

LAPORAN PRAKTIKUM STRUKTUR DATA
IMPLEMENTASI ALGORITMA SORTING SELECTION DAN INSERTION
PADA PEMROGRAMAN JAVA



Oleh :

DERIEL CHAERAHMAN

NIM 2411533007

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

ASISTEN PRAKTIKUM : RAHMAT DWIRIZKI OLDERS

FAKULTAS TEKNOLOGI INFORMASI

DEPARTEMEN INFORMATIKA

UNIVERSITAS ANDALAS

2025

A. Pendahuluan

Praktikum ini dilakukan untuk mengurutkan (sorting) data yang merupakan salah satu proses fundamental untuk mengorganisasi data secara terstruktur dan efisien. Sorting memudahkan dalam proses pencarian, analisis dan tampilan data. Membuat GUI dengan algoritma selection dan Insertion sort yang dapat menginputkan angka lalu program mengikuti (flow) dari penerapan algoritma sorting pada logic program sehingga dapat melihat visualisasi sorting secara bertahap.

1. Selection Sort

Merupakan algoritma pengurutan data yang beroperasi dengan memisahkan array menjadi 2 bagian. Pemisahan tersebut membagi subarray menjadi yang telah diurutkan dan tidak diurutkan. Memilih elemen terkecil/terbesar dari subarray yang belum diurutkan, lalu elemen ini terpilih ini dipindahkan ke akhir subarray yang sudah diurutkan. Melibatkan beberapa iteratif yang sistematis.

Kelebihannya yaitu sederhana dan mudah dipahami. Penggunaan memori efisien. Tidak sensitif terhadap urutan awal. Performa yang dapat diprediksi.

Kekurangannya efisiensi waktu yang rendah (untuk data set besar). Tidak stabil (jika ada 2 elemen dengan nilai yang sama). Tidak adaptif. Tidak Skalabel untuk data set besar.

2. Insertion Sort

Merupakan algoritma pengurutan data yang beroperasi dengan menyisipkan tiap elemen ke posisi yang sesuai pada bagian array yang sudah terurut. Ambil elemen dan simpan sebagai key, lalu bandingkan key dengan elemen-elemen di subarray, hingga menemukan posisi key lebih besar atau sama dengan elemen di sub array hingga mencapai awal array.

B. Tujuan

Tujuan dari dilakukannya praktikum ini adalah :

1. Memahami dan mengaplikasikan algoritma Sorting (Selection dan Insertion) dalam program java untuk mengurutkan data.
2. Membuat GUI dengan implementasi sorting sort.

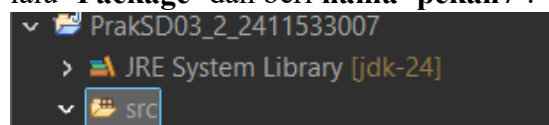
C. Langkah kerja praktikum

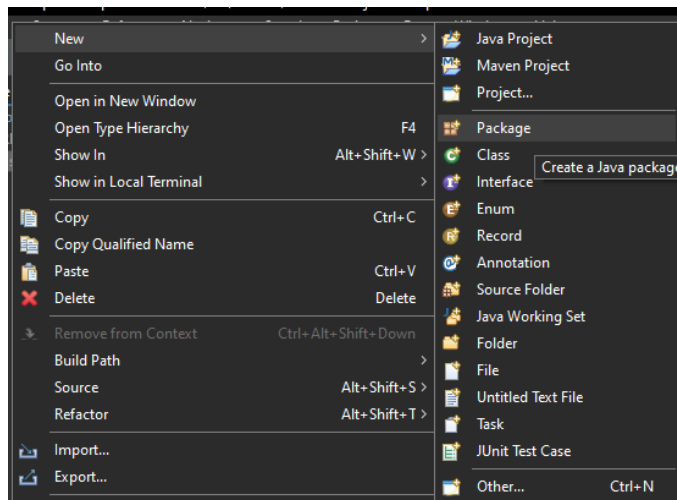
a. Alat dan Bahan

1. Perangkat computer atau laptop
2. Jaringan internet
3. IDE (Integrated Development Environment) direkomendasikan Eclipse IDE
4. Java JDK (Java Development Kit)

b. Package pekan 7

1. Buat new package. **Buka java project** yang telah dibuat sebelumnya, lalu **klik kanan** pada folder 'src', setelahnya akan muncul list option, pilih 'new', lalu 'Package' dan beri nama 'pekan7'.





