LAPORAN PRAKTIKUM

STRUKTUR DATA



DISUSUN OLEH :

Sasya Zamora

2411533014

DOSEN PENGAMPU :

Dr. Wahyudi, S.T.M.T

PROGRAM STUDI INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS ANDALAS

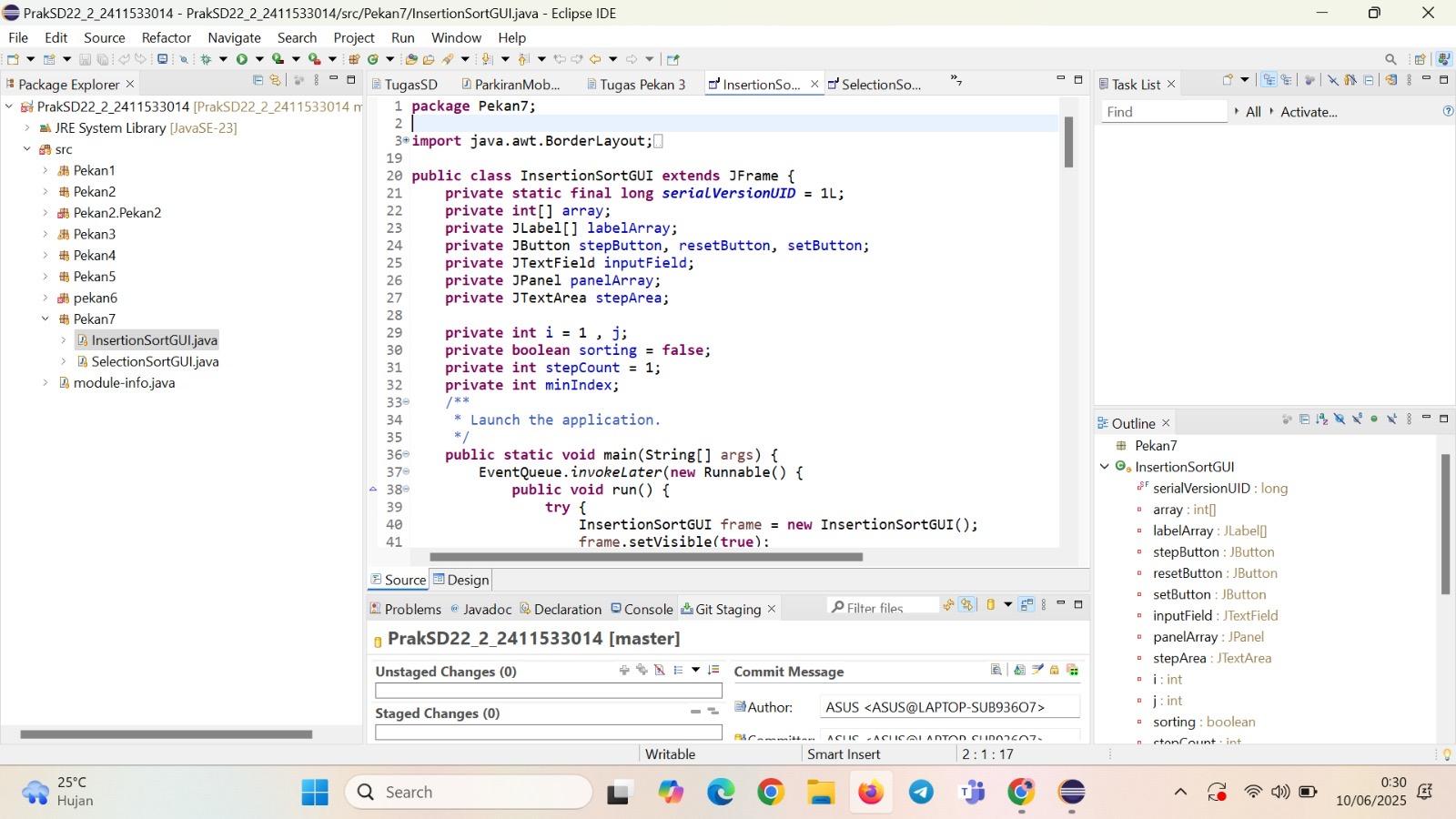
2025

1. TUJUAN
2. Mempelajari dan Menerapkan Algoritma Pengurutan.
3. Membangun Antarmuka Pengguna Grafis (GUI).
4. Menganalisis dan Membandingkan Algoritma.
5. PEMBAHASAN

Praktikum ini mengimplementasikan dan membandingkan Insertion Sort dan Selection Sort melalui aplikasi GUI Java. Kedua aplikasi, dengan GUI yang serupa, memvisualisasikan proses pengurutan secara bertahap. Insertion Sort membangun urutan terurut dengan memasukkan setiap elemen ke posisi yang tepat, lebih efisien pada data hampir terurut, tetapi kurang efisien pada data acak dan besar. Selection Sort, yang lebih mudah dipahami, menemukan dan menukar elemen terkecil secara berulang, namun performanya lebih buruk, terutama pada dataset besar, karena banyaknya pertukaran.

Visualisasi GUI memungkinkan perbandingan langsung kedua algoritma, menunjukkan trade-off antara kesederhanaan dan efisiensi. Pengguna dapat mengamati secara langsung bagaimana strategi pengurutan yang berbeda memengaruhi jumlah langkah dan efisiensi keseluruhan, memperkuat pemahaman tentang karakteristik dan batasan masing-masing algoritma.

1. LANGKAH PRAKTIKUM
2. Buka Eclipse IDE
3. Lalu, buat package pekan7
4. Klik kanan pada package7 pilih other
5. Jika belum mendownload windowbuilder, jika audah pilih windowbuilder
6. Lalu, pilih jframe
7. Masukkan nama classnya dan buat kodingan deperti dibawah ini :

* InsertionSortGUI

1. Deklarasi Package dan Import

• package Pekan7;: Menyatakan bahwa kelas ini berada dalam paket Pekan7.

• import java.awt.BorderLayout;: Mengimpor kelas BorderLayout dari paket java.awt, yang digunakan untuk mengatur tata letak komponen dalam frame.

2. Deklarasi Kelas InsertionSortGUI

Kelas InsertionSortGUI mewarisi dari JFrame, yang merupakan kelas dasar untuk jendela aplikasi Swing. serialVersionUID adalah ID unik untuk serialisasi objek.

3. Variabel Instance

Bagian ini mendeklarasikan variabel instance yang digunakan oleh kelas InsertionSortGUI:

• array: Array integer untuk menyimpan angka yang akan diurutkan.

• labelArray: Array JLabel untuk menampilkan angka-angka dalam GUI.

• stepButton, resetButton, setButton: Tombol-tombol untuk mengontrol proses pengurutan.

• inputField: JTextField untuk menerima input angka.

• panelArray, stepArea: Komponen GUI untuk menampilkan array dan langkah-langkah pengurutan.

• i, j, stepCount, minIndex: Variabel integer yang digunakan dalam algoritma pengurutan.

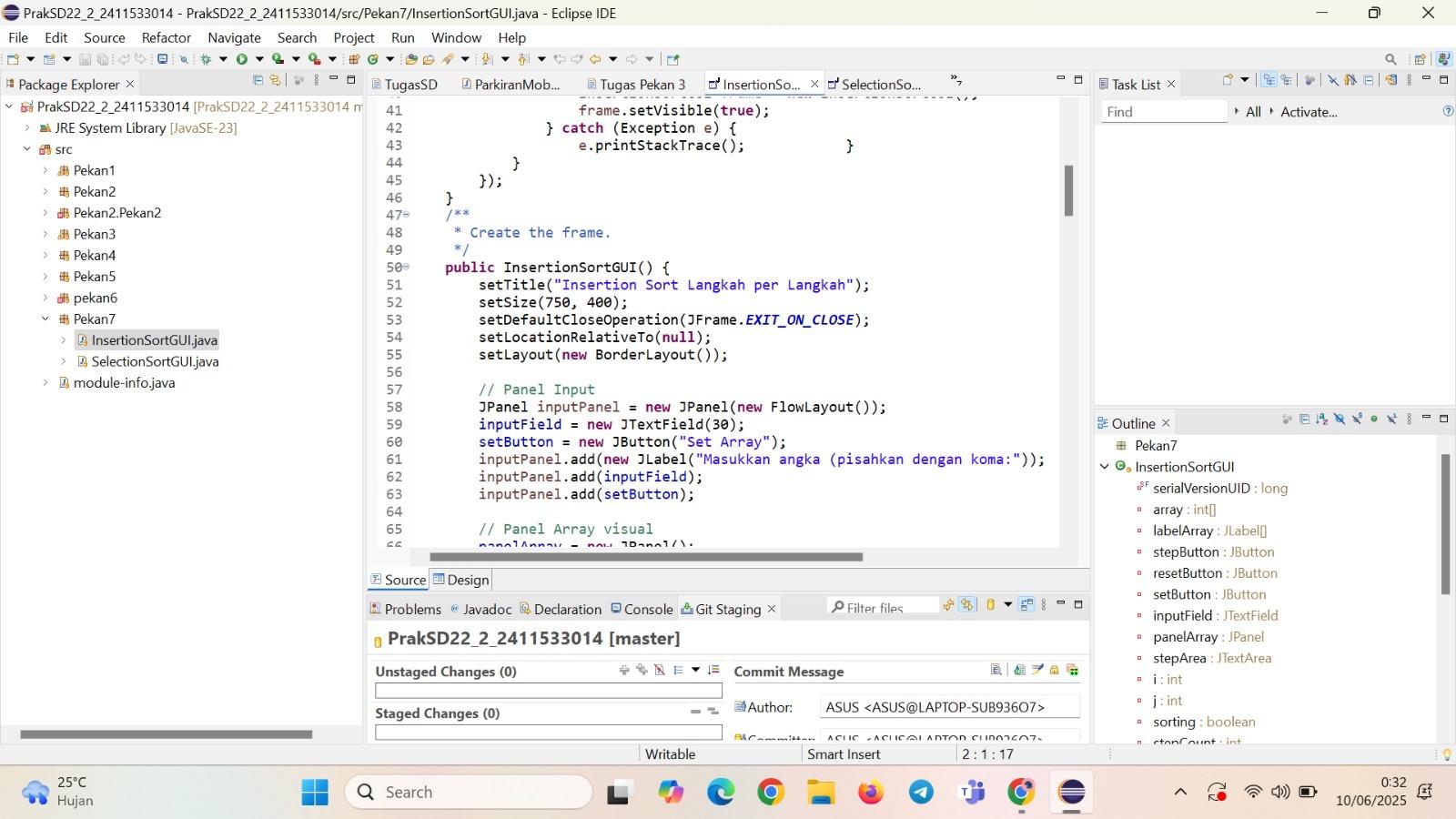
• sorting: Variabel boolean untuk menandakan apakah proses pengurutan sedang berlangsung.

4. Method main()

Method main() adalah titik masuk program. Ia melakukan hal berikut:

• EventQueue.invokeLater(...): Menjalankan kode dalam thread Event Dispatch Thread (EDT), yang penting untuk keamanan thread dalam Swing.

• Membuat instance InsertionSortGUI dan mengatur visibilitasnya menjadi true untuk menampilkan jendela.



1. Menangani Pengecualian

Blok try-catch di awal kode menangani kemungkinan pengecualian yang mungkin terjadi selama eksekusi program. e.printStackTrace() mencetak stack trace dari pengecualian ke konsol, yang membantu dalam debugging.

2. Membuat Frame Utama (InsertionSortGUI())

Konstruktor InsertionSortGUI() melakukan hal berikut:

• setTitle(...): Menetapkan judul jendela.

• setSize(...): Menetapkan ukuran jendela.

• setDefaultCloseOperation(...): Menetapkan perilaku ketika jendela ditutup (dalam hal ini, EXIT\_ON\_CLOSE).

• setLayout(...): Menetapkan tata letak jendela menggunakan BorderLayout.

