**LAPORAN PRAKTIKUM 8**

**STRUKTUR DATA**

**A logo of a university

AI-generated content may be incorrect.**

Dosen Pengampu:  
Dr. Wahyudi, S.T, M.Kom.

Disusun Oleh:  
Muhammad Luthfi Kautsar Rizata – 2311532020

**DEPARTEMEN INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**UNIVERSITAS ANDALAS**

**2025**

**A screen shot of a computer program

AI-generated content may be incorrect. 1. Class BubbleSortGUI**

Program ini merupakan aplikasi Java berbasis GUI dengan class utamanya denngan nama BubbleSortGUI yang diturunkan dari JFrame, sehingga memiliki kemampuan untuk menampilkan jendela aplikasi. Di bagian awal, terdapat main() method yang menjalankan objek GUI dalam event-dispatching thread menggunakan EventQueue.invokeLater. Pendekatan ini merupakan best practice di Java Swing untuk memastikan komponen GUI berjalan aman dan responsif. Di dalam konstruktor BubbleSortGUI, berbagai komponen antarmuka diatur menggunakan layout seperti BorderLayout, FlowLayout, dan JPanel, yang kemudian dikombinasikan untuk membentuk struktur tampilan aplikasi. Selain itu, beberapa variabel penting seperti array data (array), label tampilan (labelArray), dan indeks untuk perulangan bubble sort (i, j) juga diinisialisasi di bagian awal class. Seluruh komponen visual dan variabel inilah yang menjadi dasar dari logika dan tampilan aplikasi bubble sort ini.

**A computer screen shot of text

AI-generated content may be incorrect.**

Bagian interface GUI dalam program dirancang dengan menggunakan kombinasi beberapa layout manager untuk mengatur komponen agar lebih rapi dan mudah digunakan. Di bagian atas jendela (`BorderLayout.NORTH`), terdapat `inputPanel` yang berisi label instruksi, sebuah `JTextField` untuk memasukkan angka-angka array, serta tombol “Set Array” yang berfungsi untuk memproses input tersebut. Di bagian tengah jendela (`BorderLayout.CENTER`), ada `panelArray` yang menampilkan elemen-elemen array dalam bentuk kotak visual (`JLabel`) dengan batas dan warna latar tertentu, mempermudah user memahami proses sorting secara visual. Bagian bawah jendela (`BorderLayout.SOUTH`) menampilkan dua tombol kontrol: “Langkah Selanjutnya” untuk menjalankan proses bubble sort secara bertahap, dan “Reset” untuk mengembalikan semua tampilan dan data ke kondisi awal. Terakhir, di sisi kanan jendela (`BorderLayout.EAST`), terdapat `stepArea`, yaitu `JTextArea` di dalam `JScrollPane` yang berfungsi sebagai area log, mencatat setiap langkah yang dilakukan selama proses sorting. Komponen-komponen ini disusun dengan baik agar dapat berinteraksi dengan program secara intuitif dan melihat perkembangan proses sorting dengan jelas.

A screen shot of a computer program

AI-generated content may be incorrect.A screen shot of a computer program

AI-generated content may be incorrect.Fungsi `setArrayFromInput` bertanggung jawab untuk membaca input dari , memprosesnya menjadi array integer, lalu menampilkannya secara visual di panel utama aplikasi. Proses dimulai dengan mengambil teks dari `inputField`, lalu memotong string tersebut berdasarkan koma untuk memisahkan setiap nilai. Setiap bagian kemudian dikonversi menjadi integer dan disimpan ke dalam array. Jika terdapat karakter non-angka, program akan menampilkan pesan error menggunakan `JOptionPane`, dan proses dihentikan agar tidak terjadi kesalahan saat parsing.

Setelah array berhasil dibentuk, nilai-nilai indeks `i`, `j`, dan `stepCount` diinisialisasi ulang sebagai persiapan untuk proses sorting. Tombol langkah (`stepButton`) diaktifkan, area log dikosongkan, dan panel array dibersihkan dari elemen sebelumnya. Selanjutnya, setiap elemen array ditampilkan sebagai label (`JLabel`) di dalam `panelArray`, lengkap dengan border, ukuran tetap, dan warna latar putih. Tampilan array ini dibuat agar mudah dibaca dan memudahkan mengikuti proses sorting secara visual. Terakhir, elemen pertama yang akan dibandingkan langsung diberi highlight warna sebagai penanda bahwa sorting siap dimulai. Fungsi ini menjadi pintu masuk utama interaksi sebelum algoritma sorting dijalankan.

