

# **LAPORAN PRATIKUM**

**“PEKAN 7”**

***Disusun Untuk Memenuhi Tugas Mata Kuliah Struktur Data***

**DOSEN PENGAMPU:**

**Wahyudi, Dr. S.T. M.T.**



**DISUSUN OLEH:**

**Karimah Irsyadiyah (2411533018)**

**PROGRAM STUDI INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS ANDALAS  
T.A 2024/2025**

## Daftar Pustaka

BAB I PENDAHULUAN.....	3
1.1    Latar Belakang .....	3
1.2    Tujuan .....	3
1.3    Alat dan Bahan.....	3
BAB II PEMBAHASAN .....	4
2.1    Langkah-langkah Praktikum dan Pembahasan Program .....	4
BAB III PENUTUP .....	23
3.1    Kesimpulan .....	23

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Struktur data merupakan fondasi penting dalam dunia pemrograman, khususnya dalam pengolahan dan pengurutan data. Salah satu bentuk implementasinya adalah penggunaan algoritma pengurutan (sorting), seperti *Insertion Sort* dan *Selection Sort*. Dalam praktikum ini, kita menggabungkan pemahaman algoritma dengan visualisasi menggunakan antarmuka grafis berbasis *Java Swing*. Visualisasi membantu mahasiswa memahami langkah-langkah sorting secara nyata dan interaktif.

### **1.2 Tujuan**

- Memahami cara kerja algoritma Insertion Sort dan Selection Sort.
- Mengimplementasikan algoritma tersebut dalam bentuk antarmuka grafis (GUI).
- Menampilkan proses sorting langkah demi langkah secara visual untuk memperkuat pemahaman logika program.

### **1.3 Alat dan Bahan**

- Perangkat Keras: Laptop/PC
- Perangkat komputer dengan IDE (contoh: IntelliJ IDEA / Eclipse).
- Java Development Kit (JDK).

## BAB II

### PEMBAHASAN

#### 2.1 Langkah-langkah Praktikum dan Pembahasan Program

##### A. Class : InsertionSortGUI.java

###### a) Tujuan Program:

Aplikasi GUI Java untuk melakukan **algoritma Insertion Sort** secara **langkah per langkah**.

###### b) Struktur dan Penjelasan Komponen Program:

1. package pekan7;
  - Menandakan file berada dalam folder (package) pekan7.
2. import java.awt.\* dan import javax.swing.\*;
  - Mengimpor pustaka GUI AWT dan Swing (seperti JFrame, JButton, JLabel, JTextField, dll).
3. public class InsertionSortGUI extends JFrame
  - Membuat class utama untuk GUI, turunan dari JFrame (jendela utama GUI).
4. Deklarasi variabel:
  - int[] array: Menyimpan data yang akan disorting.
  - JLabel[] labelArray: Label visual untuk setiap elemen array.
  - JButton stepButton, resetButton, setButton: Tombol aksi.
  - JTextField inputField: Kolom untuk input array.
  - JPanel panelArray: Panel untuk menampilkan elemen array.
  - JTextArea stepArea: Menampilkan langkah-langkah sorting.
  - int i, j: Indeks loop Insertion Sort.
  - boolean sorting: Status proses sorting.
  - int stepCount: Menghitung jumlah langkah.
5. Method main()
  - Menjalankan GUI menggunakan EventQueue.invokeLater() agar thread GUI berjalan aman.
6. Constructor InsertionSortGUI()
  - Menyiapkan layout GUI:

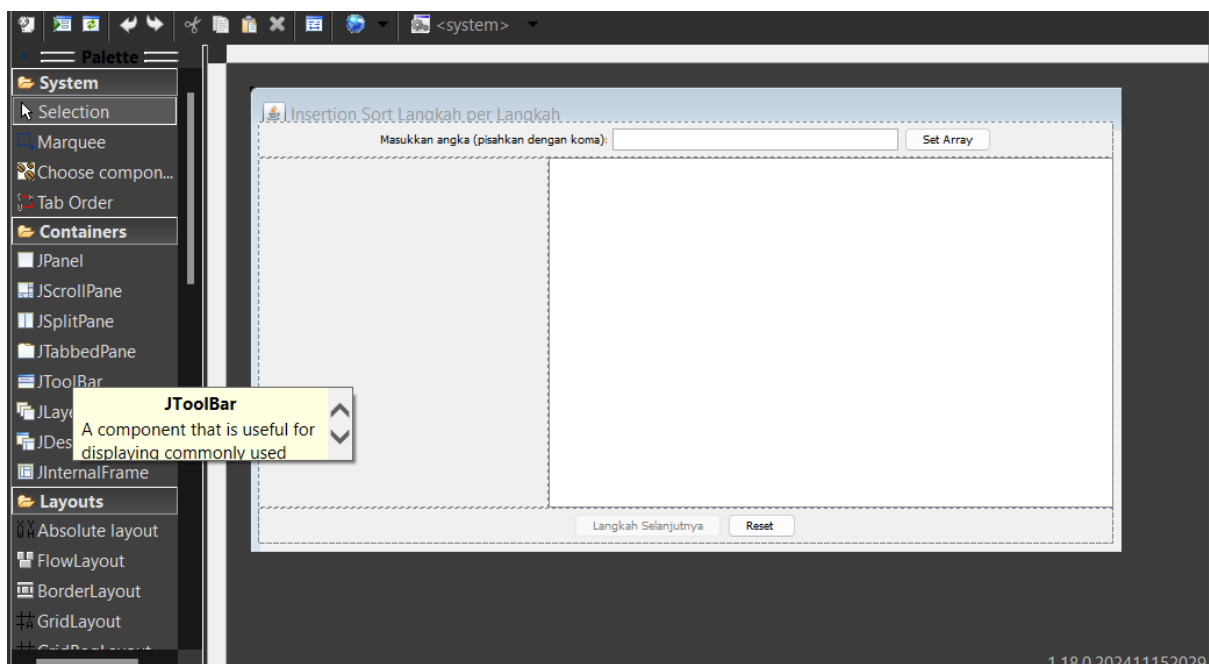
- Panel Input (atas): Untuk masukkan data array.
- Panel Array (tengah): Menampilkan array secara visual.
- Panel Kontrol (bawah): Tombol aksi Step, Reset.
- Panel Log (kanan): Area teks untuk mencatat langkah sorting.
- Menambahkan aksi tombol (event listeners) untuk:
  - Set Array: Membaca input, memisah jadi integer, mem-visualisasikan array.
  - Step: Melakukan satu langkah sorting.
  - Reset: Mengembalikan GUI ke kondisi awal.