3. Panel Input (inputPanel)

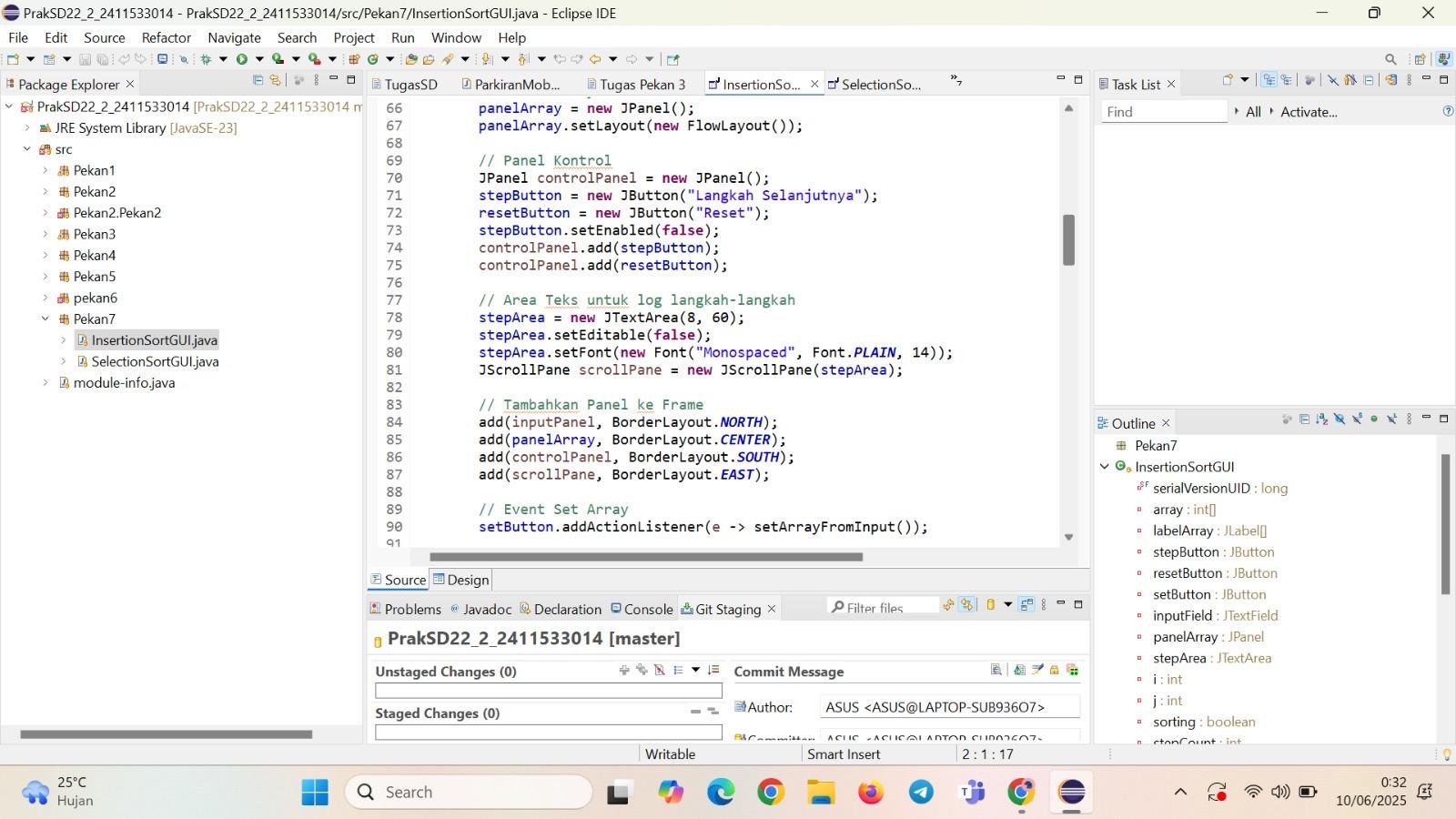
Bagian ini membuat panel input (inputPanel) yang berisi:

• inputField: Sebuah JTextField untuk memasukkan angka.

• setButton: Sebuah JButton untuk memproses input angka.

• Label petunjuk: Sebuah JLabel yang menginstruksikan pengguna untuk memasukkan angka, dipisahkan dengan koma.

Elemen-elemen ini ditambahkan ke inputPanel yang menggunakan FlowLayout.



1. Inisialisasi Komponen GUI

Kode ini memulai dengan membuat beberapa komponen GUI, termasuk:

• panelArray: Sebuah JPanel untuk menampilkan array angka. FlowLayout digunakan untuk mengatur tata letak elemen di dalam panel ini.

• controlPanel: Sebuah JPanel untuk tombol kontrol, yaitu stepButton (untuk menjalankan satu langkah pengurutan) dan resetButton (untuk mereset).

• stepArea: Sebuah JTextArea untuk menampilkan langkah-langkah proses pengurutan. setEditable(false) mencegah pengguna mengubah teks ini. Font "Monospaced" digunakan untuk menjaga agar angka-angka tetap sejajar.

• scrollPane: Sebuah JScrollPane untuk membuat stepArea dapat digulir jika teksnya terlalu panjang.

2. Menambahkan Komponen ke Frame

Bagian ini menambahkan komponen-komponen GUI yang telah dibuat ke dalam frame utama aplikasi menggunakan BorderLayout:

• inputPanel (tidak ditampilkan dalam kode yang diberikan, tetapi diasumsikan ada): Diposisikan di bagian utara (BorderLayout.NORTH).

• panelArray: Diposisikan di tengah (BorderLayout.CENTER).

• controlPanel: Diposisikan di bagian selatan (BorderLayout.SOUTH).

• scrollPane: Diposisikan di bagian timur (BorderLayout.EAST).

3. Event Handling: setArrayFromInput()

setButton.addActionListener(e -> setArrayFromInput()); menetapkan action listener ke tombol setButton. Ketika tombol ini ditekan, fungsi setArrayFromInput() (yang tidak ditampilkan dalam kode yang diberikan) akan dipanggil. Fungsi ini kemungkinan besar bertanggung jawab untuk:

• Mengambil input angka dari inputField (yang juga tidak ditampilkan).

• Mengubah input string menjadi array integer.

• Menginisialisasi labelArray (array label untuk menampilkan angka-angka dalam panelArray)

• Membersihkan stepArea

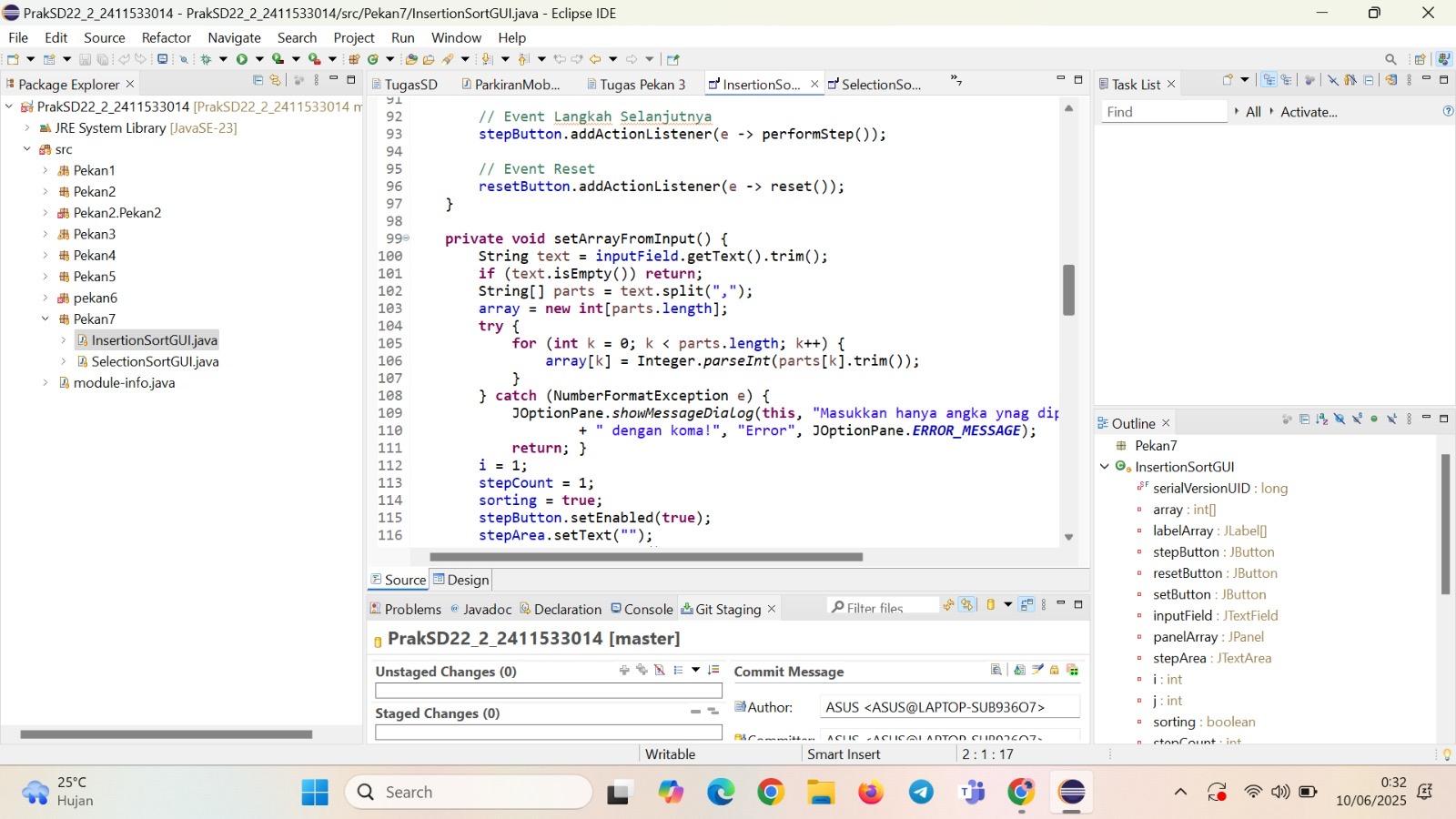
• Mengaktifkan stepButton

4. Event Handling: Tombol Langkah Selanjutnya (stepButton) dan Reset (resetButton)

Kode untuk menangani event stepButton dan resetButton tidak ditampilkan. Namun, fungsi-fungsi yang dipanggil oleh tombol-tombol ini akan bertanggung jawab untuk:

• stepButton: Melakukan satu langkah algoritma Insertion Sort, memperbarui tampilan panelArray dan stepArea untuk menunjukkan perubahan.

• resetButton: Mengatur ulang semua komponen ke keadaan awal, menonaktifkan stepButton, membersihkan stepArea, dan mempersiapkan untuk input array baru.



Method setArrayFromInput()

Method ini bertanggung jawab untuk mengambil input angka dari inputField, memvalidasinya, dan kemudian mempersiapkan array untuk pengurutan. Berikut langkah-langkahnya:

1. Mendapatkan Input: Ia mengambil teks dari inputField, menghapus spasi di awal dan akhir (trim()), dan menyimpannya dalam variabel text.

2. Memeriksa Input Kosong: Jika text kosong, method tersebut langsung kembali.

3. Membagi Input: Input dibagi menjadi bagian-bagian individual berdasarkan koma (split(",")). Hasilnya disimpan dalam array string parts.

4. Mengubah ke Integer: Loop for mencoba mengubah setiap bagian string dalam parts menjadi integer menggunakan Integer.parseInt(). trim() digunakan untuk menghapus spasi tambahan. Hasilnya disimpan dalam array integer array.

5. Penanganan Pengecualian: Blok try-catch menangani kemungkinan NumberFormatException jika pengguna memasukkan input yang bukan angka. Jika terjadi pengecualian, pesan kesalahan ditampilkan, dan method tersebut kembali.