A screen shot of a computer program

AI-generated content may be incorrect. Selanjutnya ada fungsi `performStep` yang mana merupakan inti dari program karena di sinilah logika bubble sort dijalankan secara bertahap setiap kali menekan tombol “Langkah Selanjutnya.” Fungsi ini pertama-tama memeriksa apakah proses sorting masih berlangsung dan apakah indeks `j` belum melewati batas array. Jika tidak memenuhi kondisi, maka proses dianggap selesai dan akan mendapatkan pesan bahwa sorting telah selesai, serta tombol langkah akan dinonaktifkan.

Saat proses berjalan, fungsi akan mereset warna latar dari semua elemen agar tidak terjadi tumpang tindih highlight. Kemudian, dua elemen yang sedang dibandingkan diberi warna cyan sebagai penanda visual. Jika elemen ke-`j` lebih besar dari elemen ke-`j+1`, maka program melakukan pertukaran (swap) dan memberikan warna merah pada kedua elemen yang ditukar. Proses ini juga dicatat ke dalam `stepArea` dengan informasi detail: nomor langkah, elemen yang dibandingkan, apakah ada pertukaran atau tidak, serta hasil array setelah langkah tersebut. Jika tidak ada pertukaran, maka hanya dicatat bahwa tidak ada perubahan posisi.

Setelah log dicetak, array diperbarui secara visual menggunakan `updateLabels`, lalu indeks `j` digeser ke kanan. Jika `j` telah mencapai batas akhir dari iterasi luar bubble sort (`array.length - i - 1`), maka `j` direset ke nol dan `i` bertambah satu, menandakan bahwa satu “putaran” sudah selesai. Ketika seluruh array sudah dalam kondisi terurut (`i >= array.length - 1`), tombol langkah akan dinonaktifkan dan akan mendapat pesan bahwa sorting telah selesai. Fungsi ini membuat bisa melihat proses sorting dengan sangat jelas dan terstruktur, satu langkah dalam satu klik.

A screen shot of a computer code

AI-generated content may be incorrect. Lalu ada bagian yang mencakup tiga metode yang mengatur tampilan visual elemen array selama proses sorting, yaitu `highlightCompare`, `resetHighlights`, dan `updateLabels`. Fungsi `highlightCompare` digunakan untuk memberi penanda visual pada dua elemen yang sedang dibandingkan. Elemen pertama akan diberi warna kuning dan elemen kedua berwarna oranye, membuat langsung tahu bagian mana dari array yang sedang aktif. Ini sangat membantu untuk mengikuti proses bubble sort secara intuitif karena membedakan langkah aktif dari elemen lainnya yang tidak berubah.

Sebelum memberi warna baru, program akan memanggil `resetHighlights`, yaitu fungsi yang mengembalikan semua elemen array ke warna putih sebagai warna dasar. Ini mencegah warna dari langkah sebelumnya tertinggal di layar dan membuat tampilan lebih bersih dan konsisten setiap kali langkah baru dilakukan.

A computer screen shot of a program code

AI-generated content may be incorrect.

Sementara itu, `updateLabels` bertugas memperbarui isi teks dari setiap label agar sesuai dengan kondisi array terbaru setelah pertukaran (swap). Jadi meskipun nilai dalam array di memori sudah berubah, juga bisa melihat perubahan itu langsung di tampilan label. Dalam fungsi ini juga terdapat baris yang mencetak nama dan NIM ke konsol, sebagai bentuk identitas dan pengenal program buatan mahasiswa. Ketiga fungsi ini bekerja saling mendukung untuk memastikan bahwa proses sorting tidak hanya berjalan logis, tetapi juga tampil rapi dan mudah dipahami secara visual.