c. Program InsertionSortGUI

1. Klik kanan pada package pekan7, new, other, windowBuilder lalu JFrame dan beri nama file GUI “**InsertionSortGUI**”.
2. Import package java util untuk memanipulasi array dan struktur data, java awt untuk komponen GUI dasar berupa layout, dan java swing untuk membuat komponen GUI seperti JButton dan JLabel.

```
1 package pekan7;
2 import java.util.*;
3 import java.awt.*;
4 import javax.swing.*;
```

3. Deklarasi class “InsertionSortGUI” dengan menggunakan JFrame (jendela utama aplikasi Swing).

```
6 public class InsertionSortGUI extends JFrame {
```

4. Membuat Variabel Global, array untuk menyimpan angka yang akan diurutkan. Label Array yang menampilkan secara visual. StepButton, reset dan setButton sebagai tombol GUI. InputField tempat input angka dari user. panelArray untuk menampilkan angka dalam bentuk label. stepArea untuk mencetak langkah-langkah sorting. Variabel i dan j variabel iterasi dalam insertion sort. Sorting boolean untuk memeriksa sorting sedang berjalan. StepCount sebagai nomor langkah sorting.

```
7 private static final long serialVersionUID = 1L;
8 private int[] array;
9 private JLabel[] labelArray;
10 private JButton stepButton, resetButton, setButton;
11 private JTextField inputField;
12 private JPanel panelArray;
13 private JTextArea stepArea;
14
15 private int i = 1, j;
16 private boolean sorting = false;
17 private int stepCount = 1;
```

5. **Main method.** Menjalankan GUI. EventQueue untuk menjalankan kode GUI, menjalakkannya di thread yang benar. new InsertionSortGUI(i) membuat object

baru dari kelas InsertionSortGUI turunan JFrame. `frame.setVisible(true);` untuk menampilkan GUI ke display.

```
22 public static void main(String[] args) {
23     EventQueue.invokeLater(new Runnable() {
24         public void run() {
25             try {
26                 InsertionSortGUI frame = new InsertionSortGUI();
27                 frame.setVisible(true);
28             } catch (Exception e) {
29                 e.printStackTrace();
30             }
31         }
32     });
33 }
```

6. Konstruktor GUI, menentukan model/tampilan dari GUInya berupa ukuran, layout dan semua komponen GUI.

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

Panel input, terdiri dari inputField untuk tempat user mengetik angka dan setButton tombol yang memproses input dan menampilkan array.

```
45 //Panel input
46 JPanel inputPanel = new JPanel(new FlowLayout());
47 inputField = new JTextField(30);
48 setButton = new JButton("Set Array");
49 inputPanel.add(new JLabel("Masukan angka (pisahkan dengan koma:"));
50 inputPanel.add(inputField);
51 inputPanel.add(setButton);
```

Panel array visual digunakan untuk menampilkan elemen array sebagai label(JLabel) secara horizontal.

```
53 //Panel array visual
54 panelArray = new JPanel();
55 panelArray.setLayout(new FlowLayout());
```

Panel tombol kontrol terdiri dari tombol step dan reset button.

```
57 //Panel kontrol
58 JPanel controlPanel = new JPanel();
59 stepButton = new JButton("Langkah selanjutnya");
60 resetButton = new JButton("Reset");
61 stepButton.setEnabled(false);
62 controlPanel.add(stepButton);
63 controlPanel.add(resetButton);
```

Area teks yang menampilkan log langkah-langkah sorting dari tiap penyelesaian array.

```
65 //Area teks untuk log langkah-langkah
66 stepArea = new JTextArea(8, 60);
67 stepArea.setEditable(false);
68 stepArea.setFont(new Font("Monospaced", Font.PLAIN, 14));
69 JScrollPane scrollPane = new JScrollPane(stepArea);
```