6. Inisialisasi: Setelah array integer array berhasil dibuat, variabel-variabel yang relevan diinisialisasi:

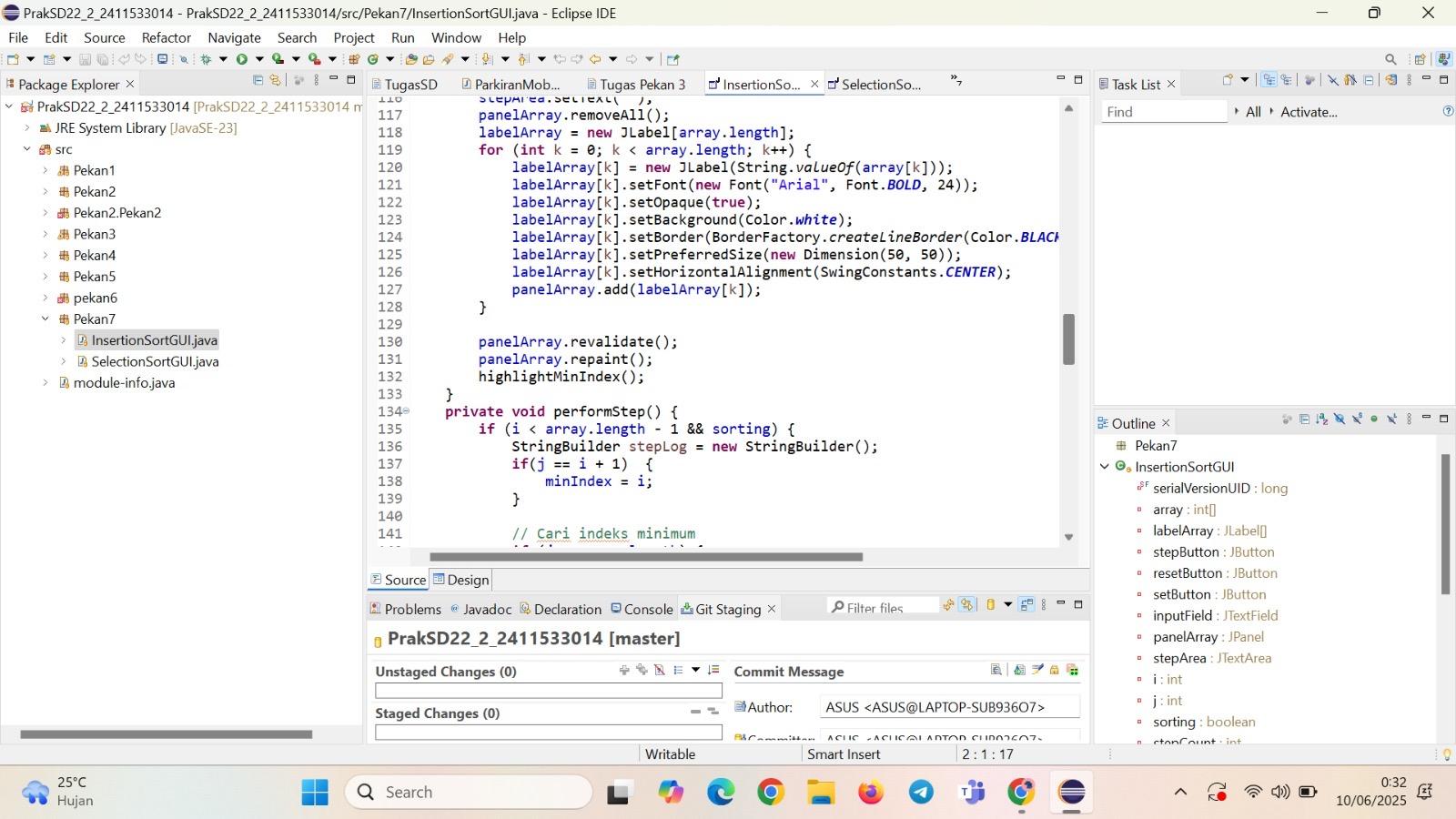
◦ i = 1; j = 1;: Variabel pencacah untuk algoritma pengurutan.

◦ stepCount = 1;: Pencacah untuk langkah pengurutan.

◦ sorting = true;: Menandakan bahwa proses pengurutan akan dimulai.

◦ stepButton.setEnabled(true);: Mengaktifkan tombol "langkah selanjutnya".

◦ stepArea.setText("");: Membersihkan area langkah-langkah pengurutan.



1. Inisialisasi Label Array (labelArray)

Bagian ini menghapus semua label sebelumnya dari panelArray, lalu membuat label baru untuk setiap elemen dalam array array. Setiap label dikonfigurasi dengan:

• Text: Nilai dari elemen array yang bersangkutan.

• Font: "Arial", Bold, ukuran 24.

• Latar Belakang: Putih (Color.white).

• Border: Border hitam (Color.BLACK).

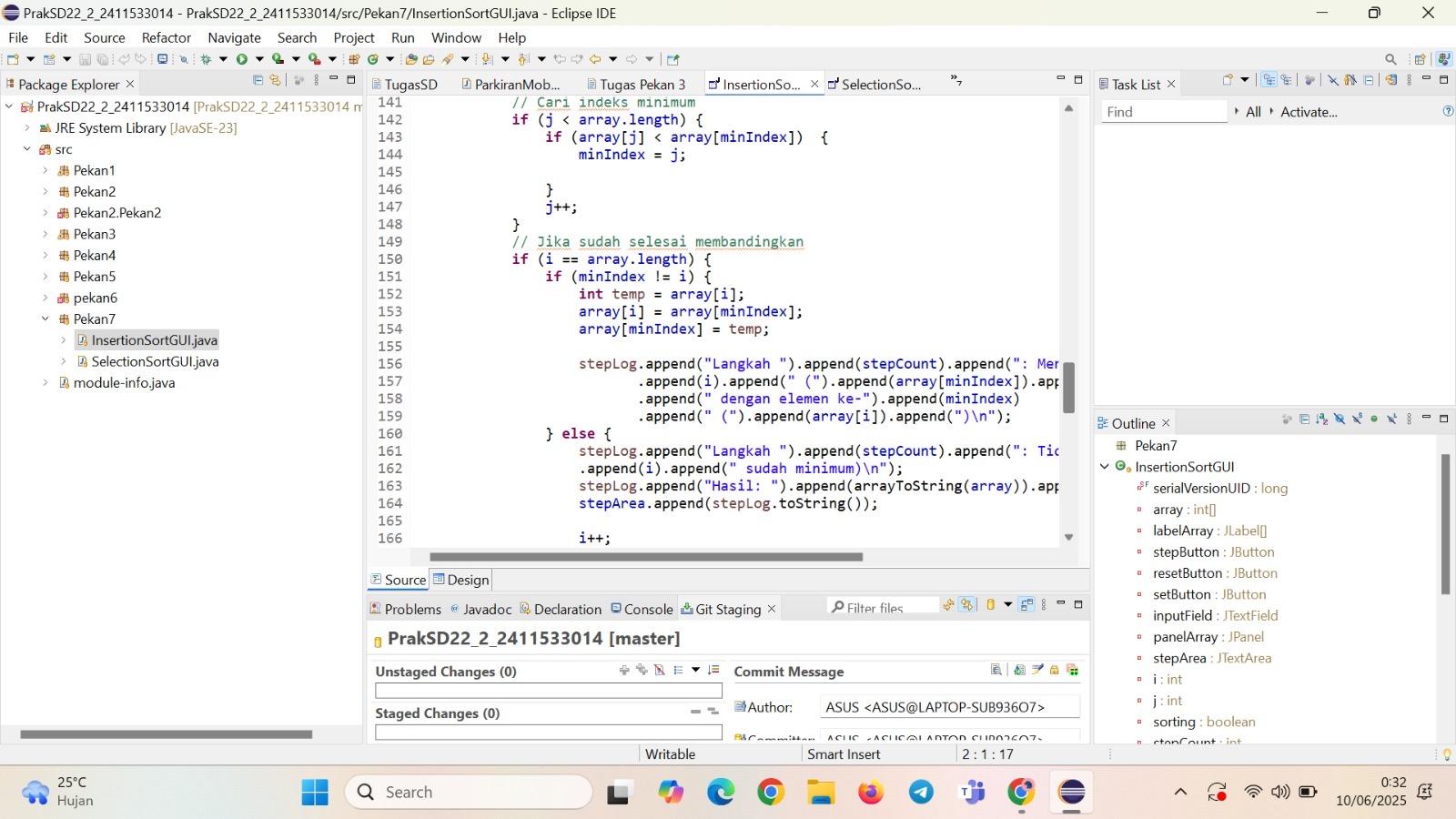
• Ukuran Preferred: 50 x 58 piksel.

• Perataan Horizontal: Di tengah (SwingConstants.CENTER).

Label-label ini kemudian ditambahkan ke panelArray. panelArray.revalidate() dan panelArray.repaint() memastikan bahwa perubahan tata letak dan tampilan dirender dengan benar.

2. Method performStep() (bagian)

Method ini menangani eksekusi satu langkah algoritma pengurutan. Bagian kode yang ditampilkan hanya menangani inisialisasi stepLog, sebuah StringBuilder untuk menyimpan log langkah pengurutan. Logika utama algoritma pengurutan (yang meliputi penentuan minIndex) berada di luar bagian kode yang ditampilkan di sini.



1. Mencari Indeks Minimum

Loop for pertama mencari indeks minimum (minIndex) dalam sub-array yang belum diurutkan (array[i]...array[array.length-1]). Perhatikan bahwa minIndex diinisialisasi dengan i pada setiap iterasi loop luar.

2. Menukar Elemen

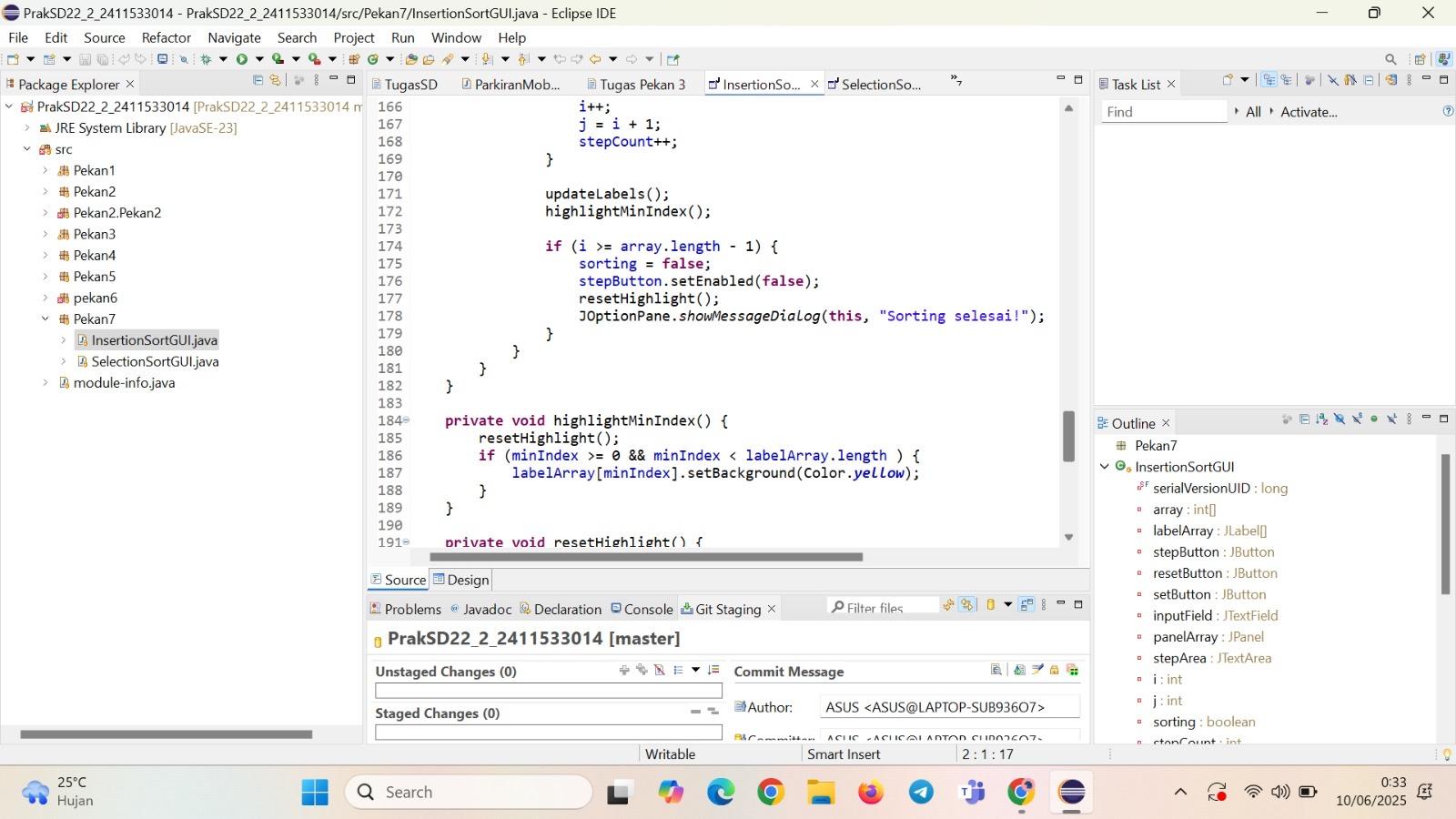
Jika indeks minimum (minIndex) ditemukan berbeda dari i (berarti ada elemen yang lebih kecil perlu dipindahkan ke posisi yang tepat), maka elemen pada indeks i dan minIndex ditukar. Variabel sementara temp digunakan untuk menyimpan nilai elemen sementara.

