



DAFTAR ISI

DAFTAR ISI.....	1
A. IDENTITAS.....	2
B. HASIL PRAKTIKUM/ TUGAS.....	2
1. PRAKTIKUM	3
a. Demo 1	3
b. Demo 2	8
c. Demo 3	13
d. Demo 4	18
e. Hitung Mundur Worker.....	22
f. Demo 5	24
2. LATIHAN	30
a. Task Worker.....	30
b. Demo 6	32
C. Hasil Pembelajaran	38



LAPORAN PRAKTIKUM

Halaman
2 of 38

A. IDENTITAS

NIM : 2403001
Nama Lengkap : Siti Sa'adah
Kelas : D3 TI 2A
Program Studi : D3 TEKNIK INFORMATIKA
Jurusan : TEKNIK INFORMATIKA

B. HASIL PRAKTIKUM/ TUGAS

[foto]



1. PRAKTIKUM

a. Demo 1

- Kode Program

```
package Praktikum;
import java.awt.*;
import java.awt.event.ActionEvent;

import javax.swing.*;

import net.miginfocom.swing.MigLayout;

public class Demo1 extends JFrame {

    private JPanel panelUtama;
    private JButton buttonMulaiTask;
    private JLabel labelStatus;
    private JTextArea textAreaHasil;

    public Demo1() {
        initializeUI();
        setupEventHandlers();
    }

    private void initializeUI() {
        setTitle("Demo 1: UI Freeze");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setPreferredSize(new Dimension(560, 420));

        panelUtama = new JPanel(
            new MigLayout(
                "fill, wrap 1, insets 20",
                "[grow]"
            )
        );

        buttonMulaiTask = new JButton("Mulai Task (8 detik)");
        buttonMulaiTask.setFont(new Font("Inter", Font.BOLD, 18));
        panelUtama.add(buttonMulaiTask, "growx, h 20%");

        labelStatus = new JLabel("Status: Siap", JLabel.CENTER);
        labelStatus.setFont(new Font("Inter", Font.PLAIN, 22));
        labelStatus.setForeground(new Color(0, 102, 204));
        panelUtama.add(labelStatus, "growx, h 20%");
    }
}
```



LAPORAN PRAKTIKUM

Halaman
4 of 38

```
textAreaHasil = new JTextArea(10, 50);
textAreaHasil.setEditable(false);
textAreaHasil.setFont(new Font("Consolas", Font.PLAIN, 15));
JScrollPane scroll = new JScrollPane(textAreaHasil);
panelUtama.add(scroll, "growx, h 60%");

add(panelUtama);
pack();
setLocationRelativeTo(null);
}

private void setupEventHandlers() {
    buttonMulaiTask.addActionListener((ActionEvent e) -> {
        buttonMulaiTask.setEnabled(false);
        labelStatus.setText("Status: Sedang bekerja");
        textAreaHasil.setText("Mengerjakan Background Task (8 detik)\n");

        for (int i = 8; i >= 0; i--) {
            final int seconds = i;
            SwingUtilities.invokeLater(() -> {
                textAreaHasil.append("Hitung mundur: " + seconds + "\n");
                textAreaHasil.setCaretPosition(textAreaHasil.getDocument()
.length());
                labelStatus.setText("Sisa " + seconds + " detik");
            });

            try {
                Thread.sleep(1000);
            } catch (InterruptedException ex) {
                Thread.currentThread().interrupt();
            }
        }
        textAreaHasil.append("Background Task selesai\n");
        labelStatus.setText("Status: Selesai");
        JOptionPane.showMessageDialog(
            this,
            "Background Task Selesai",
            "Informasi",
            JOptionPane.INFORMATION_MESSAGE
        );
        buttonMulaiTask.setEnabled(true);
    });
}

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



LAPORAN PRAKTIKUM

Halaman
5 of 38

```
        UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
    } catch (Exception e) { }

    new Demo1().setVisible(true);

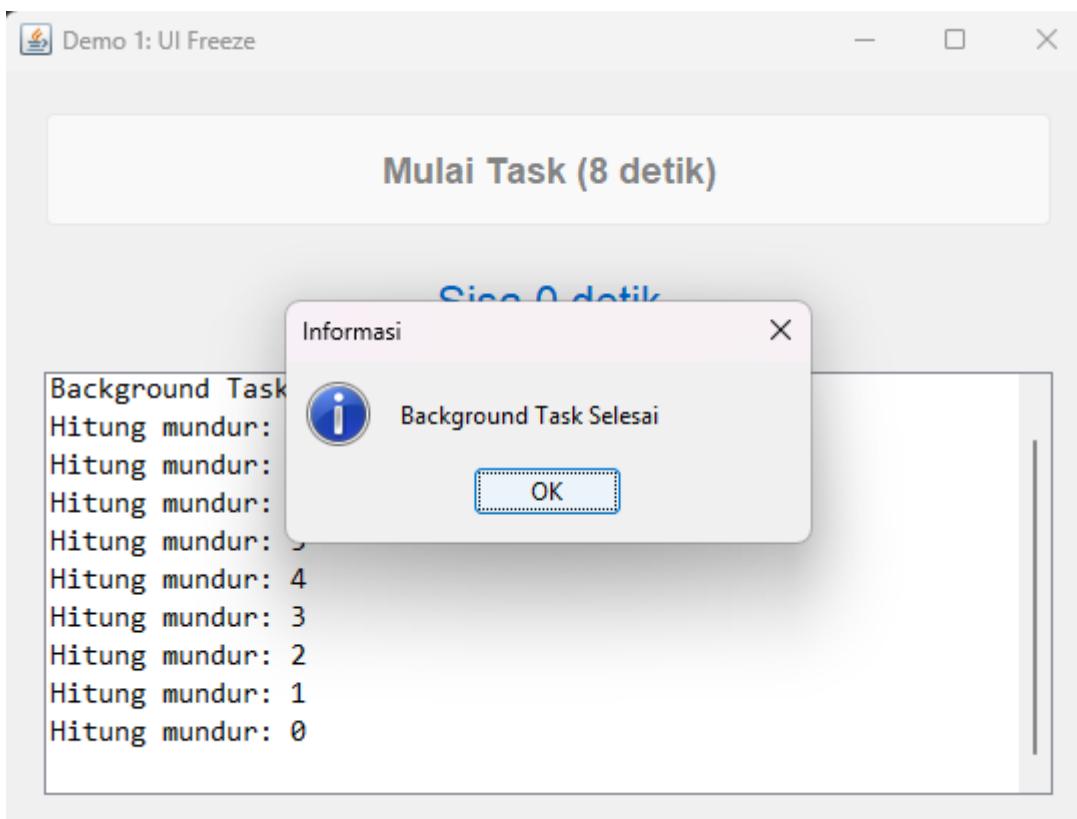
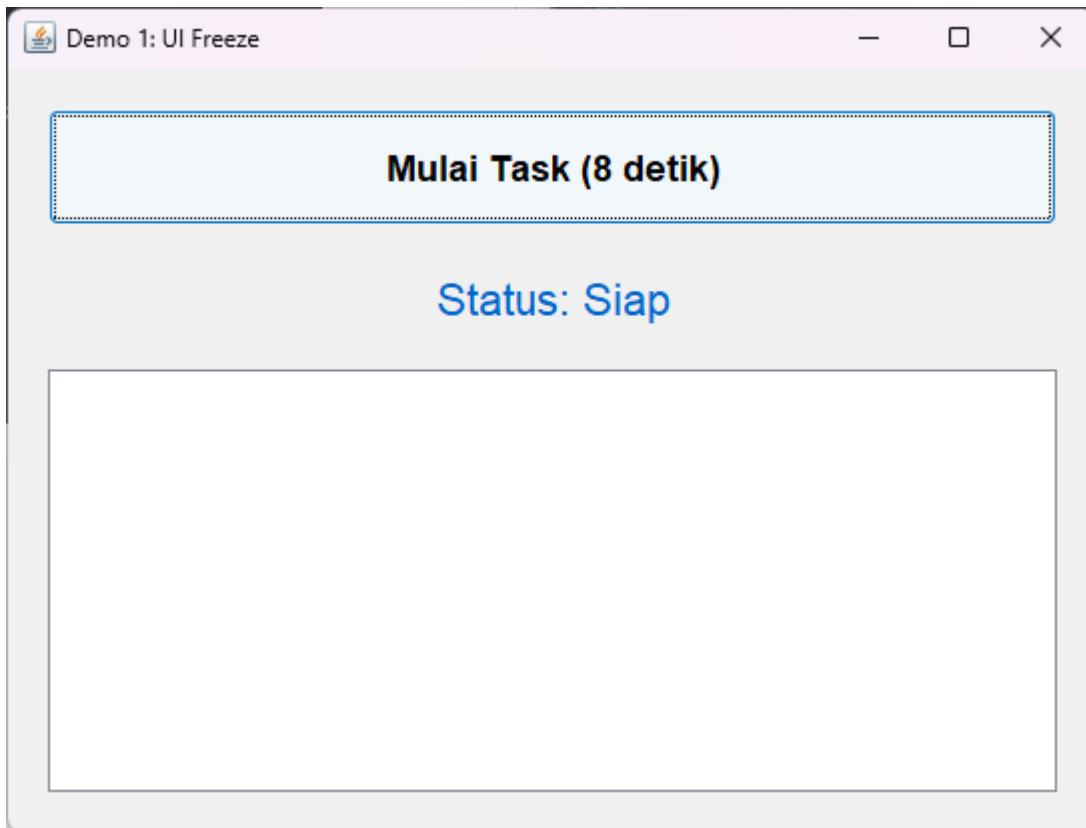
}
}
```



LAPORAN PRAKTIKUM

Halaman
6 of 38

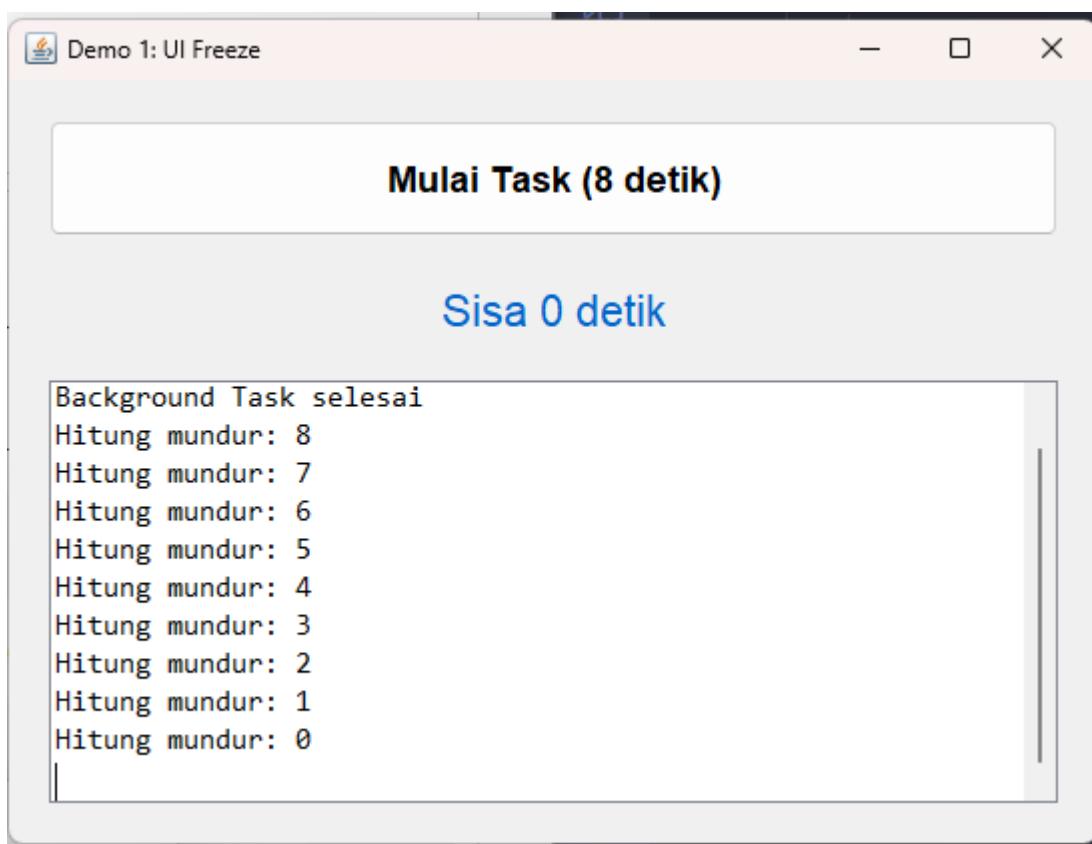
- Hasil Run





LAPORAN PRAKTIKUM

Halaman
7 of 38





b. Demo 2

- Kode Program