Menambahkan panel ke frame, fungsi add() untuk menempatkan panel ke dalam frame menggunakan layout manager “Border layout”, yang membagi frame menjadi 5 bagian utama yaitu north, south, east, west dan center.

```
71 //Tambahkan panel ke frame
72 add(inputPanel, BorderLayout.NORTH);
73 add(panelArray, BorderLayout.CENTER);
74 add(controlPanel, BorderLayout.SOUTH);
75 add(scrollPane, BorderLayout.EAST);
76
```

Menambahkan event listener ke tombol, yaitu event handler ke setButton, yang ketika ditekan akan menjalankan method.

```
77 //Event Set Array
78 setButton.addActionListener(e -> setArrayFromInput());
79
80 //Event Langkah Selanjutnya
81 stepButton.addActionListener(e -> performStep());
82
83 //Event reset
84 resetButton.addActionListener(e -> reset());
85 }
```

7. **Method setArrayFromInput** berfungsi mengambil input array dari user, memprosesnya, dan menampilkannya secara visual. Modifier “private” artinya method ini hanya bisa dipanggil di dalam kelas ini. “void” method tidak mengembalikan nilai.

```
87 //Function setArray
88 private void setArrayFromInput() {
```

Mengambil input teks input field, ‘trim()’ berfungsi menghapus spasi di awal/akhir. Jika input kosong maka method berhenti.

```
89 String text = inputField.getText().trim();
90 if (text.isEmpty()) return;
```

Memisahkan input menjadi array string. ‘split(“,”)’ berfungsi memisahkan input berdasarkan koma. ‘array = new int[parts.length];’ Membuat array integer sebanyak jumlah angka yang diinput user.

```
91 String[] parts = text.split(",");
92 array = new int[parts.length];
```

Try{} and catch {} berfungsi untuk menangani kesalahan/eror saat program berjalan, agar program tidak langsung berhenti/crash melainkan menampilkan pesan error ke user.

Konversi String ke Integer dari array “parts” menggunakan for loop yang blok kode nya akan mengubah tiap elemen hasil looping menjadi integer ‘Integer.parseInt()’. ‘trim()’ menghapus spasi jika ada.

‘catch (NumberFormatException e)’ jika eror terjadi, maka program tidak akan langsung crash, melainkan tampil popup peringatan ke user menggunakan ‘JOptionPane.showMessageDialog’, lalu menghentikan eksekusi method dengan ‘return;’ agar tidak lanjut ke bagian lain.

Block try-catch memastikan input tidak valid yang menyebabkan program eror, dengan memberitahu user dengan popup message, proses parsing hanya dilakukan jika semua input valid. Membuat program lebih user-frienly dan dapat handle eror dengan baik.

```

93         try {
94             for (int k = 0; k < parts.length; k++) {
95                 array[k] = Integer.parseInt(parts[k].trim()); }
96         } catch (NumberFormatException e) {
97             JOptionPane.showMessageDialog(this, "Masukkan hanya angka yang dipisahkan "
98             + "dengan koma!", "Error", JOptionPane.ERROR_MESSAGE);
99             return;
100         }

```

Inisialisasi variabel sorting. Variabel *i* dan *stepCount* berfungsi untuk melacak posisi langkah dalam algoritma sorting. 'Sorting = true' menandakan proses sorting telah dimulai. Mengaktifkan proses tombol "Step". Mengosongkan log langkah sebelumnya (*stepArea*).

```

101         i = 1;
102         stepCount = 1;
103         sorting = true;
104         stepButton.setEnabled(true);
105         stepArea.setText("");

```

Reset tampilan panel array. Menghapus semua komponen lama dari *panelArray*, siapkan array *JLabel* untuk menampilkan elemen array ke user.

```

106         panelArray.removeAll();
107         labelArray = new JLabel [array.length];

```

Tampilkan elemen array ke GUI, loop setiap elemen array, buat *JLabel* berisi angka array.

```

108         for (int k = 0; k < array.length; k++) {
109             labelArray[k] = new JLabel (String.valueOf(array[k]));
110             labelArray[k].setFont(new Font("Arial", Font.BOLD, 24));
111             labelArray[k].setBorder(BorderFactory.createLineBorder(Color.BLACK));
112             labelArray[k].setPreferredSize(new Dimension(50,50));
113             labelArray[k].setHorizontalAlignment(SwingConstants.CENTER);
114             panelArray.add(labelArray[k]);
115         }

