocessing

- Banyak proses terpisah dan independen.

#### 3. Single-threaded concurrency

- Menggunakan event loop (mirip JavaScript).

### 4. Asynchronous Programming

Teknik menjalankan task berat tanpa memblokir alur utama (EDT).

Ciri-ciri:

- Task berat dimulai,
- Program langsung melanjutkan task lain,
- Tidak menunggu task berat selesai,
- Hasil dikembalikan ketika sudah siap.

Dalam konteks Swing:

- Task berat berjalan di background thread,
- Update UI dilakukan di EDT.

## 5. Concurrency pada Swing

### 5.1. Masalahnya

EDT tidak boleh digunakan untuk:

- komputasi berat,
- looping panjang,
- sleep,
- akses database,
- I/O jaringan.

Semua operasi ini memblokir EDT → UI freeze.

### 5.2. Solusi: SwingWorker

SwingWorker digunakan untuk:

- menjalankan task berat di background thread,
- memberikan mekanisme aman untuk update UI di EDT,
- mencegah freeze,
- tetap mematuhi aturan single-thread pada Swing.

## 6. SwingWorker — Analisis Kelas

Struktur Generic <T, V>

SwingWorker menggunakan dua parameter generic:

T (Return Type)

- Hasil akhir dari doInBackground(),
- Dikembalikan ke done() melalui get().

V (Intermediate Progress Type)

- Data progress sementara,
- Dikirim dari publish() ke process().

Jika tidak membutuhkan progress, gunakan tipe Void.

## 7. Metode-Metode Utama SwingWorker

Modul menjelaskan 6 metode utama:

1. doInBackground()

- Tempat menulis task berat.
- Dieksekusi di worker thread.
- Harus mengembalikan nilai bertipe T.
- Bisa berisi:
  - komputasi berat,
  - sleep,
  - operasi network,
  - akses database.

2. publish(V...)

- Dipanggil di dalam doInBackground().
- Mengirimkan data progress ke EDT.

3. process(List chunks)

- Dipanggil oleh EDT.
- Menerima daftar progress dari publish().
- Digunakan untuk update UI secara bertahap:
  - progress bar,
  - label status,
  - text area.

4. done()

- Dipanggil setelah doInBackground() selesai:
  - sukses,
  - gagal,
  - atau dibatalkan.
- Dieksekusi di EDT.
- Tempat memanggil get() untuk mengambil hasil akhir T.

5. get()

- Mengambil nilai return dari doInBackground().

- Biasanya dipanggil di done().
- Meng-handle:
  - ExecutionException,
  - InterruptedException,
  - CancellationException.

#### 6. execute()

- Memulai worker thread.
- Tanpa ini, SwingWorker tidak berjalan.

### 8. Contoh Concurrency Menggunakan SwingWorker (Demo 2–4)

Modul memberikan 3 cara implementasi:

#### 8.1. Anonymous Inner Class

- Paling sederhana.
- SwingWorker ditulis langsung dalam event listener.
- Cocok untuk contoh kecil.

#### 8.2. Static Nested Class

- SwingWorker ditulis sebagai inner class statis.
- Memisahkan kode background dari UI.
- Lebih terstruktur dibanding anonymous.

#### 8.3. Separate Top-Level Class

- SwingWorker ditempatkan di file terpisah.
- Pendekatan paling baik.
- Arsitektur paling rapi dan mudah dirawat.

### 9. Cancelation pada SwingWorker

SwingWorker mendukung pembatalan task berat.

Untuk membatalkan:

```
worker.cancel(true);
```

Di dalam doInBackground() harus ada:

```
if (isCancelled()) break;
```

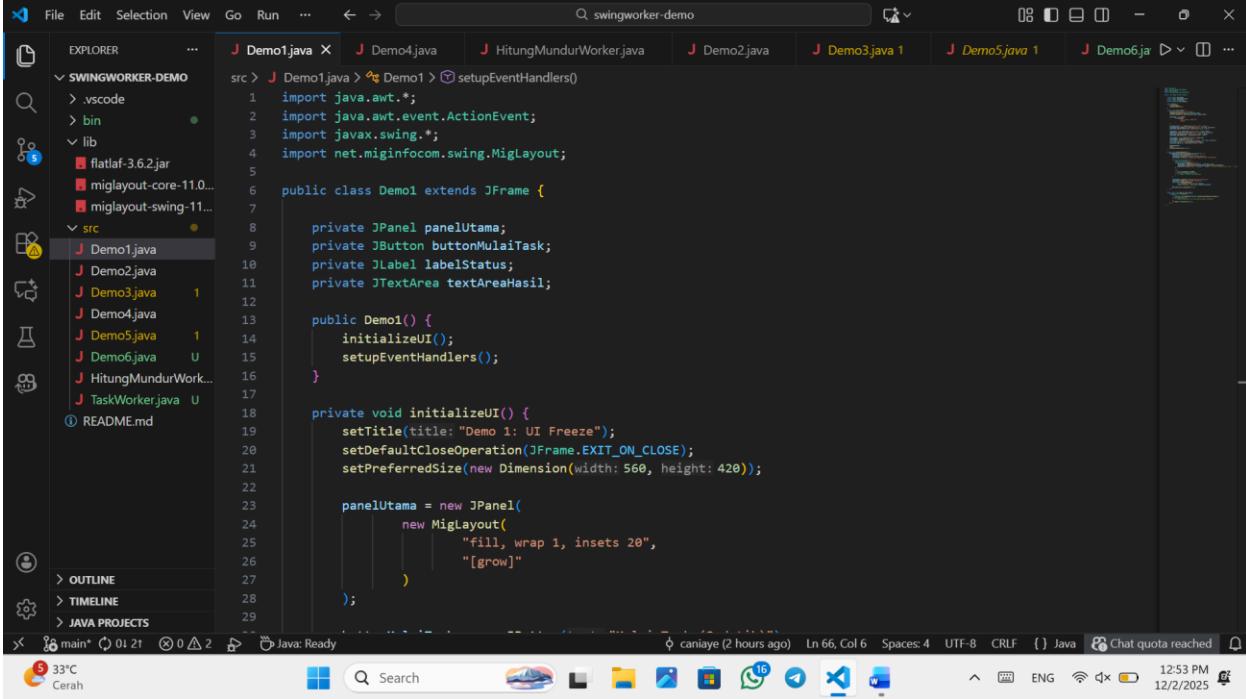
Jika tidak, pembatalan tidak akan bekerja.

Jika dibatalkan:

- process() berhenti menerima progress,
- done() akan menangani kondisi cancel,
- get() akan memunculkan CancellationException.

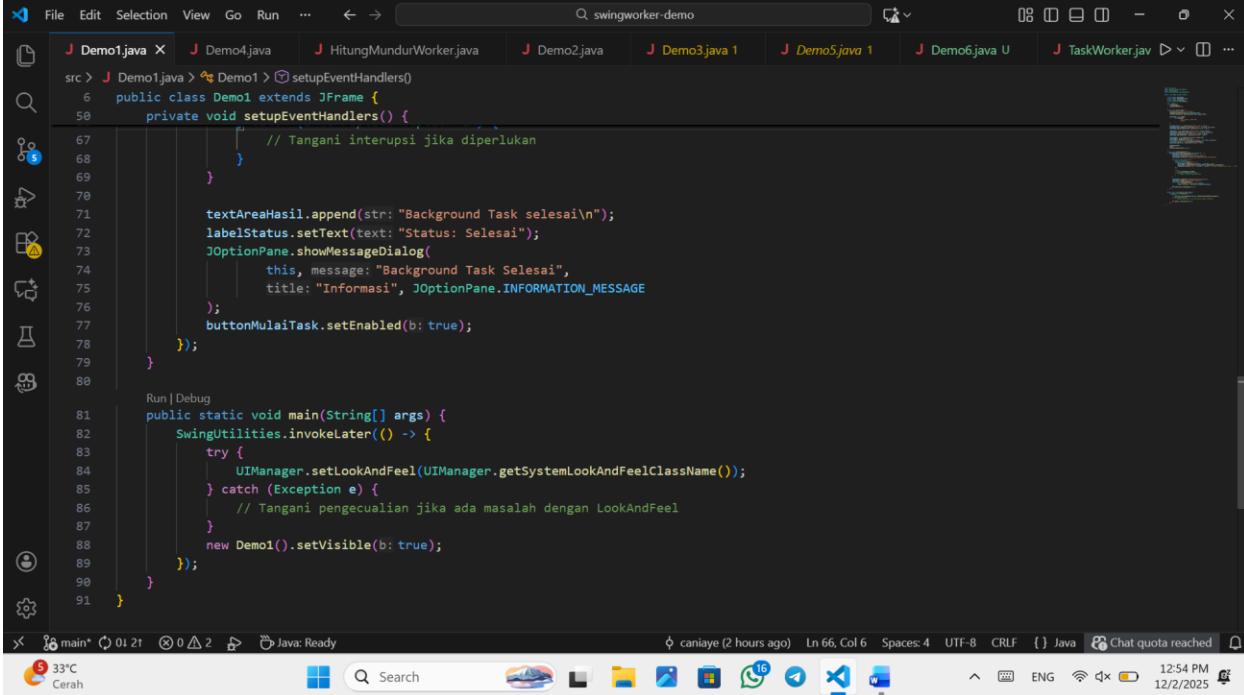
2. Terdapat penyelesaian Praktikum Demo 1 UI Freeze yang disertai bukti hasil running dan pembuktian UI Freeze

a. Kode Program



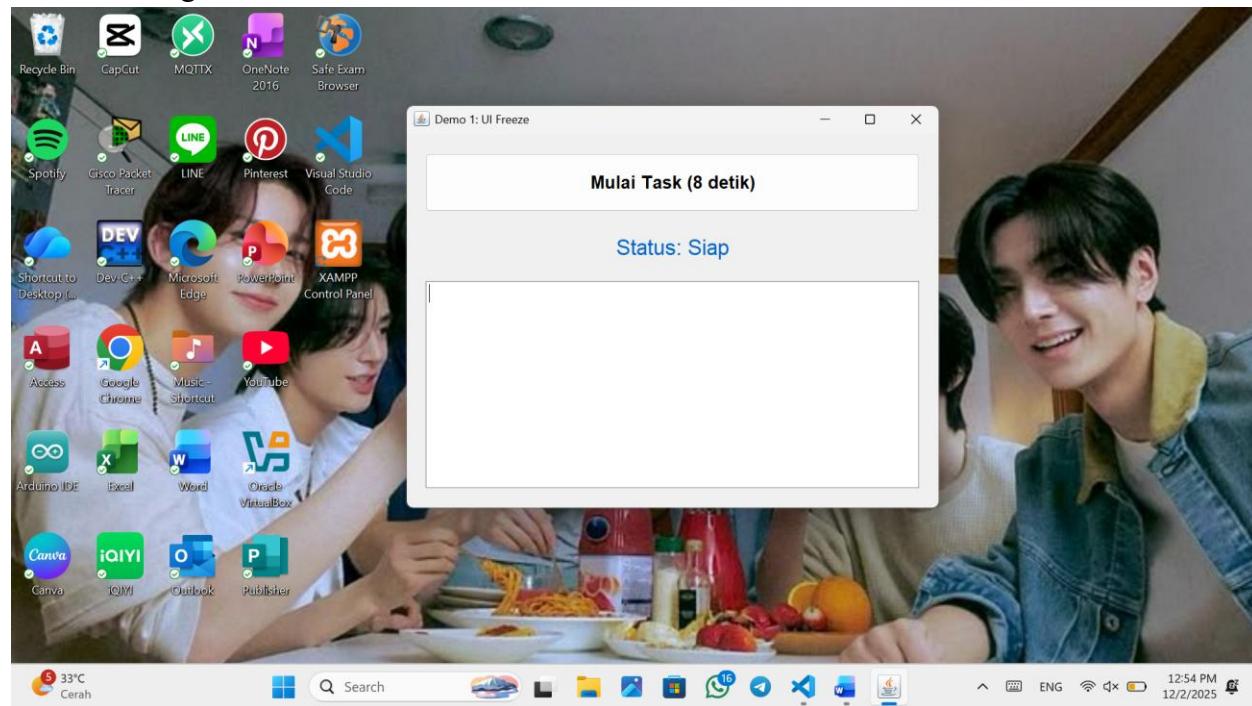
```
File Edit Selection View Go Run ... < > Q swingworker-demo
EXPLORER ... J Demo1.java X J Demo4.java J HitungMundurWorker.java J Demo2.java J Demo3.java 1 J Demo5.java 1 J Demo6.java D ...
SWINGWORKER-DEMO > .vscode
> bin
> lib
flatlatf-3.6.2.jar
miglayout-core-11.0...
miglayout-swing-11...
src > J Demo1.java > Demo1.java > setupEventHandlers()
1 import java.awt.*;
2 import java.awt.event.ActionEvent;
3 import javax.swing.*;
4 import net.miginfocom.swing.MigLayout;
5
6 public class Demo1 extends JFrame {
7
8     private JPanel panelUtama;
9     private JButton buttonMulaiTask;
10    private JLabel labelStatus;
11    private JTextArea textAreaHasil;
12
13    public Demo1() {
14        initializeUI();
15        setupEventHandlers();
16    }
17
18    private void initializeUI() {
19        setTitle("Demo 1: UI Freeze");
20        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
21        setPreferredSize(new Dimension(width: 560, height: 420));
22
23        panelUtama = new JPanel(
24            new MigLayout(
25                "fill, wrap 1, insets 20",
26                "[grow]"
27            );
28        );
29    }
30
31    private void setupEventHandlers() {
32        buttonMulaiTask.addActionListener(e -> {
33            TaskWorker taskWorker = new TaskWorker(panelUtama);
34            SwingWorker<String> swingWorker = new SwingWorker<String>() {
35                protected String doInBackground() throws Exception {
36                    return taskWorker.getHasil();
37                }
38
39                protected void done() {
40                    String hasil = get();
41                    if (hasil != null) {
42                        textAreaHasil.setText(hasil);
43                    }
44                }
45            };
46            swingWorker.execute();
47        });
48    }
49
50    class TaskWorker {
51        JPanel panelUtama;
52
53        public TaskWorker(JPanel panelUtama) {
54            this.panelUtama = panelUtama;
55        }
56
57        public String getHasil() {
58            try {
59                Thread.sleep(5000);
60            } catch (InterruptedException e) {
61                e.printStackTrace();
62            }
63
64            return "Hasil yang dihasilkan oleh task worker";
65        }
66    }
67
68    public static void main(String[] args) {
69        SwingUtilities.invokeLater(() -> {
70            new Demo1().setVisible(true);
71        });
72    }
73}
```

```
File Edit Selection View Go Run ... swingworker-demo
J Demo1.java X J Demo4.java J HitungMundurWorker.java J Demo2.java J Demo3.java 1 J Demo5.java 1 J Demo6.java U J TaskWorker.java Dv ...
src > J Demo1.java > Demo1 > setupEventHandlers()
  6   public class Demo1 extends JFrame {
  18     private void initializeUI() {
  29
  30       buttonMulaiTask = new JButton(text: "Mulai Task (8 detik)");
  31       buttonMulaiTask.setFont(new Font(name: "Inter", Font.BOLD, size: 18));
  32       panelUtama.add(buttonMulaiTask, constraints: "growx, h 20%");
  33
  34       labelStatus = new JLabel(text: "Status: Siap", JLabel.CENTER);
  35       labelStatus.setFont(new Font(name: "Inter", Font.PLAIN, size: 22));
  36       labelStatus.setForeground(new Color(r: 0, g: 102, b: 204));
  37       panelUtama.add(labelStatus, constraints: "growx, h 20%");
  38
  39       textAreaHasil = new JTextArea(rows: 10, columns: 50);
  40       textAreaHasil.setEditable(b: false);
  41       textAreaHasil.setFont(new Font(name: "Consolas", Font.PLAIN, size: 15));
  42       JScrollPane scroll = new JScrollPane(textAreaHasil);
  43       panelUtama.add(scroll, constraints: "growx, h 60%");
  44
  45       add(panelUtama);
  46       pack();
  47       setLocationRelativeTo(c: null);
  48
  49
  50     private void setupEventHandlers() {
  51       buttonMulaiTask.addActionListener((ActionEvent e) -> {
  52         buttonMulaiTask.setEnabled(b: false);
  53         labelStatus.setText(text: "Status: Sedang bekerja");
  54         textAreaHasil.setText(t: "Mengerjakan Background Task (8 detik)\n");
  55
  56       for (int i = 8; i >= 0; i--) {
  57         final int seconds = i;
  58         SwingUtilities.invokeLater(() -> {
  59           textAreaHasil.append("Hitung mundur: " + seconds + "\n");
  60           textAreaHasil.setCaretPosition(textAreaHasil.getDocument().getLength());
  61           labelStatus.setText("Sisa " + seconds + " detik"); // Asumsi: perbaikan dari "Sisa " + seconds + 11 detik"
  62         });
  63
  64         try {
  65           Thread.sleep(millis: 1000);
  66         } catch (InterruptedException ex) {
  67           // Tangani interupsi jika diperlukan
  68         }
  69       }
  70
  71       textAreaHasil.append(str: "Background Task selesai\n");
  72       labelStatus.setText(text: "Status: Selesai");
  73       JOptionPane.showMessageDialog(
  74         this, message: "Background Task Selesai",
  75         title: "Informasi", JOptionPane.INFORMATION_MESSAGE
  76       );
  77       buttonMulaiTask.setEnabled(b: true);
  78     });
  79   }
  80 }
```