```
package Praktikum;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.Font;
import java.util.List;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import javax.swing.SwingUtilities;
import javax.swing.SwingWorker;
import javax.swing.UIManager;

import net.miginfocom.swing.MigLayout;

public class Demo2 extends JFrame {

    private JPanel panelUtama;
    private JButton buttonMulaiTask;
    private JLabel labelStatus;
    private JTextArea textAreaHasil;
    private SwingWorker<String, Integer> worker;

    public Demo2() {
        initializeUI();
        setupEventHandlers();
    }

    private void initializeUI() {
        setTitle("Demo 2: SwingWorker (Anonymous Inner Class)");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setPreferredSize(new Dimension(600, 450));

        panelUtama = new JPanel(
            new MigLayout("fill, wrap 1, insets 20")
        );

        buttonMulaiTask = new JButton("Mulai Task (8 detik)");
        buttonMulaiTask.setFont(new Font("Inter", Font.BOLD, 18));
    }
}
```



LAPORAN PRAKTIKUM

Halaman
9 of 38

```
panelUtama.add(buttonMulaiTask, "growx, h 20%");

labelStatus = new JLabel("Status: Siap", JLabel.CENTER);
labelStatus.setFont(new Font("Inter", Font.PLAIN, 22));
labelStatus.setForeground(new Color(0, 128, 0));
panelUtama.add(labelStatus, "growx, h 20%");

textAreaHasil = new JTextArea();
textAreaHasil.setEditable(false);
textAreaHasil.setFont(new Font("Consolas", Font.PLAIN, 15));
JScrollPane scroll = new JScrollPane(textAreaHasil);
panelUtama.add(scroll, "growx, h 60%");

add(panelUtama);
pack();
setLocationRelativeTo(null);
}

private void setupEventHandlers() {
    buttonMulaiTask.addActionListener(e -> {
        buttonMulaiTask.setEnabled(false);
        labelStatus.setText("Status: Mulai bekerja");
        textAreaHasil.setText("Mengerjakan Background Task (8 detik)\n");

        worker = new SwingWorker<String, Integer>() {
            @Override
            protected String doInBackground() throws Exception {
                for (int i = 8; i >= 0; i--) {
                    Thread.sleep(1000);
                    publish(i);
                }
                return "Background Task Selesai";
            }

            @Override
            protected void process(List<Integer> chunks) {
                int detik = chunks.get(chunks.size() - 1);
                textAreaHasil.append("Hitung mundur: " + detik + "\n");
                textAreaHasil.setCaretPosition(textAreaHasil.getDocument()
                .getLength());
                labelStatus.setText("Sisa " + detik + " detik");
            }

            @Override
            protected void done() {
                try {
                    String hasil = get();
                }
                catch (Exception e) {
                }
            }
        };
    }
}
```



LAPORAN PRAKTIKUM

Halaman
10 of 38

```
        textAreaHasil.append(hasil);
        labelStatus.setText("Status: Selesai");
        JOptionPane.showMessageDialog(
            Demo2.this,
            hasil,
            "Informasi",
            JOptionPane.INFORMATION_MESSAGE
        );
    } catch (Exception ex) {
        labelStatus.setText("Status: Error");
    } finally {
        buttonMulaiTask.setEnabled(true);
    }
}

worker.execute();
});

}

public static void main(String[] args) {
    SwingUtilities.invokeLater(() -> {
        try {
            UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
        } catch (Exception e) {
            System.err.println(e.getMessage());
        }
        new Demo2().setVisible(true);
    });
}
```



LAPORAN PRAKTIKUM

Halaman
11 of 38

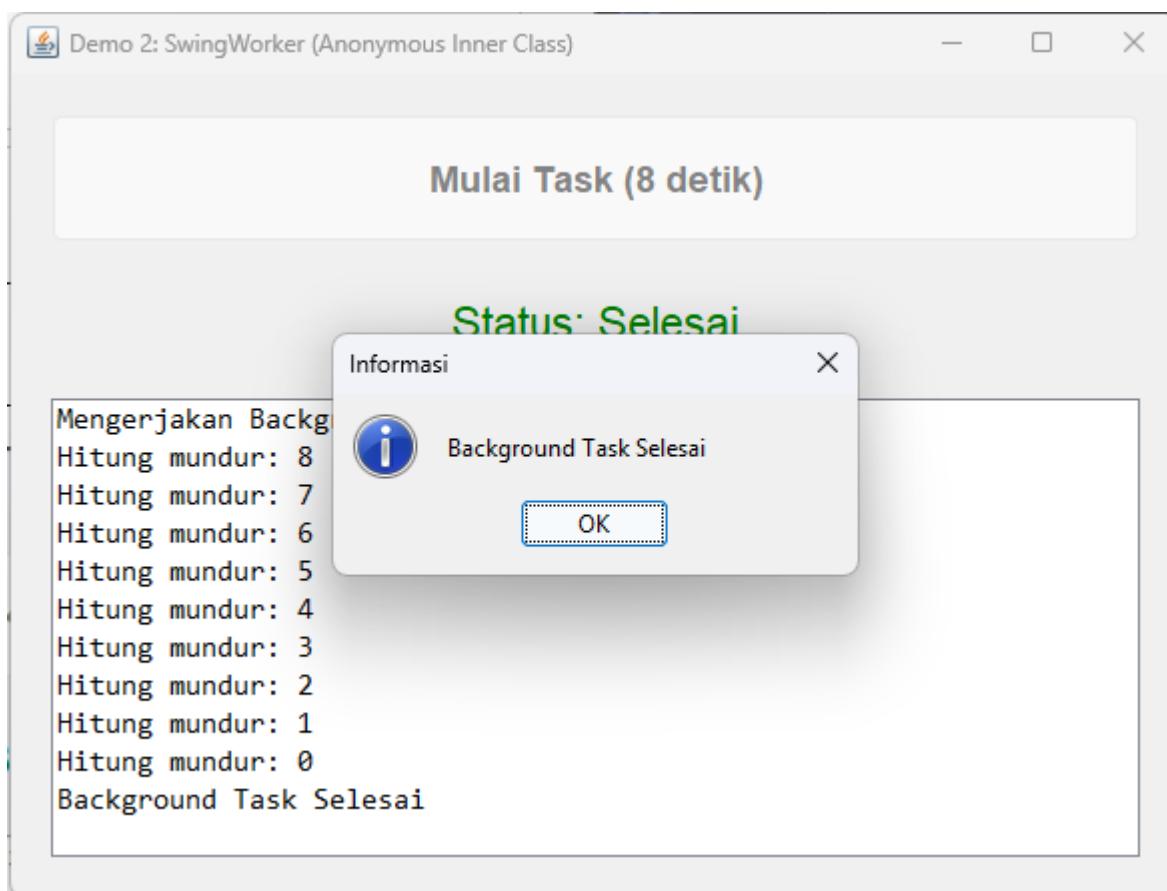
- Hasil Run





LAPORAN PRAKTIKUM

Halaman
12 of 38





c. Demo 3

- Kode Program