Fungsi reset bertugas mengembalikan seluruh elemen program ke kondisi awal, seolah-olah baru pertama kali membuka aplikasi. Ketika tombol “Reset” ditekan, kotak input (inputField) akan dikosongkan, semua label elemen array yang ada di panel tampilan (panelArray) akan dihapus, dan panel tersebut langsung direfresh agar tampilan benar-benar bersih. Selain itu, area teks log (stepArea) yang sebelumnya mencatat langkah-langkah sorting juga akan dihapus isinya.

Tombol langkah (stepButton) akan dinonaktifkan agar tidak bisa melanjutkan proses sorting yang sudah tidak relevan. Variabel-variabel penting seperti i, j, dan stepCount juga dikembalikan ke nilai awal (nol dan satu), serta status sorting diatur menjadi false untuk memastikan tidak ada proses sorting yang sedang berlangsung. Intinya, fungsi reset ini memberi kesempatan untuk mulai lagi dari awal dengan array baru, tanpa harus menutup dan membuka kembali aplikasi. Ini membuat pengalaman an jadi lebih fleksibel dan nyaman.

Fungsi arrayToString digunakan untuk mengubah isi array integer menjadi bentuk string yang lebih mudah dibaca dan ditampilkan di area log teks (stepArea). Fungsi ini bekerja dengan cara mengiterasi setiap elemen dalam array dan menambahkan setiap angka ke dalam objek StringBuilder yang mana merupakan cara efisien untuk membangun string di Java. Setelah setiap angka ditambahkan, fungsi akan menyisipkan tanda koma dan spasi, kecuali untuk elemen terakhir agar hasil akhirnya tidak diakhiri dengan koma. Hasil dari fungsi ini adalah representasi array dalam bentuk seperti “4, 2, 5, 1, 3”, yang kemudian bisa digunakan untuk menampilkan keadaan array setelah tiap langkah sorting. Ini membantu melihat bagaimana isi array berubah secara bertahap, sehingga lebih mudah memahami jalannya algoritma.

A computer screen shot of a program code

AI-generated content may be incorrect.**2. Class MergeSortGUI**

Pada bagian inisialisasi dan struktur umum program `MergeSortGUI`, dapat dilihat bahwa program ini dirancang sebagai aplikasi Java berbasis GUI menggunakan library Swing, dengan class utama `MergeSortGUI` yang merupakan turunan dari `JFrame`. Ini berarti program akan menghasilkan jendela aplikasi yang dapat ditampilkan secara visual dan interaktif. Eksekusi program dimulai dari method `main`, yang menggunakan `EventQueue.invokeLater` untuk memastikan bahwa pembuatan tampilan GUI dijalankan secara aman dalam thread yang sesuai. Di dalam method ini, objek `MergeSortGUI` dibuat dan ditampilkan ke layar.

Di dalam class, terdapat beberapa atribut penting yang menjadi dasar bagi fungsi program, seperti `array` yang menyimpan data bilangan bulat yang akan diurutkan, `labelArray` yang menampung elemen-elemen visual dari array tersebut dalam bentuk `JLabel`, dan komponen-komponen tampilan seperti `inputField`, `stepButton`, `resetButton`, `stepArea`, serta `panelArray`. Selain itu, terdapat juga struktur data `mergeQueue` berupa `Queue<int[]>` yang digunakan untuk menyimpan urutan langkah-langkah merge yang harus dilakukan selama proses sorting. Variabel tambahan seperti `temp`, `left`, `mid`, `right`, `isMerging`, `copying`, dan `k` juga digunakan untuk mengatur proses penggabungan data per langkah. Dengan inisialisasi seperti ini, struktur program sudah disiapkan untuk menerima input, memprosesnya dengan merge sort secara bertahap, dan menampilkannya dalam bentuk visual interaktif kepada .

A screen shot of a computer code

AI-generated content may be incorrect.