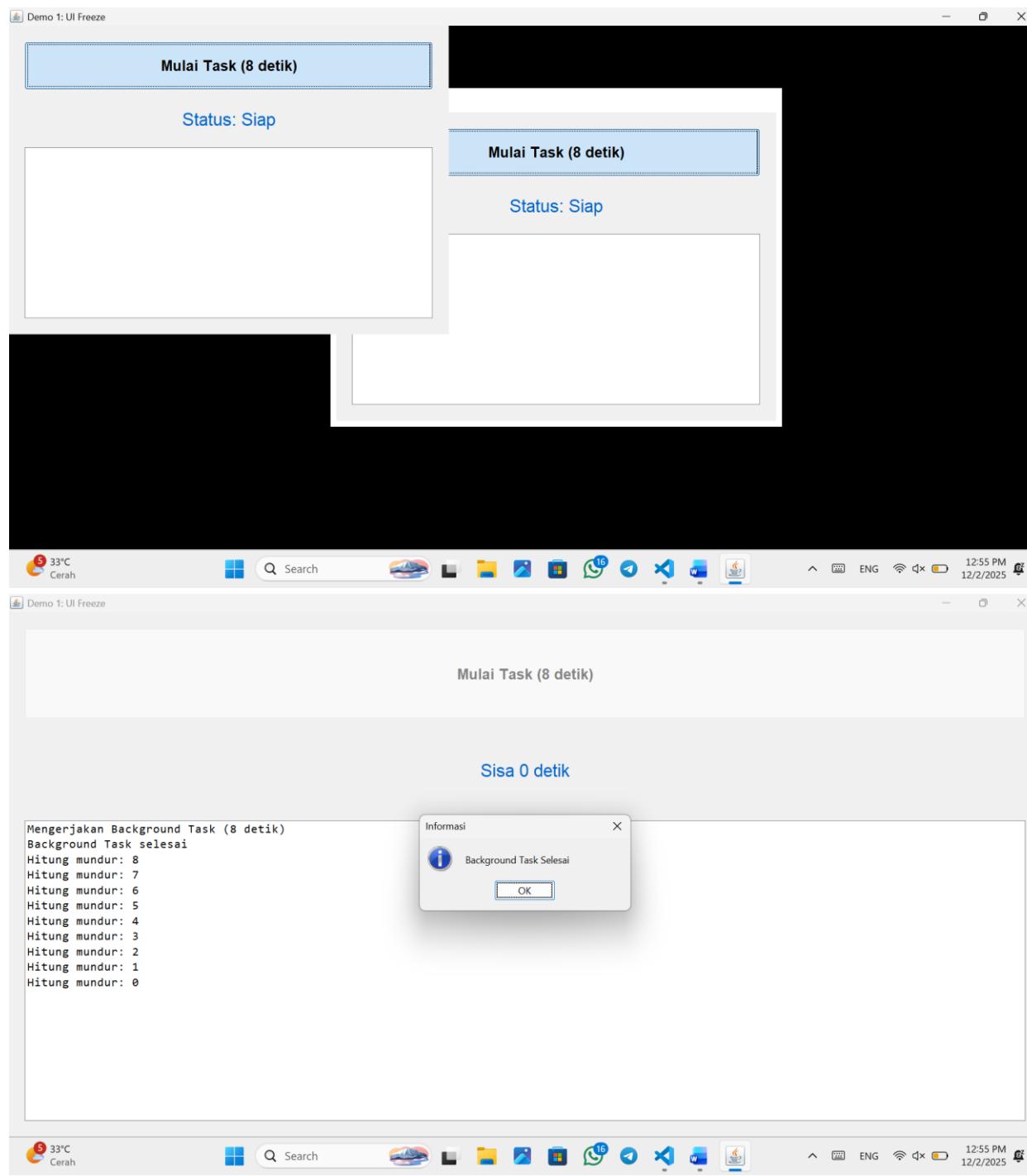


```
File Edit Selection View Go Run ... swingworker-demo
J Demo1.java X J Demo4.java J HitungMundurWorker.java J Demo2.java J Demo3.java 1 J Demo5.java 1 J Demo6.java U J TaskWorker.java D ...
src > J Demo1.java > Demo1 > setupEventHandlers()
6   public class Demo1 extends JFrame {
50     private void setupEventHandlers() {
67       // Tangani interupsi jika diperlukan
68     }
69   }
70
71   textAreaHasil.append(str: "Background Task selesai\n");
72   labelStatus.setText(text: "Status: Selesai");
73   JOptionPane.showMessageDialog(
74     this, message: "Background Task Selesai",
75     title: "Informasi", JOptionPane.INFORMATION_MESSAGE
76   );
77   buttonMulaiTask.setEnabled(b: true);
78 }
79
80
81 Run | Debug
82 public static void main(String[] args) {
83   SwingUtilities.invokeLater(() -> {
84     try {
85       UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
86     } catch (Exception e) {
87       // Tangani pengecualian jika ada masalah dengan LookAndFeel
88     }
89     new Demo1().setVisible(b: true);
90   });
91 }
```

### b. Hasil Running



### c. Pembuktian UI Freeze



3. Terdapat penyelesaian Praktikum Demo 2 SwingWorker (Anonymous Inner Class) yang disertai bukti hasil running dan pembuktian UI Interaktif (tidak freeze)
  - a. Kode Program

File Edit Selection View Go Run ... ↵ → 🔍 swingworker-demo

EXPLORER ... J Demo1.java J Demo4.java J HitungMundurWorker.java J Demo2.java X J Demo3.java 1 J Demo5.java 1 J Demo6.java Dv

SWINGWORKER-DEMO > .vscode > bin > lib > src > J Demo1.java J Demo2.java J Demo3.java 1 J Demo4.java J Demo5.java 1 J Demo6.java U J HitungMundurWorker.java J TaskWorker.java U README.md

J Demo2.java

```
src > J Demo2.java > Demo2
1 import java.awt.Color;
2 import java.awt.Dimension;
3 import java.awt.Font;
4 import java.awt.event.ActionEvent;
5 import java.util.List;
6 import javax.swing.JButton;
7 import javax.swing.JFrame;
8 import javax.swing.JLabel;
9 import javax.swing.JOptionPane;
10 import javax.swing.JPanel;
11 import javax.swing.JScrollPane;
12 import javax.swing.JTextArea;
13 import javax.swing.SwingUtilities;
14 import javax.swing.SwingWorker;
15 import javax.swing.UIManager;
16 import net.miginfocom.swing.MigLayout;
17
18 public class Demo2 extends JFrame {
19
20     private JPanel panelUtama;
21     private JButton buttonMulaiTask;
22     private JLabel labelStatus;
23     private JTextArea textAreaHasil;
24     private SwingWorker<String, Integer> worker;
25
26     public Demo2() {
27         initializeUI();
28         setupEventHandlers();
29     }
}
```

> OUTLINE > TIMELINE > JAVA PROJECTS

main\* 01 2t 0 △ 2 Java: Ready

33°C Cerah

Search

caniaye (4 hours ago) Ln 118, Col 2 Spaces: 4 UTF-8 CRLF {} Java Chat quota reached

12:56 PM 12/2/2025

File Edit Selection View Go Run ... ↵ → 🔍 swingworker-demo

EXPLORER ... J Demo1.java J Demo4.java J HitungMundurWorker.java J Demo2.java X J Demo3.java 1 J Demo5.java 1 J Demo6.java U J TaskWorker.java Dv

J Demo2.java

```
src > J Demo2.java > Demo2
18 public class Demo2 extends JFrame {
30
31     private void initializeUI() {
32         setTitle("Demo 2: SwingWorker (Anonymous Inner Class)");
33         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
34         setPreferredSize(new Dimension(width: 600, height: 450));
35
36         panelUtama = new JPanel(
37             new MigLayout("fill, wrap 1, insets 20")
38         );
39
40         buttonMulaiTask = new JButton(text: "Mulai Task (8 detik)");
41         buttonMulaiTask.setFont(new Font(name: "Inter", Font.BOLD, size: 18));
42         panelUtama.add(buttonMulaiTask, constraints: "growx, h 20%");
43
44         labelStatus = new JLabel(text: "Status: Siap", JLabel.CENTER);
45         labelStatus.setFont(new Font(name: "Inter", Font.PLAIN, size: 22));
46         labelStatus.setForeground(new Color(r: 0, g: 128, b: 0));
47         panelUtama.add(labelStatus, constraints: "growx, h 20%");
48
49         textAreaHasil = new JTextArea();
50         textAreaHasil.setEditable(b: false);
51         textAreaHasil.setFont(new Font(name: "Consolas", Font.PLAIN, size: 15));
52         JScrollPane scroll = new JScrollPane(textAreaHasil);
53         panelUtama.add(scroll, constraints: "growx, h 60%");
54
55         add(panelUtama);
56         pack();
57         setLocationRelativeTo(c: null);
}
```

main\* 01 2t 0 △ 2 Java: Ready

33°C Cerah

Search

caniaye (4 hours ago) Ln 118, Col 2 Spaces: 4 UTF-8 CRLF {} Java Chat quota reached

12:56 PM 12/2/2025

```
File Edit Selection View Go Run ... ↶ ↷ 🔍 swingworker-demo
```

```
src > J Demo1.java J Demo4.java J HitungMundurWorker.java J Demo2.java X J Demo3.java 1 J Demo5.java 1 J Demo6.java U J TaskWorker.java D...
```

```
18 public class Demo2 extends JFrame {  
31     private void initializeUI() {  
58         }  
59  
60         private void setupEventHandlers() {  
61             buttonMulaiTask.addActionListener((ActionEvent e) -> {  
62                 buttonMulaiTask.setEnabled(b: false);  
63                 labelStatus.setText(text: "Status: Mulai bekerja");  
64                 textAreaHasil.setText(t: "Mengerjakan Background Task (8 detik)\n");  
65  
66             worker = new SwingWorker<String, Integer>() {  
67                 @Override  
68                 protected String doInBackground() throws Exception {  
69                     for (Int i = 8; i >= 0; i--) {  
70                         Thread.sleep(millis: 1000);  
71                         publish(i);  
72                     }  
73                     return "Background Task Selesai";  
74                 }  
75  
76                 @Override  
77                 protected void process(List<Integer> chunks) {  
78                     // Mengambil nilai terakhir dari daftar chunk  
79                     int detik = chunks.get(chunks.size() - 1);  
80  
81                     textAreaHasil.append("Hitung mundur: " + detik + "\n");  
82                     textAreaHasil.setCaretPosition(textAreaHasil.getDocument().getLength());  
83                     labelStatus.setText("Sisa " + detik + " detik");  
84                 }  
85         }  
86         @Override  
87         protected void done() {  
88             try {  
89                 String hasil = get();  
90                 textAreaHasil.append(hasil + "\n");  
91                 labelStatus.setText(text: "Status: Selesai");  
92  
93                 JOptionPane.showMessageDialog(  
94                     Demo2.this, hasil, title: "Informasi", JOptionPane.INFORMATION_MESSAGE  
95                 );  
96             } catch (Exception ex) {  
97                 labelStatus.setText(text: "Status: Error");  
98                 textAreaHasil.append("Error: " + ex.getMessage() + "\n");  
99             } finally {  
100                 buttonMulaiTask.setEnabled(b: true);  
101             }  
102         }  
103     };  
104     worker.execute();  
105 };
```

```
Run | Debug  
public static void main(String[] args) {  
    SwingUtilities.invokeLater(() -> {  
        ...
```

⌚ 33°C Cerah

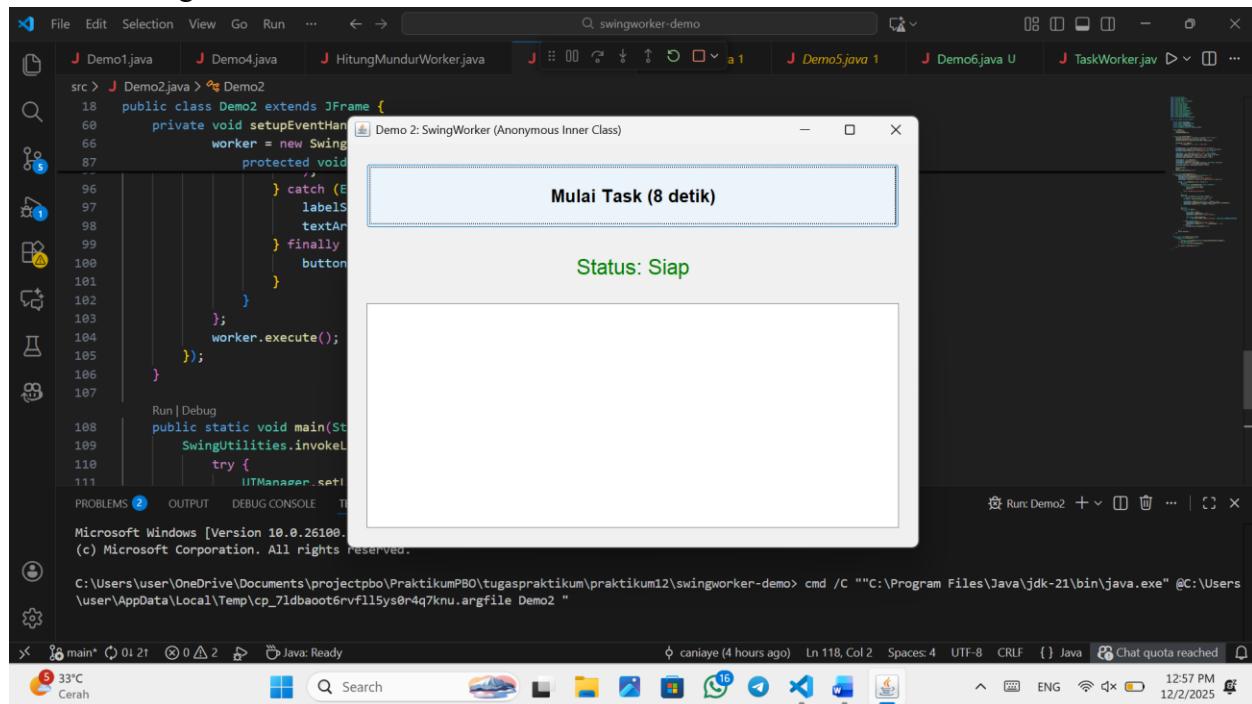
Search

caniaye (4 hours ago) Ln 118, Col 2 Spaces: 4 UTF-8 CRLF {} Java Chat quota reached

12:56 PM 12/2/2025

```
File Edit Selection View Go Run ... < > swingworker-demo
src > J Demo1.java J Demo4.java J HitungMundurWorker.java J Demo2.java X J Demo3.java 1 J Demo5.java 1 J Demo6.java U J TaskWorker.java D ...
18 public class Demo2 extends JFrame {
60     private void setupEventHandlers() {
66         worker = new SwingWorker<String, Integer>() {
87             protected void done() {
95                 ...
96                 } catch (Exception ex) {
97                     labelStatus.setText("Status: Error");
98                     textAreaHasil.append("Error: " + ex.getMessage() + "\n");
99                 } finally {
100                     buttonMulaiTask.setEnabled(b: true);
101                 }
102             };
103         worker.execute();
104     }
105 }
106 }
107 }
108 public static void main(String[] args) {
109     SwingUtilities.invokeLater(() -> {
110         try {
111             UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
112         } catch (Exception e) {
113             System.err.println(e.getMessage());
114         }
115         new Demo2().setVisible(b: true);
116     });
117 }
118 }
```

### b. Hasil Running



### c. Pembuktian UI Interaktif (tidak freeze)

The screenshot displays the Eclipse IDE interface with two Java code editors open. Both editors contain the same code for a SwingWorker demonstration:

```

src > J Demo2.java > Demo2
18  public class Demo2 extends JFrame {
60   private void setupEventHandle...
66   worker = new SwingWorker<...> {
72     protected void do...
78       ...
84     } catch (Exception e) {
85       ...
86     }
87   }
88   ...
89   worker.execute();
90 }
91
92 Run | Debug
93 public static void main(String[] args) {
94   SwingUtilities.invokeLater(new Runnable() {
95     public void run() {
96       ...
97       ...
98       ...
99       ...
100      ...
101    }
102  });
103 }
104 }
105 }
106 }
107 }
108
109 try {
110   UIManager.setLookAndFeel("Nimbus");
111 }

```

Below the code editors, the Eclipse interface includes tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL. The OUTPUT tab shows command-line logs:

```

Microsoft Windows [Version 10.0.26100.717]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user\OneDrive\Documents\projectpb0\PraktikumPBO\tugaspraktikum\praktikum12\swingworker-demo> cmd /C "C:\Program Files\Java\jdk-21\bin\java.exe" @C:\Users\user\AppData\Local\Temp\cp_7idbaoot6rvfl5ys0r4q7knu.argvfile Demo2

```

The Eclipse status bar at the bottom indicates the current user is 'caniaye' (4 hours ago), line 118, column 2, spaces: 4, UTF-8, CRLF, Java selected, and a 'Chat quota reached' notification.