3. Menambahkan Log ke stepArea

Setelah pertukaran (atau jika minIndex sama dengan i, berarti tidak ada pertukaran), log langkah-langkah pengurutan ditambahkan ke stepArea. stepLog digunakan sebagai StringBuilder untuk efisiensi dan kemudian ditambahkan ke stepArea. Log ini menjelaskan langkah yang dilakukan (pertukaran atau tidak ada pertukaran) dan kondisi array saat ini.

4. Iterasi dan Kondisi Pengakhiran

Loop for luar (for (int i = 0; i < array.length - 1; i++)) melakukan iterasi melalui array. Pengulangan berhenti ketika i mencapai array.length - 1, menandakan bahwa array telah diurutkan sepenuhnya.



1. Logika Pengurutan (bagian)

Potongan kode ini menunjukkan sebagian dari implementasi algoritma Insertion Sort. Detail lengkap algoritma tidak ditampilkan, tetapi kita dapat melihat beberapa langkah kunci:

• i++, j++, stepCount++: Menginkremen variabel pencacah untuk melacak kemajuan pengurutan.

• updateLabels(): Memanggil method untuk memperbarui tampilan visual array di GUI.

• Kondisi if (i >= array.length - 1): Memeriksa apakah pengurutan telah selesai. Jika ya, proses pengurutan dihentikan, stepButton dinonaktifkan, dan pesan dialog ditampilkan.

2. Method highlightMinIndex()

Method ini kemungkinan besar digunakan untuk menyoroti indeks minimum dalam array selama proses pengurutan. Ia mengubah latar belakang label pada indeks minIndex menjadi kuning (Color.yellow). Ini akan memberikan visualisasi yang lebih baik tentang langkah-langkah pengurutan.

3. Method resetHighlight() (tidak ditampilkan)

Method ini, yang dideklarasikan tetapi tidak ditampilkan, mungkin digunakan untuk mereset penyorotan dari label-label di labelArray. Ini akan membersihkan penyorotan setelah setiap langkah pengurutan.

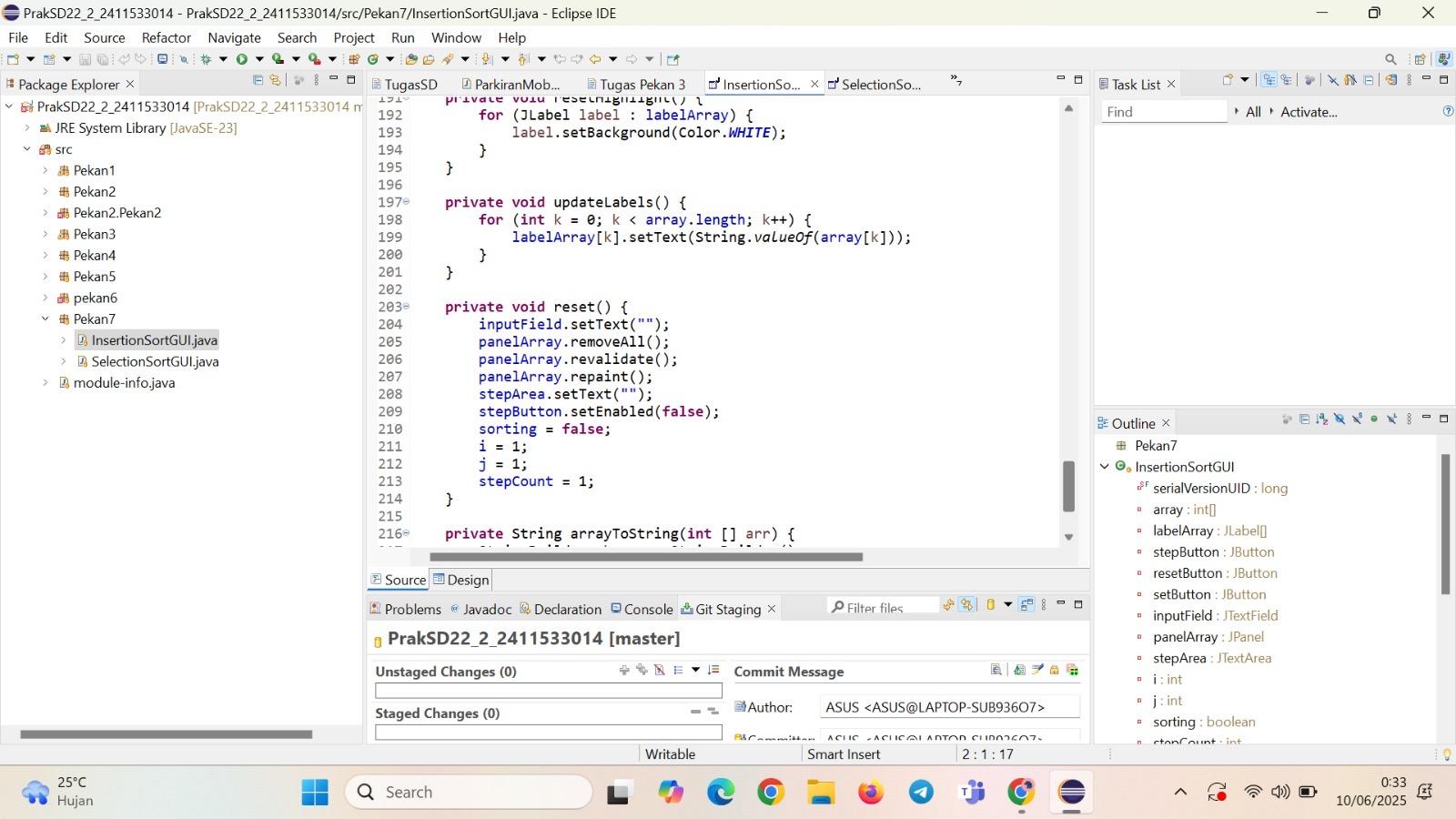
4. Penanganan Akhir Pengurutan

Blok if memeriksa apakah pengurutan telah selesai (i >= array.length - 1). Jika selesai, maka:

• sorting = false;: Variabel sorting diset menjadi false.

• stepButton.setEnabled(false);: Tombol "langkah selanjutnya" dinonaktifkan.

• JOptionPane.showMessageDialog(...): Sebuah pesan dialog ditampilkan untuk memberitahu pengguna bahwa pengurutan telah selesai.



Method updateLabels()

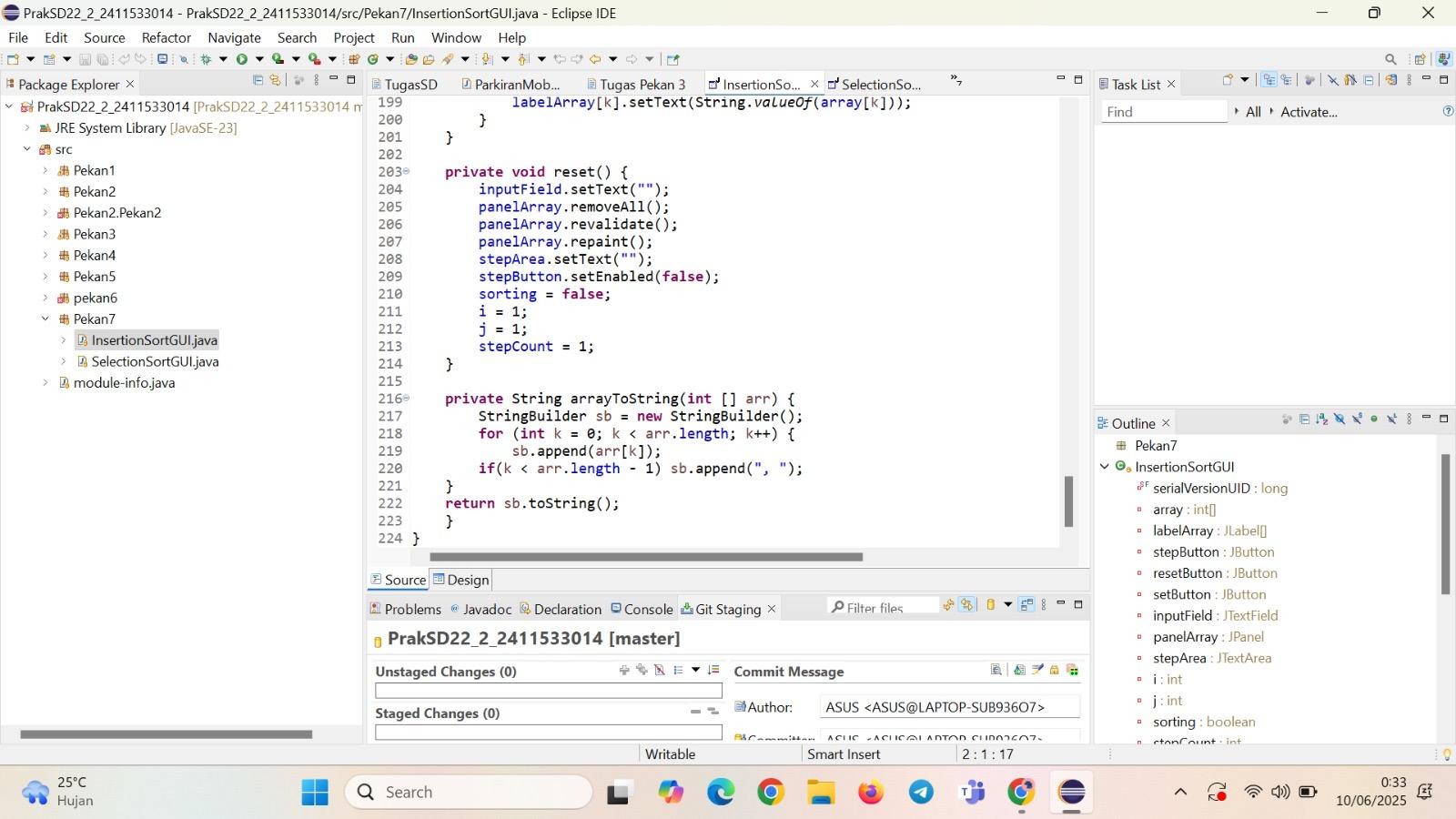
Method updateLabels() memperbarui teks dari setiap label dalam labelArray dengan nilai yang sesuai dari array array. Ia melakukan iterasi melalui array array dan menetapkan teks setiap label dengan nilai dari elemen array yang sesuai. Kode ini penting karena ia bertanggung jawab untuk menampilkan data array yang diurutkan secara visual di GUI. Dengan demikian, setiap perubahan pada array akan tercermin dalam tampilan GUI melalui pembaruan label-label ini.

Ringkasan:

• updateLabels() memperbarui tampilan visual array dengan menetapkan teks label sesuai dengan nilai array yang diurutkan. Ini memastikan sinkronisasi antara data internal dan representasi visualnya.

• reset() mereset aplikasi ke kondisi awal, termasuk membersihkan field input, panel array, dan area langkah-langkah, serta menonaktifkan tombol langkah selanjutnya.

Kedua method ini bekerja bersama untuk menyediakan fungsionalitas inti dari aplikasi GUI Insertion Sort. updateLabels() menangani pembaruan tampilan visual, sementara reset() mengelola penyetelan ulang aplikasi untuk siklus pengurutan yang baru.



1. Method reset()

Method reset() digunakan untuk mereset keadaan aplikasi ke keadaan awal. Ia melakukan hal berikut:

• inputField.setText("");: Membersihkan inputField.

• panelArray.removeAll();: Menghapus semua komponen dari panelArray.

• panelArray.revalidate();: Memvalidasi ulang tata letak panelArray.

• panelArray.repaint();: Menggambar ulang panelArray.

• stepArea.setText("");: Membersihkan stepArea.

• stepButton.setEnabled(false);: Menonaktifkan stepButton.

• sorting = false;: Mengatur variabel sorting menjadi false.

• i = 1; j = 1; stepCount = 1;: Mereset variabel-variabel pencacah yang digunakan dalam algoritma Insertion Sort.

Method ini penting untuk mempersiapkan aplikasi agar dapat menerima input dan menjalankan algoritma pengurutan baru.

