

# **LAPORAN PRAKTIKUM 7**

## **STRUKTUR DATA**



**Oleh:**  
**Ibrahim Mousa Dhani**  
**2411532010**

**Dosen Pengampu : Dr. Wahyudi S.T, M.T.**  
**Mata Kuliah : Struktur Data**

**FAKULTAS TEKNOLOGI INFORMASI**  
**DEPARTEMEN INFORMATIKA**  
**UNIVERSITAS ANDALAS**  
**PADANG**

## **A. Pendahuluan**

Algoritma pengurutan atau sorting merupakan salah satu konsep fundamental dalam ilmu komputer yang berfungsi untuk mengatur kumpulan data ke dalam urutan tertentu, seperti menaik (ascending) atau menurun (descending). Pemahaman yang baik tentang berbagai jenis algoritma sorting sangat penting karena menjadi dasar bagi pengembangan aplikasi yang lebih kompleks dan efisien.

Salah satu contoh algoritma pengurutan yang digunakan pada praktikum ini yaitu Insertion dan Selection Sort. Insertion Sort adalah algoritma pengurutan yang bekerja dengan cara menyisipkan setiap elemen satu per satu ke posisi yang tepat dalam sebuah daftar (array) yang sedang dibangun. Sedangkan Selection Sort adalah algoritma pengurutan yang bekerja dengan cara berulang kali memilih elemen terkecil dari bagian yang belum terurut dan menukarnya dengan elemen paling awal dari bagian tersebut.

Untuk menjembatani antara konsep teoretis algoritma dan implementasi praktisnya, praktikum ini berfokus pada pembuatan sebuah aplikasi visual. Dengan menggunakan bahasa pemrograman Java dan GUI (Graphical User Interface),. Aplikasi ini memungkinkan pengguna untuk memasukkan serangkaian angka dan melihat proses pengurutan Insertion dan Selection Sort berjalan langkah demi langkah secara interaktif, mulai dari pencarian elemen minimum hingga proses pertukaran posisi.

## **B. Tujuan Praktikum**

1. Mengetahui dan memahami konsep dasar Insertion dan Selection Sort
2. Mengimplementasikan Insertion dan Selection Sort dalam bahasa pemrograman Java, menggunakan GUI (Graphical User Interface)
3. Memahami dan mengetahui komponen-komponen penting pada GUI seperti JFrame, JPanel, JLabel, JButton dan masih banyak lagi, serta penggunaan ActionListener untuk menangani interaksi pengguna.

## C. Langkah Langkah

### a. InsertionSortGUI

1. Pertama buat package baru dan beri nama pekan7, karna kita akan menggunakan GUI kita buat JFrame terlebih dahulu , caranya klik kanan pada bagian package klik new, lalu others, pilih WindowBuilder, pilih swing, dan pilih JFrame lalu tuliskan nama kelas nya yaitu InsertionSortGUI.

Tampilan awalnya akan seperti berikut :

```
1 package pekan7;
2
3 import java.awt.EventQueue[];
4
5 public class InsertionSortGUI_ extends JFrame {
6
7     private static final long serialVersionUID = 1L;
8     private JPanel contentPane;
9
10    /**
11     * Launch the application.
12     */
13    public static void main(String[] args) {
14        EventQueue.invokeLater(new Runnable() {
15            public void run() {
16                try {
17                    InsertionSortGUI_frame = new InsertionSortGUI_();
18                    frame.setVisible(true);
19                } catch (Exception e) {
20                    e.printStackTrace();
21                }
22            }
23        });
24    }
25
26    /**
27     * Create the frame.
28     */
29    public InsertionSortGUI () {
30        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
31        setBounds(100, 100, 450, 300);
32        contentPane = new JPanel();
33        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
34
35        setContentPane(contentPane);
36    }
37
38 }
```

2. Import Library yang dibutuhkan

```
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;
```

3. Deklarasikan semua komponen GUI dan variabel agar bisa diakses oleh semua metode di dalam kelas tersebut.

```
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;
```

4. Buat Constructor dan Desain GUI, pertama atur ppproperty Frame Utama: beri judul, ukuran, dan operasi saat ditutup.

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

5. Buat panel-panel yaitu Inisialisasi JPanel untuk setiap bagian: input, visualisasi array, kontrol, dan log. Buat komponen yaitu inisialisasi semua komponen seperti JTextField, JButton, dan JTextArea.

```
// 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);
```

6. Masukkan komponen ke dalam panel yang sesuai. Kemudian, masukkan panel-panel tersebut ke dalam JFrame utama menggunakan BorderLayout.