Two Java applications are running in the foreground. Both show a window titled 'Demo 2: SwingWorker (Anonymous Inner Class)'. The first window has a progress bar labeled 'Mulai Task (8 detik)' and a message box titled 'Status: Selesai' containing the text 'Background Task Selesai'. The second window has a similar progress bar and message box.

4. Terdapat penyelesaian Praktikum Demo 3 SwingWorker (Static Nested Class) yang disertai bukti hasil running dan pembuktian UI Interaktif (tidak freeze)

a. Kode Program

```
src > J Demo3.java > Demo3
1 import java.awt.Color;
2 import java.awt.Dimension;
3 import java.awt.Font;
4 import java.awt.event.ActionEvent;
5 import java.util.List;
6 import javax.swing.JButton;
7 import javax.swing.JFrame;
8 import javax.swing.JLabel;
9 import javax.swing.JOptionPane;
10 import javax.swing.JPanel;
11 import javax.swing.JScrollPane;
12 import javax.swing.JTextArea;
13 import javax.swing.SwingUtilities;
14 import javax.swing.SwingWorker;
15 import javax.swing.UIManager;
16 import net.miginfocom.swing.MigLayout;
17
18 public class Demo3 extends JFrame {
19
20     private JPanel panelUtama;
21     private JButton buttonMulaiTask;
22     private JLabel labelStatus;
23     private JTextArea textAreaHasil;
24
25     public Demo3() {
26         initializeUI();
27         setupEventHandlers();
28     }
29
30     private void initializeUI() {
31         setTitle("Demo 3: SwingWorker (Static Nested Class)");
32         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
33         setPreferredSize(new Dimension(width: 600, height: 450));
34
35         // Perbaikan: Inisialisasi JPanel dan MigLayout digabungkan
36         panelUtama = new JPanel(new MigLayout("fill, wrap 1, insets 20"));
37
38         // Perbaikan: Penulisan inisialisasi
39         buttonMulaiTask = new JButton(text: "Mulai Task (8 detik)");
40         buttonMulaiTask.setFont(new Font(name: "Inter", Font.BOLD, size: 18));
41         panelUtama.add(buttonMulaiTask, constraints: "growx, h 20%");
42
43         labelStatus = new JLabel(text: "Status: Siap", JLabel.CENTER);
44         labelStatus.setFont(new Font(name: "Inter", Font.PLAIN, size: 22));
45         labelStatus.setForeground(new Color(r: 0, g: 128, b: 0));
46         panelUtama.add(labelStatus, constraints: "growx, h 20%");
47
48         // Perbaikan: Penulisan inisialisasi
49         textAreaHasil = new JTextArea();
50         textAreaHasil.setEditable(b: false);
51         textAreaHasil.setFont(new Font(name: "Consolas", Font.PLAIN, size: 15));
52         JScrollPane scroll = new JScrollPane(textAreaHasil);
53         panelUtama.add(scroll, constraints: "growx, h 60%");
54
55         add(panelUtama);
56         pack();
57
58     }
59
60     private void setupEventHandlers() {
61         buttonMulaiTask.addActionListener {
62             JButton button = it;
63             SwingWorker<String> worker = new SwingWorker<String>() {
64                 protected String doInBackground() throws Exception {
65                     return "Hasil Task";
66                 }
67
68                 protected void done() {
69                     String result = get();
70                     textAreaHasil.setText(result);
71                 }
72             }
73             worker.execute();
74         }
75     }
76
77     public static void main() {
78         SwingUtilities.invokeLater {
79             () -> new Demo3().setVisible(true);
80         }
81     }
82 }
```

```
src > J Demo3.java > Demo3
18 public class Demo3 extends JFrame {
29
30     private void initializeUI() {
31         setTitle("Demo 3: SwingWorker (Static Nested Class)");
32         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
33         setPreferredSize(new Dimension(width: 600, height: 450));
34
35         // Perbaikan: Inisialisasi JPanel dan MigLayout digabungkan
36         panelUtama = new JPanel(new MigLayout("fill, wrap 1, insets 20"));
37
38         // Perbaikan: Penulisan inisialisasi
39         buttonMulaiTask = new JButton(text: "Mulai Task (8 detik)");
40         buttonMulaiTask.setFont(new Font(name: "Inter", Font.BOLD, size: 18));
41         panelUtama.add(buttonMulaiTask, constraints: "growx, h 20%");
42
43         labelStatus = new JLabel(text: "Status: Siap", JLabel.CENTER);
44         labelStatus.setFont(new Font(name: "Inter", Font.PLAIN, size: 22));
45         labelStatus.setForeground(new Color(r: 0, g: 128, b: 0));
46         panelUtama.add(labelStatus, constraints: "growx, h 20%");
47
48         // Perbaikan: Penulisan inisialisasi
49         textAreaHasil = new JTextArea();
50         textAreaHasil.setEditable(b: false);
51         textAreaHasil.setFont(new Font(name: "Consolas", Font.PLAIN, size: 15));
52         JScrollPane scroll = new JScrollPane(textAreaHasil);
53         panelUtama.add(scroll, constraints: "growx, h 60%");
54
55         add(panelUtama);
56         pack();
57
58     }
59
50     private void setupEventHandlers() {
51         buttonMulaiTask.addActionListener {
52             JButton button = it;
53             SwingWorker<String> worker = new SwingWorker<String>() {
54                 protected String doInBackground() throws Exception {
55                     return "Hasil Task";
56                 }
57
58                 protected void done() {
59                     String result = get();
60                     textAreaHasil.setText(result);
61                 }
62             }
63             worker.execute();
64         }
65     }
66
67     public static void main() {
68         SwingUtilities.invokeLater {
69             () -> new Demo3().setVisible(true);
70         }
71     }
72 }
```

```
File Edit Selection View Go Run ... ← → ⌂ swingworker-demo
J Demo1.java J Demo4.java J HitungMundurWorker.java J Demo2.java J Demo3.java 1 X J Demo5.java 1 J Demo6.java U J TaskWorker.java Dv ⌂
src > J Demo3.java > Demo3
18  public class Demo3 extends JFrame {
30  private void initializeUI() {
56      pack();
57      setLocationRelativeTo(null);
58  }
59
60  private void setupEventHandlers() {
61      buttonMulaiTask.addActionListener(
62          // Perbaikan: Penggunaan lambda untuk ActionEvent
63          e -> new HitungMundurWorker(this).execute()
64      );
65  }
66
67  // Static Nested Class untuk operasi background
68  private static class HitungMundurWorker extends SwingWorker<String, Integer> {
69
70      private final Demo3 frame;
71
72      private HitungMundurWorker(Demo3 frame) {
73          this.frame = frame;
74      }
75
76      @Override
77      protected String doInBackground() throws Exception {
78          // Akses komponen UI di awal task (diperbolehkan karena EDT belum sibuk)
79          frame.buttonMulaiTask.setEnabled(false);
80          frame.labelStatus.setText("Status: Mulai bekerja");
81          frame.textAreaHasil.setText(t: "Mengerjakan Background Task (8 detik)\n");
82      }
83
84      @Override
85      protected void process(List<Integer> chunks) {
86          for (int i = 8; i >= 0; i--) {
87              Thread.sleep(millis: 1000); // Operasi lama di thread background
88              publish(i); // Kirim update ke EDT
89          }
90      }
91
92      @Override
93      protected void done() {
94          // Berjalan di EDT: Update UI setelah task selesai
95          try {
96              String hasil = get(); // Ambil hasil dari doInBackground
97              frame.textAreaHasil.append(hasil + "\n");
98              frame.textAreaHasil.setCaretPosition(frame.textAreaHasil.getDocument().getLength());
99          } catch (Exception e) {
100             e.printStackTrace();
101         }
102     }
103
104     @Override
105     protected void done() {
106         // Berjalan di EDT: Update UI setelah task selesai
107         try {
108             String hasil = get(); // Ambil hasil dari doInBackground
109             frame.textAreaHasil.append(hasil + "\n");
110             frame.labelStatus.setText("Status: Selesai!");
111         } catch (Exception e) {
112             e.printStackTrace();
113         }
114     }
}
main* ⌂ 0 2t ⌂ 0 △ 2 ⌂ Java: Ready ⌂ caniaye (4 hours ago) Ln 118, Col 1 Spaces: 4 UTF-8 CRLF {} Java Chat quota reached
33°C Cerah Search ⌂ ENG ⌂ 12:59 PM 12/2/2025
```

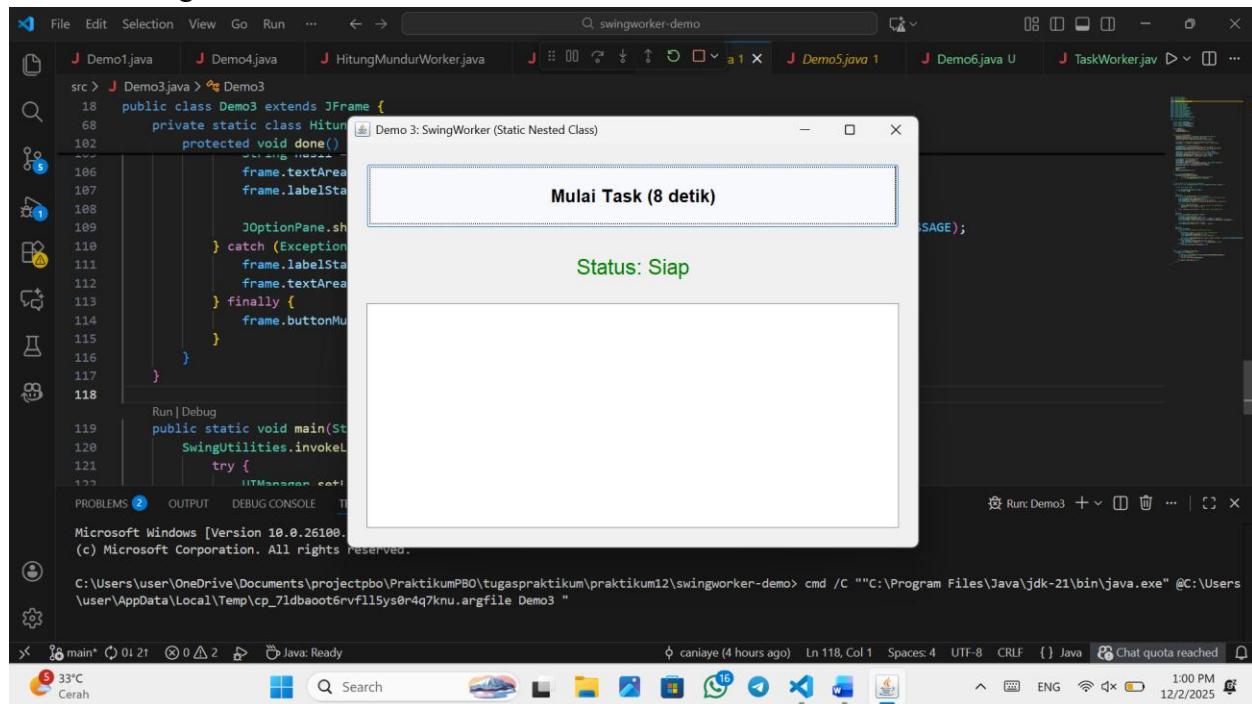
```
File Edit Selection View Go Run ... ← → ⌂ swingworker-demo
J Demo1.java J Demo4.java J HitungMundurWorker.java J Demo2.java J Demo3.java 1 X J Demo5.java 1 J Demo6.java U J TaskWorker.java Dv ⌂
src > J Demo3.java > Demo3
18  public class Demo3 extends JFrame {
68  private static class HitungMundurWorker extends SwingWorker<String, Integer> {
77      protected String doInBackground() throws Exception {
83          for (int i = 8; i >= 0; i--) {
84              Thread.sleep(millis: 1000); // Operasi lama di thread background
85              publish(i); // Kirim update ke EDT
86          }
87          return "Background Task Selesai"; // Hasil akhir (untuk method done)
88      }
89
90      @Override
91      protected void process(List<Integer> chunks) {
92          // Berjalan di EDT: Update UI
93          int detik = chunks.get(chunks.size() - 1);
94          frame.textAreaHasil.append("Hitung mundur: " + detik + " detik\n");
95          frame.textAreaHasil.setCaretPosition(frame.textAreaHasil.getDocument().getLength());
96
97          // Perbaikan: Kesalahan penulisan "detik"
98          frame.labelStatus.setText("Sisa " + detik + " detik");
99      }
100
101      @Override
102      protected void done() {
103          // Berjalan di EDT: Update UI setelah task selesai
104          try {
105              String hasil = get(); // Ambil hasil dari doInBackground
106              frame.textAreaHasil.append(hasil + "\n");
107              frame.labelStatus.setText("Status: Selesai!");
108          } catch (Exception e) {
109              e.printStackTrace();
110          }
111      }
}
main* ⌂ 0 2t ⌂ 0 △ 2 ⌂ Java: Ready ⌂ caniaye (4 hours ago) Ln 118, Col 1 Spaces: 4 UTF-8 CRLF {} Java Chat quota reached
33°C Cerah Search ⌂ ENG ⌂ 12:59 PM 12/2/2025
```

The screenshot shows a Java IDE interface with multiple tabs open. The main code editor tab contains the following Java code:

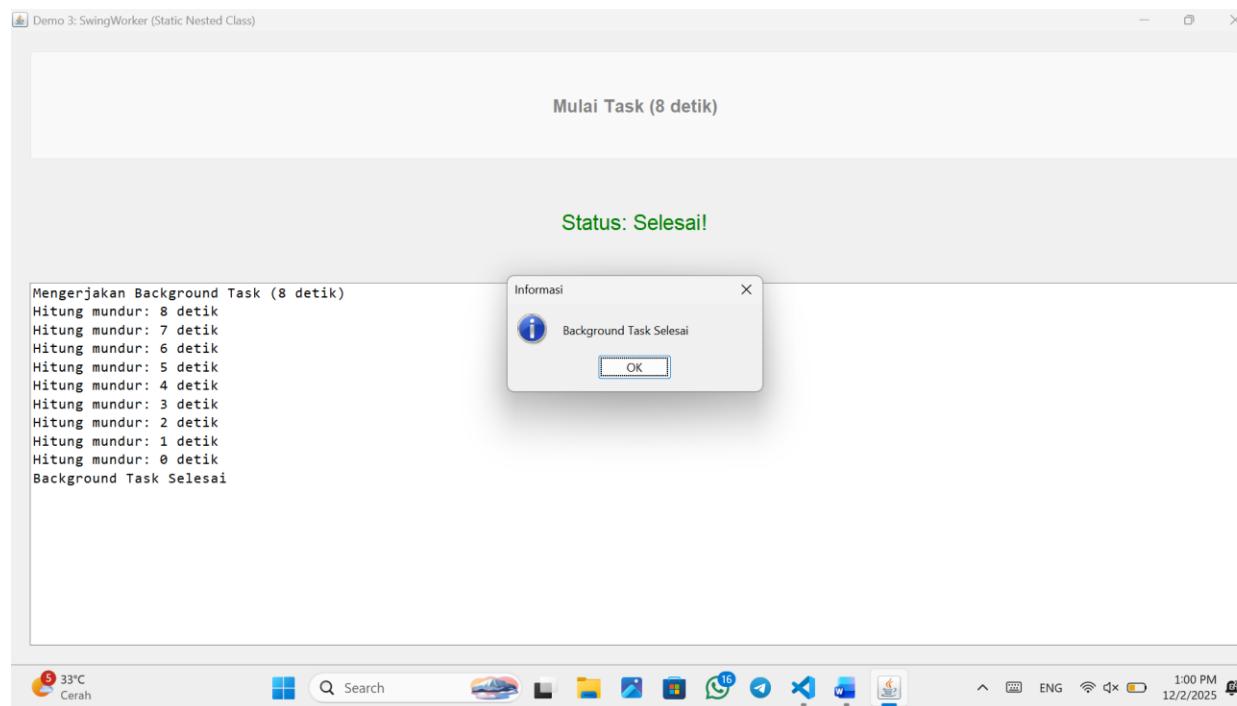
```
File Edit Selection View Go Run ... ← → 🔍 swingworker-demo
src > J Demo1.java J Demo4.java J HitungMundurWorker.java J Demo2.java J Demo3.java 1 J Demo5.java 1 J Demo6.java U J TaskWorker.java D ...
18     public class Demo3 extends JFrame {
68         private static class HitungMundurWorker extends SwingWorker<String, Integer> {
102             protected void done() {
103                 String hasil = get();
104                 frame.textAreaHasil.append(hasil + "\n");
105                 frame.labelStatus.setText(text: "Status: Selesai!");
106 
107                 JOptionPane.showMessageDialog(frame, hasil, title: "Informasi", JOptionPane.INFORMATION_MESSAGE);
108             } catch (Exception ex) {
109                 frame.labelStatus.setText(text: "Status: Error");
110                 frame.textAreaHasil.append("Error: " + ex.getMessage() + "\n");
111             } finally {
112                 frame.buttonMulaiTask.setEnabled(b: true);
113             }
114         }
115     }
116 
117 }
118 
119 public static void main(String[] args) {
120     SwingUtilities.invokeLater(() -> {
121         try {
122             UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
123         } catch (Exception e) {
124             System.err.println(e.getMessage());
125         }
126         new Demo3().setVisible(b: true);
127     });
128 }
129 }
```

The status bar at the bottom indicates the file is ready for Java compilation.