2. Method arrayToString()

Method arrayToString() mengubah array integer menjadi string yang diformat dengan baik. Ia menggunakan StringBuilder untuk efisiensi dan melakukan hal berikut:

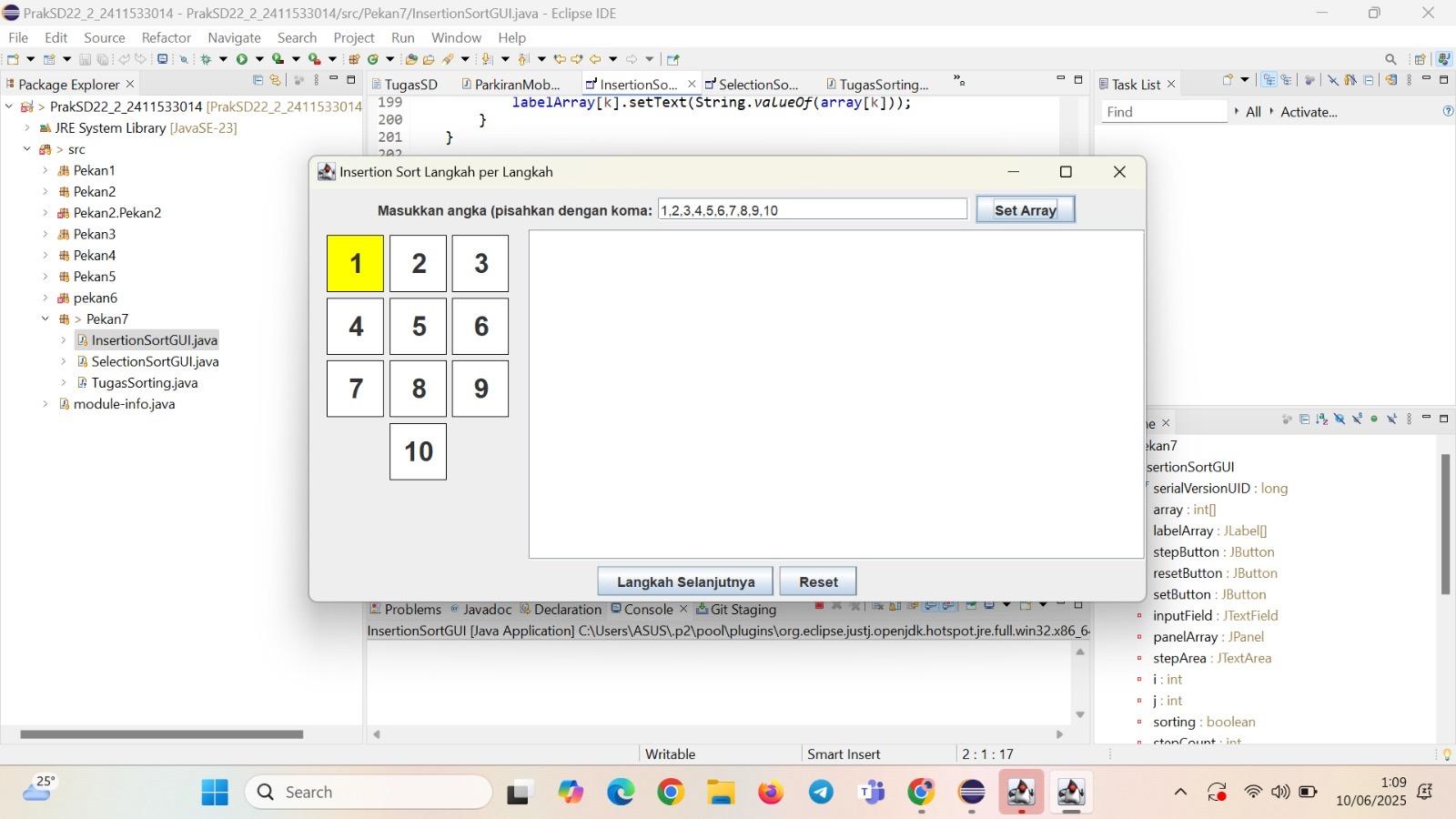
• Membuat instance StringBuilder.

• Iterasi melalui array arr.

• Menambahkan setiap elemen array ke StringBuilder, dipisahkan dengan koma.

• Mengembalikan string yang telah dibangun.

Method ini digunakan untuk menampilkan isi array dalam GUI, kemungkinan besar di stepArea atau di panel yang menampilkan visualisasi array.



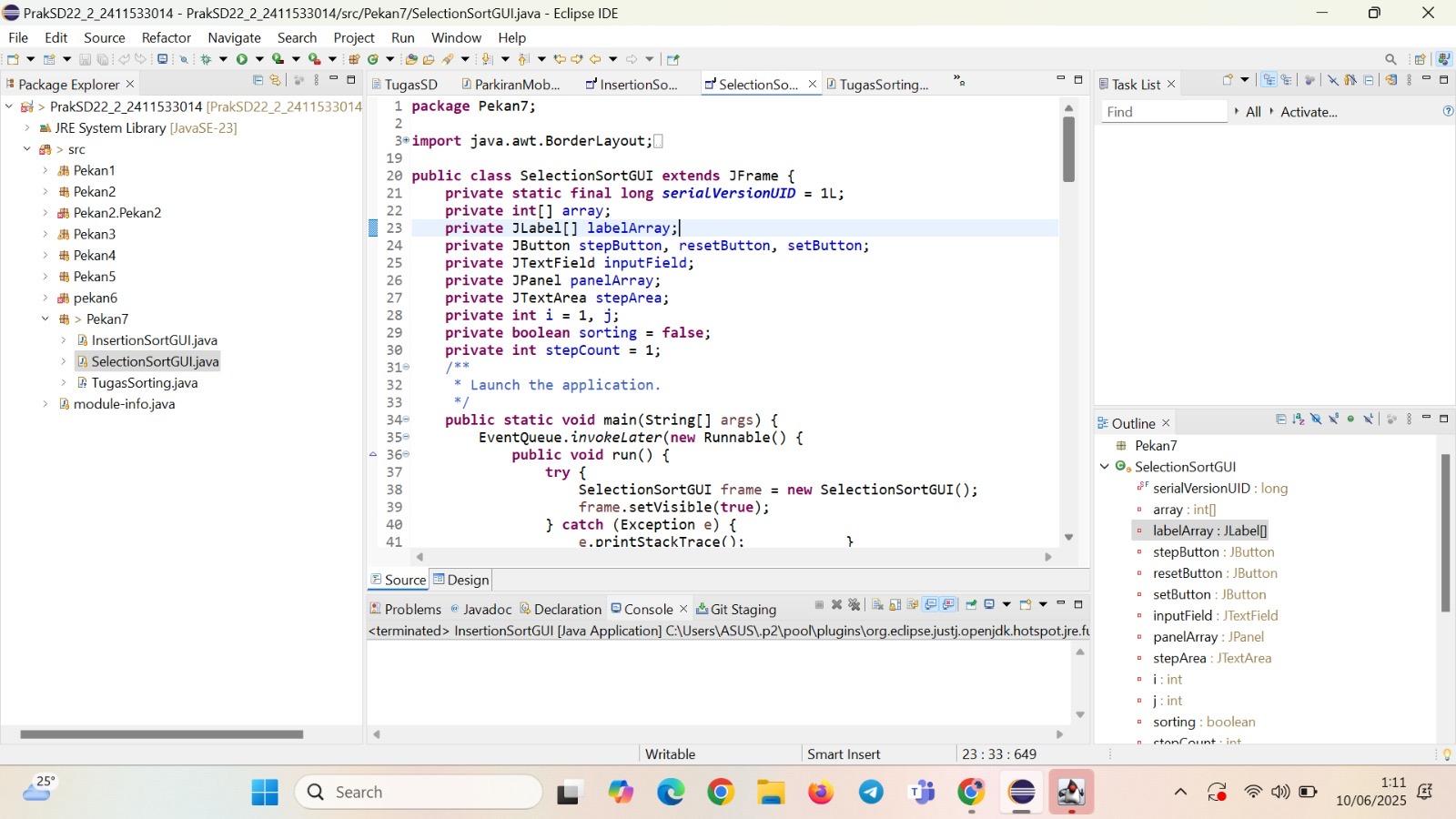
Gambar tersebut menunjukkan antarmuka pengguna grafis (GUI) dari aplikasi Insertion Sort setelah pengguna memasukkan angka 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 dan menekan tombol "Set Array". Berikut observasi dari gambar:

• Input Field: Field input menunjukkan angka-angka yang telah dimasukkan, dipisahkan oleh koma.

• Panel Array Visual: Angka-angka tersebut ditampilkan secara visual sebagai label-label dalam grid. Label dengan angka 1 memiliki latar belakang kuning, menunjukkan bahwa angka ini mungkin sedang disorot sebagai bagian dari algoritma pengurutan (kemungkinan besar ini adalah langkah awal dari algoritma).

• Tombol Kontrol: Tombol "Langkah Selanjutnya" dan "Reset" tersedia. Tombol "Langkah Selanjutnya" akan menjalankan satu langkah algoritma Insertion Sort, sedangkan tombol "Reset" akan mengatur ulang aplikasi ke keadaan awal.

• Area Langkah: Area di bagian bawah masih kosong. Area ini akan menunjukkan langkah-langkah yang dilakukan selama proses pengurutan.

* SelectionSortGUI

1. Struktur Kode:

Kode ini mirip dengan kode Insertion Sort sebelumnya, tetapi mengimplementasikan algoritma Selection Sort. Struktur dasarnya sama:

• Package dan Import: Menentukan paket dan mengimpor kelas yang dibutuhkan (khususnya BorderLayout).

• Kelas SelectionSortGUI: Mewarisi dari JFrame, membentuk kerangka utama aplikasi GUI. Menggunakan variabel-variabel instance yang serupa dengan implementasi Insertion Sort, seperti array, labelArray, stepButton, resetButton, inputField, stepArea, dll.

• Method main(): Menjalankan aplikasi dengan membuat instance dari SelectionSortGUI dan menampilkannya.

• Method-method lain: Method-method tambahan untuk menangani input, mereset, memperbarui label, dan menjalankan algoritma Selection Sort langkah demi langkah. Method-method ini akan serupa fungsinya dengan yang ada di implementasi Insertion Sort, tetapi logika pengurutannya berbeda.

2. Perbedaan dengan Insertion Sort:

Perbedaan utama terletak pada logika pengurutan di dalam method yang menjalankan algoritma Selection Sort. Algoritma Selection Sort bekerja dengan cara menemukan elemen minimum dalam array yang belum diurutkan, dan kemudian menukarnya dengan elemen pertama dalam sub-array yang belum diurutkan. Proses ini diulang sampai seluruh array terurut.

3. Bagian-bagian Kode yang Mungkin Ada (tidak ditampilkan):

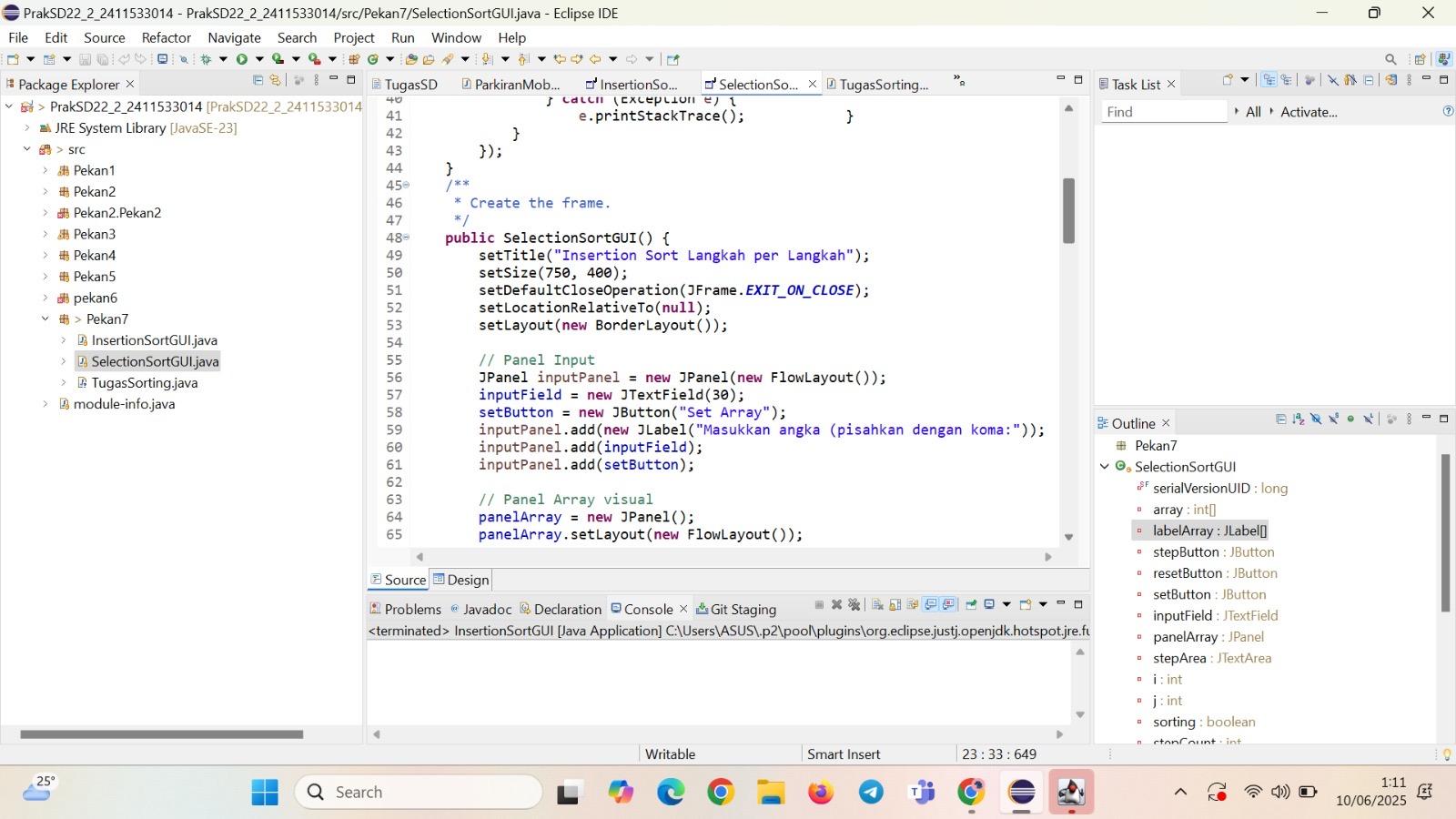
Kode yang Anda berikan tidak lengkap. Untuk menjalankan aplikasi dan memahami sepenuhnya, bagian-bagian kode berikut ini mungkin diperlukan:

