

**LAPORAN PRAKTIKUM STRUKTUR DATA**  
**SORTING**



OLEH:  
AHMAD ZAHRAN  
2411532004

DOSEN PENGAMPU:  
DR. WAHYUDI, S.T, M.T

FAKULTAS TEKNOLOGI INFORMASI  
DEPARTEMEN INFORMATIKA  
UNIVERSITAS ANDALAS

2025

## A. Pendahuluan

Struktur Data Sorting (pengurutan) adalah dasar dalam pemrograman yang sangat penting untuk mengatur data agar lebih mudah diproses. Ada dua jenis sorting yang sering diajarkan adalah **Insertion Sort** dan **Selection Sort**.

## B. Tujuan

Tujuan dilakukannya praktikum ini adalah sebagai berikut:

1. Memahami Struktur Data Sorting pada java
2. Mengimplementasikan Sorting pada java dalam bentuk GUI

## C. Langkah Kerja Praktikum

### a. Kelas InsertionSortGUI

```
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;
```

1. Pertama buat class InsertionSortGUI, lalu buatlah array yang bertipe data Integer, dan deklarasikan fitur-fitur yang ingin dipakai pada GUI seperti JLabel, JButton, JTextField, JPanel, JTextArea, lalu buat variabel untuk penanda proses dalam insertion sort, dan buat penanda posisi dalam insertion sort, lalu buat penanda apakah sortingnya sedang berlangsung

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

2. Method main, untuk menjalankan program GUI di java agar dapat berjalan dalam thread GUI

```

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

```

3. Membuat window utama untuk JFrame

```

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);

```

4. Untuk menambahkan Panel untuk menginputkan Array

```

panelArray = new JPanel(new FlowLayout());

```

5. Menambahkan panel untuk menampilkan nilai-nilai array

```

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

```

6. Menambahkan panel untuk mengontrol proses sorting yang mana disini terdiri dari dua tombol yaitu StepButton untuk menjalankan langkah dari sorting dan resetButton untuk mereset.

```

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

```

7. Menambahkan JTextArea untuk menampilkan langkah-langkah dari sorting

```

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

```

8. Lalu terakhir tambahkan semua panel ke JFrame sesuai dengan posisinya



```

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

```

9. Kode ini untuk menambahkan EventListener agar Tombol bias berfungsi

```

private void setArrayFromInput() {
    String text = inputField.getText().trim();
    if (text.isEmpty()) return;
    String[] parts = text.split(",");
    array = new int[parts.length];

```

10. Buat method setArrayFromInput, dengan penjelasan sebagai berikut pertama untuk mengambil teks dari inputField, dan pisahkan teks berdasarkan koma menjadi array dengan typedata String

```

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;
}

```

11. Lalu konversi String ke Integer

```

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

```

12. Dan buat variabel untuk langkah-langkah Sorting

```

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]);
}

```

13. Lalu kode ini untuk Membuat label untuk semua element array, dengan cara setiap element array dikonversi menjadi JLabel

```
private void performStep() {
    if (i < array.length && sorting) {
```

14. Method `performStep`, ini untuk menjalankan setiap langkah dari insertion sort, pertama cek apakah sorting masih berlangsung dan 'i' kecil dari panjang array

```
int key = array[i];
j = i - 1;
```

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

15. Lalu buat key untuk menjalankan sorting untuk meletakkan nilai di tempat yang sesuai, dan membuat `stepLog` untuk menampilkan langkah ke berapa dan "key" yang sedang dimasukkan

```
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++;
```

16. Kode ini untuk menggeser array yang lebih besar dari key ke kanan hingga posisinya sesuai, dan jika sudah sesuai dengan tempatnya Update tampilan sesuai urutan terbaru

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

17. Dan cek apakah nilai sudah sesuai dengan panjang array

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

18. Method `updateLabels`, untuk mengupdate teks dari `JLabel` di 'panelArray' untuk membuat kondisi array terbaru

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

19. Method reset, untuk mereset seluruh tampilan dan data agar bisa mengulang proses dari awal

b. Kelas SelectionSort

```
public class SelectionSort extends JFrame {
    //Ahmad Zahran:)

    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;
```

1. Sama seperti insertion sort tadi untuk GUI dari Selection Sort ini masih sama, Pertama buat class SelectionSort, lalu buatlah array yang bertipe data Integer, dan deklarasikan fitur-fitur yang ingin dipakai pada GUI seperti JLabel, JButton, JTextField, JPanel, JTextArea, lalu buat variabel untuk penanda proses dalam insertion sort, dan buat penanda posisi dalam insertion sort, lalu buat penanda apakah sortingnya sedang berlangsung



```

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();
            }
        }
    });
}

```

2. Method main, untuk menjalankan program GUI di java agar dapat berjalan dalam thread GUI

```

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

```

3. Membuat window utama untuk JFrame

```

// 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);

```

4. Untuk menambahkan Panel untuk menginputkan Array

```

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

```

5. Menambahkan panel untuk menampilkan nilai-nilai array

```

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

```

6. Menambahkan panel untuk mengontrol proses sorting yang mana disini terdiri dari dua tombol yaitu StepButton untuk menjalankan langkah dari sorting dan resetButton untuk mereset.

```
// 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);
```

7. Menambahkan JTextArea untuk menampilkan langkah-langkah dari sorting

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

8. Lalu terakhir tambahkan semua panel ke JFrame sesuai dengan posisinya

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

9. Kode ini untuk menambahkan EventListener agar Tombol bisa berfungsi

```
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;
    }
}
```

10. Method setArrayFromInput, dengan penjelasan sebagai berikut pertama untuk mengambil teks dari inputField, dan pisahkan teks berdasarkan koma menjadi array dengan typedata String

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

11. Dan buat variabel untuk langkah-langkah Sorting



```

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

```

12. Lalu kode ini untuk Membuat label untuk semua element array, dengan cara setiap element array dikonversi menjadi JLabel

```

panelArray.revalidate();
panelArray.repaint();
highlightMinIndex();

```

13. Dan memanggil method highlightMinIndex

```

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

```

14. Method performStep, untuk melakukan langkah-langkah selection sort pertama cek apakah i belum mencapai akhir array dan jika belum berarti proses masih berlangsung, dan inisialisasi minIndex saat pertama kali memulai langkahnya

```

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

```

15. Cari array terkecil dari i ke kanan

```

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

```

16. Jika sudah memukannya, dan jika minIndex tidak sama dengan i berarti ada yang lebih kecil maka kita perlu menukar element i dengan element terkecil di minIndex dan simpan di variabel temp

```

// 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();

```

17. Lalu menambahkan log yang terjadi pertukaran ke area teks, dan jika tak ada pertukaran lalu tampilkan hasil sementara, dan update tampilan pada GUI, dan memanggil method highlightMinIndex

```

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

```

18. Lalu kode ini untuk memeriksa bagian akhir sorting

```

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

```

19. Buat method highlightMinIndex untuk menandai elemen minimum array

```

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

```

20. Method resetHighlights untuk menghapus kembali warna yang ditandai sebelumnya

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

21. Method updateLabels untuk mengupdate isi dari labelArray

```
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();
}
```

22. Dan terakhir untuk mengubah array ke string dengan koma dan menampilkan hasil sorting di stepArea

#### D. Kesimpulan

Melalui praktikum ini, mahasiswa memperoleh pemahaman yang lebih dalam tentang bagaimana struktur data Sorting bekerja. Dan bagaimana cara kerja insertion sort dan selection sort bekerja dengan visualisasi tiap-tiap langkah dari sorting tersebut