### b. Hasil Running



### c. Pembuktian UI Interaktif (tidak freeze)



5. Terdapat penyelesaian Praktikum Demo 4 SwingWorker (Separate Top-Level Class) yang terdiri dari 2 file (Demo4.java dan HitungMundurWorker.java) disertai bukti hasil running dan pembuktian UI Interaktif (tidak freeze)

#### a. Kode Program Demo4.java

```

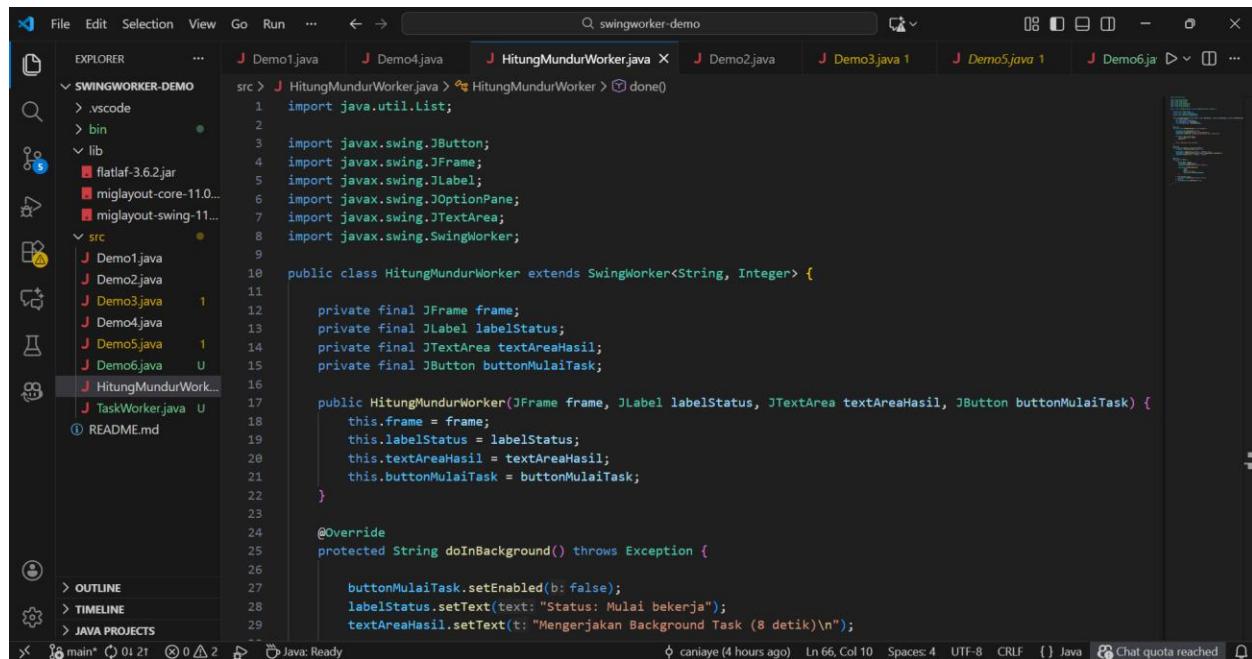
File Edit Selection View Go Run ... < > swingworker-demo
EXPLORER J Demo1.java J Demo4.java X J HitungMundurWorker.java J Demo2.java J Demo3.java 1 J Demo4.java J Demo5.java 1 J Demo5.java 1 J Demo6.java D ...
src > J Demo4.java > Demo4
1 import java.awt.Color;
2 import java.awt.Dimension;
3 import java.awt.Font;
4 import javax.swing.JButton;
5 import javax.swing.JFrame;
6 import javax.swing.JLabel;
7 import javax.swing.JPanel;
8 import javax.swing.JScrollPane;
9 import javax.swing.JTextArea;
10 import javax.swing.SwingUtilities;
11 import javax.swing.UIManager;
12
13 import net.miginfocom.swing.MigLayout;
14
15 public class Demo4 extends JFrame {
16
17     private JPanel panelUtama;
18     private JButton buttonMulaiTask;
19     private JLabel labelStatus;
20     private JTextArea textAreaHasil;
21
22     public Demo4() {
23         initializeUI();
24         setupEventHandlers();
25     }
26
27     private void initializeUI() {
28         setTitle(title: "Demo 4: SwingWorker (Separate Top-Level Class)");
29         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    
```

```

File Edit Selection View Go Run ... ↵ → swingworker-demo
J Demo1.java J Demo4.java X J HitungMundurWorker.java J Demo2.java J Demo3.java 1 J Demo5.java 1 J Demo6.java U J TaskWorker.java D ...
src > J Demo4.java > Demo4
15  public class Demo4 extends JFrame {
27  private void initializeUI() {
30      setPreferredSize(new Dimension(width: 600, height: 450));
31
32      panelUtama = new JPanel(new MigLayout("fill, wrap 1, insets 20"));
33
34      // Tombol mulai task
35      buttonMulaiTask = new JButton(text: "Mulai Task (8 detik)");
36      buttonMulaiTask.setFont(new Font(name: "Inter", Font.BOLD, size: 18));
37      panelUtama.add(buttonMulaiTask, constraints: "growx, h 20%");
38
39      // Label status
40      labelStatus = new JLabel(text: "Status: Siap", JLabel.CENTER);
41      labelStatus.setFont(new Font(name: "Inter", Font.PLAIN, size: 22));
42      labelStatus.setForeground(new Color(r: 0, g: 128, b: 0));
43      panelUtama.add(labelStatus, constraints: "growx, h 20%");
44
45      // Text area hasil
46      textAreaHasil = new JTextArea();
47      textAreaHasil.setEditable(b: false);
48      textAreaHasil.setFont(new Font(name: "Consolas", Font.PLAIN, size: 15));
49      JScrollPane scroll = new JScrollPane(textAreaHasil);
50      panelUtama.add(scroll, constraints: "growx, h 68%");
51
52      add(panelUtama);
53      pack();
54      setLocationRelativeTo(c: null);
55  }
56
57  private void setupEventHandlers() {
58      buttonMulaiTask.addActionListener(e -> {
59          buttonMulaiTask.setEnabled(b: false);
60          new HitungMundurWorker(this, labelStatus, textAreaHasil, buttonMulaiTask).execute();
61      });
62  }
63
64  public static void main(String[] args) {
65      SwingUtilities.invokeLater(() -> {
66          try {
67              UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
68          } catch (Exception e) {
69              System.err.println(e.getMessage());
70          }
71          new Demo4().setVisible(b: true);
72      });
73  }
74 }

```

b. Kode Program HitungMundurWorker.java



The screenshot shows the Visual Studio Code interface with the title bar "swingworker-demo". The left sidebar displays the file structure of a Java project named "SWINGWORKER-DEMO". The "src" folder contains several Java files: Demo1.java, Demo2.java, Demo3.java, Demo4.java, Demo5.java, Demo6.java, HitungMundurWorker.java, and TaskWorker.java. The "HitungMundurWorker.java" file is currently open in the editor. The code implements a SwingWorker to perform a countdown task. It includes imports for java.util.List, javax.swing.JButton, javax.swing.JFrame, javax.swing.JLabel, javax.swing.JOptionPane, javax.swing.JTextArea, and javax.swing.SwingWorker. The class defines a constructor that initializes a JFrame, a JLabel, a JTextArea, and a JButton. It overrides the doInBackground() method to perform a loop that sleeps for 1000ms and publishes each iteration to the UI thread. The process() method updates a JTextArea with the current value. The done() method retrieves the final result from the worker and displays it in an JOptionPane dialog.

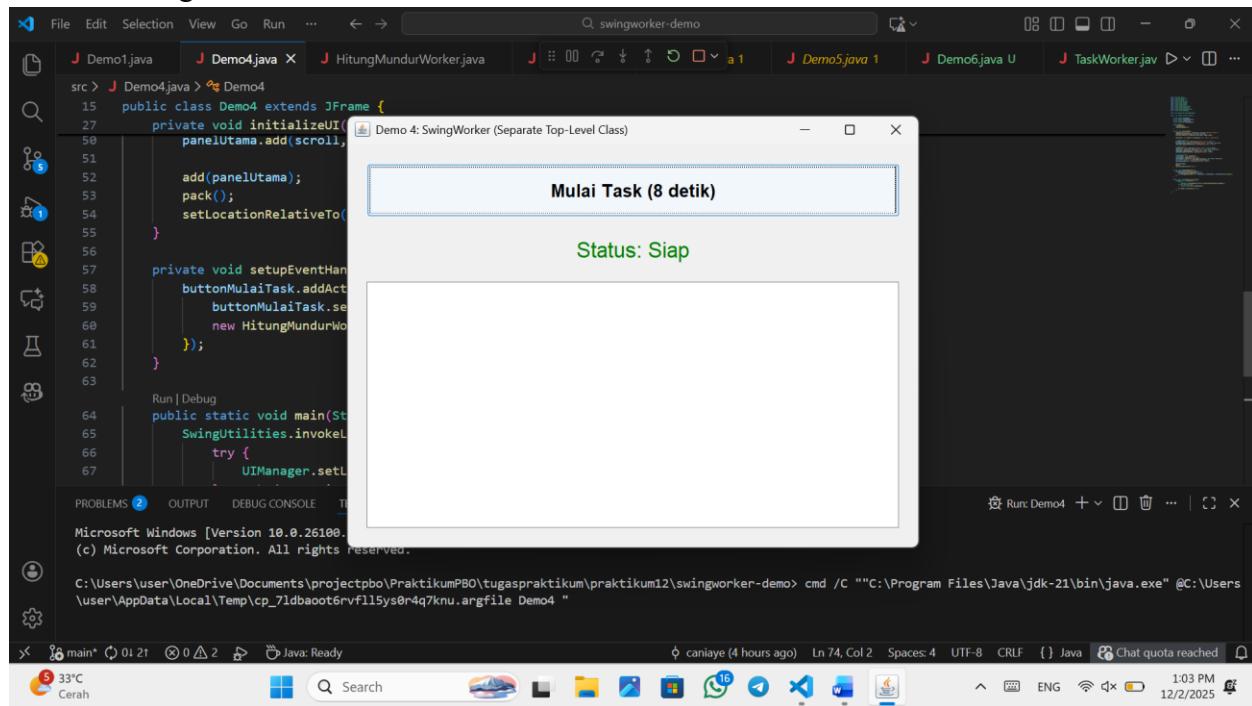
```
File Edit Selection View Go Run ... < > swingworker-demo
EXPLORER J Demo1.java J Demo4.java J HitungMundurWorker.java X J Demo2.java J Demo3.java 1 J Demo5.java 1 J Demo6.java D ...
src > J HitungMundurWorker.java > HitungMundurWorker > done()
1 import java.util.List;
2
3 import javax.swing.JButton;
4 import javax.swing.JFrame;
5 import javax.swing.JLabel;
6 import javax.swing.JOptionPane;
7 import javax.swing.JTextArea;
8 import javax.swing.SwingWorker;
9
10 public class HitungMundurWorker extends SwingWorker<String, Integer> {
11
12     private final JFrame frame;
13     private final JLabel labelStatus;
14     private final JTextArea textAreaHasil;
15     private final JButton buttonMulaiTask;
16
17     public HitungMundurWorker(JFrame frame, JLabel labelStatus, JTextArea textAreaHasil, JButton buttonMulaiTask) {
18         this.frame = frame;
19         this.labelStatus = labelStatus;
20         this.textAreaHasil = textAreaHasil;
21         this.buttonMulaiTask = buttonMulaiTask;
22     }
23
24     @Override
25     protected String doInBackground() throws Exception {
26
27         buttonMulaiTask.setEnabled(false);
28         labelStatus.setText("Status: Mulai bekerja");
29         textAreaHasil.setText("Mengerjakan Background Task (8 detik)\n");
30
31         for (int i = 8; i >= 0; i--) {
32             Thread.sleep(1000);
33             publish(i);
34         }
35
36         return "Background Task Selesai\n";
37     }
38
39     @Override
40     protected void process(List<Integer> chunks) {
41         int detik = chunks.get(chunks.size() - 1);
42
43         textAreaHasil.append("Hitung mundur: " + detik + "\n");
44         textAreaHasil.setCaretPosition(textAreaHasil.getDocument().getLength());
45         labelStatus.setText("Sisa " + detik + " detik");
46     }
47
48     @Override
49     protected void done() {
50         try {
51             String hasil = get();
52             textAreaHasil.append(hasil);
53             labelStatus.setText("Status: Selesai");
54
55             JOptionPane.showMessageDialog(
56                 frame,
57
58             );
59         } catch (Exception e) {
60             e.printStackTrace();
61         }
62     }
63
64 }
```

File Edit Selection View Go Run ... < > swingworker-demo
J Demo1.java J Demo4.java J HitungMundurWorker.java X J Demo2.java J Demo3.java 1 J Demo5.java 1 J Demo6.java U J TaskWorker.java D ...
src > J HitungMundurWorker.java > HitungMundurWorker > done()
10 public class HitungMundurWorker extends SwingWorker<String, Integer> {
25 protected String doInBackground() throws Exception {
26
27 for (int i = 8; i >= 0; i--) {
28 Thread.sleep(1000);
29 publish(i);
30 }
31
32 return "Background Task Selesai\n";
33 }
34
35 @Override
36 protected void process(List<Integer> chunks) {
37 int detik = chunks.get(chunks.size() - 1);
38
39 textAreaHasil.append("Hitung mundur: " + detik + "\n");
40 textAreaHasil.setCaretPosition(textAreaHasil.getDocument().getLength());
41 labelStatus.setText("Sisa " + detik + " detik");
42 }
43
44 @Override
45 protected void done() {
46 try {
47 String hasil = get();
48 textAreaHasil.append(hasil);
49 labelStatus.setText("Status: Selesai");
50
51 JOptionPane.showMessageDialog(
52 frame,
53
54 );
55 } catch (Exception e) {
56 e.printStackTrace();
57 }
58 }
59
60 }