A screen shot of a computer code

AI-generated content may be incorrect. Fungsi `setArrayFromInput` dalam program ini berperan penting sebagai langkah awal untuk memulai proses merge sort. Fungsi ini dipanggil ketika menekan tombol "Set Array", dan tugas utamanya adalah membaca input berupa deretan angka yang dipisahkan oleh koma dari `inputField`, lalu mengubah input tersebut menjadi array bilangan bulat (`int[]`). Sebelum diproses, program akan memeriksa apakah input kosong; jika tidak, input akan dipisahkan dengan metode `split(",")` dan diubah menjadi integer melalui parsing. Apabila ditemukan karakter non-angka atau input tidak valid, maka akan muncul dialog error agar segera memperbaikinya. Setelah input berhasil diubah menjadi array, program akan menghapus semua data sebelumnya dari `mergeQueue`, lalu memanggil fungsi `generateMergeSteps` untuk menyusun urutan langkah-langkah merge sort yang akan dilakukan. Kemudian, status program diatur agar siap memulai proses sorting: variabel `isMerging` diatur menjadi true, `copying` menjadi false, dan `stepCount` direset ke angka satu. Komponen visual seperti `panelArray` dan `stepArea` juga dibersihkan untuk menghindari sisa data sebelumnya. Lalu, setiap elemen dari array diubah menjadi label (`JLabel`) yang ditampilkan dalam bentuk kotak visual di `panelArray`, lengkap dengan warna latar putih dan border hitam. Setelah semua label ditampilkan, teks “Langkah 1: Mulai Merge Sort” akan dituliskan ke area log sebagai penanda bahwa proses sudah siap untuk dijalankan secara bertahap. Fungsi ini menjadi jembatan utama dari input menuju visualisasi sorting yang interaktif.

Fungsi generateMergeSteps bertugas menyusun urutan langkah-langkah yang akan dilakukan dalam proses merge sort, dan hasilnya disimpan dalam struktur data antrian (mergeQueue). Fungsi ini bekerja secara rekursif, menerima dua parameter: l (left) dan r (right), yang menunjukkan indeks batas kiri dan kanan dari bagian array yang sedang diproses. Bila l masih lebih kecil dari r, artinya bagian array tersebut masih bisa dipecah lagi menjadi dua subbagian. Di sinilah merge sort bekerja: membagi-bagi array hingga menjadi bagian-bagian kecil yang terdiri dari satu elemen.

Dalam setiap pemanggilan, fungsi ini menghitung nilai tengah m dengan rumus (l + r) / 2, lalu memanggil dirinya sendiri untuk bagian kiri (l sampai m) dan bagian kanan (m + 1 sampai r). Setelah pembagian selesai, pasangan indeks [l, m, r] akan dimasukkan ke dalam mergeQueue. Ini adalah informasi yang akan digunakan nanti untuk menggabungkan kembali bagian kiri dan kanan yang telah diurutkan. Dengan menyusun semua langkah merge terlebih dahulu ke dalam queue, program dapat menjalankan proses penggabungan (merge) secara bertahap saat tombol “Langkah Selanjutnya” ditekan.

A screen shot of a computer program

AI-generated content may be incorrect. Fungsi performStep merupakan inti dari program ini karena di sinilah proses merge sort dijalankan secara bertahap sesuai dengan urutan langkah yang telah disusun sebelumnya dalam mergeQueue. Setiap kali menekan tombol “Langkah Selanjutnya”, fungsi ini dipanggil untuk menjalankan satu bagian dari proses sorting. Langkah pertama yang dilakukan adalah mereset warna dari semua label agar highlight dari langkah sebelumnya tidak tertinggal. Setelah itu, program memeriksa apakah proses merge masih aktif (isMerging) dan apakah ada data yang perlu disalin atau digabung.

Jika belum dalam kondisi menyalin (copying == false) dan mergeQueue masih memiliki langkah yang tersisa, maka program akan mengambil tiga nilai dari queue, yaitu batas kiri (left), tengah (mid), dan kanan (right). Nilai-nilai ini menandakan rentang bagian array yang akan digabung. Label-label yang berada dalam rentang ini akan diberi highlight kuning untuk menunjukkan bahwa mereka sedang diproses. Proses selanjutnya adalah membandingkan elemen dari bagian kiri dan kanan. Jika elemen kiri lebih kecil atau sama, maka nilai tersebut disalin ke array sementara (temp) dan diberi warna cyan. Hal yang sama juga berlaku untuk elemen kanan. Setelah setiap elemen disalin, fungsi langsung kembali agar satu langkah bisa ditampilkan dengan jelas dalam setiap klik.