• Method untuk memproses input: Method untuk mengambil input dari inputField, memvalidasi input, dan membuat array integer.

• Method untuk menjalankan Selection Sort: Method utama yang mengimplementasikan logika algoritma Selection Sort langkah demi langkah. Method ini akan memperbarui labelArray dan stepArea setelah setiap langkah.

• Method untuk mereset aplikasi: Method untuk mereset semua variabel dan komponen GUI ke keadaan awal.

• Method untuk memperbarui label: Method untuk memperbarui tampilan visual array di GUI.



1. Penanganan Pengecualian:

Sama seperti kode sebelumnya, blok try-catch di awal menangani potensi pengecualian selama pembuatan dan inisialisasi frame.

2. Pembuatan Frame (SelectionSortGUI):

Konstruktor SelectionSortGUI membuat frame utama aplikasi. Perhatikan kesamaan dengan konstruktor di kode Insertion Sort:

• setTitle(...): Menetapkan judul jendela.

• setSize(...): Menetapkan ukuran jendela.

• setDefaultCloseOperation(...): Menetapkan perilaku saat jendela ditutup.

• setLayout(...): Menetapkan tata letak jendela menggunakan BorderLayout.

3. Panel Input (inputPanel):

Kode ini menciptakan panel input yang identik dengan kode Insertion Sort:

• inputField: Sebuah JTextField untuk memasukkan angka.

• setButton: Sebuah JButton untuk memproses input.

• Label instruksi: Sebuah JLabel yang menjelaskan cara memasukkan angka.

Elemen-elemen ini ditambahkan ke inputPanel yang menggunakan FlowLayout.

4. Panel Array Visual (panelArray):

Kode ini juga membuat panel untuk visualisasi array, tetapi menggunakan FlowLayout. Ini berbeda dengan implementasi Insertion Sort sebelumnya, yang mungkin menggunakan tata letak yang berbeda untuk visualisasi array.

Perbedaan Utama dan Bagian yang Hilang:

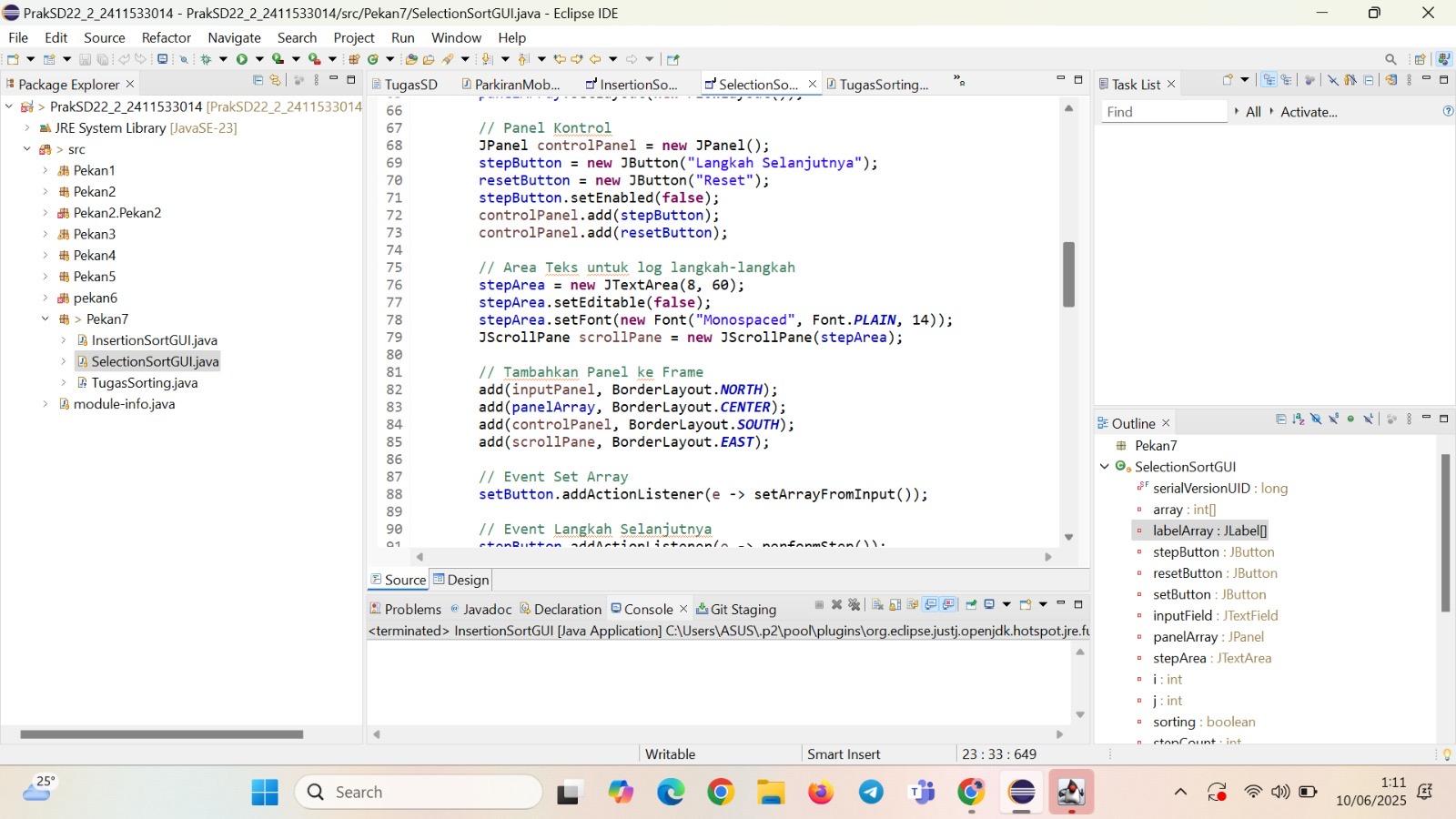
Perbedaan utama antara kode ini dan kode Insertion Sort terletak pada implementasi algoritma pengurutan itu sendiri. Kode yang ditampilkan hanya mencakup pembuatan frame dan panel. Bagian-bagian kode berikut ini tidak ditampilkan, tetapi sangat penting untuk menyelesaikan aplikasi:

• Method untuk memproses input angka dari inputField dan membuat array array.

• Method untuk menjalankan algoritma Selection Sort langkah demi langkah. Method ini akan mencakup logika utama algoritma Selection Sort, yaitu menemukan elemen minimum dalam sub-array yang belum terurut, menukarnya dengan elemen pertama dalam sub-array tersebut, dan mengulang proses ini hingga seluruh array terurut.

• Method untuk memperbarui labelArray dan stepArea setelah setiap langkah pengurutan.

• Method untuk menangani event dari tombol "Langkah Selanjutnya" dan "Reset".



1. Panel Kontrol (controlPanel):

Kode ini membuat panel untuk tombol kontrol:

• stepButton: Tombol "Langkah Selanjutnya" yang diinisialisasi dinonaktifkan (setEnabled(false)).

• resetButton: Tombol "Reset".

Kedua tombol ini kemudian ditambahkan ke controlPanel, yang kemungkinan menggunakan FlowLayout.

2. Area Teks untuk Log (stepArea):

Kode ini membuat JTextArea untuk menampilkan log langkah-langkah algoritma. stepArea diinisialisasi dinonaktifkan (setEditable(false)) dan menggunakan font Monospaced untuk tampilan yang rapi. stepArea dibungkus dalam JScrollPane untuk memungkinkan scrolling jika teks log melebihi ukuran area tampilan.

3. Menambahkan Panel ke Frame:

Kode ini menambahkan panel-panel (inputPanel, panelArray, controlPanel) dan JScrollPane yang berisi stepArea ke dalam frame utama menggunakan BorderLayout. Posisi masing-masing panel ditentukan:

• inputPanel: BorderLayout.NORTH (di atas).

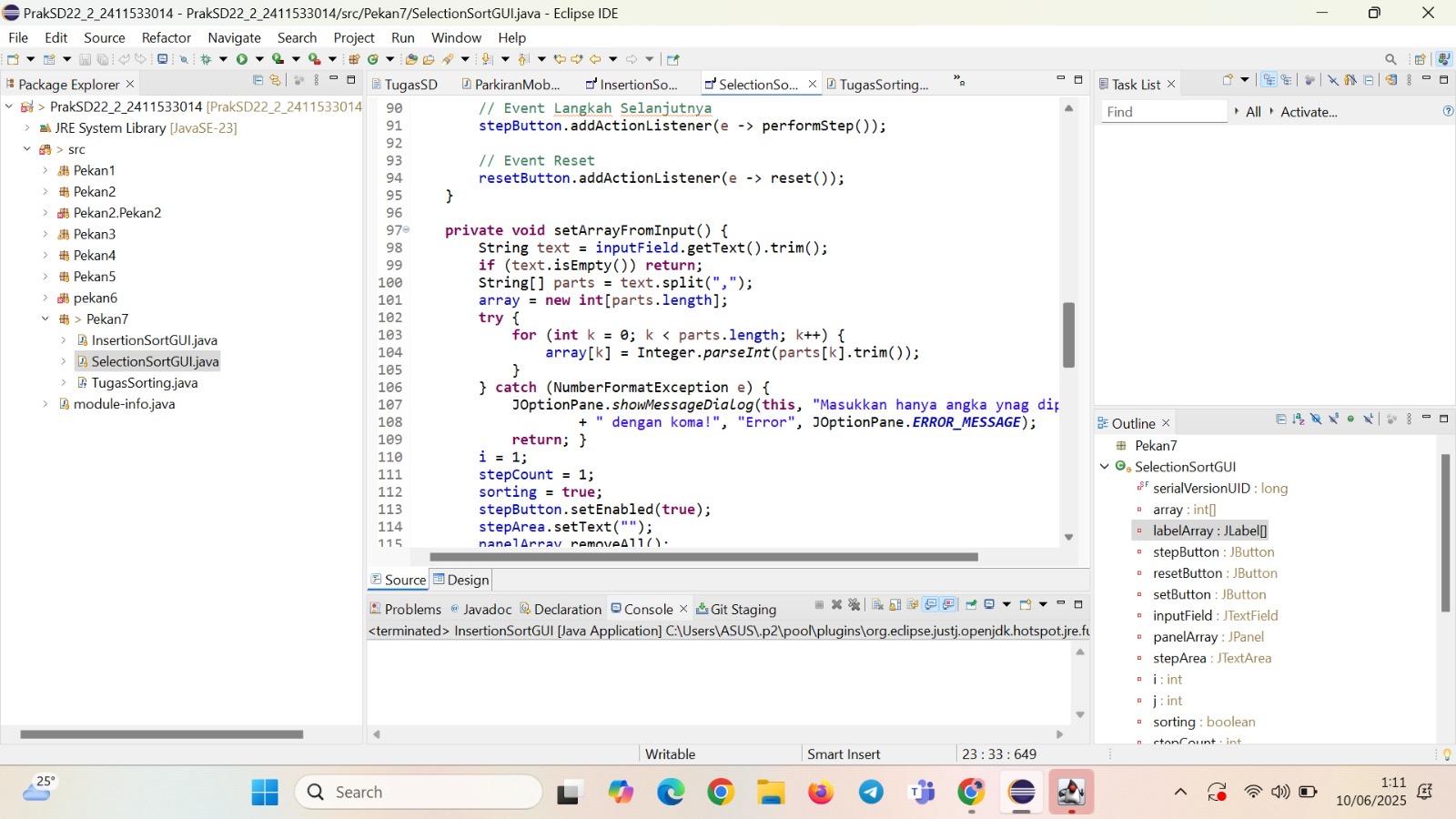
• panelArray: BorderLayout.CENTER (di tengah).