#### 7. Fungsi-fungsi utama:

- `updateArrayDisplay()`: Memperbarui tampilan label sesuai isi array.
- `performStep()`: Melakukan satu langkah Insertion Sort, memperbarui array, label, dan mencatat langkah ke `stepArea`.
- `resetSort()`: Menghapus semua data & mengatur ulang semua elemen GUI.

#### c) Fitur Utama Program:

- Pengguna dapat memasukkan angka seperti 5,2,9,1.
- Dapat menjalankan langkah sorting satu demi satu (step-by-step).
- Menampilkan log proses sorting secara real time.
- Tampilan elemen array menggunakan label visual yang berubah warna.



Insertion Sort Langkah per Langkah

Masukkan angka (pisahkan dengan koma):

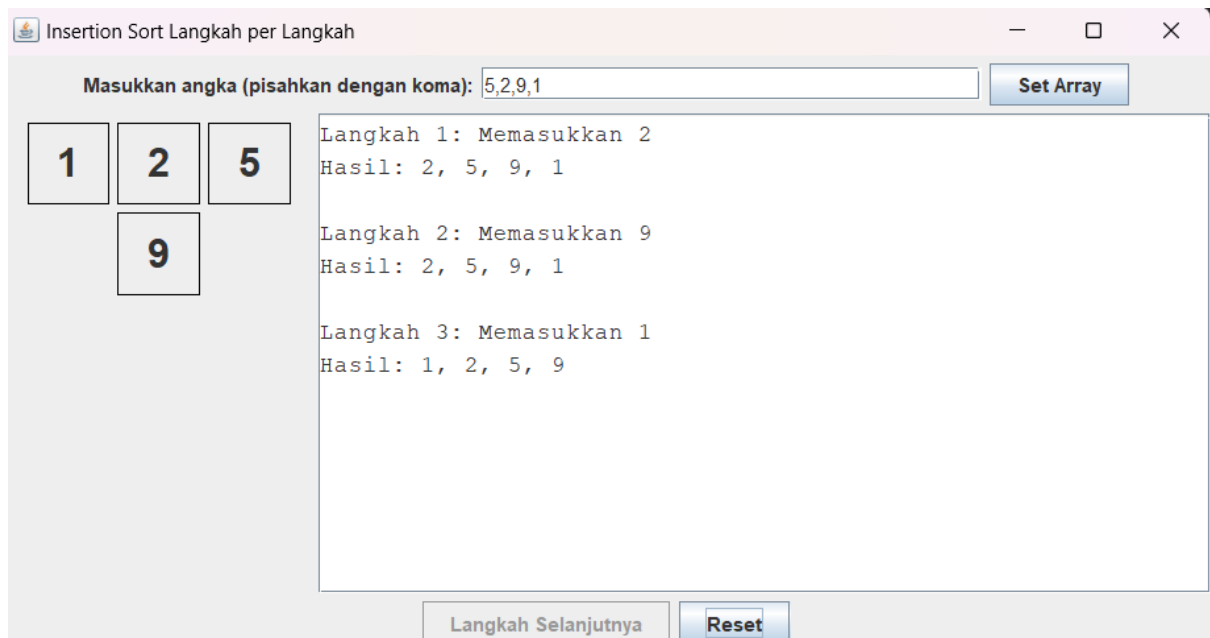
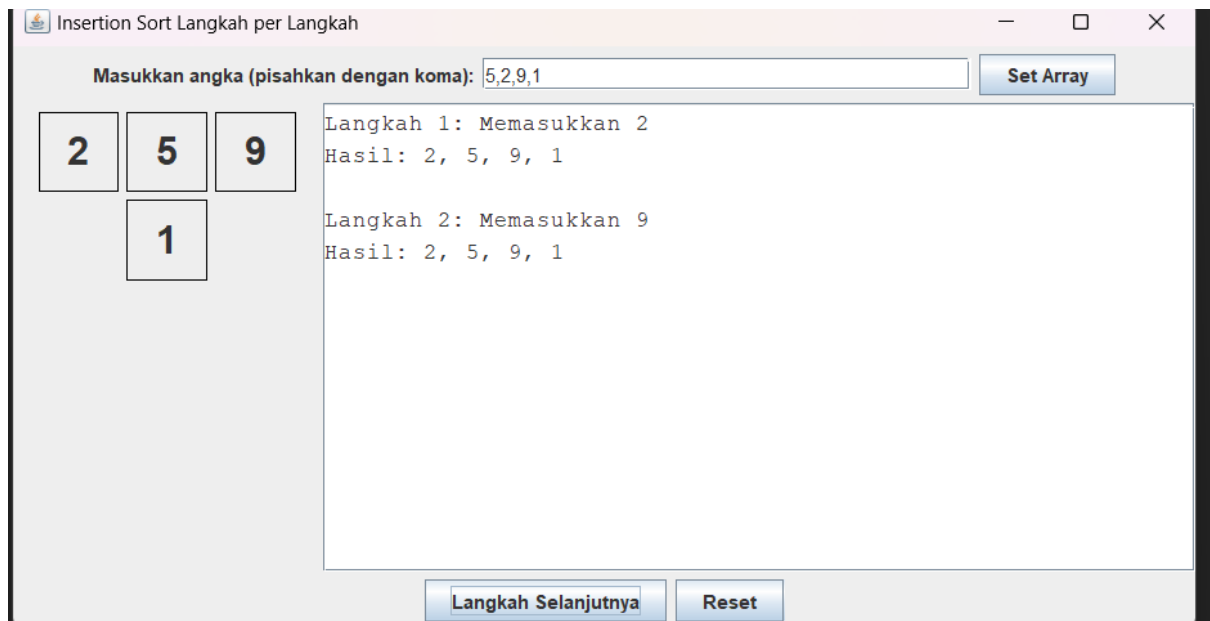
5	2	9
	1	