```

Refresh tampilan panel. *revalidate()*, Memberitahu layout manager untuk memperbarui posisi/ukuran komponen. *repaint()*, Memaksa komponen untuk di-redraw ulang, agar perubahan terlihat.

```

115         }
116         panelArray.revalidate();
117         panelArray.repaint();
118     }

```

Penjelasan method *setArrayFromInput()* : Mengambil input angka dari user dalam bentuk string, mengonversinya ke array integer (parsing), Menampilkan array tersebut sebagai deretan kotak angka di tampilan GUI, mengatur ulang kondisi agar siap untuk sorting visual secara bertahap. Sebagai langkah awal sebelum proses sorting dilakukan.

8. **Method *performStep()***, modifier *private* artinya hanya dapat diakses dalam class ini. 'Void' artinya method ini tidak mengembalikan nilai.

```

122     private void performStep() {

```

Memeriksa kondisi awal apakah sorting masih berlangsung. For loop mengecek apakah indeks '*i*' masih dalam batas array dan proses sorting masih aktif. '*i*' merupakan elemen array yang sedang dimasukan ke posisi yg tepat.

```

123         if (i < array.length && sorting) {

```

Mengambil nilai kunci(key). Key adalah nilai yang ingin dimasukkan ke posisi yang benar di bagian array yang sudah terurut. j adalah indeks untuk membandingkan key dengan elemen sebelumnya.

```
123         if (i < array.length && sorting) {
124             int key = array[i];
125             j = i - 1;
```

Logging langkah ke JTextArea. stepLog digunakan untuk membuat teks penjelasan tiap langkah sorting. Mencatat langkah ke berapa, dan elemen mana yang sedang dimasukkan (key)

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

Proses Sorting (Insertion Sort). Loop ini **menggeser elemen** yang lebih besar dari key ke kanan untuk memberi ruang bagi key. Loop ini **menggeser elemen** yang lebih besar dari key ke kanan untuk memberi ruang bagi key.

```
131             while(j >= 0 && array[j] > key) {
132                 array [j + 1] = array[j];
133                 j--;
134             }
135             array[j + 1] = key;
```

Memperbarui tampilan JLabel array di panel. Menambahkan hasil perubahan array ke stepArea agar user bisa melihat hasil tiap langkah.

```
135         updateLabels();
136         stepLog.append("Hasil: ").append(arrayToString(array)).append("\n\n");
137         stepArea.append(stepLog.toString());
```

Naikkan Indeks dan Langkah.

```
139             i++;
140             stepCount++;
```

Akhiri Jika Sorting Selesai. Jika 'i' sudah mencapai akhir array, proses sorting selesai, stepButton dimatikan, menampilkan popup bahwa sorting selesai.

```
142         if (i == array.length) {
143             sorting = false;
144             stepButton.setEnabled(false);
145             JOptionPane.showMessageDialog(this, "Sorting selesai!");
146         }
147     }
148 }
```

9. **Method updateLabels()**, berfungsi untuk memperbarui tampilan angka pada komponen GUI(labelArray) sesuai dengan isi terkini dari array. Melakukan loop untuk tiap elemen dalam array, mengubah teks dari label ke-k agar menampilkan angka baru setelah proses sorting. Mengubah integer menjadi string agar bisa ditampilkan sebagai teks. Dipakai setelah perubahan nilai array seperti saat proses sorting berlangsung.

```
150     //Function
151     private void updateLabels() {
152         for (int k = 0; k < array.length; k++) {
153             labelArray[k].setText(String.valueOf(array[k]));
154         }
155     }
```


10. **Method reset()**, berfungsi untuk mengatur ulang tampilan dan variabel ke kondisi awal (menghapus semua inputan dan proses). Mengatur ulang index dan penghitung langkah.