```
package Praktikum;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.Font;
import java.util.List;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import javax.swing.SwingUtilities;
import javax.swing.SwingWorker;
import javax.swing.UIManager;

import net.miginfocom.swing.MigLayout;

public class Demo3 extends JFrame {

    private JPanel panelUtama;
    private JButton buttonMulaiTask;
    private JLabel labelStatus;
    private JTextArea textAreaHasil;

    public Demo3() {
        initializeUI();
        setupEventHandlers();
    }

    private void initializeUI() {
        setTitle("Demo 3: SwingWorker (Static Nested Class)");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setPreferredSize(new Dimension(600, 450));

        panelUtama = new JPanel(
            new MigLayout("fill, wrap 1, insets 20")
        );

        buttonMulaiTask = new JButton("Mulai Task (8 detik)");
        buttonMulaiTask.setFont(new Font("Inter", Font.BOLD, 18));
        panelUtama.add(buttonMulaiTask, "growx, h 20%");
    }
}
```



LAPORAN PRAKTIKUM

Halaman
14 of 38

```
labelStatus = new JLabel("Status: Siap", JLabel.CENTER);
labelStatus.setFont(new Font("Inter", Font.PLAIN, 22));
labelStatus.setForeground(new Color(0, 128, 0));
panelUtama.add(labelStatus, "growx, h 20%");

textAreaHasil = new JTextArea();
textAreaHasil.setEditable(false);
textAreaHasil.setFont(new Font("Consolas", Font.PLAIN, 15));
JScrollPane scroll = new JScrollPane(textAreaHasil);
panelUtama.add(scroll, "growx, h 60%");

add(panelUtama);
pack();
setLocationRelativeTo(null);
}

private void setupEventHandlers() {
    buttonMulaiTask.addActionListener(
        e -> new HitungMundurWorker(this).execute()
    );
}

private static class HitungMundurWorker extends SwingWorker<String,
Integer> {
    private final Demo3 frame;

    private HitungMundurWorker(Demo3 frame) {
        this.frame = frame;
    }

    @Override
    protected String doInBackground() throws Exception {
        frame.buttonMulaiTask.setEnabled(false);
        frame.labelStatus.setText("Status: Mulai bekerja");
        frame.textAreaHasil.setText("Mengerjakan Background Task (8
detik)\n");

        for (int i = 8; i >= 0; i--) {
            Thread.sleep(1000);
            publish(i);
        }
        return "Background Task Selesai";
    }

    @Override
    protected void process(List<Integer> chunks) {
        int detik = chunks.get(chunks.size() - 1);
```



LAPORAN PRAKTIKUM

Halaman
15 of 38

```
        frame.textAreaHasil.append("Hitung mundur: " + detik + "\n");
        frame.textAreaHasil.setCaretPosition(frame.textAreaHasil.getDocument().getLength());
        frame.labelStatus.setText("Sisa " + detik + " detik");
    }

@Override
protected void done() {
    try {
        String hasil = get();
        frame.textAreaHasil.append(hasil);
        frame.labelStatus.setText("Status: Selesai");
        JOptionPane.showMessageDialog(frame, hasil, "Informasi",
JOptionPane.INFORMATION_MESSAGE);
    } catch (Exception ex) {
        frame.labelStatus.setText("Status: Error");
    } finally {
        frame.buttonMulaiTask.setEnabled(true);
    }
}

public static void main(String[] args) {
    SwingUtilities.invokeLater(() -> {
        try {
            UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
        } catch (Exception e) {
            System.err.println(e.getMessage());
        }
        new Demo3().setVisible(true);
    });
}
```



LAPORAN PRAKTIKUM

Halaman
16 of 38

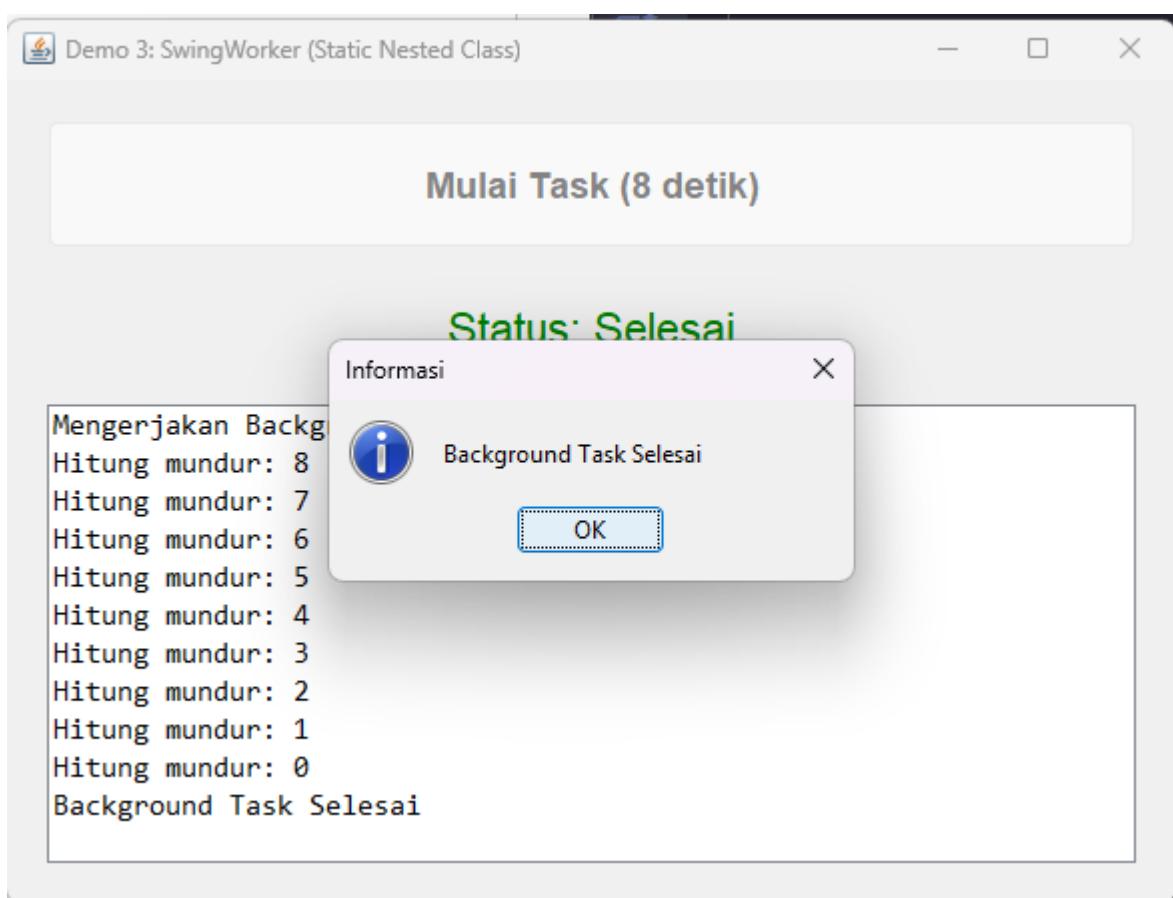
- Hasil Run





LAPORAN PRAKTIKUM

Halaman
17 of 38





d. Demo 4

- Kode Program

```
package Praktikum;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.Font;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import javax.swing.SwingUtilities;
import javax.swing.UIManager;

import net.miginfocom.swing.MigLayout;

public class Demo4 extends JFrame {

    private JPanel panelUtama;
    private JButton buttonMulaiTask;
    private JLabel labelStatus;
    private JTextArea textAreaHasil;

    public Demo4() {
        initializeUI();
        setupEventHandlers();
    }

    private void initializeUI() {
        setTitle("Demo 4: SwingWorker (Separate Top-Level Class)");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setPreferredSize(new Dimension(600, 450));

        panelUtama = new JPanel(new MigLayout("fill, wrap 1, insets 20"));

        buttonMulaiTask = new JButton("Mulai Task (8 detik)");
        buttonMulaiTask.setFont(new Font("Inter", Font.BOLD, 18));
        panelUtama.add(buttonMulaiTask, "growx, h 20%");

        labelStatus = new JLabel("Status: Siap", JLabel.CENTER);
        labelStatus.setFont(new Font("Inter", Font.PLAIN, 22));
        labelStatus.setForeground(new Color(0, 128, 0));
        panelUtama.add(labelStatus, "growx, h 20%");
    }
}
```



LAPORAN PRAKTIKUM

Halaman
19 of 38