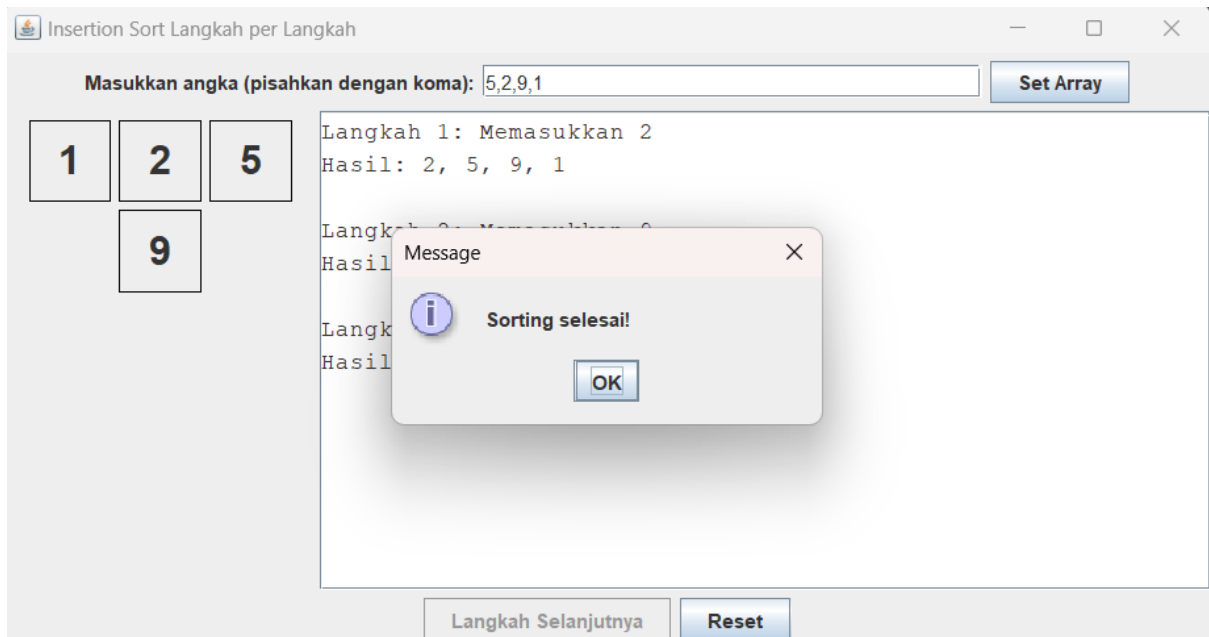
Insertion Sort Langkah per Langkah

Masukkan angka (pisahkan dengan koma):