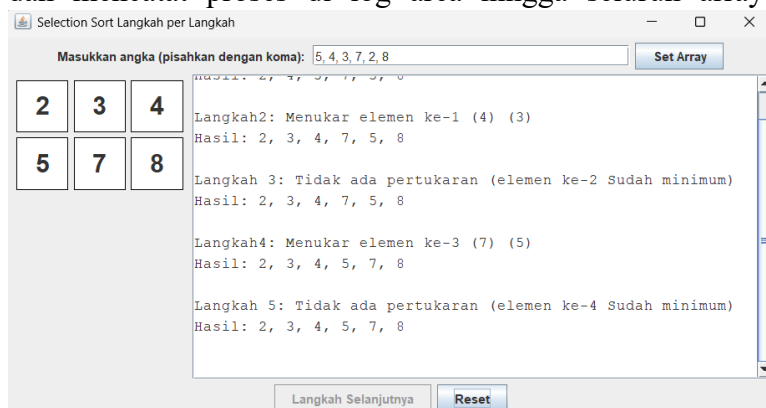
```
158 private void reset () {  
159     inputField.setText("");  
160     panelArray.removeAll();  
161     panelArray.revalidate();  
162     panelArray.repaint();  
163     stepArea.setText("");  
164     stepButton.setEnabled(false);  
165     sorting = false;  
166     i = 1;  
167     stepCount = 1;  
168 }
```

11. **Method arrayToString(int[] arr)**, berfungsi mengubah sebuah array integer menjadi string berformat seperti 1, 2, 3. Menampilkan array ke dalam stepArea selama proses sorting.

```
171 private String arrayToString(int[] arr) {  
172     StringBuilder sb = new StringBuilder();  
173     for (int k = 0; k < arr.length; k++) {  
174         sb.append(arr[k]);  
175         if (k < arr.length - 1) sb.append(", ");  
176     }  
177     return sb.toString();  
178 }  
179 }
```

12. **Output program** : Program InsertionSortGUI adalah aplikasi Java Swing yang mengimplementasikan algoritma Insertion Sort dengan visualisasi step-by-step. Program ini memiliki interface yang terdiri dari text field untuk input array (angka dipisahkan koma), panel visual yang menampilkan elemen array dalam bentuk kotak-kotak berlabel, tombol kontrol untuk menjalankan langkah sorting, dan area teks untuk menampilkan log setiap langkah proses.

Alur kerja program dimulai dengan pengguna memasukkan angka-angka yang akan diurutkan melalui text field, kemudian sistem akan memvalidasi input dan membuat visualisasi array. Setelah itu, pengguna dapat menekan tombol "Langkah selanjutnya" untuk menjalankan algoritma insertion sort secara bertahap - di setiap langkah, program akan mengambil elemen berikutnya sebagai kunci, mencari posisi yang tepat di bagian array yang sudah terurut, menggeser elemen yang lebih besar, dan memasukkan kunci di posisi yang benar. Setiap langkah akan memperbarui tampilan visual dan mencatat proses di log area hingga seluruh array terurut sempurna.



d. Program SelectionSortGUI

1. Klik kanan pada package pekan7, new, other, windowBuilder lalu JFrame dan beri nama file GUI “SelectionSortGUI”.
2. Import package java util untuk memanipulasi array dan struktur data, java awt untuk komponen GUI dasar berupa layout, java swing untuk membuat komponen GUI seperti JButton dan JLabel, dan java eventQueue untuk menjalankan GUI secara thread-safe (agar aman dan konsisten).

```
1 package pekan7;
2 import java.util.*;
3 import javax.swing.*;
4 import java.awt.*;
5 import java.awt.EventQueue;
```

3. Deklarasi class “InsertionSortGUI” dengan menggunakan JFrame (jendela utama aplikasi Swing).

```
8 public class SelectionSortGUI extends JFrame {
```

4. Deklarasi variabel. Menyimpan data array, elemen visual GUI, tombol interaktif GUI, dan variabel kontrol sorting. Variabel i sebagai index utama iterasi luar pada algoritma sorting, variabel j digunakan untuk inner loop saat membandingkan dan menukar elemen. Boolean sorting sebagai penanda proses sorting sedang berlangsung atau tidak. StepCount untuk menghitung proses sorting dilakukan. minIndex khusus untuk selection sort, menyimpan index dari nilai terkecil yang ditemukan selama satu putaran iterasi.

```
9 private static final long serialVersionUID = 1L;
10 private int[] array;
11 private JLabel[] labelArray;
12 private JButton stepButton, resetButton, setButton;
13 private JTextField inputField;
14 private JPanel panelArray;
15 private JTextArea stepArea;
16
17 private int i = 0, j;
18 private boolean sorting = false;
19 private int stepCount = 1;
20 private int minIndex;
```