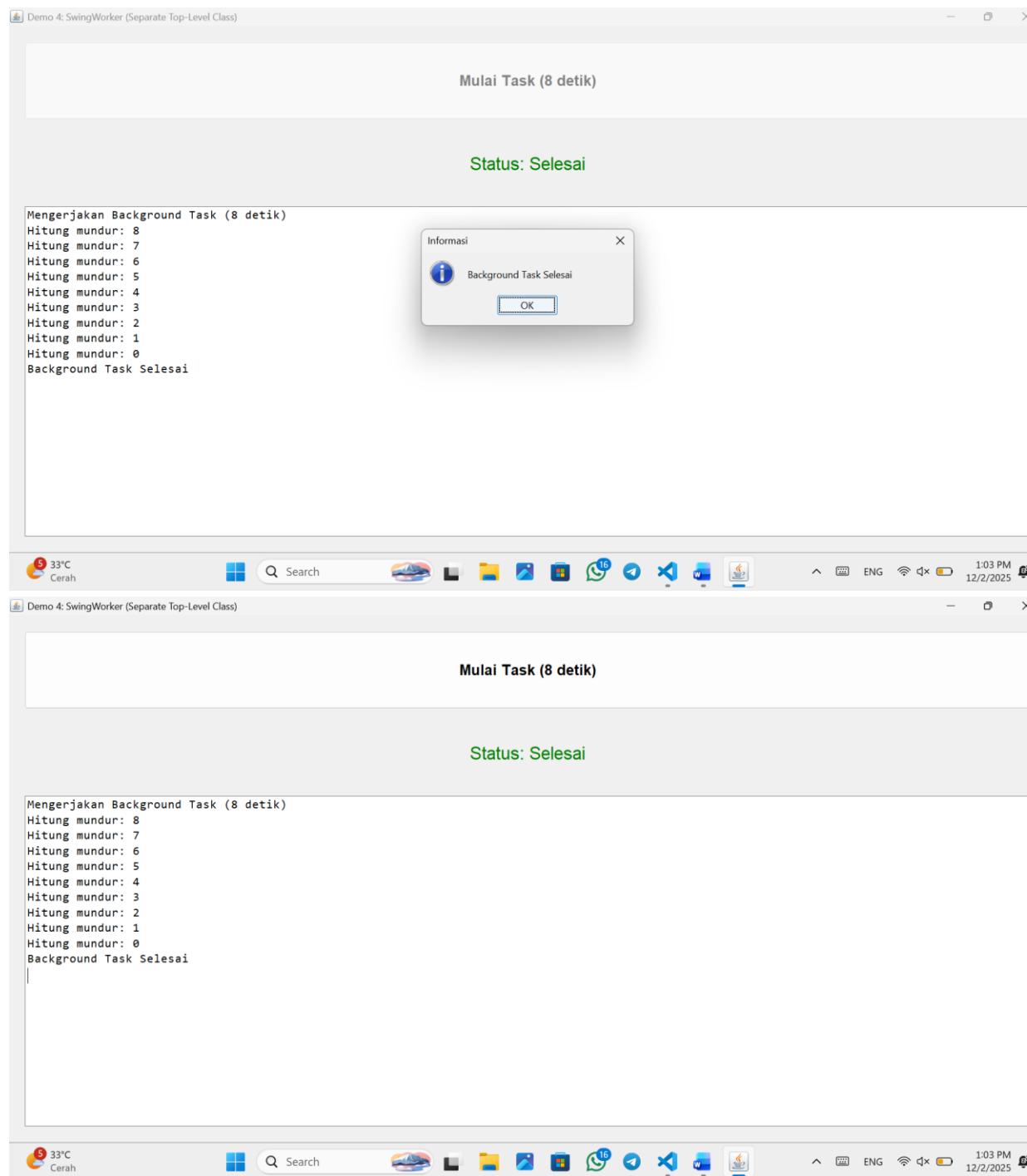
The screenshot shows an IDE interface with multiple tabs open. The active tab contains Java code for a class named `HitungMundurWorker`. The code implements the `SwingWorker<String, Integer>` interface. It has a `process` method that appends text to a `JTextArea` and updates a `JLabel` with the remaining time. It also overrides the `done` method to show an information dialog and update the status label. The code is annotated with `@Override` and `@SuppressWarnings("serial")`.

```
src > J HitungMundurWorker.java > HitungMundurWorker > done()
10  public class HitungMundurWorker extends SwingWorker<String, Integer> {
40      protected void process(List<Integer> chunks) {
43          textAreaHasil.append("Hitung mundur: " + detik + "\n");
44          textAreaHasil.setCaretPosition(textAreaHasil.getDocument().getLength());
45          labelStatus.setText("Sisa " + detik + " detik");
46      }
47
48      @Override
49      protected void done() {
50          try {
51              String hasil = get();
52              textAreaHasil.append(hasil);
53              labelStatus.setText(text: "Status: Selesai");
54
55              JOptionPane.showMessageDialog(
56                  frame,
57                  hasil,
58                  title: "Informasi",
59                  JOptionPane.INFORMATION_MESSAGE
60              );
61
62          } catch (Exception ex) {
63              labelStatus.setText(text: "Status: Error");
64          } finally [
65              buttonMulaiTask.setEnabled(b: true);
66          ]
67      }
68 }
```

### c. Hasil Running



### d. Pembuktian UI Interaktif (tidak freeze)



6. Terdapat penyelesaian Praktikum Demo 5 SwingWorker + Cancelation (Anonymous Inner Class) yang disertai bukti hasil running dan pembuktian UI Interaktif (tidak freeze)
  - a. Kode Program

```
File Edit Selection View Go Run ... < > swingworker-demo
EXPLORER ... J Demo1.java J Demo4.java J HitungMundurWorker.java J Demo2.java J Demo3.java 1 J Demo5.java 1 X J Demo6.java Dv ...
src > J Demo5.java > Demo5
1 import java.awt.Color;
2 import java.awt.Dimension;
3 import java.awt.Font;
4 import java.awt.event.ActionEvent;
5 import java.util.List;
6 import java.util.concurrent.CancellationException;
7 import java.util.concurrent.ExecutionException;
8 import javax.swing.JButton;
9 import javax.swing.JFrame;
10 import javax.swing.JLabel;
11 import javax.swing.JOptionPane;
12 import javax.swing.JPanel;
13 import javax.swing.JProgressBar;
14 import javax.swing.JScrollPane;
15 import javax.swing.JTextArea;
16 import javax.swing.SwingUtilities;
17 import javax.swing.SwingWorker;
18 import javax.swing.UIManager;
19 import net.miginfocom.swing.MigLayout;
20
21 public class Demo5 extends JFrame {
22
23     private JPanel panelUtama;
24     private JButton buttonMulaiTask;
25     private JButton buttonCancel;
26     private JLabel labelStatus;
27     private JProgressBar progressBar;
28     private JTextArea textAreaHasil;
29     private SwingWorker<Integer, Integer> worker;
30
31     public Demo5() {
32         initializeUI();
33         setupEventHandlers();
34     }
35
36     private void initializeUI() {
37         setTitle("Demo 5: SwingWorker + Cancellation");
38         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
39         setPreferredSize(new Dimension(width: 720, height: 520));
40
41         panelUtama = new JPanel(
42             new MigLayout("fill, wrap 1, insets 20"));
43
44         addPanelTombol();
45         addLabelStatus();
46         addProgressBar();
47         addScrollTextArea();
48
49         add(panelUtama);
50         pack();
51         setLocationRelativeTo(null);
52     }
53
54     private void addPanelTombol() {
55         JPanel panelTombol = new JPanel(
56             new MigLayout("fill", "[grow] [grow]"));
57
58         JButton buttonMulai = new JButton("Mulai");
59         JButton buttonCancel = new JButton("Cancel");
60
61         panelTombol.add(buttonMulai);
62         panelTombol.add(buttonCancel);
63
64         panelUtama.add(panelTombol, "center");
65     }
66
67     private void addLabelStatus() {
68         JLabel labelStatus = new JLabel("Status: ");
69         panelUtama.add(labelStatus, "left");
70
71         labelStatus.setText("Tunggu...");
72     }
73
74     private void addProgressBar() {
75         JProgressBar progressBar = new JProgressBar(0, 100);
76         panelUtama.add(progressBar, "below, center");
77
78         progressBar.setStringPainted(true);
79     }
80
81     private void addScrollTextArea() {
82         JTextArea textAreaHasil = new JTextArea("Hasil: ");
83         panelUtama.add(textAreaHasil, "bottom, center");
84
85         textAreaHasil.setEditable(false);
86     }
87
88     protected void processTask(List<Integer> results) {
89         if (!buttonCancel.isEnabled()) {
90             return;
91         }
92
93         int sum = 0;
94         for (Integer result : results) {
95             sum += result;
96         }
97
98         textAreaHasil.append("Hasil: " + sum + "\n");
99
100        if (buttonCancel.isEnabled()) {
101            buttonCancel.setEnabled(false);
102            buttonMulai.setEnabled(true);
103        }
104    }
105
106    protected void cancel() {
107        if (buttonCancel.isEnabled()) {
108            worker.cancel(true);
109        }
110    }
111
112    public static void main(String[] args) {
113        SwingUtilities.invokeLater(() -> {
114            new Demo5().setVisible(true);
115        });
116    }
117}
```

Java Ready caniaye (4 hours ago) Ln 186, Col 2 Spaces: 4 UTF-8 CRLF {} Java Chat quota reached

S 33°C Cerah

File Edit Selection View Go Run ... < > swingworker-demo
J Demo1.java J Demo4.java J HitungMundurWorker.java J Demo2.java J Demo3.java 1 J Demo5.java 1 X J Demo6.java U J TaskWorker.java Dv ...
src > J Demo5.java > Demo5
21 public class Demo5 extends JFrame {
22
23 public Demo5() {
24 initializeUI();
25 setupEventHandlers();
26 }
27
28 private void initializeUI() {
29 setTitle("Demo 5: SwingWorker + Cancellation");
30 setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);
31 setPreferredSize(new Dimension(width: 720, height: 520));
32
33 panelUtama = new JPanel(
34 new MigLayout("fill, wrap 1, insets 20"));
35
36 addPanelTombol();
37 addLabelStatus();
38 addProgressBar();
39 addScrollTextArea();
40
41 add(panelUtama);
42 pack();
43 setLocationRelativeTo(null);
44 }
45
46 private void addPanelTombol() {
47 JPanel panelTombol = new JPanel(
48 new MigLayout("fill", "[grow] [grow]"));
49
50 JButton buttonMulai = new JButton("Mulai");
51 JButton buttonCancel = new JButton("Cancel");
52
53 panelTombol.add(buttonMulai);
54 panelTombol.add(buttonCancel);
55
56 panelUtama.add(panelTombol, "center");
57
58 buttonMulai.addActionListener(e -> {
59 if (!buttonCancel.isEnabled()) {
60 return;
61 }
62
63 worker.execute();
64 });
65
66 buttonCancel.addActionListener(e -> {
67 if (!buttonCancel.isEnabled()) {
68 return;
69 }
70
71 worker.cancel(true);
72 });
73 }
74
75 private void addLabelStatus() {
76 JLabel labelStatus = new JLabel("Status: ");
77 panelUtama.add(labelStatus, "left");
78
79 labelStatus.setText("Tunggu...");
80 }
81
82 private void addProgressBar() {
83 JProgressBar progressBar = new JProgressBar(0, 100);
84 panelUtama.add(progressBar, "below, center");
85
86 progressBar.setStringPainted(true);
87 }
88
89 private void addScrollTextArea() {
90 JTextArea textAreaHasil = new JTextArea("Hasil: ");
91 panelUtama.add(textAreaHasil, "bottom, center");
92
93 textAreaHasil.setEditable(false);
94 }
95
96 protected void processTask(List<Integer> results) {
97 if (!buttonCancel.isEnabled()) {
98 return;
99 }
100
101 int sum = 0;
102 for (Integer result : results) {
103 sum += result;
104 }
105
106 textAreaHasil.append("Hasil: " + sum + "\n");
107
108 if (buttonCancel.isEnabled()) {
109 buttonCancel.setEnabled(false);
110 buttonMulai.setEnabled(true);
111 }
112 }
113
114 protected void cancel() {
115 if (buttonCancel.isEnabled()) {
116 worker.cancel(true);
117 }
118 }
119
120 public static void main(String[] args) {
121 SwingUtilities.invokeLater(() -> {
122 new Demo5().setVisible(true);
123 });
124 }
125}

Java Ready caniaye (4 hours ago) Ln 186, Col 2 Spaces: 4 UTF-8 CRLF {} Java Chat quota reached

S 33°C Cerah

```
swingworker-demo
```

```
File Edit Selection View Go Run ... ↵ → 🔍 swingworker-demo
```

```
J Demo1.java J Demo4.java J HitungMundurWorker.java J Demo2.java J Demo3.java 1 J Demo5.java 1 X J Demo6.java U J TaskWorker.java D...
```

```
src > J Demo5java > Demo5
```

```
21  public class Demo5 extends JFrame {  
54  private void addPanelTombol() {  
57  buttonMulaiTask = new JButton(text: "Mulai Task (8 detik)");  
58  buttonCancel = new JButton(text: "Cancel");  
59  buttonMulaiTask.setFont(new Font(name: "Inter", Font.BOLD, size: 18));  
60  buttonCancel.setFont(new Font(name: "Inter", Font.BOLD, size: 18));  
61  buttonCancel.setEnabled(b: false);  
62  
63  panelTombol.add(buttonMulaiTask, constraints: "w 90%, growy");  
64  panelTombol.add(buttonCancel, constraints: "w 10%, growy");  
65  panelUtama.add(panelTombol, constraints: "h 20%, hmin pref, grow");  
66  
67 }  
68  
69 private void addLabelStatus() {  
70  labelStatus = new JLabel(text: "Status: Siap", JLabel.CENTER);  
71  labelStatus.setFont(new Font(name: "Inter", Font.BOLD, size: 24));  
72  labelStatus.setForeground(new Color(r: 0, g: 100, b: 0));  
73  panelUtama.add(labelStatus, constraints: "h 10%, hmin pref, growx");  
74  
75 }  
76  
77 private void addProgressBar() {  
78  progressBar = new JProgressBar(min: 0, max: 100);  
79  progressBar.setStringPainted(b: true);  
80  progressBar.setFont(new Font(name: "Inter", Font.BOLD, size: 18));  
81  progressBar.setValue(n: 0);  
82  panelUtama.add(progressBar, constraints: "h 10%, hmin pref, growx");  
83 }
```

Loading...

```
main* 01 2t 0 0 △ 2 🔍 Java: Ready
```

```
File Edit Selection View Go Run ... ↵ → 🔍 swingworker-demo
```

```
J Demo1.java J Demo4.java J HitungMundurWorker.java J Demo2.java J Demo3.java 1 J Demo5.java 1 X J Demo6.java U J TaskWorker.java D...
```

```
src > J Demo5java > Demo5
```

```
21  public class Demo5 extends JFrame {  
83  
84  private void addScrollTextArea() {  
85  textAreaHasil = new JTextArea();  
86  textAreaHasil.setEditable(b: false);  
87  textAreaHasil.setFont(new Font(name: "Consolas", Font.PLAIN, size: 14));  
88  JScrollPane scroll = new JScrollPane(textAreaHasil);  
89  panelUtama.add(scroll, constraints: "h 60%, growx");  
90 }  
91  
92 private void setupEventHandlers() {  
93  buttonMulaiTask.addActionListener(e -> mulaiProses());  
94  buttonCancel.addActionListener(e -> {  
95   if (worker != null && !worker.isDone()) {  
96    worker.cancel(mayInterruptIfRunning: true);  
97    addLog(text: "Proses dibatalkan oleh user...");  
98    buttonCancel.setEnabled(b: false);  
99    buttonMulaiTask.setEnabled(b: true);  
100 }  
101 })  
102 }  
103  
104 private void mulaiProses() {  
105  buttonMulaiTask.setEnabled(b: false);  
106  buttonCancel.setEnabled(b: true);  
107  progressBar.setValue(n: 0);  
108  textAreaHasil.setText(t: "");  
109  labelStatus.setText(text: "Status: Mulai bekerja");  
110  addLog(text: "Mengerjakan Background Task (8 detik)");  
111 }
```

caniaye (4 hours ago) Ln 186, Col 2 Spaces: 4 UTF-8 CRLF {} Java Chat quota reached

```
33°C Cerah 🔍 Search
```

106 PM 12/2/2025

The screenshot displays two Java code editors side-by-side, both titled "swingworker-demo".