• controlPanel: BorderLayout.SOUTH (di bawah).

• JScrollPane (yang berisi stepArea): BorderLayout.EAST (di samping kanan).

4. Event Listener:

Kode ini menambahkan ActionListener ke setButton. Ketika tombol "Set Array" ditekan, method setArrayFromInput() akan dipanggil untuk memproses input angka dari pengguna. Ini mirip dengan fungsi yang ada di kode Insertion Sort.



Method setArrayFromInput():

Method ini bertanggung jawab untuk memproses input angka dari pengguna dan mempersiapkan array untuk pengurutan. Langkah-langkahnya adalah:

1. Mendapatkan Input: Membaca teks dari inputField, menghapus spasi tambahan (trim()).

2. Memeriksa Input Kosong: Mengembalikan jika inputField kosong.

3. Membagi Input: Membagi teks menjadi array string (parts) berdasarkan koma.

4. Konversi ke Integer: Mengubah setiap elemen dalam parts menjadi integer dalam array array. Blok try-catch menangani potensi kesalahan (NumberFormatException) jika input bukan angka. Jika terjadi kesalahan, pesan error ditampilkan dan method kembali.

5. Inisialisasi Variabel: Setelah konversi berhasil, variabel-variabel diinisialisasi:

◦ i = 1; j = 1;: Pencacah untuk algoritma Selection Sort.

◦ stepCount = 1;: Pencacah langkah.

◦ sorting = true;: Menandakan bahwa proses pengurutan aktif.

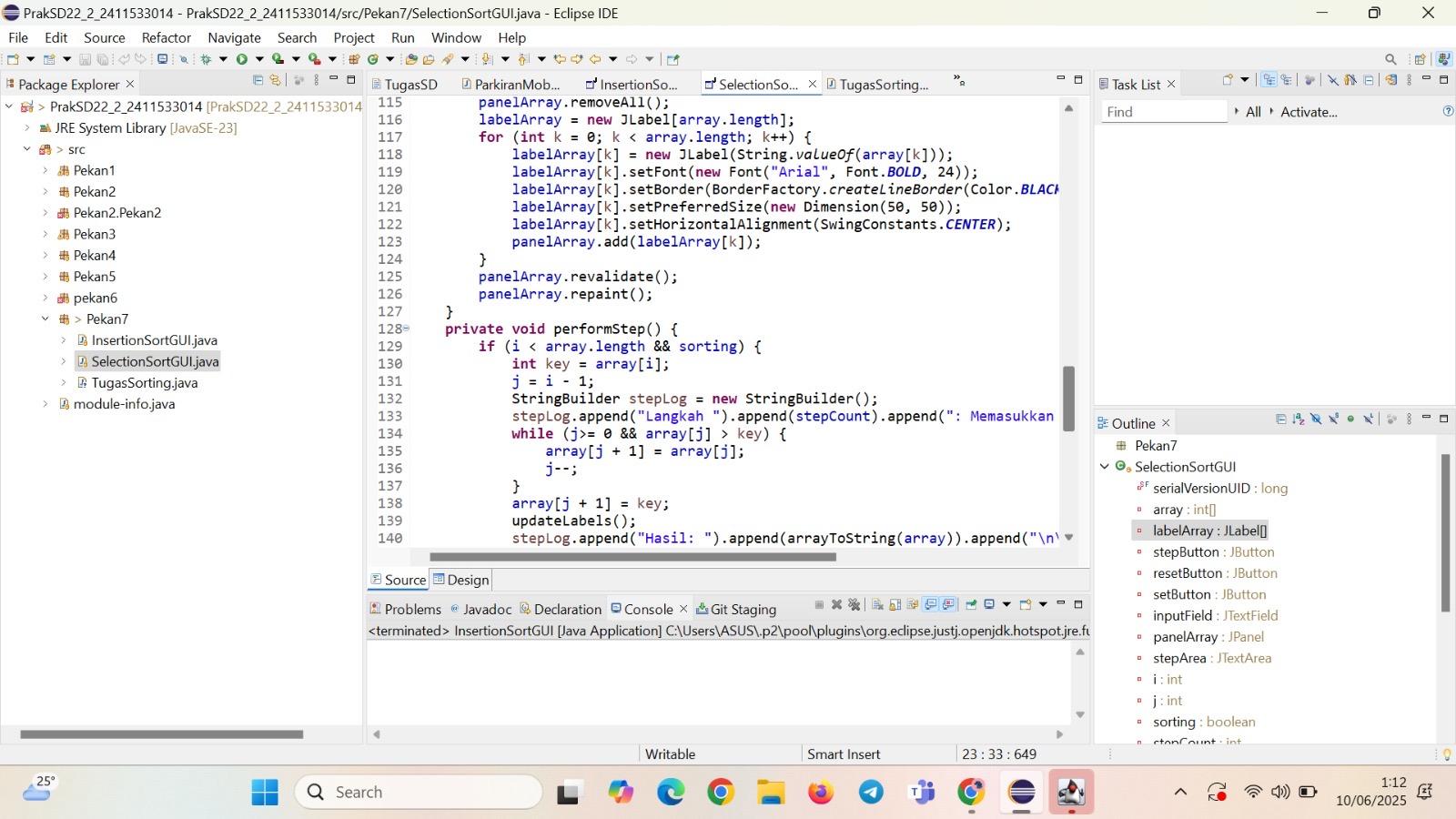
◦ stepButton.setEnabled(true);: Mengaktifkan tombol "Langkah Selanjutnya".

◦ stepArea.setText("");: Membersihkan area log.

◦ panelArray.removeAll();: Menghapus label-label array sebelumnya.

Perbedaan dengan Insertion Sort:

Tidak ada perbedaan fungsional yang signifikan antara setArrayFromInput() dalam aplikasi Selection Sort dan Insertion Sort. Kedua method melakukan tugas yang sama: mengambil input angka, memvalidasinya, dan mempersiapkan array untuk pengurutan. Perbedaannya hanya terletak pada algoritma pengurutan yang digunakan setelah method ini dipanggil. Algoritma Selection Sort akan digunakan dalam aplikasi ini, sedangkan Insertion Sort digunakan pada aplikasi sebelumnya.



Method performStep():

Method ini menjalankan satu langkah algoritma Selection Sort. Berikut uraiannya:

1. Kondisi Awal: Method memeriksa apakah proses pengurutan (sorting) masih aktif dan jika i belum mencapai akhir array.

2. Mencari Minimum: Variabel key menyimpan indeks minimum dari sub-array yang belum terurut (array[i]...array[array.length-1]). Loop while mencari elemen terkecil dalam sub-array ini.

3. Menukar Elemen: Jika key tidak sama dengan i (artinya ditemukan elemen yang lebih kecil), elemen pada indeks i dan key ditukar.

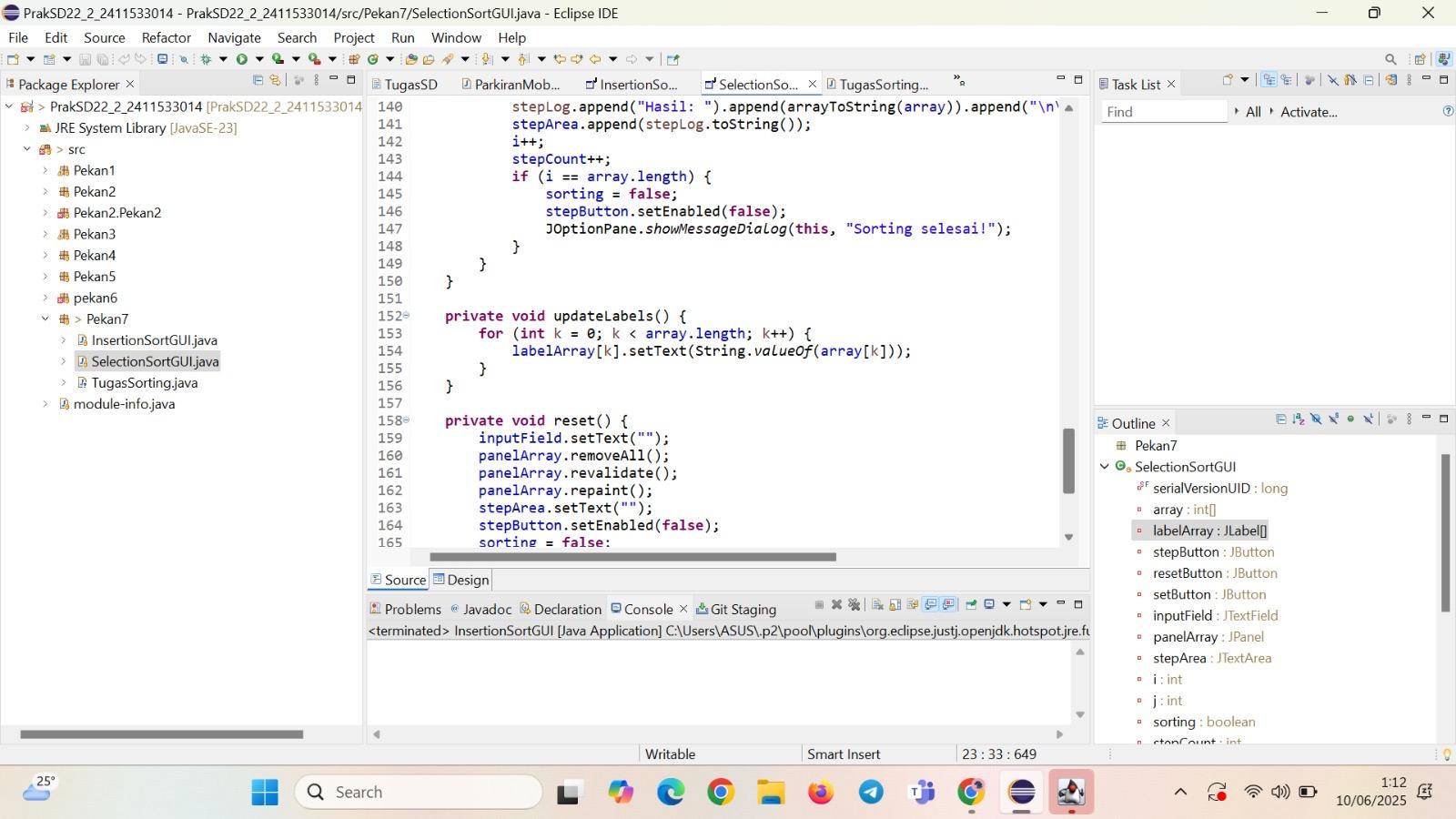
4. Memperbarui Tampilan: Setelah pertukaran (atau jika tidak ada pertukaran), updateLabels() dipanggil untuk memperbarui tampilan visual array di GUI.

5. Menambahkan Log: Informasi langkah yang dilakukan ditambahkan ke stepLog.

6. Iterasi: i diinkremen untuk melanjutkan ke langkah selanjutnya dalam algoritma Selection Sort.

Perbandingan dengan Insertion Sort:

Method performStep() untuk Selection Sort berbeda secara signifikan dengan method yang sama di aplikasi Insertion Sort. Logika pencarian minimum dan penukaran elemen mencerminkan algoritma Selection Sort. Algoritma Insertion Sort memiliki pendekatan yang berbeda dalam mengurutkan elemen.



1. Method updateLabels():

Method ini bertanggung jawab untuk memperbarui teks dari setiap label di labelArray agar sesuai dengan nilai yang terkini dalam array array. Loop for melakukan iterasi melalui labelArray, dan setText() mengatur teks setiap label menjadi nilai integer yang sesuai dari array array.