5. **Main method.** Menjalankan GUI. EventQueue untuk menjalankan kode GUI, menjalankannya di thread yang benar. new SelectionSortGUI(i) membuat object baru dari kelas SelectionSortGUI turunan JFrame. frame.setVisible(true); untuk menampilkan GUI ke display

```
25 public static void main(String[] args) {
26     EventQueue.invokeLater(new Runnable() {
27         public void run() {
28             try {
29                 SelectionSortGUI frame = new SelectionSortGUI();
30                 frame.setVisible(true);
31             } catch (Exception e) {
32                 e.printStackTrace();
33             }
34         }
35     });
36 }
```

6. Konstruktor GUI, menentukan model/tampilan dari GUInya berupa ukuran, layout dan semua komponen GUI.

```
41 public SelectionSortGUI() {  
42     setTitle("Selection Sort Langkah per Langkah");  
43     setSize(750, 400);  
44     setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
45     setLocationRelativeTo(null);  
46     setLayout(new BorderLayout());  
47 }
```

Panel input, terdiri dari inputField untuk tempat user mengetik angka dan setButton tombol yang memproses input dan menampilkan array.

```
45 //Panel input  
46 JPanel inputPanel = new JPanel(new FlowLayout());  
47 inputField = new JTextField(30);  
48 setButton = new JButton("Set Array");  
49 inputPanel.add(new JLabel("Masukan angka (pisahkan dengan koma):"));  
50 inputPanel.add(inputField);  
51 inputPanel.add(setButton);
```

Panel array visual digunakan untuk menampilkan elemen array sebagai label(JLabel) secara horizontal.

```
53 //Panel array visual  
54 panelArray = new JPanel();  
55 panelArray.setLayout(new FlowLayout());
```

Panel tombol kontrol terdiri dari tombol step dan reset button.

```
57 //Panel kontrol  
58 JPanel controlPanel = new JPanel();  
59 stepButton = new JButton("Langkah selanjutnya");  
60 resetButton = new JButton("Reset");  
61 stepButton.setEnabled(false);  
62 controlPanel.add(stepButton);  
63 controlPanel.add(resetButton);
```

Area teks yang menampilkan log langkah-langkah sorting dari tiap penyelesaian array.

```
65 //Area teks untuk log langkah-langkah  
66 stepArea = new JTextArea(8, 60);  
67 stepArea.setEditable(false);  
68 stepArea.setFont(new Font("Monospaced", Font.PLAIN, 14));  
69 JScrollPane scrollPane = new JScrollPane(stepArea);
```

Menambahkan panel ke frame, fungsi add() untuk menempatkan panel ke dalam frame menggunakan layout manager "Border layout", yang membagi frame menjadi 5 bagian utama yaitu north, south, east, west dan center.

```
71 //Tambahkan panel ke frame  
72 add(inputPanel, BorderLayout.NORTH);  
73 add(panelArray, BorderLayout.CENTER);  
74 add(controlPanel, BorderLayout.SOUTH);  
75 add(scrollPane, BorderLayout.EAST);  
76
```

Menambahkan event listener ke tombol, yaitu event handler ke setButton, yang ketika ditekan akan menjalankan method.

```

77 //Event Set Array
78 setButton.addActionListener(e -> setArrayFromInput());
79
80 //Event Langkah Selanjutnya
81 stepButton.addActionListener(e -> performStep());
82
83 //Event reset
84 resetButton.addActionListener(e -> reset());
85 }