```
textAreaHasil = new JTextArea();
textAreaHasil.setEditable(false);
textAreaHasil.setFont(new Font("Consolas", Font.PLAIN, 15));
JScrollPane scroll = new JScrollPane(textAreaHasil);
panelUtama.add(scroll, "growx, h 60%");

add(panelUtama);
pack();
setLocationRelativeTo(null);
}

private void setupEventHandlers() {
    buttonMulaiTask.addActionListener(e -> {
        buttonMulaiTask.setEnabled(false);
        new HitungMundurWorker(this, labelStatus, textAreaHasil,
buttonMulaiTask).execute();
    });
}

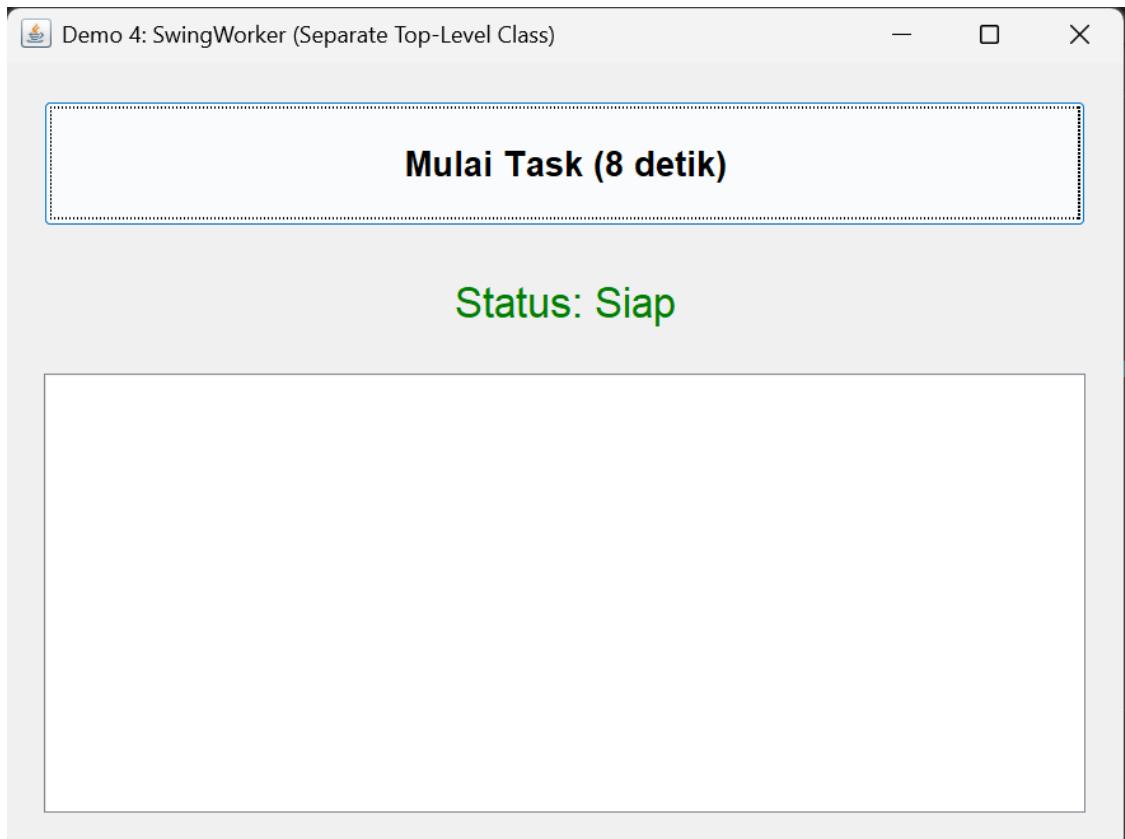
public static void main(String[] args) {
    SwingUtilities.invokeLater(() -> {
        try {
            UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
        } catch (Exception e) {
            System.err.println(e.getMessage());
        }
        new Demo4().setVisible(true);
    });
}
```



LAPORAN PRAKTIKUM

Halaman
20 of 38

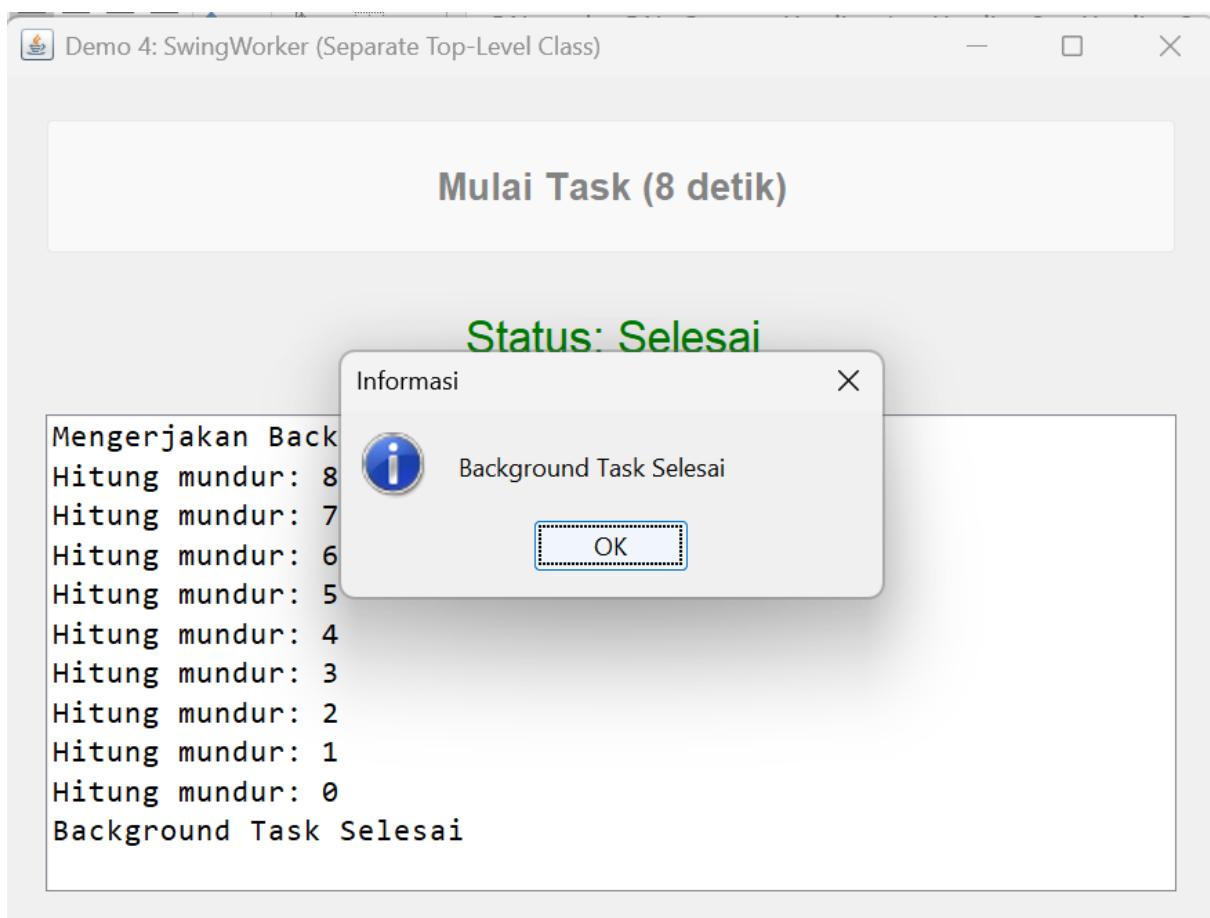
- Hasil Run





LAPORAN PRAKTIKUM

Halaman
21 of 38





e. Hitung Mundur Worker

- Kode Program

```
package Praktikum;
import java.util.List;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JTextArea;
import javax.swing.SwingWorker;

public class HitungMundurWorker extends SwingWorker<String, Integer> {
    private final JFrame frame;
    private final JLabel labelStatus;
    private final JTextArea textAreaHasil;
    private final JButton buttonMulaiTask;

    public HitungMundurWorker(JFrame frame, JLabel labelStatus, JTextArea textAreaHasil, JButton buttonMulaiTask) {
        this.frame = frame;
        this.labelStatus = labelStatus;
        this.textAreaHasil = textAreaHasil;
        this.buttonMulaiTask = buttonMulaiTask;
    }

    @Override
    protected String doInBackground() throws Exception {
        buttonMulaiTask.setEnabled(false);
        labelStatus.setText("Status: Mulai bekerja");
        textAreaHasil.setText("Mengerjakan Background Task (8 detik)\n");

        for (int i = 8; i >= 0; i--) {
            Thread.sleep(1000);
            publish(i);
        }
        return "Background Task Selesai";
    }

    @Override
    protected void process(List<Integer> chunks) {
        int detik = chunks.get(chunks.size() - 1);
        textAreaHasil.append("Hitung mundur: " + detik + "\n");
        textAreaHasil.setCaretPosition(textAreaHasil.getDocument().getLength());
    }

    labelStatus.setText("Sisa " + detik + " detik");
}
```



LAPORAN PRAKTIKUM

Halaman
23 of 38