2. Method reset():

Method ini mereset aplikasi ke keadaan awal. Berikut langkah-langkahnya:

• inputField.setText("");: Membersihkan field input.

• panelArray.removeAll();: Menghapus semua label dari panel array.

• panelArray.revalidate();: Memperbarui tata letak panel array.

• panelArray.repaint();: Menggambar ulang panel array.

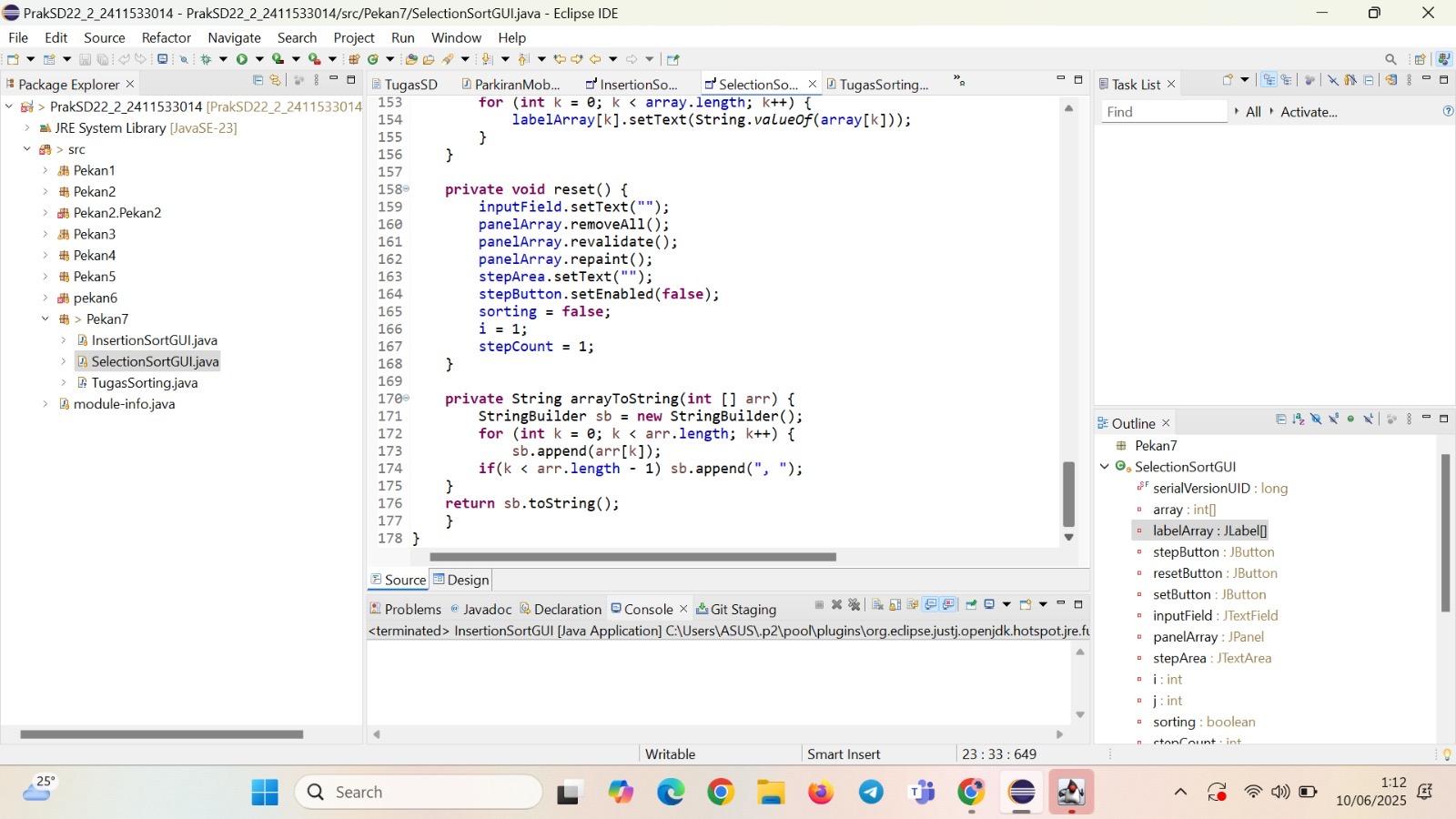
• stepArea.setText("");: Membersihkan area log.

• stepButton.setEnabled(false);: Menonaktifkan tombol "Langkah Selanjutnya".

• sorting = false;: Mengatur variabel sorting menjadi false untuk menghentikan proses pengurutan.

Perbandingan dengan Insertion Sort:

Kedua method ini memiliki fungsi yang serupa dalam aplikasi Insertion Sort GUI. updateLabels() selalu sama, karena tugasnya adalah memperbarui label visual array. reset() juga melakukan tugas yang sama, yaitu mengembalikan aplikasi ke keadaan awal.



Method arrayToString():

Method ini menerima array integer (arr) sebagai input dan mengembalikan representasi string dari array tersebut. Berikut detailnya:

1. Inisialisasi StringBuilder: Sebuah StringBuilder bernama sb diinisialisasi. StringBuilder lebih efisien daripada String untuk manipulasi string yang berulang.

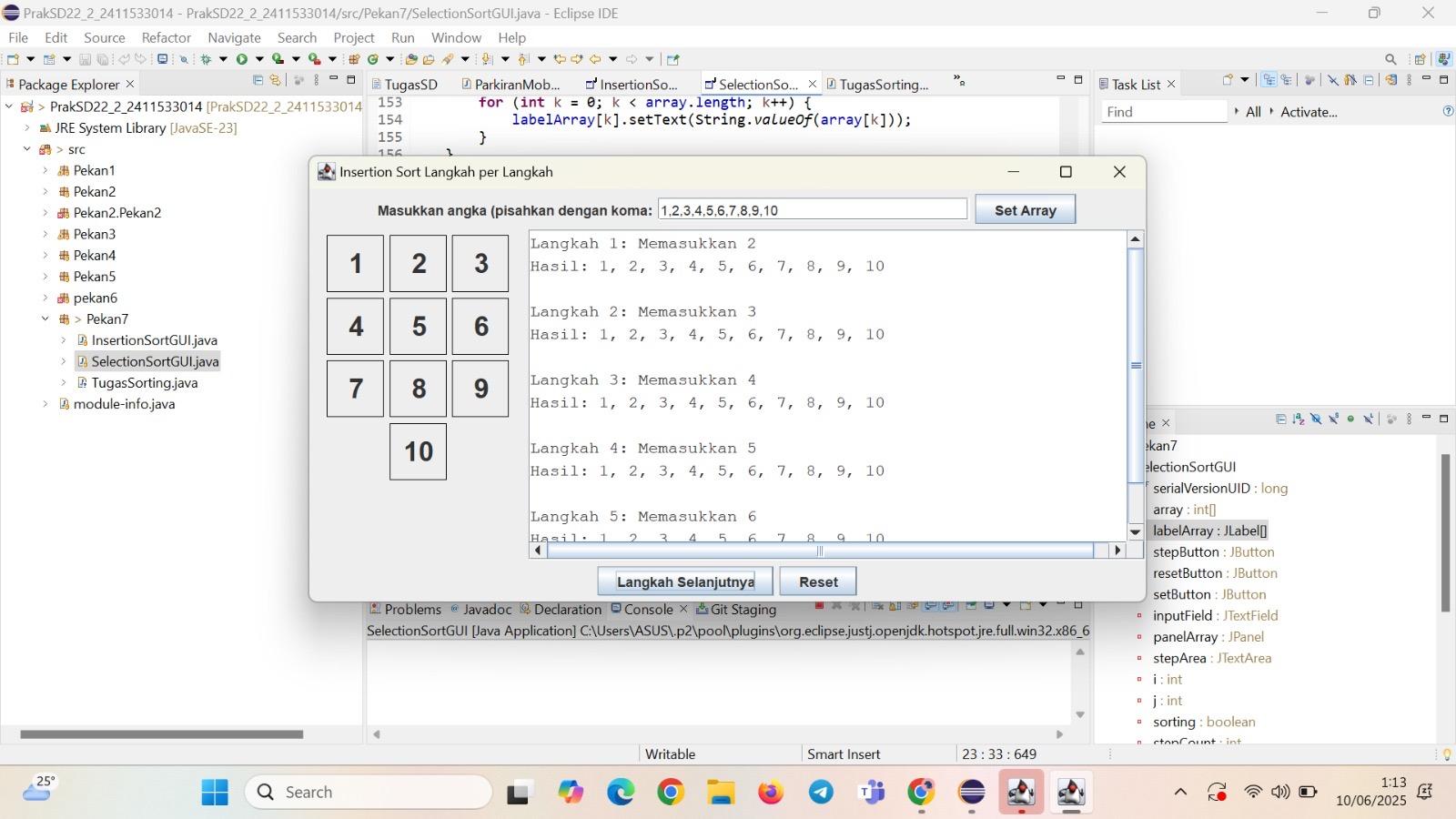
2. Iterasi: Loop for melakukan iterasi melalui array arr.

3. Menambahkan Elemen: Dalam setiap iterasi, elemen array (arr[k]) ditambahkan ke sb. Sebuah koma (,) ditambahkan sebagai pemisah antar elemen, kecuali untuk elemen terakhir. Kondisi if (k < arr.length - 1) memastikan bahwa koma tidak ditambahkan setelah elemen terakhir.

4. Mengembalikan String: Setelah loop selesai, sb.toString() mengembalikan representasi string dari array.

Perbandingan dengan Insertion Sort:

Method arrayToString() ini identik dengan method yang digunakan dalam aplikasi Insertion Sort. Method ini bersifat umum dan dapat digunakan untuk mengubah array integer menjadi string dalam berbagai konteks, bukan hanya untuk visualisasi Selection Sort.



Gambar diatas menunjukkan aplikasi GUI Selection Sort yang sedang berjalan, beberapa langkah pengurutan telah dilakukan. Berikut analisisnya:

• Visualisasi Array: Angka-angka dalam panel array menunjukkan keadaan array setelah beberapa langkah Selection Sort. Angka-angka belum sepenuhnya terurut.

• Log Langkah-langkah: Area teks di sebelah kanan menampilkan log dari langkah-langkah yang telah dilakukan. Setiap langkah menunjukkan array sebelum dan sesudah pertukaran elemen. Penulisan log dengan format "Langkah n: Memasukkan x, Hasil: [array]" menunjukkan bahwa algoritma Selection Sort divisualisasikan langkah demi langkah dengan menampilkan array setelah setiap iterasi.

• Tombol Kontrol: Tombol "Langkah Selanjutnya" dan "Reset" masih aktif. Pengguna dapat melanjutkan pengurutan dengan mengklik "Langkah Selanjutnya" atau memulai lagi dengan mengklik "Reset".

1. KESIMPULAN

Implementasi algoritma pengurutan Insertion Sort dan Selection Sort dengan antarmuka pengguna grafis (GUI) menggunakan Java Swing telah berhasil divisualisasikan. Kedua aplikasi memiliki struktur yang serupa, dengan komponen GUI yang terdiri dari field input untuk angka, panel visualisasi array, panel kontrol (tombol "Langkah Selanjutnya" dan "Reset"), dan area teks untuk menampilkan log langkah-langkah pengurutan. Meskipun struktur GUI serupa, logika pengurutan di dalam kedua aplikasi sangat berbeda, mencerminkan perbedaan mendasar antara algoritma Insertion Sort dan Selection Sort.

Method-method utama dalam kedua aplikasi, seperti setArrayFromInput(), performStep(), updateLabels(), dan reset(), menunjukkan bagaimana input diproses, pengurutan dilakukan langkah demi langkah, tampilan array diperbarui secara dinamis, dan aplikasi dapat direset. Method arrayToString() digunakan untuk mengubah array integer menjadi representasi string untuk tampilan di GUI dan log. Penanganan pengecualian juga diimplementasikan untuk menangani input yang tidak valid dari pengguna.

Visualisasi langkah demi langkah dari algoritma pengurutan memungkinkan pengguna untuk memahami bagaimana setiap algoritma bekerja. Perbedaan dalam kecepatan dan efisiensi antara Insertion Sort dan Selection Sort dapat diamati secara langsung melalui visualisasi ini. Aplikasi GUI ini menyediakan alat yang efektif untuk pembelajaran dan pemahaman yang lebih mendalam tentang algoritma pengurutan.