```

7. **Method setArrayFromInput ()**, berfungsi mengambil data angka dari input pengguna (berupa string), mengubahnya menjadi array integer, menyiapkan tampilan interaktif visual GUI untuk sorting, dan mengatur kondisi awal untuk proses selection sort.

```

91 private void setArrayFromInput() {
92     String text = inputField.getText().trim();
93     if (text.isEmpty()) return;
94     String[] parts = text.split(",");
95     array = new int[parts.length];
96     try {
97         for (int k = 0; k < parts.length; k++) {
98             array[k] = Integer.parseInt(parts[k].trim());
99         } catch (NumberFormatException e) {
100             JOptionPane.showMessageDialog(this, "Masukkan hanya angka yang dipisahkan "
101                 + "dengan koma!", "Error", JOptionPane.ERROR_MESSAGE);
102             return;
103         }
104         i = 0;
105         j = j + 1;
106         stepCount = 1;
107         sorting = true;
108         stepButton.setEnabled(true);
109         stepArea.setText("");
110         panelArray.removeAll();
111         labelArray = new JLabel[array.length];
112         for (int k = 0; k < array.length; k++) {
113             labelArray[k] = new JLabel(String.valueOf(array[k]));
114             labelArray[k].setFont(new Font("Arial", Font.BOLD, 24));
115             labelArray[k].setOpaque(true);
116             labelArray[k].setBackground(Color.WHITE);
117             labelArray[k].setBorder(BorderFactory.createLineBorder(Color.BLACK));
118             labelArray[k].setPreferredSize(new Dimension(50, 50));
119             labelArray[k].setHorizontalAlignment(SwingConstants.CENTER);
120             panelArray.add(labelArray[k]);
121         }
122         panelArray.revalidate();
123         panelArray.repaint();
124         highlightMinIndex();
125     }
126 }

```

8. **Method performStep()**, berfungsi menjalankan satu langkah (step) dari algoritma Selection Sort, setiap kali tombol "Step" ditekan. Proses dilakukan secara bertahap dan interaktif dalam GUI.

```

127 if (i < array.length - 1 && sorting) {
128     StringBuilder stepLog = new StringBuilder();
129     if (j == i + 1) {
130         minIndex = i;
131     }
132     if (j < array.length) {
133         if (array[j] < array[minIndex]) {
134             minIndex = j;
135         }
136         j++;
137     }
138     if (j == array.length) {
139         if (minIndex != i) {
140             int temp = array[i];
141             array[i] = array[minIndex];
142             array[minIndex] = temp;
143
144             stepLog.append("Langkah ").append(stepCount).append(": Menukar elemen ke-")
145                 .append(i).append(" (").append(array[minIndex]).append(") ")
146                 .append(" ").append(array[i]).append(")\n");
147         } else {
148             stepLog.append("Langkah ").append(stepCount)
149                 .append(": Tidak ada pertukaran (elemen ke-")
150                 .append(i).append(" Sudah minimum)\n");
151         }
152     }
153 }

```

9. **Method highlightMinIndex()**, berfungsi untuk menyorot (highlight) elemen terkecil (minimum) dalam array pada satu langkah Selection Sort, menggunakan warna kuning pada antarmuka grafis (GUI).

```
174 private void highlightMinIndex() {
175     resetHighlights();
176     if(minIndex >= 0 && minIndex < labelArray.length) {
177         labelArray[minIndex].setBackground(Color.YELLOW);
178     }
179 }
```

10. **Method resetHighlights()**, berfungsi untuk **menghapus warna highlight sebelumnya**, agar proses highlight baru (seperti di highlightMinIndex()) tidak menumpuk atau membuat tampilan membingungkan. dipanggil setiap kali sebelum memberi highlight baru, supaya hanya satu elemen yang diberi warna khusus.

```
182 private void resetHighlights() {
183     for(JLabel label : labelArray) {
184         label.setBackground(Color.WHITE);
185     }
186 }
```

11. **Method updateLabels()**, digunakan setiap kali ada perubahan pada isi array, seperti saat proses penukaran (swap) dalam sorting. Tujuannya untuk menyinkronkan tampilan GUI dengan data array yang terbaru.

```
189 private void updateLabels() {
190     for(int k = 0; k < array.length; k++) {
191         labelArray[k].setText(String.valueOf(array[k]));
192     }
193 }
```