Setelah semua elemen dari kedua sisi disalin ke temp, program masuk ke tahap copying, yang berarti saatnya menyalin isi dari array sementara ke array utama. Di tahap ini, elemen-elemen dari temp dimasukkan kembali ke posisi awal di array satu per satu, sambil mengubah tampilan label agar mencerminkan nilai terbaru dan memberi warna hijau sebagai penanda bahwa nilai tersebut telah berhasil “ditanamkan kembali” ke array utama. Setelah seluruh proses merge pada rentang tersebut selesai, log akan diperbarui dengan pesan bahwa langkah merge selesai. Jika antrian mergeQueue sudah kosong dan tidak ada lagi yang perlu disalin, maka proses merge dianggap selesai. Tombol langkah dinonaktifkan, diberi pesan bahwa sorting sudah selesai, dan hasil akhir array ditampilkan di area log.

A screen shot of a computer program

AI-generated content may be incorrect.

resetHighlights dan updateLabels, yang masing-masing berperan dalam memperbarui tampilan visual elemen array selama proses merge sort berlangsung. Fungsi resetHighlights bertugas menghapus semua highlight warna pada elemen array dengan cara mengembalikan latar belakang setiap label ke warna putih. Ini penting dilakukan sebelum memulai langkah baru, agar warna dari langkah sebelumnya tidak membingungkan . Fungsi ini cukup sederhana namun sangat esensial dalam menjaga tampilan aplikasi tetap bersih dan terfokus pada langkah aktif saat ini.

Sementara itu, updateLabels digunakan untuk memperbarui teks dari setiap elemen label agar sesuai dengan isi terbaru dari array. Fungsi ini dipanggil setelah terjadi perubahan pada nilai array, terutama setelah proses penyalinan dari array sementara (temp) kembali ke array utama

A computer screen shot of a program code

AI-generated content may be incorrect.

Fungsi reset dalam program ini berperan sebagai tombol penyegar yang mengembalikan seluruh komponen dan variabel ke kondisi awal, memungkinkan untuk memulai proses merge sort dari awal dengan input baru. Saat fungsi ini dipanggil—biasanya lewat klik tombol “Reset”—program akan langsung mengosongkan kolom input (inputField) dan menghapus semua label elemen array dari panel tampilan (panelArray). Panel tersebut kemudian diperbarui agar tampak bersih secara visual. Area log (stepArea) juga dikosongkan agar catatan langkah-langkah sebelumnya tidak muncul lagi.

Secara internal, fungsi ini juga menonaktifkan tombol “Langkah Selanjutnya” untuk mencegah menjalankan proses saat belum ada data baru. Antrian langkah (mergeQueue) akan dikosongkan agar tidak menyisakan sisa urutan dari proses sebelumnya. Selain itu, status boolean isMerging dan copying diatur menjadi false, dan penghitung langkah (stepCount) direset ke 1. Semua inisialisasi ulang ini bertujuan untuk memastikan bahwa tidak ada bagian program yang tertinggal dari proses sebelumnya. Dengan kata lain, fungsi reset menyediakan ruang kosong yang segar untuk bereksperimen kembali, membuat pengalaman an menjadi lebih fleksibel dan terkontrol.

A screen shot of a computer code

AI-generated content may be incorrect.Fungsi arrayToString dalam program ini digunakan untuk mengubah isi array integer menjadi format string yang rapi dan mudah dibaca, khususnya untuk ditampilkan di area log (stepArea). Fungsi ini bekerja dengan menggunakan StringBuilder, yang merupakan cara efisien dalam Java untuk menyusun string panjang dari potongan-potongan kecil. Melalui perulangan, setiap elemen array ditambahkan ke StringBuilder satu per satu. Jika elemen tersebut bukan elemen terakhir, maka setelah nilainya ditambahkan, akan diikuti dengan tanda koma dan spasi sebagai pemisah antar angka.