2	5	9
	1	

Langkah 1: Memasukkan 2  
Hasil: 2, 5, 9, 1





```
package pekan7;

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.EventQueue;
import java.awt.FlowLayout;
import java.awt.Font;

import javax.swing.BorderFactory;
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.JTextField;
import javax.swing.SwingConstants;

public class InsertionSortGUI extends JFrame {

    private static final long serialVersionUID = 1L;
    private int[] array;
    private JLabel[] labelArray;
    private JButton stepButton, resetButton, setButton;
    private JTextField inputField;
    private JPanel panelArray;
    private JTextArea stepArea;
    private int i = 1, j;
```



```

private boolean sorting = false;
private int stepCount = 1;

public static void main(String[] args) {
    EventQueue.invokeLater(() -> {
        try {
            InsertionSortGUI frame = new InsertionSortGUI();
            frame.setVisible(true);
        } catch (Exception e) {
            e.printStackTrace();
        }
    });
}

public InsertionSortGUI() {
    setTitle("Insertion Sort Langkah per Langkah");
    setSize(750, 400);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setLocationRelativeTo(null);
    setLayout(new BorderLayout());

    JPanel inputPanel = new JPanel(new FlowLayout());
    inputField = new JTextField(30);
    setButton = new JButton("Set Array");
    inputPanel.add(new JLabel("Masukkan angka (pisahkan dengan koma):"));
    inputPanel.add(inputField);
    inputPanel.add(setButton);

    panelArray = new JPanel(new FlowLayout());

    JPanel controlPanel = new JPanel();
    stepButton = new JButton("Langkah Selanjutnya");
    resetButton = new JButton("Reset");
    stepButton.setEnabled(false);
    controlPanel.add(stepButton);
    controlPanel.add(resetButton);

    stepArea = new JTextArea(8, 60);
    stepArea.setEditable(false);
    stepArea.setFont(new Font("Monospaced", Font.PLAIN, 14));
    JScrollPane scrollPane = new JScrollPane(stepArea);

    add(inputPanel, BorderLayout.NORTH);
    add(panelArray, BorderLayout.CENTER);
    add(controlPanel, BorderLayout.SOUTH);
    add(scrollPane, BorderLayout.EAST);
}

```

```

        setButton.addActionListener(e -> setArrayFromInput());
        stepButton.addActionListener(e -> performStep());
        resetButton.addActionListener(e -> reset());
    }

    private void setArrayFromInput() {
        String text = inputField.getText().trim();
        if (text.isEmpty()) return;

        String[] parts = text.split(",");
        array = new int[parts.Length];

        try {
            for (int k = 0; k < parts.Length; k++) {
                array[k] = Integer.parseInt(parts[k].trim());
            }
        } catch (NumberFormatException e) {
            JOptionPane.showMessageDialog(this,
                "Masukkan hanya angka yang dipisahkan dengan koma!",
                "Error",
                JOptionPane.ERROR_MESSAGE);
            return;
        }

        i = 1;
        stepCount = 1;
        sorting = true;
        stepButton.setEnabled(true);
        stepArea.setText("");
        panelArray.removeAll();

        labelArray = new JLabel[array.Length];
        for (int k = 0; k < array.Length; k++) {
            labelArray[k] = new JLabel(String.valueOf(array[k]));
            labelArray[k].setFont(new Font("Arial", Font.BOLD, 24));
            labelArray[k].setBorder(BorderFactory.createLineBorder(Color.
r.BLACK));
            labelArray[k].setPreferredSize(new Dimension(50, 50));
            labelArray[k].setHorizontalAlignment(SwingConstants.CENTER)
;
            panelArray.add(labelArray[k]);
        }

        panelArray.revalidate();
        panelArray.repaint();
    }

    private void performStep() {

```

```

        if (i < array.Length && sorting) {
            int key = array[i];
            j = i - 1;

            StringBuilder stepLog = new StringBuilder();
            stepLog.append("Langkah ").append(stepCount).append(":
Memasukkan ").append(key).append("\n");

            while (j >= 0 && array[j] > key) {
                array[j + 1] = array[j];
                j--;
            }

            array[j + 1] = key;
            updateLabels();
            stepLog.append("Hasil:
").append(arrayToString(array)).append("\n\n");
            stepArea.append(stepLog.toString());

            i++;
            stepCount++;

            if (i == array.Length) {
                sorting = false;
                stepButton.setEnabled(false);
                JOptionPane.showMessageDialog(this, "Sorting
selesai!");
            }
        }
    }

    private void updateLabels() {
        for (int k = 0; k < array.Length; k++) {
            labelArray[k].setText(String.valueOf(array[k]));
        }
    }

    private void reset() {
        inputField.setText("");
        panelArray.removeAll();
        panelArray.revalidate();
        panelArray.repaint();
        stepArea.setText("");
        stepButton.setEnabled(false);
        sorting = false;
        i = 1;
        stepCount = 1;
    }
}

```

```

private String arrayToString(int[] arr) {
    StringBuilder sb = new StringBuilder();
    for (int k = 0; k < arr.Length; k++) {
        sb.append(arr[k]);
        if (k < arr.Length - 1) sb.append(", ");
    }
    return sb.toString();
}
}

```

## B. Class : SelectionSortGUI.java

### a) Tujuan Program:

Aplikasi GUI Java untuk melakukan algoritma **Selection Sort** secara langkah per langkah.

### b) Struktur dan Penjelasan Komponen Program:

- package pekan7;  
Menandakan file berada dalam folder (package) pekan7.
- import java.awt.\* dan import javax.swing.\*;  
Mengimpor pustaka GUI AWT dan Swing (seperti JFrame, JButton, JLabel, JTextField, dll).
- public class SelectionSortGUI extends JFrame  
Membuat class utama untuk GUI, turunan dari JFrame (jendela utama GUI).