12. **Method reset()**, digunakan untuk **mengembalikan tampilan dan variabel program ke kondisi awal**, seolah-olah pengguna belum memasukkan data apa pun. Biasanya dipakai setelah proses sorting selesai atau saat pengguna ingin mengulang.

```
196 private void reset() {
197     inputField.setText("");
198     panelArray.removeAll();
199     panelArray.revalidate();
200     panelArray.repaint();
201     stepArea.setText("");
202     stepButton.setEnabled(false);
203     sorting = false;
204     i=0;
205     j= 1;
206     stepCount = 1;
207 }
```

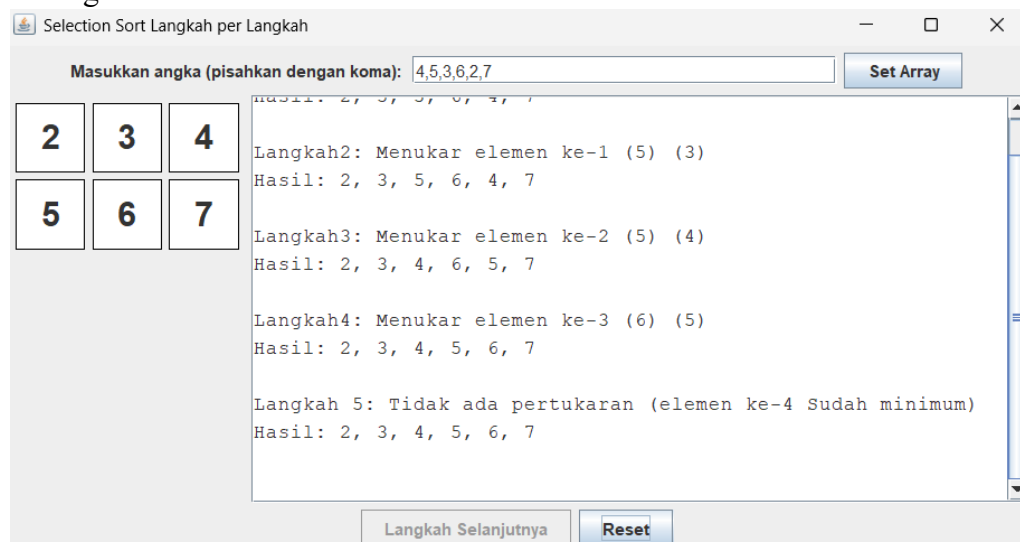
13. **Method arrayToString(int[] arr)**, berfungsi untuk **menampilkan isi array ke pengguna** dalam format yang rapi di JTextArea, seperti ketika menampilkan hasil langkah-langkah dalam proses sorting

```

210     private String arrayToString(int[] arr) {
211         StringBuilder sb = new StringBuilder();
212         for (int k = 0; k<arr.length; k++) {
213             sb.append(arr[k]);
214             if(k < arr.length - 1) sb.append(", ");
215         }
216         return sb.toString();
217     }
218
219 }

```

14. Output program : Program SelectionSortGUI berbasis GUI menampilkan proses pengurutan data menggunakan algoritma **Selection Sort** secara visual dan interaktif. Pengguna dapat memasukkan angka yang dipisahkan koma ke dalam kolom input, lalu menekan tombol “Step” untuk melihat setiap langkah pengurutan. Program akan mencari elemen terkecil dari sisa array, menukarnya dengan elemen saat ini (jika perlu), dan menampilkan hasil setiap langkah dalam tampilan teks serta memperbarui label angka pada panel secara visual. Pewarnaan kuning digunakan untuk menandai elemen terkecil yang sedang dipilih, memberikan pemahaman visual yang jelas tentang proses sorting.



D. Kesimpulan

Setelah melakukan praktikum disimpulkan implementasi algoritma sorting Selection Sort dan Insertion Sort dalam pemrograman Java untuk memahami proses pengurutan data secara lebih visual dan interaktif, terutama dengan penggunaan GUI (Graphical User Interface). Selection Sort bekerja dengan cara memilih elemen terkecil dari array dan menukarnya ke posisi yang tepat, sedangkan Insertion Sort menyisipkan elemen ke posisi yang sesuai di bagian array yang sudah terurut. Keduanya menunjukkan bagaimana logika algoritma bekerja secara bertahap, sehingga mempermudah proses pembelajaran konsep dasar struktur data dan algoritma.