**Left Editor (Demo5.java):**

```
src > J Demo5java > Demo5
21  public class Demo5 extends JFrame {
104    private void mulaiProses() {
111        worker = new SwingWorker<Integer, Integer>() {
112            @Override
113            protected Integer doInBackground() throws Exception {
114                int total = 100;
115                for (int i = 1; i <= total; i++) {
116                    if (isCancelled()) return null;
117                    Thread.sleep(100);
118                    publish(i);
119                }
120                return total;
121            }
122            @Override
123            protected void process(List<Integer> chunks) {
124                if (isCancelled()) return;
125                int progress = chunks.get(chunks.size() - 1);
126                progressBar.setValue(progress);
127
128                // Perbaikan perhitungan sisa detik
129                int remainingSeconds = (8 * (100 - progress)) / 100;
130
131                labelStatus.setText("Sisa " + remainingSeconds + " detik");
132                addLog("Persentase penggeraan task: " + progress + "%");
133            }
134            @Override
135            protected void done() {
136                try {
137                    Integer result = get();
138                    labelStatus.setText(text: "Status: Selesai");
139                    addLog("Background Task Selesai: " + result + "%");
140
141                    JOptionPane.showMessageDialog(Demo5.this,
142                        message: "Background Task Selesai",
143                        title: "Informasi", JOptionPane.INFORMATION_MESSAGE);
144
145                } catch (ExecutionException ex) {
146                    labelStatus.setText(text: "Status: Error");
147                    addLog("Error: " + ex.getCause().getMessage());
148                    JOptionPane.showMessageDialog(Demo5.this,
149                        message: "Terjadi kesalahan: " + ex.getMessage(),
150                        title: "Error", JOptionPane.ERROR_MESSAGE);
151
152                } catch (CancellationException ce) {
153                    labelStatus.setText(text: "Status: Dibatalkan");
154
155                } catch (InterruptedException e) {
156                    Thread.currentThread().interrupt();
157
158                } finally {
159                    buttonCancel.setEnabled(b: false);
160                    buttonMulaiTask.setEnabled(b: true);
161
162                }
163            }
164        }
165    }
166}
```

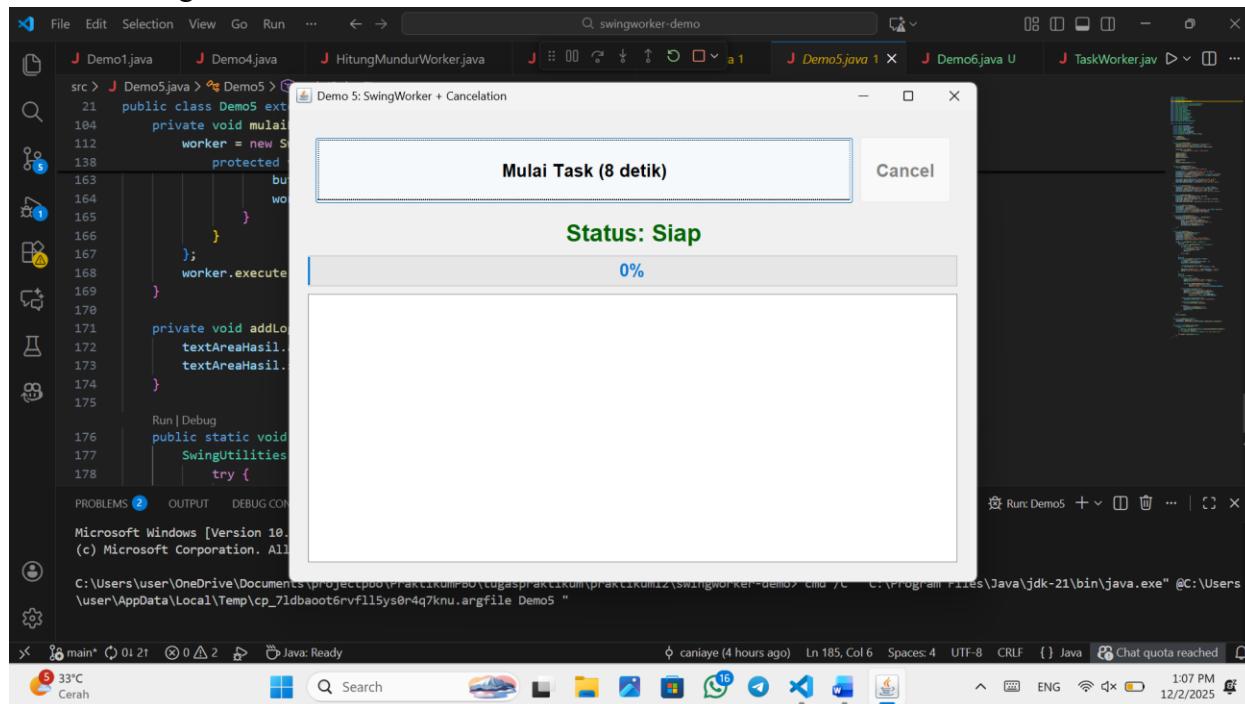
**Right Editor (Demo5.java):**

```
src > J Demo5java > Demo5
21  public class Demo5 extends JFrame {
104    private void mulaiProses() {
111        worker = new SwingWorker<Integer, Integer>() {
112            @Override
113            protected Integer doInBackground() throws Exception {
114                int total = 100;
115                for (int i = 1; i <= total; i++) {
116                    if (isCancelled()) return null;
117                    Thread.sleep(100);
118                    publish(i);
119                }
120                return total;
121            }
122            @Override
123            protected void process(List<Integer> chunks) {
124                if (isCancelled()) return;
125                int progress = chunks.get(chunks.size() - 1);
126                progressBar.setValue(progress);
127
128                // Perbaikan perhitungan sisa detik
129                int remainingSeconds = (8 * (100 - progress)) / 100;
130
131                labelStatus.setText("Sisa " + remainingSeconds + " detik");
132                addLog("Persentase penggeraan task: " + progress + "%");
133            }
134            @Override
135            protected void done() {
136                try {
137                    Integer result = get();
138                    labelStatus.setText(text: "Status: Selesai");
139                    addLog("Background Task Selesai: " + result + "%");
140
141                    JOptionPane.showMessageDialog(Demo5.this,
142                        message: "Background Task Selesai",
143                        title: "Informasi", JOptionPane.INFORMATION_MESSAGE);
144
145                } catch (ExecutionException ex) {
146                    labelStatus.setText(text: "Status: Error");
147                    addLog("Error: " + ex.getCause().getMessage());
148                    JOptionPane.showMessageDialog(Demo5.this,
149                        message: "Terjadi kesalahan: " + ex.getMessage(),
150                        title: "Error", JOptionPane.ERROR_MESSAGE);
151
152                } catch (CancellationException ce) {
153                    labelStatus.setText(text: "Status: Dibatalkan");
154
155                } catch (InterruptedException e) {
156                    Thread.currentThread().interrupt();
157
158                } finally {
159                    buttonCancel.setEnabled(b: false);
160                    buttonMulaiTask.setEnabled(b: true);
161
162                }
163            }
164        }
165    }
166}
```

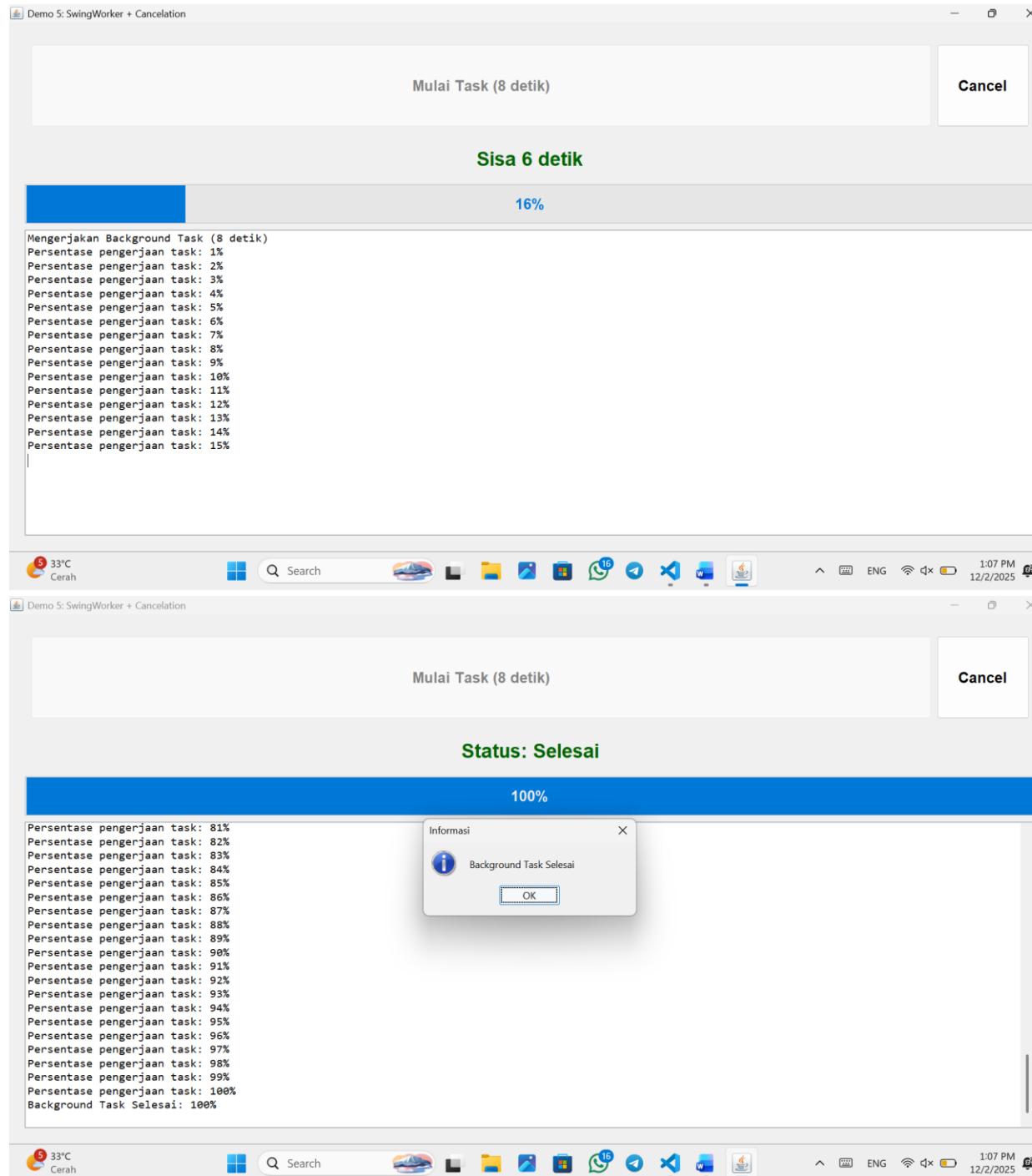
```
File Edit Selection View Go Run ... swingworker-demo
J Demo1.java J Demo4.java J HitungMundurWorker.java J Demo2.java J Demo3.java 1 J Demo5.java 1 X J Demo6.java U J TaskWorker.java D ...
src > J Demo5java > Demo5 > main(String[])
21  public class Demo5 extends JFrame {
104  private void mulaiProses() {
112    worker = new SwingWorker<Integer, Integer>() {
138      protected void done() {
163        buttonMulaiTask.setEnabled(b: true);
164        worker = null;
165      }
166    };
167    worker.execute();
168  }
169
170  private void addLog(String text) {
171    textAreaHasil.append(text + "\n");
172    textAreaHasil.setCaretPosition(textAreaHasil.getDocument().getLength());
173  }
174
175
Run | Debug
public static void main(String[] args) {
  SwingUtilities.invokeLater(() -> {
    try {
      UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
    } catch (Exception ignored) {
      // Mengabaikan pengecualian UIManager jika tidak dapat diatur
    }
    new Demo5().setVisible(b: true);
  });
}
185
186 }
```

main\* 01 2t 0 ▲ 2 Java: Ready caniaye (4 hours ago) Ln 185, Col 6 Spaces: 4 UTF-8 CRLF {} Java Chat quota reached 33°C Cerah 1:06 PM 12/2/2025

### b. Hasil Running

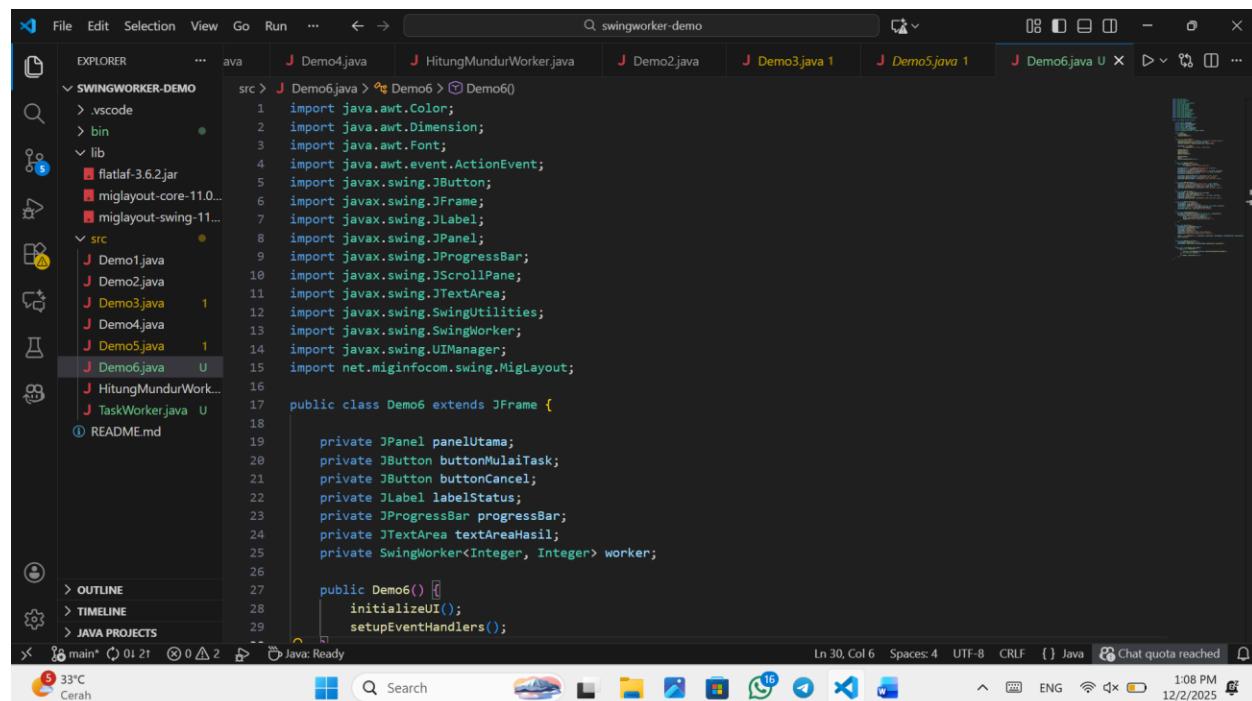


### c. Pembuktian UI Interaktif (tidak freeze)

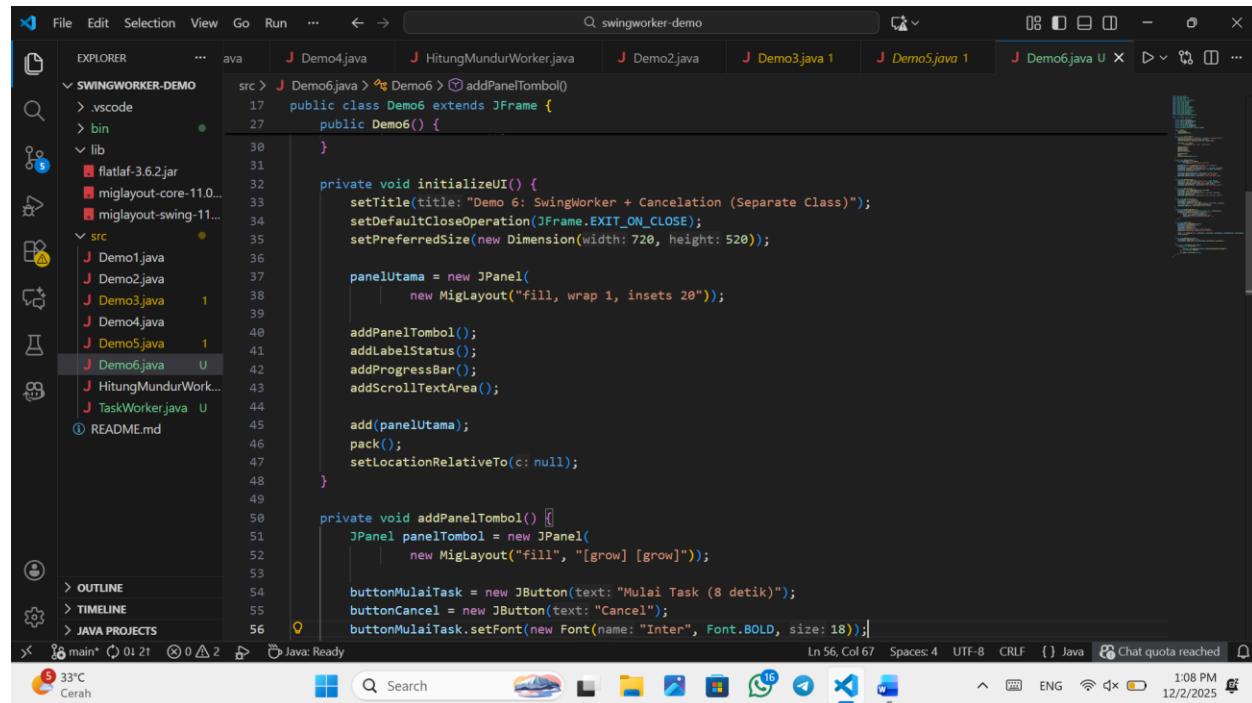


7. Terdapat penyelesaian Latihan Demo 6 SwingWorker + Cancelation (Separate Top-Level Class) yang terdiri dari 2 file (Demo6.java dan TaskWorker.java) disertai bukti hasil running dan pembuktian UI Interaktif (tidak freeze)