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

7. Tambahkan addActionListener pada setiap tombol yang sudah dibuat tadi yang nanti akan terhubung dengan metode yang akan dibuat setelah ini, bisa disable terlebih dahulu agar tidak terjadi error sampai semua metode nya selesai dibuat

```
// Event Set Array  
setButton.addActionListener(e -> setArrayFromInput());  
  
// Event Langkah Selanjutnya  
stepButton.addActionListener(e -> performStep());  
  
// Event Reset  
resetButton.addActionListener(e -> reset());
```

8. Lalu buat metode setArrayFromInput yang berfungsi untuk membaca input, mengubahnya menjadi array berupa angka, dan menampilkannya. Ambil teks dari inputField. gunakan split(",") untuk memisahkan angka. lakukan parsing dari String ke int (gunakan try-catch untuk menangani error jika input bukan angka). bersihkan panelArray dari visualisasi sebelumnya. buat JLabel baru untuk setiap angka, beri style, dan tambahkan ke panelArray. panggil revalidate() dan repaint() pada panel agar perubahan terlihat. aktifkan tombol "Langkah Selanjutnya".

```

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

```

9. Lalu buat metode performStep yang berfungsi untuk menjalankan satu iterasi dari algoritma Insertion Sort, metode ini akan bekerja saat tombol “Langkah Selanjutnya” di klik. Metode ini mengambil satu elemen (key) dan menyisipkannya ke posisi yang benar di bagian array yang sudah diurutkan. Dan memanggil updateLabels() untuk memperbarui tampilan visual serta menulis log tentang apa yang terjadi pada langkah tersebut ke stepArea lalu menghentikan

```

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!");
        }
    }
}

```

proses dan menampilkan pesan saat array sudah terurut sepenuhnya.

10. Buat metode updateLabels yang berfungsi untuk memperbarui teks pada setiap JLabel di panelArray agar sesuai dengan nilai terbaru dalam int[] array. Ini dipanggil setiap kali ada perubahan urutan angka.

```

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

```

11. Selanjutnya buat metode reset yang berfungsi untuk mengembalikan/mereset angka dan semua operasi yang telah dilakukan seperti semula

```

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

```

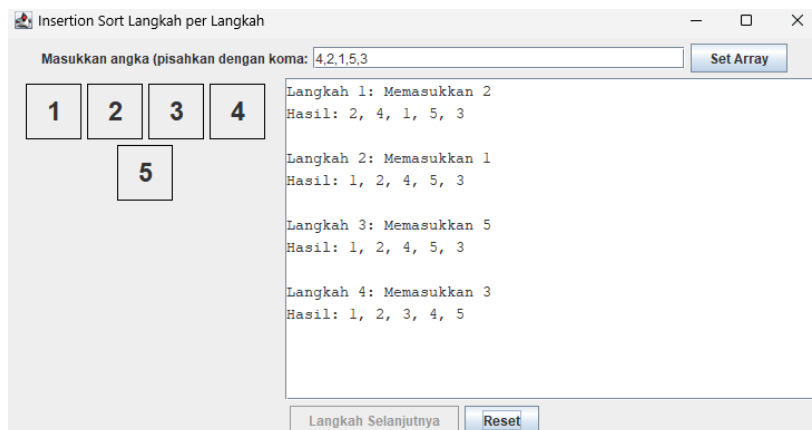
12. Buat metode main, yang berfungsi untuk membuat dan menampilkan jendela aplikasi, metode main ini otomatis muncul saat membuat class dengan JFrame menggunakan WindowBuilder saat awal tadi