```
}

@Override
protected void done() {
    try {
        String hasil = get();
        textAreaHasil.append(hasil);
        labelStatus.setText("Status: Selesai");
        JOptionPane.showMessageDialog(frame, hasil, "Informasi",
JOptionPane.INFORMATION_MESSAGE);
    } catch (Exception ex) {
        labelStatus.setText("Status: Error");
    } finally {
        buttonMulaiTask.setEnabled(true);
    }
}

}
```



f. Demo 5

- Kode Program

```
package Praktikum;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.Font;
import java.util.List;
import java.util.concurrent.CancellationException;
import java.util.concurrent.ExecutionException;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JProgressBar;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import javax.swing.SwingUtilities;
import javax.swing.SwingWorker;
import javax.swing.UIManager;

import net.miginfocom.swing.MigLayout;

public class Demo5 extends JFrame {

    private JPanel panelUtama;
    private JButton buttonMulaiTask;
    private JButton buttonCancel;
    private JLabel labelStatus;
    private JProgressBar progressBar;
    private JTextArea textAreaHasil;
    private SwingWorker<Integer, Integer> worker;

    public Demo5() {
        initializeUI();
        setupEventHandlers();
    }

    private void initializeUI() {
        setTitle("Demo 5: SwingWorker + Cancelation");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setPreferredSize(new Dimension(720, 520));

        panelUtama = new JPanel(
            new MigLayout("fill, wrap 1, insets 20")
        )
    }
}
```



LAPORAN PRAKTIKUM

Halaman
25 of 38

```
);

addPanelTombol();
addLabelStatus();
addProgressBar();
addScrollTextArea();

add(panelUtama);
pack();
setLocationRelativeTo(null);
}

private void addPanelTombol() {
    JPanel panelTombol = new JPanel(
        new MigLayout("fill", "[grow][grow]")
    );
    buttonMulaiTask = new JButton("Mulai Task (8 detik)");
    buttonCancel = new JButton("Cancel");
    buttonMulaiTask.setFont(new Font("Inter", Font.BOLD, 18));
    buttonCancel.setFont(new Font("Inter", Font.BOLD, 18));
    buttonCancel.setEnabled(false);
    panelTombol.add(buttonMulaiTask, "w 90%, growy");
    panelTombol.add(buttonCancel, "w 10%, growy");
    panelUtama.add(panelTombol, "h 20%, hmin pref, grow");
}

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

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

private void addScrollTextArea() {
    textAreaHasil = new JTextArea();
    textAreaHasil.setEditable(false);
    textAreaHasil.setFont(new Font("Consolas", Font.PLAIN, 14));
    JScrollPane scroll = new JScrollPane(textAreaHasil);
    panelUtama.add(scroll, "h 60%, growx");
}
```



LAPORAN PRAKTIKUM

Halaman
26 of 38

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

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

    worker = new SwingWorker<Integer, Integer>() {
        @Override
        protected Integer doInBackground() throws Exception {
            int total = 100;
            for (int i = 1; i <= total; i++) {
                if (isCancelled()) return null;
                Thread.sleep(80);
                publish(i);
            }
            return total;
        }

        @Override
        protected void process(List<Integer> chunks) {
            if (isCancelled()) return;
            int progress = chunks.get(chunks.size() - 1);
            progressBar.setValue(progress);
            int remainingSeconds = 8 - (8 * progress / 100);
            labelStatus.setText("Sisa " + remainingSeconds + " detik");
            addLog("Percentase penggerjaan task: " + progress + "%");
        }

        @Override
        protected void done() {
            try {
                Integer result = get();
            
```



LAPORAN PRAKTIKUM

Halaman
27 of 38

```
labelStatus.setText("Status: Selesai");
addLog("Background Task Selesai " + result + "%");
JOptionPane.showMessageDialog(Demo5.this,
    "Background Task Selesai",
    "Informasi", JOptionPane.INFORMATION_MESSAGE);
} catch (ExecutionException ex) {
    labelStatus.setText("Status: Error");
    addLog("Error: " + ex.getCause().getMessage());
    JOptionPane.showMessageDialog(Demo5.this,
        "Terjadi kesalahan: " + ex.getMessage(),
        "Error", JOptionPane.ERROR_MESSAGE);
} catch (CancellationException ee){
    labelStatus.setText("Status: Dibatalkan");
} catch (InterruptedException e) {
    Thread.currentThread().interrupt();
} finally {
    buttonCancel.setEnabled(false);
    buttonMulaiTask.setEnabled(true);
    worker = null;
}
};

worker.execute();
}

private void addLog(String text) {
    textAreaHasil.append(text + "\n");
    textAreaHasil.setCaretPosition(textAreaHasil.getDocument().getLength());
}

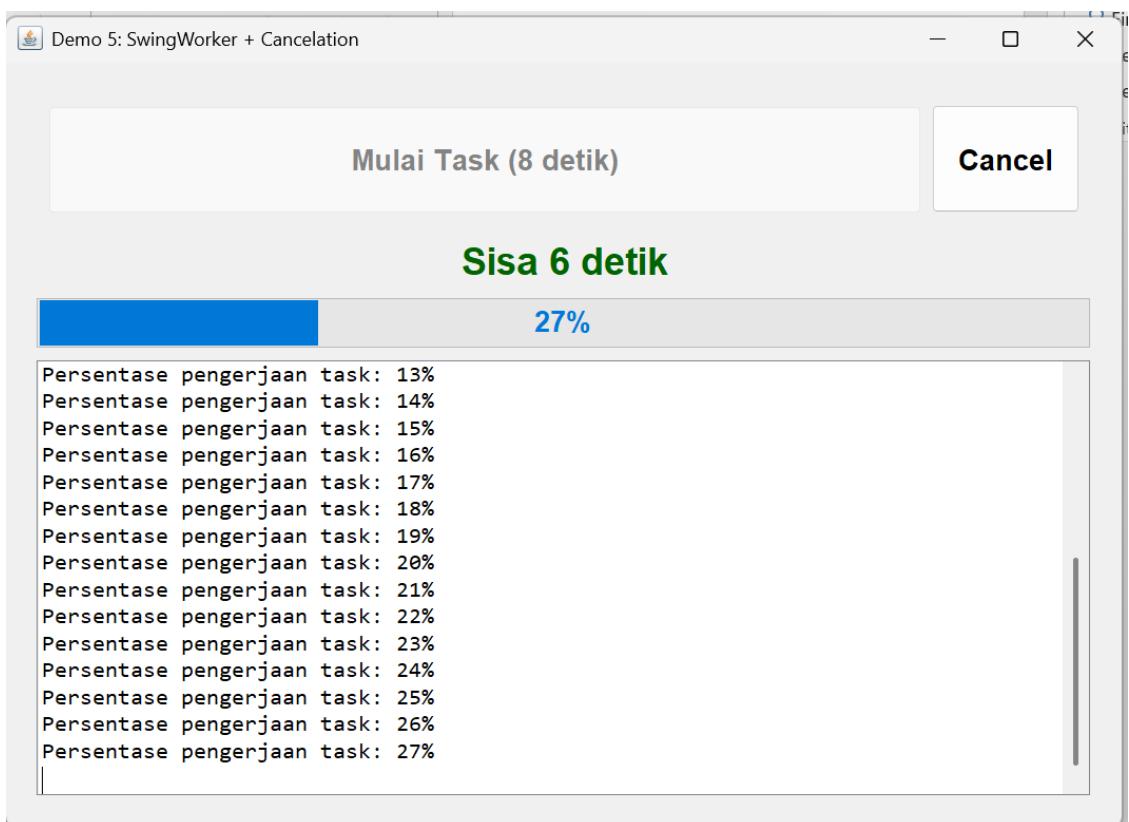
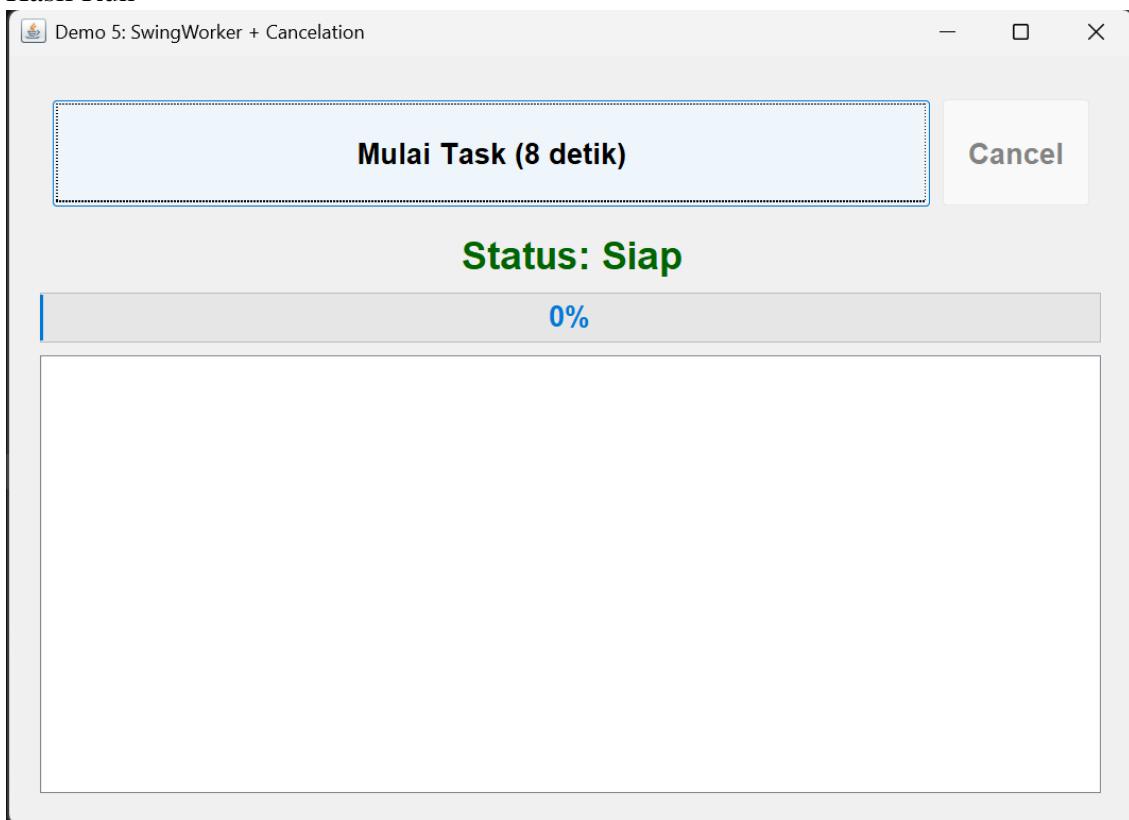
public static void main(String[] args) {
    SwingUtilities.invokeLater(() -> {
        try {
            UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
        } catch (Exception ignored) {}
        new Demo5().setVisible(true);
    });
}
```