- a. Kode Program Demo6.java



```
File Edit Selection View Go Run ... < > swingworker-demo
EXPLORER J Demo4.java J HitungMundurWorker.java J Demo2.java J Demo3.java 1 J Demo5.java 1 J Demo6.java X
src > J Demo6.java > Demo6 > Demo6()
1 import java.awt.Color;
2 import java.awt.Dimension;
3 import java.awt.Font;
4 import java.awt.event.ActionEvent;
5 import javax.swing.JButton;
6 import javax.swing.JFrame;
7 import javax.swing.JLabel;
8 import javax.swing.JPanel;
9 import javax.swing.JProgressBar;
10 import javax.swing.JScrollPane;
11 import javax.swing.JTextArea;
12 import javax.swing.SwingUtilities;
13 import javax.swing.SwingWorker;
14 import javax.swing.UIManager;
15 import net.miginfocom.swing.MigLayout;
16
17 public class Demo6 extends JFrame {
18
19     private JPanel panelUtama;
20     private JButton buttonMulaiTask;
21     private JButton buttonCancel;
22     private JLabel labelStatus;
23     private JProgressBar progressBar;
24     private JTextArea textAreaHasil;
25     private SwingWorker<Integer, Integer> worker;
26
27     public Demo6() {
28         initializeUI();
29         setupEventHandlers();
30     }
31
32     private void initializeUI() {
33         setTitle(title: "Demo 6: SwingWorker + Cancellation (Separate Class)");
34         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
35         setPreferredSize(new Dimension(width: 720, height: 520));
36
37         panelUtama = new JPanel(
38             new MigLayout("fill, wrap 1, insets 20"));
39
40         addPanelTombol();
41         addLabelStatus();
42         addProgressBar();
43         addScrollTextArea();
44
45         add(panelUtama);
46         pack();
47         setLocationRelativeTo(c: null);
48     }
49
50     private void addPanelTombol() {
51         JPanel panelTombol = new JPanel(
52             new MigLayout("fill", "[grow] [grow]"));
53
54         buttonMulaiTask = new JButton(text: "Mulai Task (8 detik)");
55         buttonCancel = new JButton(text: "Cancel");
56         buttonMulaiTask.setFont(new Font(name: "Inter", Font.BOLD, size: 18));
57
58         panelTombol.add(buttonMulaiTask);
59         panelTombol.add(buttonCancel);
60
61         panelUtama.add(panelTombol, "center");
62     }
63
64     protected void processTask(Integer progress) {
65         progressBar.setValue(progress);
66         if (progress == 100) {
67             buttonCancel.setEnabled(false);
68         }
69     }
70
71     protected void taskCancelled() {
72         buttonCancel.setEnabled(true);
73     }
74
75     protected void taskError(Throwable error) {
76         labelStatus.setText(error.getMessage());
77     }
78
79     protected void taskFinalized(Integer result) {
80         textAreaHasil.append("Hasil: " + result);
81     }
82
83     private void setupEventHandlers() {
84         buttonMulaiTask.addActionListener(e -> {
85             worker.execute();
86         });
87
88         buttonCancel.addActionListener(e -> {
89             worker.cancel(true);
90         });
91     }
92
93     public static void main(String[] args) {
94         SwingUtilities.invokeLater(() -> {
95             new Demo6().setVisible(true);
96         });
97     }
98 }
```



```
File Edit Selection View Go Run ... < > swingworker-demo
EXPLORER J Demo4.java J HitungMundurWorker.java J Demo2.java J Demo3.java 1 J Demo5.java 1 J Demo6.java X
src > J Demo6.java > Demo6 > addPanelTombol()
17 public class Demo6 extends JFrame {
18     public Demo6() {
19
20     }
21
22     private void initializeUI() {
23         setTitle(title: "Demo 6: SwingWorker + Cancellation (Separate Class)");
24         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
25         setPreferredSize(new Dimension(width: 720, height: 520));
26
27         panelUtama = new JPanel(
28             new MigLayout("fill, wrap 1, insets 20"));
29
30         addPanelTombol();
31         addLabelStatus();
32         addProgressBar();
33         addScrollTextArea();
34
35         add(panelUtama);
36         pack();
37         setLocationRelativeTo(c: null);
38     }
39
40     private void addPanelTombol() {
41         JPanel panelTombol = new JPanel(
42             new MigLayout("fill", "[grow] [grow]"));
43
44         buttonMulaiTask = new JButton(text: "Mulai Task (8 detik)");
45         buttonCancel = new JButton(text: "Cancel");
46         buttonMulaiTask.setFont(new Font(name: "Inter", Font.BOLD, size: 18));
47
48         panelTombol.add(buttonMulaiTask);
49         panelTombol.add(buttonCancel);
50
51         panelUtama.add(panelTombol, "center");
52     }
53
54     protected void processTask(Integer progress) {
55         progressBar.setValue(progress);
56         if (progress == 100) {
57             buttonCancel.setEnabled(false);
58         }
59     }
60
61     protected void taskCancelled() {
62         buttonCancel.setEnabled(true);
63     }
64
65     protected void taskError(Throwable error) {
66         labelStatus.setText(error.getMessage());
67     }
68
69     protected void taskFinalized(Integer result) {
70         textAreaHasil.append("Hasil: " + result);
71     }
72
73     private void setupEventHandlers() {
74         buttonMulaiTask.addActionListener(e -> {
75             worker.execute();
76         });
77
78         buttonCancel.addActionListener(e -> {
79             worker.cancel(true);
80         });
81     }
82
83     public static void main(String[] args) {
84         SwingUtilities.invokeLater(() -> {
85             new Demo6().setVisible(true);
86         });
87     }
88 }
```

swingworker-demo

```

File Edit Selection View Go Run ...
EXPLORER ava J Demo4.java J HitungMundurWorker.java J Demo2.java J Demo3.java 1 J Demo5.java 1 J Demo6.java U
SWINGWORKER-DEMO > vscode > bin > src > J Demo6.java > Demo6 > addPanelTombol()
src > J Demo6.java > Demo6 > addPanelTombol()
public class Demo6 extends JFrame {
    private void addPanelTombol() {
        buttonCancel.setFont(new Font(name: "Inter", Font.BOLD, size: 18));
        buttonCancel.setEnabled(b: false);

        panelTombol.add(buttonMulaiTask, constraints: "w 90%, growy");
        panelTombol.add(buttonCancel, constraints: "w 10%, growy");
        panelUtama.add(panelTombol, constraints: "h 20%, hmin pref, grow");

        private void addLabelStatus() {
            labelStatus = new JLabel(text: "Status: Siap", JLabel.CENTER);
            labelStatus.setFont(new Font(name: "Inter", Font.BOLD, size: 24));
            labelStatus.setForeground(new Color(r: 0, g: 100, b: 0));
            panelUtama.add(labelStatus, constraints: "h 10%, hmin pref, growx");
        }

        private void addProgressBar() {
            progressBar = new JProgressBar(min: 0, max: 100);
            progressBar.setStringPainted(b: true);
            progressBar.setFont(new Font(name: "Inter", Font.BOLD, size: 18));
            progressBar.setValue(n: 0);
            panelUtama.add(progressBar, constraints: "h 10%, hmin pref, growx");
        }

        private void addScrollTextArea() {
            textAreaHasil = new JTextArea();
            textAreaHasil.setEditable(b: false);
            textAreaHasil.setFont(new Font(name: "Consolas", Font.PLAIN, size: 14));
        }
    }
}

33°C Cerah

```

swingworker-demo

```

File Edit Selection View Go Run ...
EXPLORER ava J Demo4.java J HitungMundurWorker.java J Demo2.java J Demo3.java 1 J Demo5.java 1 J Demo6.java U
SWINGWORKER-DEMO > vscode > bin > lib > src > J Demo6.java > Demo6 > addPanelTombol()
src > J Demo6.java > Demo6 > addPanelTombol()
public class Demo6 extends JFrame {
    private void addPanelTombol() {
        JScrollPane scroll = new JScrollPane(textAreaHasil);
        panelUtama.add(scroll, constraints: "h 60%, growx");
    }

    private void setupEventHandlers() {
        buttonMulaiTask.addActionListener((ActionEvent e) -> mulaiProses());
        buttonCancel.addActionListener((ActionEvent e) -> {
            if (worker != null && !worker.isDone()) {
                worker.cancel(mayInterruptIfRunning: true);
                addLog(text: "Proses dibatalkan oleh user...");
            }
        });
    }

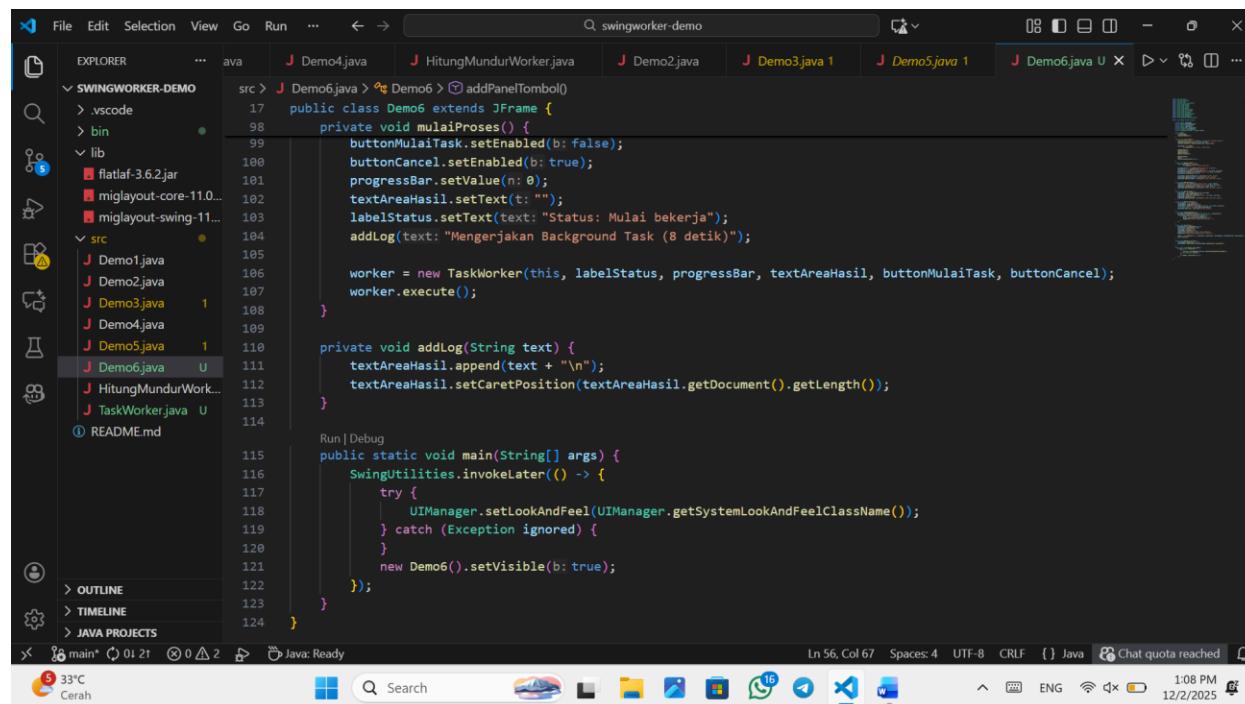
    private void mulaiProses() {
        buttonMulaiTask.setEnabled(b: false);
        buttonCancel.setEnabled(b: true);
        progressBar.setValue(n: 0);
        textAreaHasil.setText(t: "");
        labelStatus.setText(text: "Status: Mulai bekerja");
        addLog(text: "Mengerjakan Background Task (8 detik)");

        worker = new TaskWorker(this, labelStatus, progressBar, textAreaHasil, buttonMulaiTask, buttonCancel);
        worker.execute();
    }

    private void addLog(String text) {
        File Explorer
        33°C Cerah
    }
}

108 PM 12/2/2025

```



```

File Edit Selection View Go Run ... < > Q swingworker-demo
EXPLORER ... ava J Demo4.java J HitungMundurWorker.java J Demo2.java J Demo3.java 1 J Demo5.java 1 J Demo6.java U
src > J Demo6.java > addPanelTombol()
17 public class Demo6 extends JFrame {
98     private void mulaiProses() {
99         buttonMulaiTask.setEnabled(b: false);
100        buttonCancel.setEnabled(b: true);
101        progressBar.setValue(n: 0);
102        textAreaHasil.setText(t: "");
103        labelStatus.setText(text: "Status: Mulai bekerja");
104        addLog(text: "Mengerjakan Background Task (8 detik)");
105    }
106
107    private void addLog(String text) {
108        textAreaHasil.append(text + "\n");
109        textAreaHasil.setCaretPosition(textAreaHasil.getDocument().getLength());
110    }
111
112    public static void main(String[] args) {
113        SwingUtilities.invokeLater(() -> {
114            try {
115                UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
116            } catch (Exception ignored) {
117            }
118            new Demo6().setVisible(b: true);
119        });
120    }
121
122 }
123
124 }

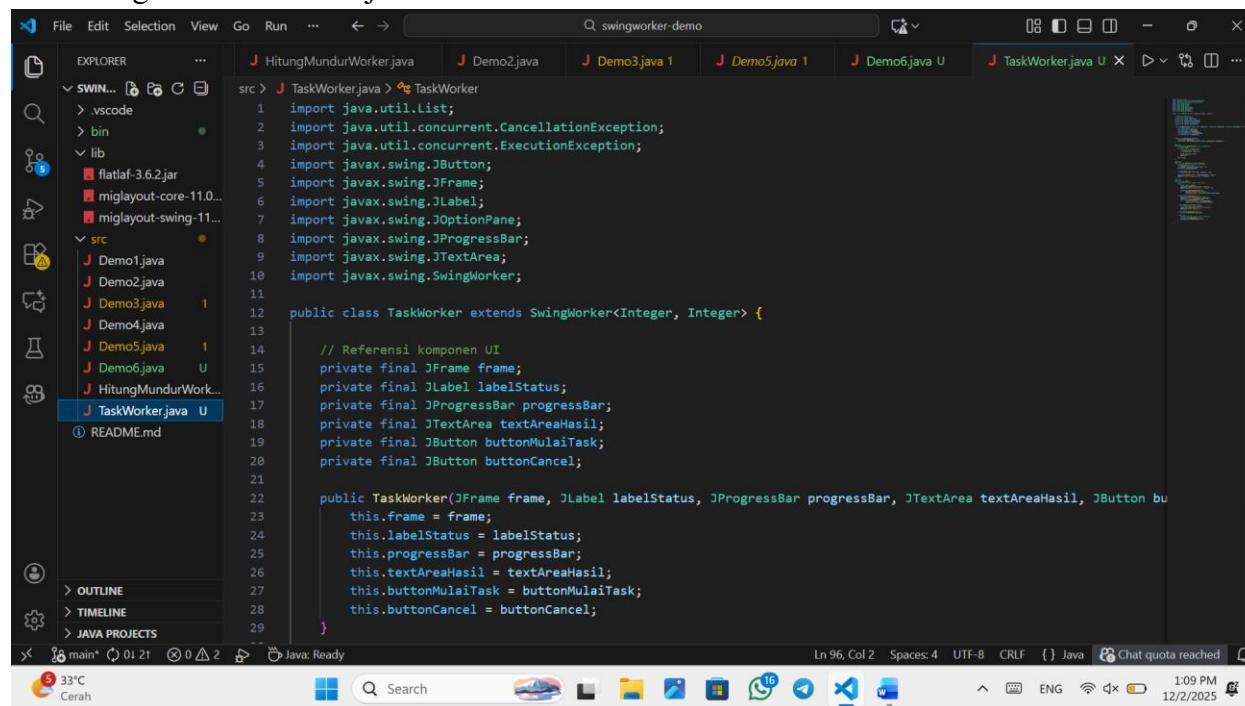
Run | Debug
public static void main(String[] args) {
    SwingUtilities.invokeLater(() -> {
        try {
            UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
        } catch (Exception ignored) {
        }
        new Demo6().setVisible(b: true);
    });
}

```

Ln 56, Col 67 Spaces: 4 UTF-8 CRLF {} Java Chat quota reached

33°C Cerah

## b. Kode Program TaskWorker.java



```

File Edit Selection View Go Run ... < > Q swingworker-demo
EXPLORER ... J HitungMundurWorker.java J Demo2.java J Demo3.java 1 J Demo5.java 1 J Demo6.java U J TaskWorker.java U
src > J TaskWorker.java > TaskWorker
1 import java.util.List;
2 import java.util.concurrent.CancellationException;
3 import java.util.concurrent.ExecutionException;
4 import javax.swing.JButton;
5 import javax.swing.JFrame;
6 import javax.swing.JLabel;
7 import javax.swing.JOptionPane;
8 import javax.swing.JProgressBar;
9 import javax.swing.JTextArea;
10 import javax.swing.SwingWorker;
11
12 public class TaskWorker extends SwingWorker<Integer, Integer> {
13
14     // Referensi komponen UI
15     private final JFrame frame;
16     private final JLabel labelStatus;
17     private final JProgressBar progressBar;
18     private final JTextArea textAreaHasil;
19     private final JButton buttonMulaiTask;
20     private final JButton buttonCancel;
21
22
23
24
25
26
27
28
29 }

public TaskWorker(JFrame frame, JLabel labelStatus, JProgressBar progressBar, JTextArea textAreaHasil, JButton bu
    this.frame = frame;
    this.labelStatus = labelStatus;
    this.progressBar = progressBar;
    this.textAreaHasil = textAreaHasil;
    this.buttonMulaiTask = buttonMulaiTask;
    this.buttonCancel = buttonCancel;
}