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

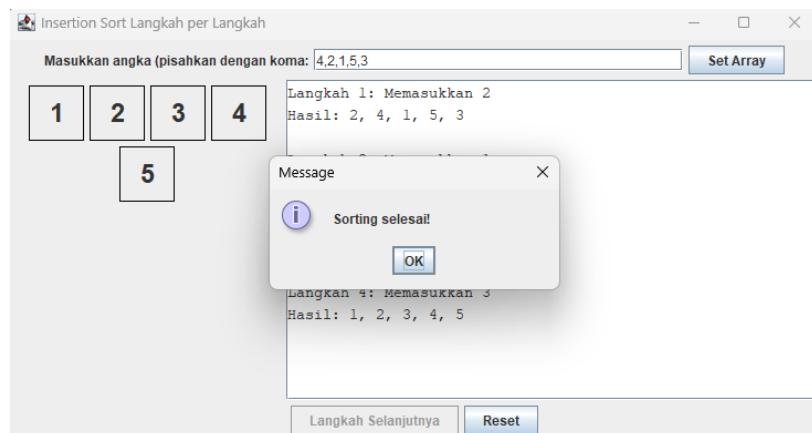
13. Terakhir metode arrayToString ini merupakan metode pembantu yang mengubah array integer menjadi String agar mudah dibaca dan ditampilkan di log

```
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();  
}
```

14. Setelah semua benar dan tidak ada eror, maka jalankan program, lalu akan menghasilkan output sebagai berikut :  
Disini saya menginputkan angka 4, 2, 1, 5, lalu klik Set Array dan akan muncul gambar di sebelah kiri layer



- lalu klik tombol Langkah Selanjutnya dan akan muncul penjelasan pada log, lakukan proses tersebut berulang kali hingga muncul pop up Sorting telah selesai



## b. SelectionSortGUI

1. Sama seperti yang sebelumnya, buat pada package pekan7 class baru, karna kita akan menggunakan GUI kita buat JFrame terlebih dahulu , caranya klik kanan pada bagian package klik new, lalu others, pilih WindowBuilder, pilih swing, dan pilih JFrame lalu tuliskan nama kelas nya yaitu SelectionSortGUI.
2. Selanjutnya import library yang dibutuhkan

```
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;
```

3. Deklarasikan semua komponen GUI dan variabel agar bisa diakses oleh semua metode di dalam kelas tersebut

```
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;
```

4. Selanjutnya kita akan mendesain antarmuka user (GUI). Dengan cara buat Constructor dan Desain GUI, pertama atur properti Frame Utama: beri judul, ukuran, dan operasi saat ditutup

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

5. Buat panel-panel yaitu Inisialisasi JPanel untuk setiap bagian: input, visualisasi array, kontrol, dan log. Buat komponen yaitu inisialisasi semua komponen seperti JTextField, JButton, dan JTextArea.

```
//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");
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);
```

6. Masukkan komponen ke dalam panel yang sesuai. Kemudian, masukkan panel-panel tersebut ke dalam JFrame utama menggunakan BorderLayout.

```
//Tambahkan panel ke frame
add(inputPanel, BorderLayout.NORTH);
add(panelArray, BorderLayout.CENTER);
add(controlPanel, BorderLayout.SOUTH);
add(scrollPane, BorderLayout.EAST);
```

7. Tambahkan addActionListener pada setiap tombol yang sudah dibuat tadi yang nanti akan terhubung dengan metode yang akan dibuat setelah ini, bisa disable terlebih dahulu agar tidak terjadi error sampai semua metode nya selesai dibuat

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

//Event langkah selanjutnya
stepButton.addActionListener(e -> performStep());

//event reset
resetButton.addActionListener(e -> reset());
```

8. Lalu buat metode setArrayFromInput yang berfungsi untuk membaca input, mengubahnya menjadi array berupa angka, dan menampilkannya. Ambil teks dari inputField. gunakan split(",") untuk memisahkan angka. lakukan parsing dari String ke int (gunakan try-catch untuk menangani error jika input bukan angka). bersihkan panelArray dari visualisasi sebelumnya. buat JLabel baru untuk setiap angka, beri style, dan tambahkan ke panelArray. panggil revalidate() dan repaint() pada panel agar perubahan terlihat. aktifkan tombol "Langkah Selanjutnya".



9. Lalu buat metode `performStep` yang berfungsi untuk menjalankan satu langkah dari algoritma Selection Sort setiap kali tombol "Langkah Selanjutnya" diklik. Cara kerjanya yaitu program akan memindai bagian array yang belum terurut dari indeks `i` sampai akhir untuk menemukan angka terkecil. Setiap klik akan memajukan pemindaian ini satu langkah `j++`. Angka terkecil yang ditemukan sejauh ini akan ditandai dengan warna kuning (`highlightMinIndex`). Setelah seluruh bagian yang belum terurut selesai dipindai, angka terkecil tersebut akan ditukar dengan angka di posisi `i` dan proses diulangi untuk sisa array `i++`.

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

10. Setelah itu buat metode `highlightMinIndex` dan `resetMinIndex`. `highlightMinIndex` berfungsi mengubah warna latar belakang JLabel dari angka minimum yang ditemukan menjadi kuning. `resetMinIndex` berfungsi mengembalikan warna latar belakang semua JLabel menjadi putih.