LAPORAN PRAKTIKUM

Halaman
28 of 38

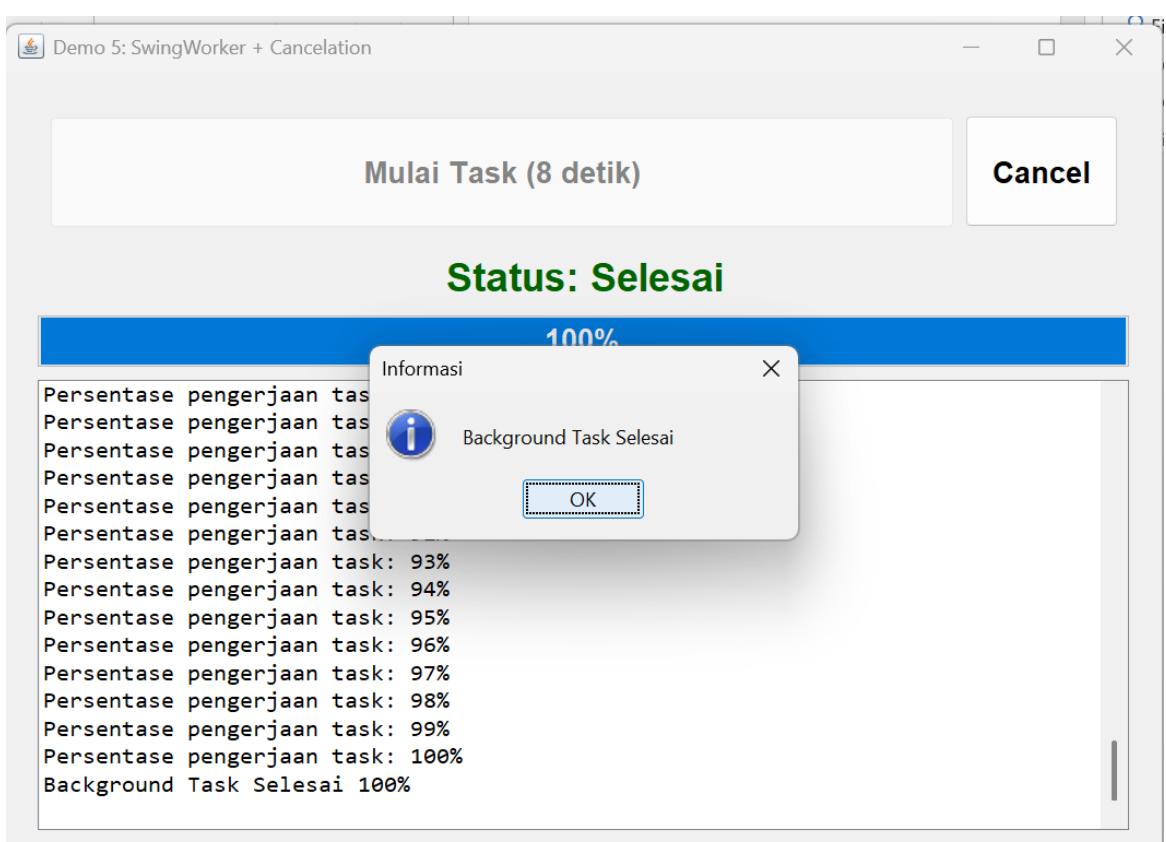
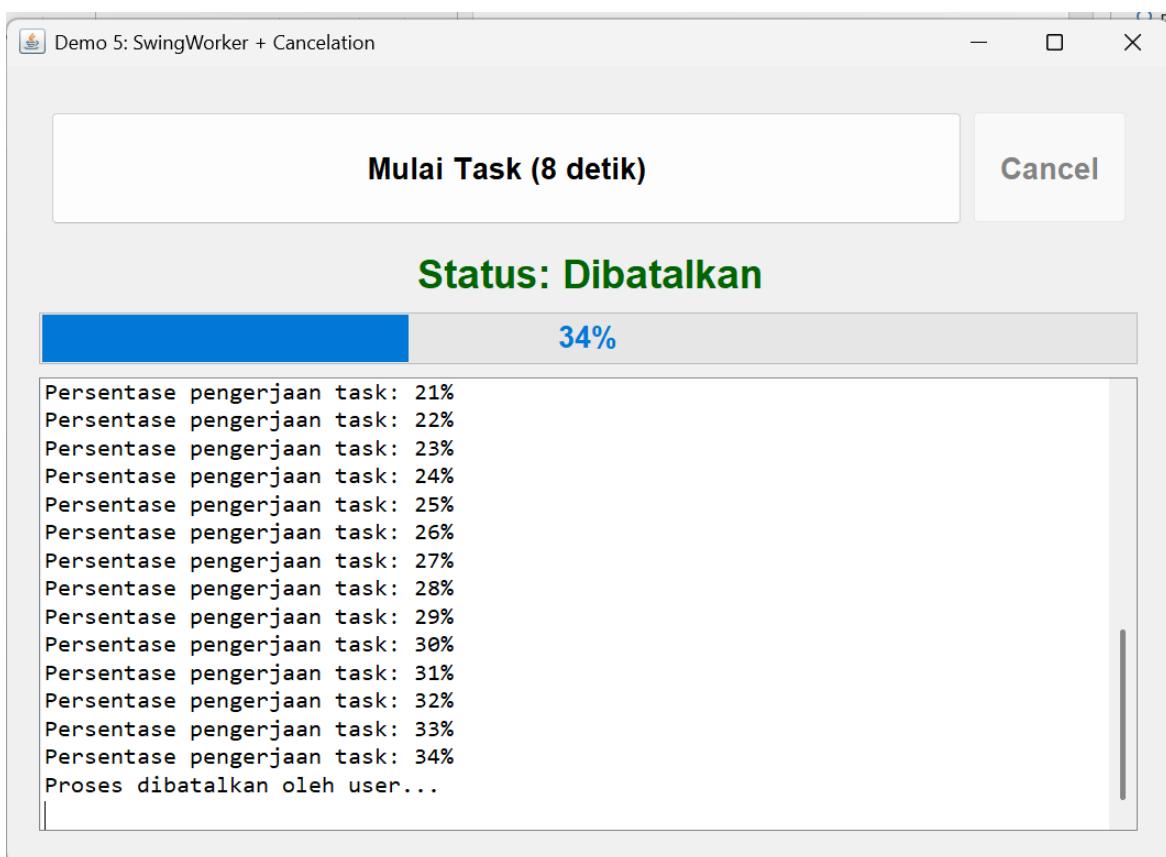
- Hasil Run