Hasil akhir dari fungsi ini adalah representasi array seperti "3, 7, 1, 4, 5", yang akan sangat membantu dalam memahami kondisi array pada setiap langkah proses sorting. Fungsi ini biasanya dipanggil saat ingin menampilkan hasil array setelah satu langkah merge selesai atau saat sorting benar-benar berakhir.

**3. Class QuickSortGUI**

**A computer screen shot of a program

AI-generated content may be incorrect.** program QuickSortGUI dirancang sebagai aplikasi berbasis GUI yang dibuat menggunakan Java Swing, dengan class utama bernama QuickSortGUI yang merupakan turunan dari JFrame. Ini artinya, program ini bisa ditampilkan dalam bentuk jendela grafis seperti aplikasi desktop pada umumnya. Eksekusi program dimulai dari method main, yang menjalankan GUI di dalam event-dispatching thread menggunakan EventQueue.invokeLater. Ini adalah cara yang aman untuk membuat dan menampilkan antarmuka di Java Swing tanpa mengalami konflik thread.

Di dalam class, terdapat beberapa variabel penting yang akan mendukung proses sorting. Variabel array menyimpan data integer yang akan diurutkan, sedangkan labelArray menyimpan representasi visualnya dalam bentuk label. Ada pula komponen tombol seperti stepButton, resetButton, dan setButton, serta input field (inputField) untuk menerima input angka dari . Untuk tampilan, digunakan panel utama (panelArray) yang menampung kotak-kotak angka, serta stepArea, yaitu kotak teks di sebelah kanan yang mencatat langkah-langkah sorting secara tertulis.

Selain komponen GUI, terdapat juga Stack<int[]> stack yang menyimpan rentang-rentang indeks untuk diurutkan, mencerminkan cara kerja Quick Sort yang menggunakan pendekatan divide and conquer. Variabel tambahan seperti pivotIndex, i, j, partitioning, dan sorting digunakan untuk mengatur alur logika sorting di setiap langkah.

A screen shot of a computer program

AI-generated content may be incorrect.Antarmuka GUI dalam program QuickSortGUI dibangun dengan pendekatan yang rapi dan user-friendly menggunakan layout standar dari Java Swing. Di bagian atas jendela (BorderLayout.NORTH), terdapat sebuah inputPanel yang berisi label instruksi, kolom teks (inputField) untuk memasukkan deretan angka yang ingin diurutkan, serta tombol “Set Array” yang akan memproses input tersebut. Komponen ini ditata menggunakan FlowLayout agar susunannya horizontal dan responsif terhadap panjang input.

Bagian tengah jendela (BorderLayout.CENTER) diisi oleh panelArray, sebuah panel yang akan menampilkan elemen-elemen array dalam bentuk label visual (JLabel). Masing-masing label merepresentasikan satu angka dan diberi border, ukuran tetap, serta warna latar putih untuk membuat tampilannya seperti kotak-kotak angka. Panel ini akan berubah setiap kali ada pertukaran elemen atau saat pivot ditentukan.

Di bagian bawah jendela (BorderLayout.SOUTH), ada controlPanel yang berisi dua tombol: “Langkah Selanjutnya” untuk menjalankan proses Quick Sort satu per satu, dan “Reset” untuk mengembalikan semua elemen ke kondisi awal. Sementara itu, di sisi kanan jendela (BorderLayout.EAST) terdapat stepArea, yaitu area teks panjang dalam JScrollPane yang mencatat semua langkah yang telah dilakukan, dari partisi sampai penempatan pivot, dengan penomoran langkah yang jelas.

A screen shot of a computer code

AI-generated content may be incorrect.

A screen shot of a computer code

AI-generated content may be incorrect. Fungsi setArrayFromInput dalam program ini merupakan pintu masuk utama bagi untuk memulai proses Quick Sort. Fungsi ini akan dijalankan saat menekan tombol “Set Array”. Pertama-tama, program akan mengambil isi dari kolom input (inputField) dan memeriksa apakah input tersebut kosong. Jika tidak, string input akan dipecah berdasarkan tanda koma, lalu setiap bagian akan dikonversi menjadi angka bulat (int). Apabila ditemukan karakter non-angka, program akan memunculkan pesan error agar memperbaiki input-nya.