### c) Deklarasi variabel:

- int[] array: Menyimpan data yang akan disorting.
- JLabel[] labelArray: Label visual untuk setiap elemen array.
- JButton stepButton, resetButton, setButton: Tombol aksi.
- JTextField inputField: Kolom untuk input array.
- JPanel panelArray: Panel untuk menampilkan elemen array.
- JTextArea stepArea: Menampilkan langkah-langkah sorting.
- int i, j: Indeks loop Selection Sort (i untuk iterasi luar, j untuk mencari nilai terkecil).
- int minIndex: Menyimpan indeks elemen terkecil sementara.
- boolean sorting: Status proses sorting.
- int stepCount: Menghitung jumlah langkah yang telah dilakukan.

### d) Method main()

Menjalankan GUI menggunakan `EventQueue.invokeLater()` agar thread GUI berjalan aman dan responsif.

**e) Constructor `SelectionSortGUI()`**

Menyiapkan layout GUI:

- Panel Input (atas): Untuk memasukkan data array.
- Panel Array (tengah): Menampilkan array secara visual dengan label.
- Panel Kontrol (bawah): Tombol aksi Set, Langkah Selanjutnya, dan Reset.
- Panel Log (kanan): Area teks (`JTextArea`) untuk mencatat langkah sorting secara real-time.

Menambahkan aksi tombol (event listeners) untuk:

- Set Array: Membaca input, memecahnya menjadi integer, dan menampilkan array secara visual.
- Langkah Selanjutnya (Step): Melakukan satu langkah algoritma Selection Sort.
- Reset: Mengembalikan GUI ke kondisi awal dan menghapus semua data.

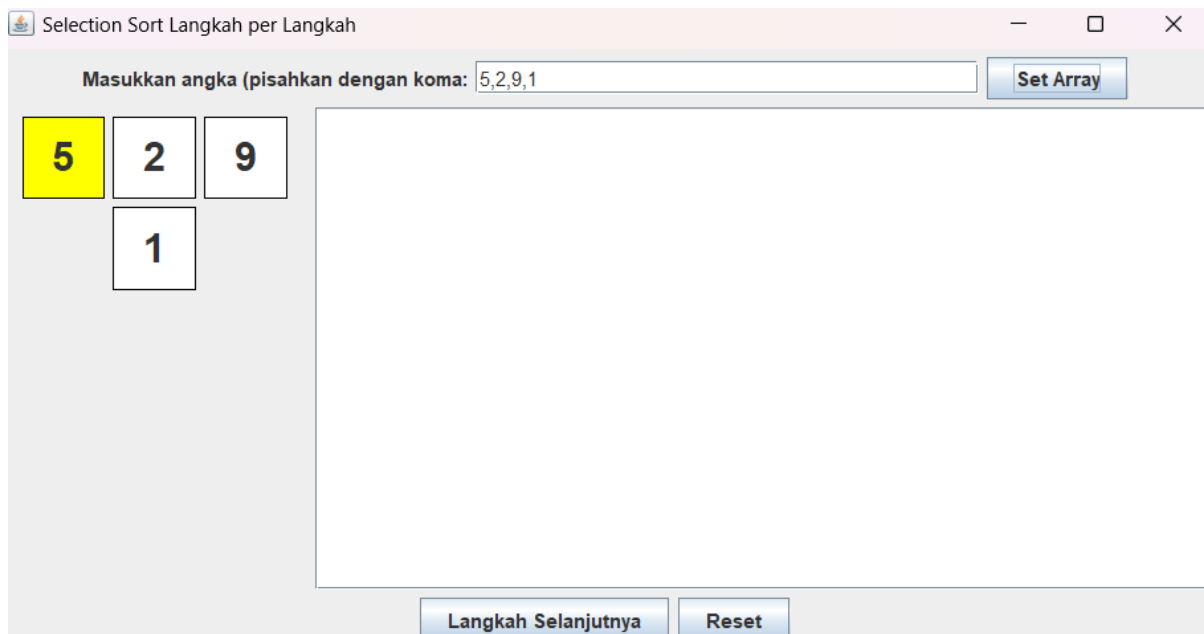
**f) Fungsi-fungsi utama:**

- `setArrayFromInput()`:  
Membaca input dari `inputField`, mengonversinya ke array integer, dan menginisialisasi tampilan array dengan `JLabel[]`. Menyiapkan variabel awal dan mulai proses sorting.
- `performStep()`:  
Melakukan satu langkah Selection Sort:
  - Mencari elemen terkecil dalam subarray yang belum diurutkan.
  - Menentukan `minIndex`, lalu melakukan pertukaran jika perlu.
  - Memperbarui tampilan array, memberi highlight elemen yang dipilih.
  - Menambahkan log langkah ke `stepArea`.
  - Menonaktifkan tombol langkah jika sorting selesai.
- `highlightMinIndex()`:  
Memberi warna kuning pada elemen array dengan indeks `minIndex`.
- `resetHighlights()`:  
Menghapus highlight pada seluruh label array (mengembalikan warna ke putih).

- `updateLabels()`:  
Memperbarui nilai pada label sesuai isi array terbaru.
- `reset()`:  
Menghapus input pengguna, label array, log langkah, dan mengatur ulang status sorting.
- `arrayToString(int[] arr)`:  
Mengubah array ke dalam format string agar bisa ditampilkan di log langkah (seperti 5,2,8).