LAPORAN PRAKTIKUM

Halaman
29 of 38





2. LATIHAN

a. Task Worker

- Kode Program

```
package Latihan;

import java.util.List;
import java.util.concurrent.CancellationException;
import java.util.concurrent.ExecutionException;

import javax.swing.SwingWorker;

public class TaskWorker extends SwingWorker<Integer, Integer> {

    public interface Listener {
        void onProgressUpdate(int progress, int remainingSeconds);
        void onLog(String msg);
        void onCompleted(int result);
        void onCancelled();
        void onError(Exception ex);
    }

    private final Listener listener;

    public TaskWorker(Listener listener) {
        this.listener = listener;
    }

    @Override
    protected Integer doInBackground() throws Exception {
        int total = 100;
        for (int i = 1; i <= total; i++) {
            if (isCancelled()) return null;

            Thread.sleep(80);
            publish(i);
        }
        return total;
    }

    @Override
    protected void process(List<Integer> chunks) {
        if (isCancelled()) return;

        int progress = chunks.get(chunks.size() - 1);
        int remainingSeconds = 8 - (8 * progress / 100);
    }
}
```



```
        listener.onProgressUpdate(progress, remainingSeconds);
        listener.onLog("Persentase pengeraaan task: " + progress + "%");
    }

@Override
protected void done() {
    try {
        Integer result = get();
        listener.onCompleted(result);
    } catch (CancellationException e) {
        listener.onCancelled();
    } catch (ExecutionException e) {
        listener.onError(e);
    } catch (InterruptedException e) {
        Thread.currentThread().interrupt();
    }
}
```



b. Demo 6

- Kode Program

```
package Latihan;

import java.awt.Color;
import java.awt.Dimension;
import java.awt.Font;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JProgressBar;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import javax.swing.SwingUtilities;
import javax.swing.UIManager;

import net.miginfocom.swing.MigLayout;

public class Demo6 extends JFrame implements TaskWorker.Listener {

    private JPanel panelUtama;
    private JButton buttonMulaiTask;
    private JButton buttonCancel;
    private JLabel labelStatus;
    private JProgressBar progressBar;
    private JTextArea textAreaHasil;

    private TaskWorker worker;

    public Demo6() {
        initializeUI();
        setupEventHandlers();
    }

    private void initializeUI() {
        setTitle("Demo 6: SwingWorker Toplevel + Cancelation");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setPreferredSize(new Dimension(720, 520));

        panelUtama = new JPanel(new MigLayout("fill, wrap 1, insets 20"));

        addPanelTombol();
        addLabelStatus();
    }

    private void addPanelTombol() {
        // Add code to add buttons to the panel
    }

    private void addLabelStatus() {
        // Add code to add status label to the panel
    }
}
```



LAPORAN PRAKTIKUM

Halaman
33 of 38

```
addProgressBar();
addScrollTextArea();

add(panelUtama);
pack();
 setLocationRelativeTo(null);
}

private void addPanelTombol() {
    JPanel panelTombol = new JPanel(new MigLayout("fill",
"[grow][grow]"));
    buttonMulaiTask = new JButton("Mulai Task (8 detik)");
    buttonCancel = new JButton("Cancel");

    buttonMulaiTask.setFont(new Font("Inter", Font.BOLD, 18));
    buttonCancel.setFont(new Font("Inter", Font.BOLD, 18));
    buttonCancel.setEnabled(false);

    panelTombol.add(buttonMulaiTask, "w 90%, growy");
    panelTombol.add(buttonCancel, "w 10%, growy");

    panelUtama.add(panelTombol, "h 20%, grow");
}

private void addLabelStatus() {
    labelStatus = new JLabel("Status: Siap", JLabel.CENTER);
    labelStatus.setFont(new Font("Inter", Font.BOLD, 24));
    labelStatus.setForeground(new Color(0, 100, 0));
    panelUtama.add(labelStatus, "h 10%, growx");
}

private void addProgressBar() {
    progressBar = new JProgressBar(0, 100);
    progressBar.setStringPainted(true);
    progressBar.setFont(new Font("Inter", Font.BOLD, 18));
    panelUtama.add(progressBar, "h 10%, growx");
}

private void addScrollTextArea() {
    textAreaHasil = new JTextArea();
    textAreaHasil.setEditable(false);
    textAreaHasil.setFont(new Font("Consolas", Font.PLAIN, 14));
    JScrollPane scroll = new JScrollPane(textAreaHasil);

    panelUtama.add(scroll, "h 60%, grow");
}
```



LAPORAN PRAKTIKUM

Halaman
34 of 38

```
private void setupEventHandlers() {

    buttonMulaiTask.addActionListener(e -> mulaiProses());

    buttonCancel.addActionListener(e -> {
        if (worker != null && !worker.isDone()) {
            worker.cancel(true);
            addLog("Proses dibatalkan oleh user...");
        }
    });
}

private void mulaiProses() {
    buttonMulaiTask.setEnabled(false);
    buttonCancel.setEnabled(true);
    progressBar.setValue(0);
    textAreaHasil.setText("");

    labelStatus.setText("Status: Mulai bekerja");
    addLog("Mengerjakan Background Task (8 detik)");

    worker = new TaskWorker(this);
    worker.execute();
}

private void addLog(String msg) {
    textAreaHasil.append(msg + "\n");
    textAreaHasil.setCaretPosition(textAreaHasil.getDocument().getLength());
};

@Override
public void onProgressUpdate(int progress, int remainingSeconds) {
    progressBar.setValue(progress);
    labelStatus.setText("Sisa " + remainingSeconds + " detik");
}

@Override
public void onLog(String msg) {
    addLog(msg);
}

@Override
public void onCompleted(int result) {
    labelStatus.setText("Status: Selesai");
    addLog("Background Task Selesai " + result + "%");
}
```



LAPORAN PRAKTIKUM

Halaman
35 of 38

```
JOptionPane.showMessageDialog(this,
    "Background Task Selesai",
    "Informasi",
    JOptionPane.INFORMATION_MESSAGE);

    resetButtons();
}

@Override
public void onCancelled() {
    labelStatus.setText("Status: Dibatalkan");
    addLog("Task dibatalkan.");
    resetButtons();
}

@Override
public void onError(Exception ex) {
    labelStatus.setText("Status: Error");
    addLog("Error: " + ex.getMessage());

    JOptionPane.showMessageDialog(this,
        "Terjadi kesalahan: " + ex.getMessage(),
        "Error",
        JOptionPane.ERROR_MESSAGE);

    resetButtons();
}

private void resetButtons() {
    buttonMulaiTask.setEnabled(true);
    buttonCancel.setEnabled(false);
    worker = null;
}

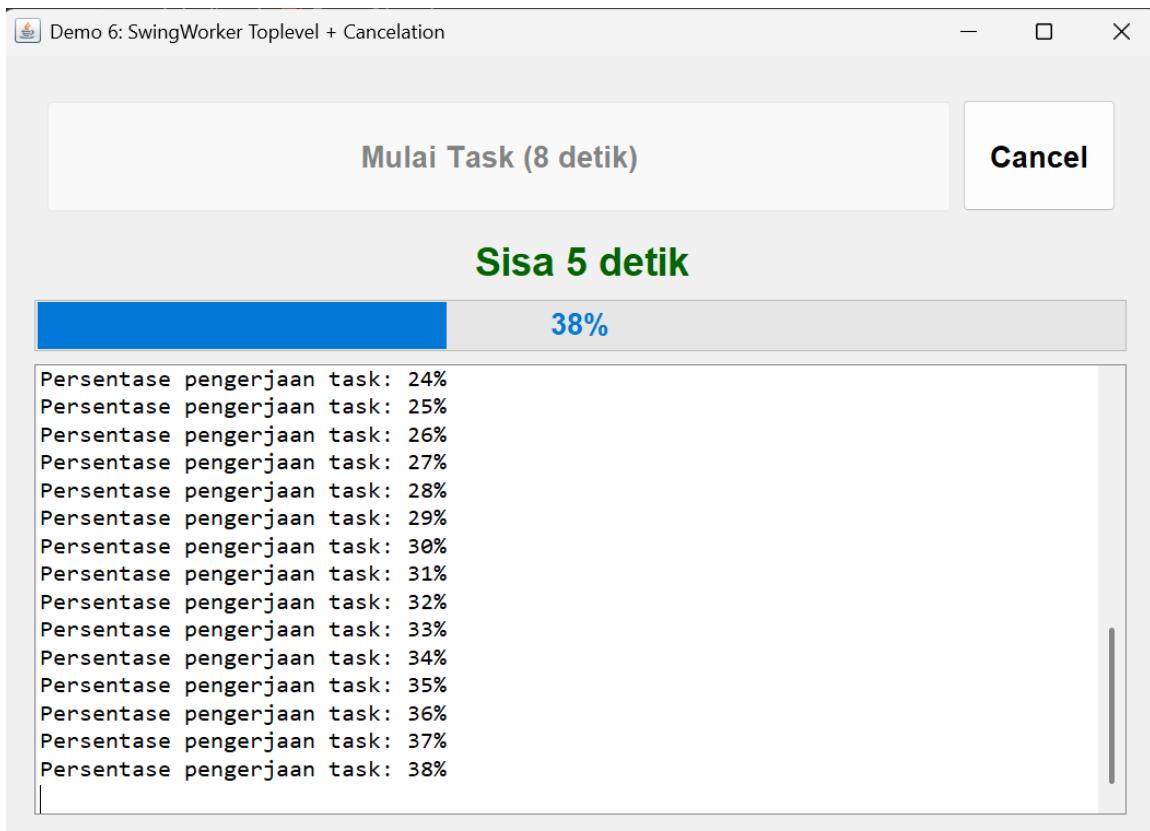
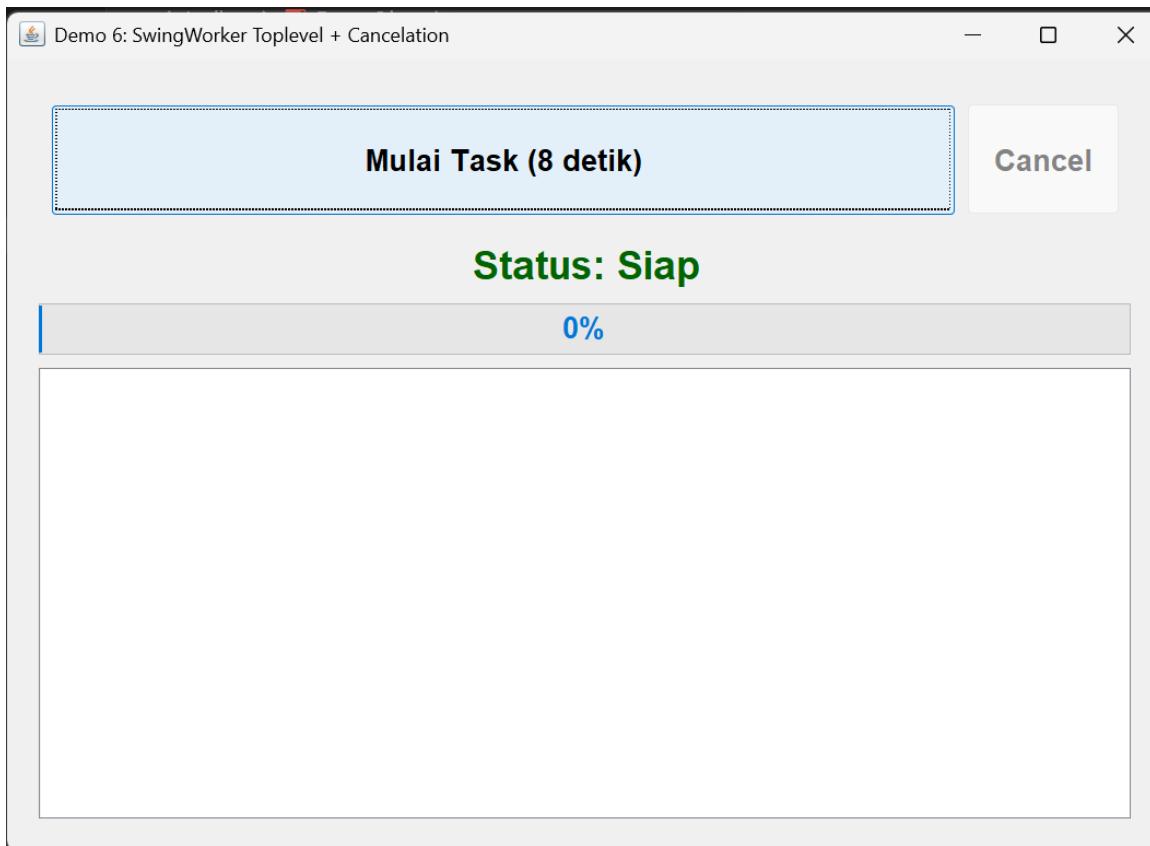
public static void main(String[] args) {
    SwingUtilities.invokeLater(() -> {
        try {
            UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
        } catch (Exception ignored) {}
        new Demo6().setVisible(true);
    });
}
```