```
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);
    }
}
```

11. Buat metode `updateLabels` yang berfungsi untuk membantu menyinkronkan tampilan, setelah terjadi pertukaran nilai di dalam array, metode ini akan memperbarui teks pada setiap JLabel agar sesuai.

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

12. Selanjutnya buat metode reset yang berfungsi untuk mengembalikan/mereset angka dan semua operasi yang telah dilakukan seperti semula

```
private void reset() {  
    inputField.setText("");  
    panelArray.removeAll();  
    panelArray.revalidate();  
    panelArray.repaint();  
    stepArea.setText("");  
    stepButton.setEnabled(false);  
    sorting = false;  
    i=0;  
    j= 1;  
    stepCount = 1;  
}
```

13. Buat metode main, yang berfungsi untuk membuat dan menampilkan jendela aplikasi, metode main ini otomatis muncul saat membuat class dengan JFrame menggunakan WindowBuilder saat awal tadi

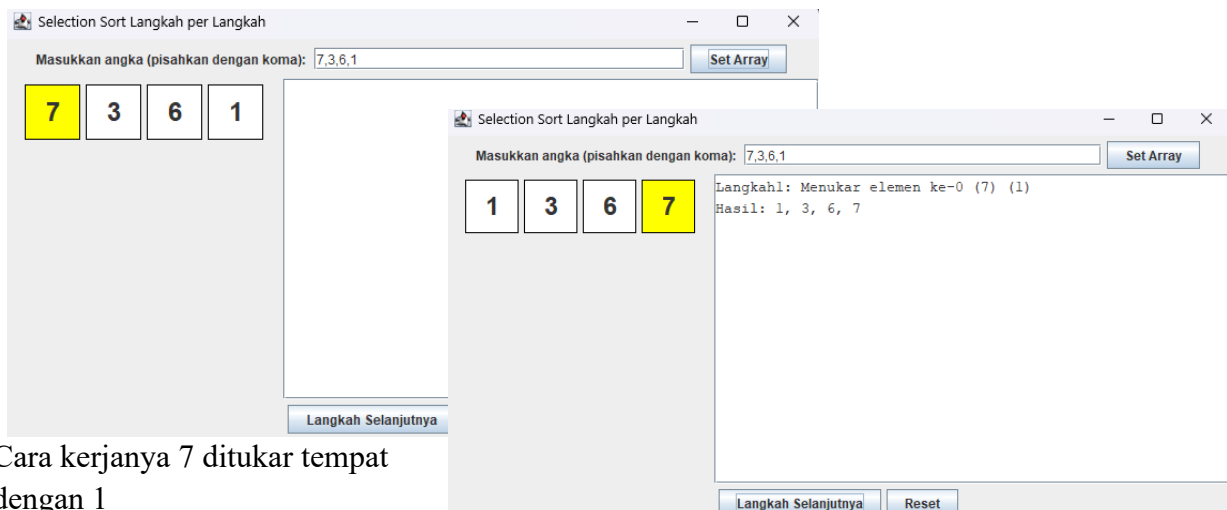
```
public static void main(String[] args) {  
    EventQueue.invokeLater(new Runnable() {  
        public void run() {  
            try {  
                SelectionSortGUI frame = new SelectionSortGUI();  
                frame.setVisible(true);  
            } catch (Exception e) {  
          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

14. Terakhir metode arrayToString ini merupakan metode pembantu yang mengubah array integer menjadi String agar mudah dibaca dan ditampilkan di log