**g) Fitur Utama Program:**

- Pengguna dapat memasukkan angka seperti 5,2,9,1.
- Dapat menjalankan langkah sorting satu demi satu (step-by-step).
- Menampilkan log proses sorting di sisi kanan aplikasi secara real time.



Selection Sort Langkah per Langkah

Masukkan angka (pisahkan dengan koma):

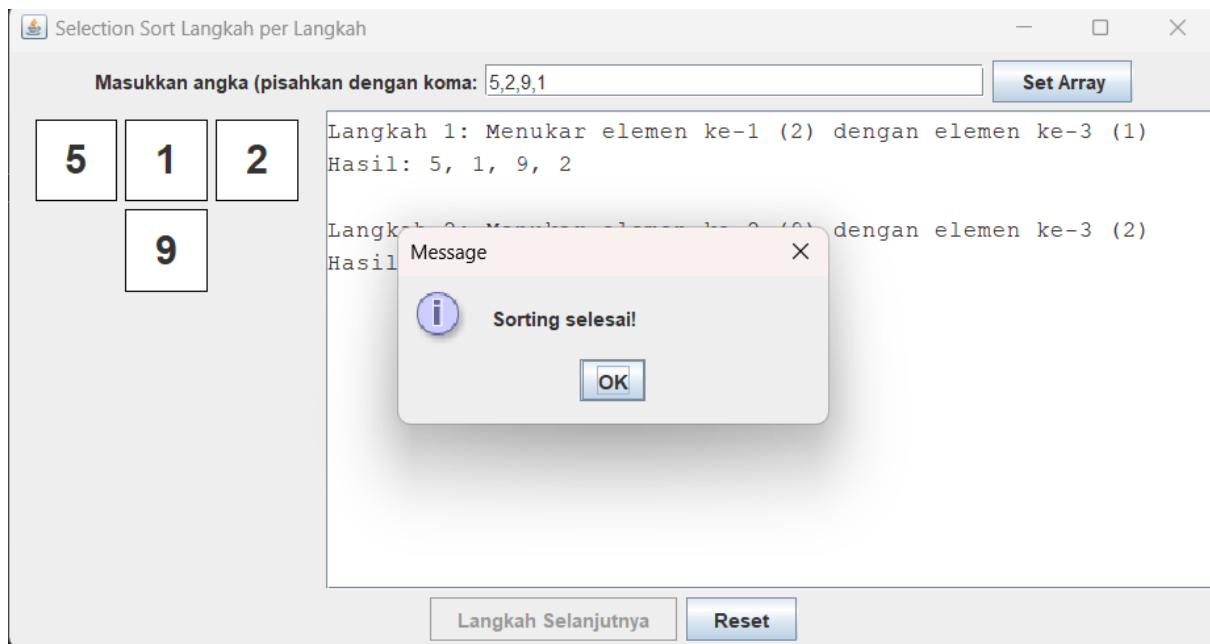
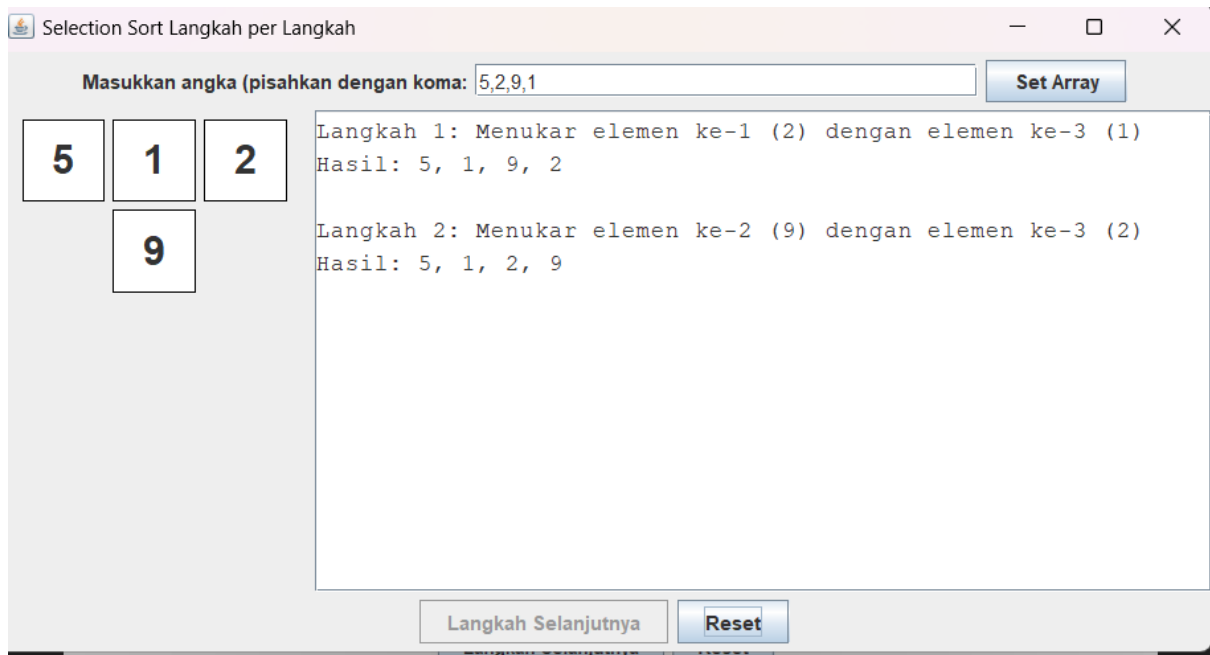
5	2	9
	1	