LAPORAN PRAKTIKUM

Halaman
36 of 38

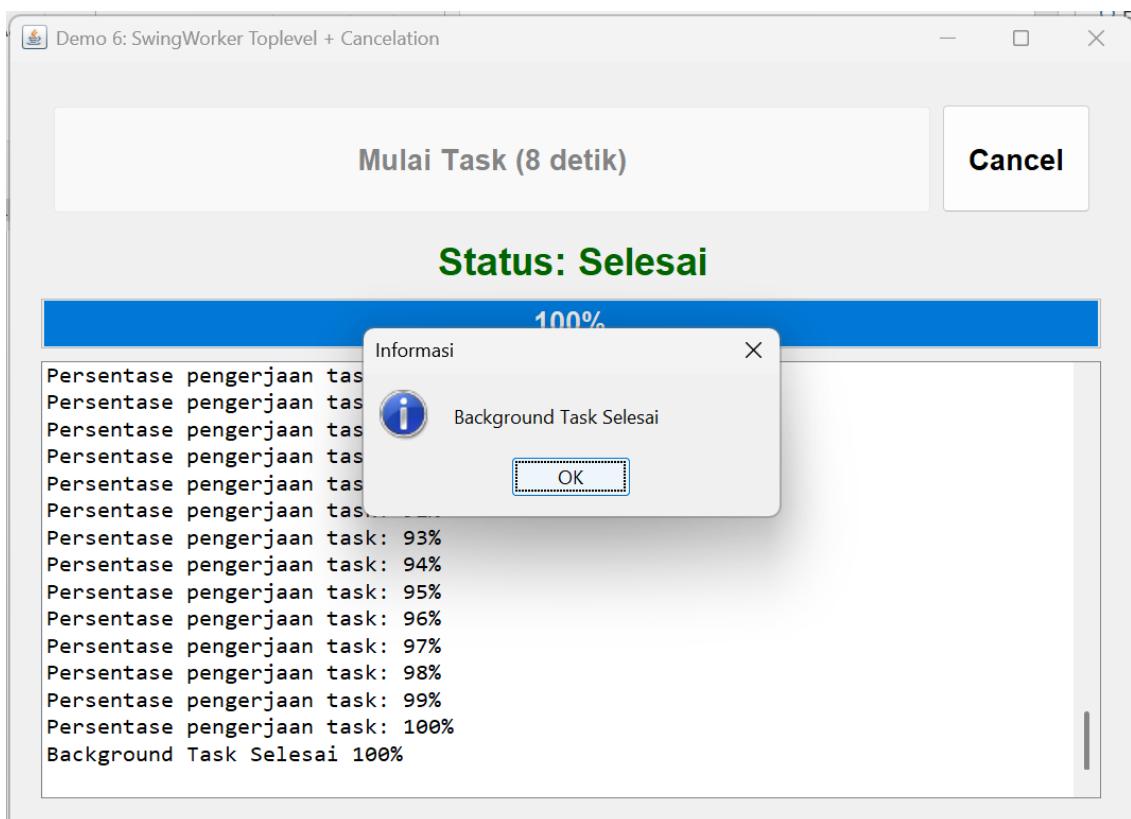
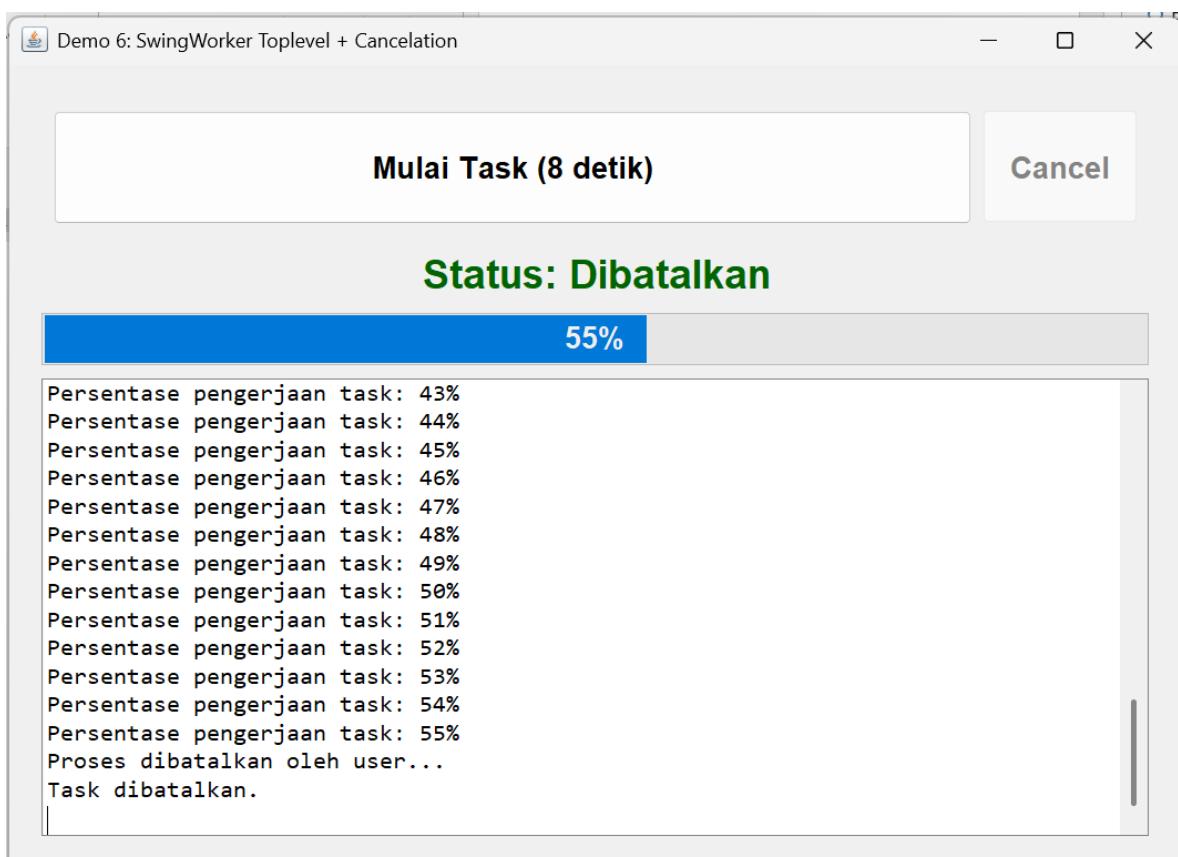
- Hasil Run





LAPORAN PRAKTIKUM

Halaman
37 of 38





C. Hasil Pembelajaran

Pada materi kali ini menbahas mengenai SWING WORKER. Dimana di dalam nya menjelaskan mengenai *Concurrency* yang di implemntasikan dengan berbagai cara;

diantaranya :

- Multithreading
- Multiprocessing
- Single-threaded concurrency.