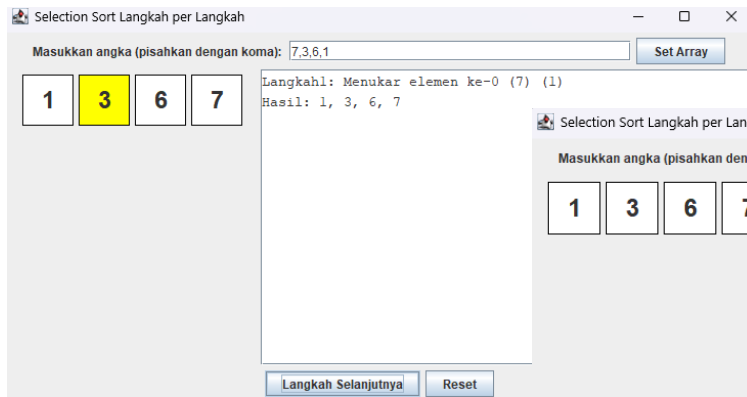
```
private String arrayToString(int [] arr) {  
    StringBuilder sb = new StringBuilder();  
    for (int k = 0; k < arr.length; k++) {  
        sb.append(arr[k]);  
        if(k < arr.length - 1) sb.append(", ");  
    }  
    return sb.toString();  
}
```

15. Setelah semua benar dan tidak ada eror, maka jalankan program, lalu akan menghasilkan output sebagai berikut :

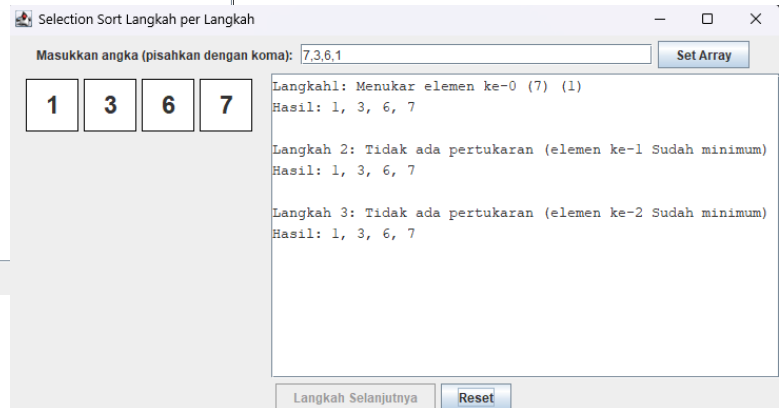
Disini saya menginputkan angka 7, 3, 6, 1, lalu klik Set Array dan akan muncul gambar di sebelah kiri layer



Cara kerjanya 7 ditukar tempat dengan 1

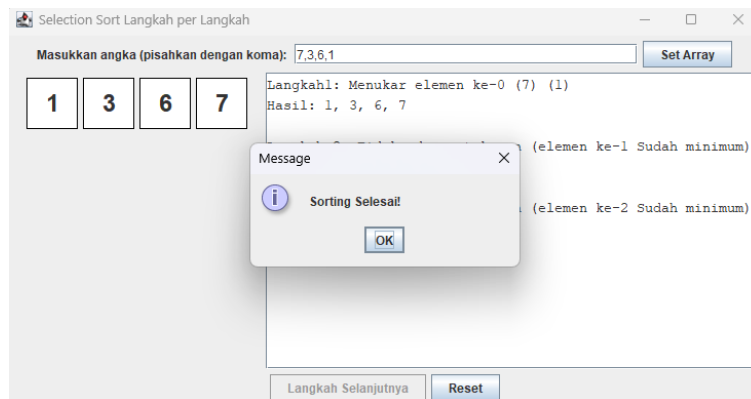


lalu di cek apakah 3 dan 6 perlu dilakukan pertukaran



setiap tombol “Langkah Selanjutnya” di klik maka akan muncul penjelasan langkah per langkah pada bagian log

lalu klik tombol Langkah Selanjutnya dan akan muncul penjelasan pada log, lakukan proses tersebut berulang kali hingga muncul pop up Sorting telah selesai



## D. Kesimpulan

Berdasarkan praktikum yang telah dilakukan, dapat disimpulkan pada praktikum ini kita dapat memahami dan mengetahui kosep dasar dan algoritma/cara kerja Insertion serta Selection Sort menggunakan GUI (Graphical User Interface) untuk membuat program yang interaktif bagi user dan menampilkan/memvisualisasikan algoritma nya secara langkah per langkah.