Selection Sort Langkah per Langkah

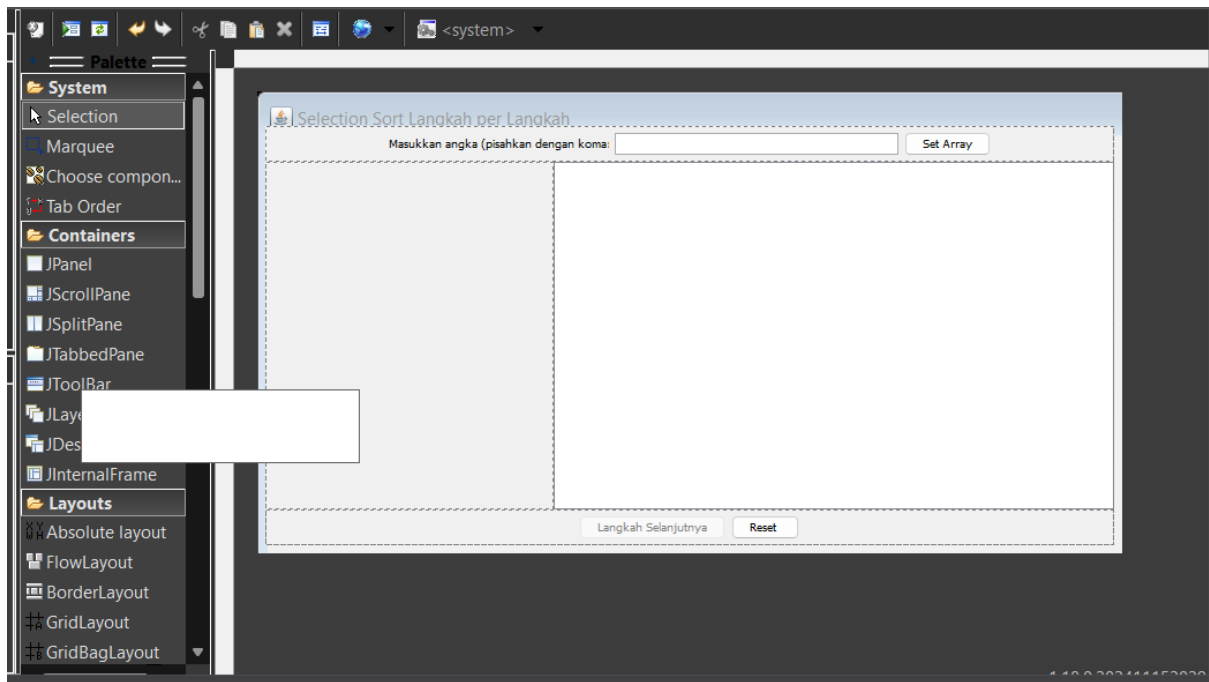
Masukkan angka (pisahkan dengan koma):

5	1	9
	2	

Langkah 1: Menukar elemen ke-1 (2) dengan elemen ke-3 (1)  
Hasil: 5, 1, 9, 2







```
package pekan7;

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.EventQueue;
import java.awt.FlowLayout;
import java.awt.Font;

import javax.swing.BorderFactory;
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.JTextField;
import javax.swing.SwingConstants;
import javax.swing.border.EmptyBorder;

public class SelectionSortGUI extends JFrame {

    private static final long serialVersionUID = 1L;
    private int[] array;
    private JLabel[] labelArray;
    private JButton stepButton, resetButton, setButton;
    private JTextField inputField;
    private JPanel panelArray;
    private JTextArea stepArea;
}
```

```

private int i = 1, j;
private boolean sorting = false;
private int stepCount = 1;
private int minIndex;

/**
 * Launch the application.
 */
public static void main(String[] args) {
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                SelectionSortGUI frame = new SelectionSortGUI();
                frame.setVisible(true);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}

/**
 * Create the frame.
 */
public SelectionSortGUI() {
    setTitle("Selection Sort Langkah per Langkah");
    setSize(750, 400);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setLocationRelativeTo(null);
    setLayout(new BorderLayout());

    // Panel Input
    JPanel inputPanel = new JPanel(new FlowLayout());
    inputField = new JTextField(30);
    setButton = new JButton("Set Array");
    inputPanel.add(new JLabel("Masukkan angka (pisahkan dengan koma:"));
    inputPanel.add(inputField);
    inputPanel.add(setButton);

    // Panel Array visual
    panelArray = new JPanel();
    panelArray.setLayout(new FlowLayout());

    // Panel Kontrol
    JPanel controlPanel = new JPanel();
    stepButton = new JButton("Langkah Selanjutnya");