```

Ln 96, Col 2 Spaces: 4 UTF-8 CRLF {} Java Chat quota reached

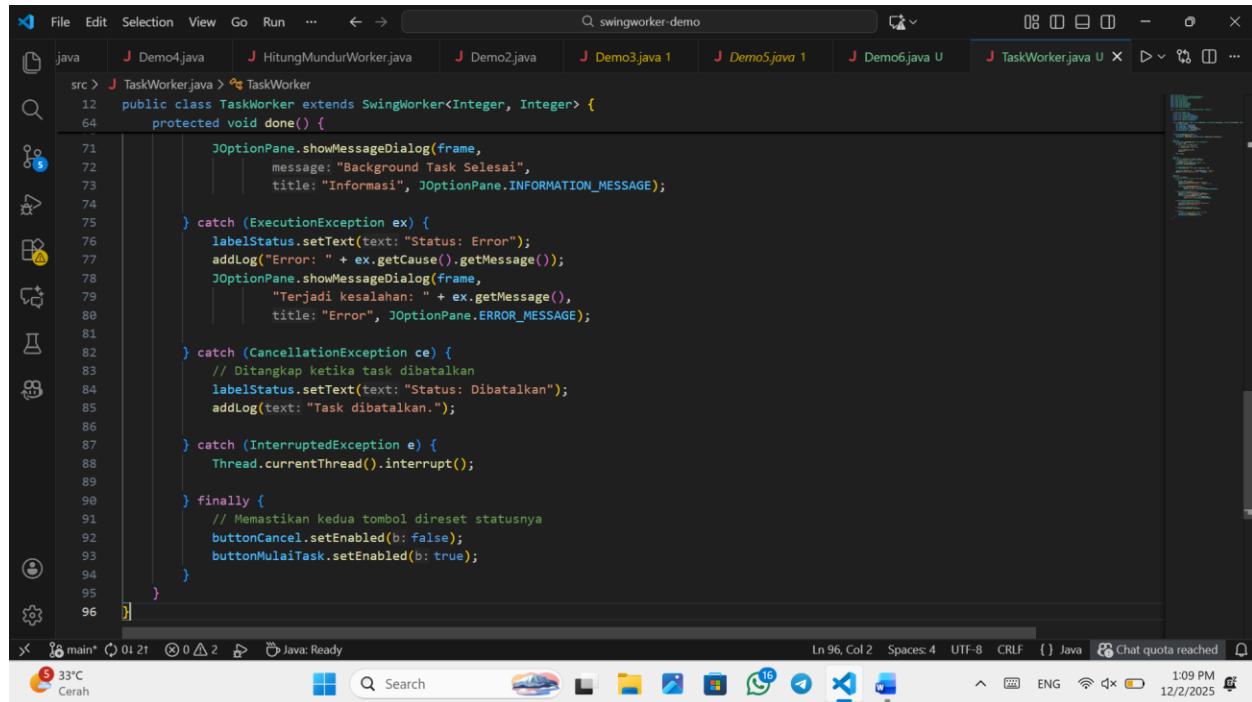
33°C Cerah

```
File Edit Selection View Go Run ... < > swingworker-demo
src > J TaskWorker.java > TaskWorker
12 public class TaskWorker extends SwingWorker<Integer, Integer> {
30
31     private void addLog(String text) {
32         textAreaHasil.append(text + "\n");
33         textAreaHasil.setCaretPosition(textAreaHasil.getDocument().getLength());
34     }
35
36     @Override
37     protected Integer doInBackground() throws Exception {
38         int total = 100;
39         for (int i = 1; i <= total; i++) {
40             // Mengacak status pembatalan
41             if (isCancelled()) return null;
42
43             Thread.sleep(millis: 80);
44             publish(i);
45         }
46         return total;
47     }
48
49     @Override
50     protected void process(List<Integer> chunks) {
51         // Berjalan di Event Dispatch Thread (EDT)
52         if (isCancelled()) return;
53         int progress = chunks.get(chunks.size() - 1);
54         progressBar.setValue(progress);
55
56         // Perhitungan sisa detik
57         int remainingSeconds = (8 * (100 - progress)) / 100;
58
59         labelStatus.setText("Sisa " + remainingSeconds + " detik");
60         addLog("Percentase pengerjaan task: " + progress + "%");
61     }
62
63     @Override
64     protected void done() {
65         // Berjalan di Event Dispatch Thread (EDT)
66         try {
67             Integer result = get();
68             labelStatus.setText(text: "Status: Selesai");
69             addLog("Background Task Selesai: " + result + "%");
70
71             JOptionPane.showMessageDialog(frame,
72                 message: "Background Task Selesai",
73                 title: "Informasi", JOptionPane.INFORMATION_MESSAGE);
74
75         } catch (ExecutionException ex) {
76             labelStatus.setText(text: "Status: Error");
77             addLog("Error: " + ex.getCause().getMessage());
78             JOptionPane.showMessageDialog(frame,
79                 "Terjadi kesalahan: " + ex.getMessage(),
80                 title: "Error", JOptionPane.ERROR_MESSAGE);
81
82         } catch (CancellationException ce) {
83             // Ditangkap ketika task dibatalkan
84             labelStatus.setText(text: "Status: Dibatalkan");
85         }
86     }
87
88 }
```

main\* 01 2t 0 ▲ 2 Java: Ready

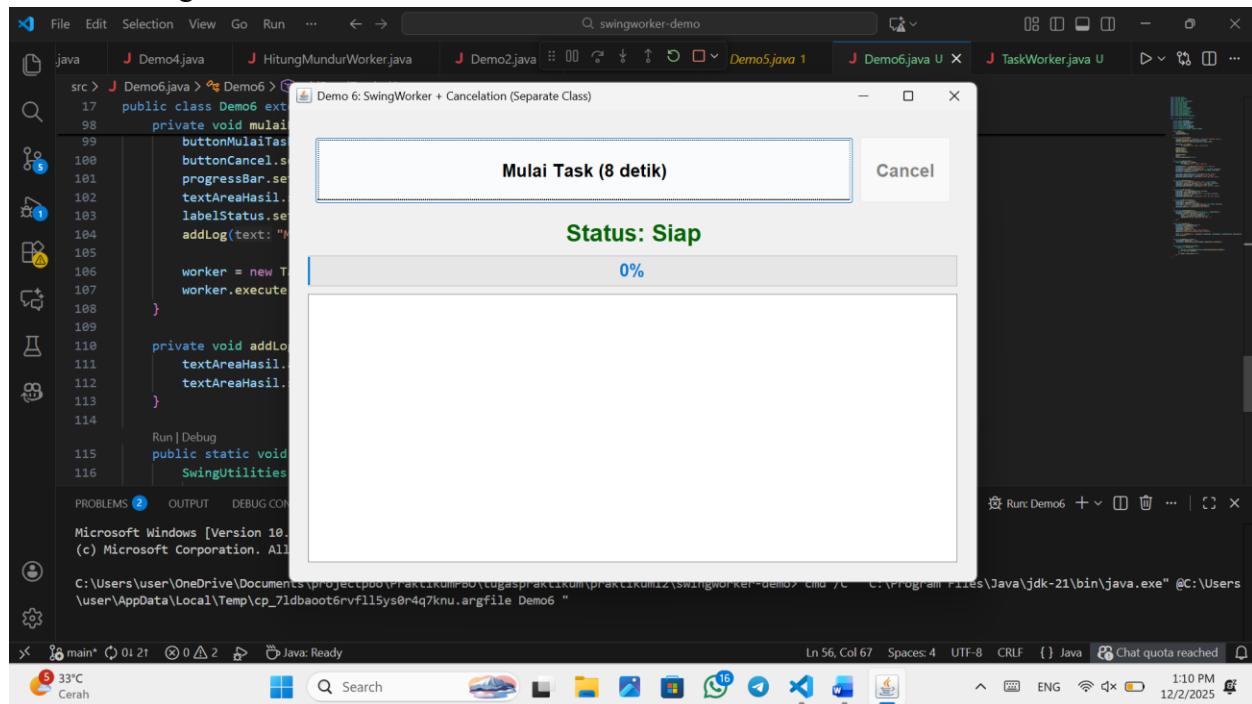
File Edit Selection View Go Run ... < > swingworker-demo
src > J TaskWorker.java > TaskWorker
12 public class TaskWorker extends SwingWorker<Integer, Integer> {
50 protected void process(List<Integer> chunks) {
58
59 labelStatus.setText("Sisa " + remainingSeconds + " detik");
60 addLog("Percentase pengerjaan task: " + progress + "%");
61 }
62
63 @Override
64 protected void done() {
65 // Berjalan di Event Dispatch Thread (EDT)
66 try {
67 Integer result = get();
68 labelStatus.setText(text: "Status: Selesai");
69 addLog("Background Task Selesai: " + result + "%");
70
71 JOptionPane.showMessageDialog(frame,
72 message: "Background Task Selesai",
73 title: "Informasi", JOptionPane.INFORMATION\_MESSAGE);
74
75 } catch (ExecutionException ex) {
76 labelStatus.setText(text: "Status: Error");
77 addLog("Error: " + ex.getCause().getMessage());
78 JOptionPane.showMessageDialog(frame,
79 "Terjadi kesalahan: " + ex.getMessage(),
80 title: "Error", JOptionPane.ERROR\_MESSAGE);
81
82 } catch (CancellationException ce) {
83 // Ditangkap ketika task dibatalkan
84 labelStatus.setText(text: "Status: Dibatalkan");
85 }
86 }
87
88 }

main\* 01 2t 0 ▲ 2 Java: Ready

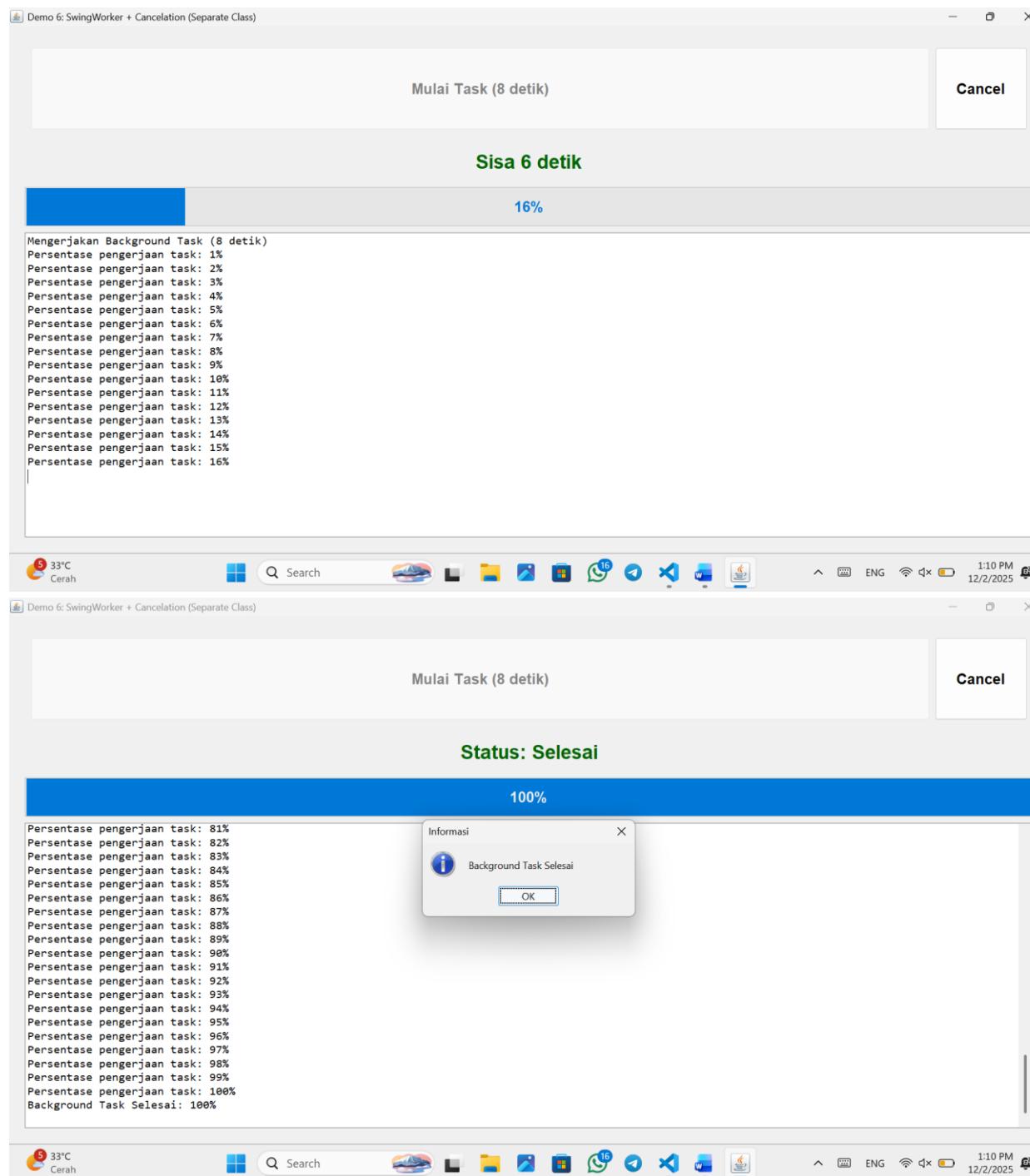


```
File Edit Selection View Go Run ... < > swingworker-demo
src > J Demo4.java J HitungMundurWorker.java J Demo2.java J Demo3.java I J Demo5.java 1 J Demo6.java U J TaskWorker.java U
java
12 public class TaskWorker extends SwingWorker<Integer, Integer> {
13     protected void done() {
14         ...
15     }
16     ...
17     ...
18     ...
19     ...
20     ...
21     ...
22     ...
23     ...
24     ...
25     ...
26     ...
27     ...
28     ...
29     ...
30     ...
31     ...
32     ...
33     ...
34     ...
35     ...
36     ...
37     ...
38     ...
39     ...
40     ...
41     ...
42     ...
43     ...
44     ...
45     ...
46     ...
47     ...
48     ...
49     ...
50     ...
51     ...
52     ...
53     ...
54     ...
55     ...
56     ...
57     ...
58     ...
59     ...
60     ...
61     ...
62     ...
63     ...
64     ...
65     ...
66     ...
67     ...
68     ...
69     ...
70     ...
71     ...
72     ...
73     ...
74     ...
75     ...
76     ...
77     ...
78     ...
79     ...
80     ...
81     ...
82     ...
83     ...
84     ...
85     ...
86     ...
87     ...
88     ...
89     ...
90     ...
91     ...
92     ...
93     ...
94     ...
95     ...
96 }
```

### c. Hasil Running



### d. Pembuktian UI Interaktif (tidak freeze)



8. Terdapat penyajian redaksi pengalaman pembelajaran yang didapat (secara pribadi) selama mempelajari materi SwingWorker secara keseluruhan (baik teori, praktikum, maupun latihan)

Dari perspektif saya sebagai mahasiswa yang baru mempelajari materi SwingWorker, pengalaman ini benar-benar memberikan pemahaman yang jelas mengenai masalah kritis dalam pengembangan aplikasi Swing, yaitu UI *freezing* akibat menempatkan operasi yang

memakan waktu pada Event Dispatch Thread (EDT). Praktikum dengan berbagai implementasi (dari *Anonymous Class* hingga *Separate Top-Level Class* seperti Demo6 dan TaskWorker) menegaskan bahwa SwingWorker adalah pola desain wajib untuk menjaga *user experience*, di mana *method doInBackground()* memindahkan beban kerja ke *background thread*, sementara *publish()* dan *process()* memastikan *progress* diperbarui secara aman dan *real-time* di EDT. Selain itu, mempelajari pembatalan (*cancellation*) dan penanganan *CancellationException* mengajarkan saya pentingnya membuat aplikasi yang tangguh dan dapat diinterupsi dengan bersih, sementara penggunaan *Separate Top-Level Class* seperti TaskWorker menjadi praktik terbaik untuk menjaga kode tetap modular dan *maintainable*.