```

```

        resetButton = new JButton("Reset");
        stepButton.setEnabled(false);
        controlPanel.add(stepButton);
        controlPanel.add(resetButton);

        // Area Teks untuk Log Langkah-Langkah
        stepArea = new JTextArea(8, 60);
        stepArea.setEditable(false);
        stepArea.setFont(new Font("Monospaced", Font.PLAIN, 14));
        JScrollPane scrollPane = new JScrollPane(stepArea);

        // Tambahkan Panel ke Frame
        add(inputPanel, BorderLayout.NORTH);
        add(panelArray, BorderLayout.CENTER);
        add(controlPanel, BorderLayout.SOUTH);
        add(scrollPane, BorderLayout.EAST);

        // Event Set Array
        setButton.addActionListener(e -> setArrayFromInput());

        // Event Langkah Selanjutnya
        stepButton.addActionListener(e -> performStep());

        // Event Reset
        resetButton.addActionListener(e -> reset());

    }

    private void setArrayFromInput() {
        String text = inputField.getText().trim();
        if (text.isEmpty()) return;
        String[] parts = text.split(",");
        array = new int[parts.length];
        try {
            for (int k = 0; k < parts.length; k++) {
                array[k] = Integer.parseInt(parts[k].trim());
            }
        } catch (NumberFormatException e) {
            JOptionPane.showMessageDialog(this, "Masukkan hanya angka yang  
dipisahkan"
                + " dengan koma!", "Error", JOptionPane.ERROR_MESSAGE);
            return;
        }
        i = 1;
        stepCount = 1;
        sorting = true;
        stepButton.setEnabled(true);
    }

```

```

        stepArea.setText("");
        panelArray.removeAll();
        labelArray = new JLabel[array.length];
        for (int k = 0; k < array.length; k++) {
            labelArray[k] = new JLabel(String.valueOf(array[k]));
            labelArray[k].setFont(new Font("Arial", Font.BOLD, 24));
            labelArray[k].setOpaque(true);
            labelArray[k].setBackground(Color.WHITE);
            labelArray[k].setBorder(BorderFactory.createLineBorder(Color.BL
ACK));

            labelArray[k].setPreferredSize(new Dimension(50, 50));
            labelArray[k].setHorizontalAlignment(SwingConstants.CENTER);
            panelArray.add(labelArray[k]);
        }
        panelArray.revalidate();
        panelArray.repaint();
        highlightMinIndex();
    }

    private void performStep() {
        if (i < array.length - 1 && sorting) {
            StringBuilder stepLog = new StringBuilder();
            if (j == i + 1) {
                minIndex = i;
            }

            // Cari indeks minimum
            if (j < array.length) {
                if (array[j] < array[minIndex]) {
                    minIndex = j;
                }
                j++;
            }

            // Jika sudah selesai membandingkan
            if (j == array.length) {
                if (minIndex != i) {
                    int temp = array[i];
                    array[i] = array[minIndex];
                    array[minIndex] = temp;

                    stepLog.append("Langkah ").append(stepCount).append(":
Menukar elemen ke-")
                        .append(i).append("
").append(array[minIndex]).append("")
                        .append(" dengan elemen ke-").append(minIndex)
                        .append(" ").append(array[i]).append(")\n");
                } else {

```

```

        stepLog.append("Langkah ").append(stepCount)
                .append(": Tidak ada pertukaran (elemen ke-")
                .append(i).append(" sudah minimum)\n");
    }
    stepLog.append("Hasil:
").append(arrayToString(array)).append("\n\n");
    stepArea.append(stepLog.toString());

    i++;
    j = i + 1;
    stepCount++;
}

updateLabels();
highlightMinIndex();

if (i >= array.length - 1) {
    sorting = false;
    stepButton.setEnabled(false);
    resetHighlights();
    JOptionPane.showMessageDialog(this, "Sorting selesai!");
}
}

private void highlightMinIndex() {
    resetHighlights();
    if (minIndex >= 0 && minIndex < labelArray.length) {
        labelArray[minIndex].setBackground(Color.YELLOW);
    }
}

private void resetHighlights() {
    for (JLabel label : labelArray) {
        label.setBackground(Color.WHITE);
    }
}

private void updateLabels() {
    for (int k = 0; k < array.length; k++) {
        labelArray[k].setText(String.valueOf(array[k]));
    }
}

private void reset() {
    inputField.setText("");
    panelArray.removeAll();
    panelArray.revalidate();
}

```

```

        panelArray.repaint();
        stepArea.setText("");
        stepButton.setEnabled(false);
        sorting = false;
        j = 0;
        stepCount = 1;
    }

    private String arrayToString(int[] arr) {
        StringBuilder sb = new StringBuilder();
        for (int k = 0; k < arr.length; k++) {
            sb.append(arr[k]);
            if (k < arr.length - 1) sb.append(", ");
        }
        return sb.toString();
    }
}

```

## BAB III

### PENUTUP

#### 3.1 Kesimpulan

Dari praktikum pekan 7 ini, dapat disimpulkan bahwa:

- Algoritma *Insertion Sort* dan *Selection Sort* dapat dipahami dengan lebih baik melalui visualisasi langkah demi langkah.
- Pemrograman GUI dengan Java Swing memungkinkan representasi visual yang interaktif dan edukatif.
- *Insertion Sort* lebih efisien untuk array kecil yang hampir terurut, sementara *Selection Sort* bekerja secara konsisten meskipun tidak seefisien untuk data besar.
- Pembuatan GUI seperti ini dapat memperdalam pemahaman terhadap konsep dasar algoritma dan struktur data.