Setelah array berhasil dibentuk, program mulai melakukan inisialisasi untuk proses sorting. Stack yang akan menyimpan rentang indeks partisi direset terlebih dahulu, lalu ditambahkan pasangan indeks awal dan akhir array sebagai tahap pertama dari proses Quick Sort. Beberapa variabel seperti sorting, partitioning, dan stepCount juga diatur ulang untuk memastikan bahwa kondisi awal program sudah bersih. Tombol “Langkah Selanjutnya” diaktifkan agar bisa langsung mulai proses sorting per langkah, dan area teks log (stepArea) dikosongkan dari catatan sebelumnya.

Selanjutnya, panelArray yang bertugas menampilkan elemen array dalam bentuk visual akan dibersihkan dan diisi ulang. Setiap elemen dalam array ditampilkan dalam bentuk JLabel berbentuk kotak, dengan ukuran dan border yang seragam, latar belakang putih, dan teks berisi nilai angka tersebut. Kotak-kotak ini akan menjadi elemen visual yang berubah warna saat proses sorting berlangsung. Setelah semuanya selesai ditampilkan, program akan menuliskan catatan pertama di area log bahwa proses Quick Sort telah dimulai.

A screen shot of a computer code

AI-generated content may be incorrect.

A screen shot of a computer code

AI-generated content may be incorrect. Fungsi performStep merupakan inti dari program ini, di mana seluruh logika Quick Sort dijalankan secara bertahap setiap kali menekan tombol “Langkah Selanjutnya.” Proses dimulai dengan memanggil resetHighlights untuk menghapus semua highlight warna dari langkah sebelumnya, memastikan tampilan bersih untuk langkah berikutnya. Selanjutnya, program akan memeriksa apakah proses sorting masih berlangsung dan apakah masih ada rentang indeks di dalam stack. Jika stack kosong dan tidak sedang dalam kondisi partisi, artinya proses sudah selesai. Maka, program akan menonaktifkan tombol langkah, menampilkan pesan bahwa sorting selesai, dan mencetak array hasil akhir di area log.

Jika masih ada langkah yang harus dijalankan dan belum dalam kondisi partisi, program akan mengambil pasangan indeks (low dan high) dari stack. Kemudian, dipilih elemen paling kanan sebagai pivot. Dua indeks bantu, i dan j, diinisialisasi untuk membantu proses partisi, lalu variabel partitioning diatur menjadi true. Program menandai posisi pivot dengan warna kuning dan mencatat langkah partisi ke dalam log. Setelah itu, fungsi keluar untuk memberi kesempatan melihat bagian ini secara jelas sebelum lanjut ke perbandingan.

Pada langkah berikutnya, ketika partitioning sudah aktif, program membandingkan elemen ke-j dengan pivot. Jika elemen tersebut lebih kecil atau sama dengan pivot, maka indeks i digeser satu posisi ke kanan dan dilakukan pertukaran antara elemen i dan j. Elemen yang ditukar diberi warna hijau dan cyan sebagai penanda visual. Jika tidak ada pertukaran, hanya log teks yang dicetak untuk mencatat bahwa perbandingan tidak menghasilkan pertukaran. Indeks j kemudian ditambah agar proses berjalan ke elemen berikutnya.

A computer screen shot of text

AI-generated content may be incorrect.

Setelah semua elemen dari low sampai high sudah dibandingkan dengan pivot, maka pivot akan dipindahkan ke posisi akhir dari partisi, yaitu i + 1, dan diberi warna kuning. Stack akan diperbarui dengan dua rentang baru: bagian kiri (sebelum pivot) dan bagian kanan (setelah pivot), yang akan diproses pada langkah-langkah selanjutnya. Variabel partitioning direset menjadi false agar langkah berikutnya bisa kembali memproses rentang baru dari stack. Seluruh proses ini akan terus berulang hingga stack kosong dan array terurut.

Bagian highlight dan penandaan pivot serta perbandingan dalam program ini bertujuan untuk memperjelas secara visual elemen mana yang sedang aktif selama proses Quick Sort berlangsung. Setiap kali menekan tombol “Langkah Selanjutnya,” elemen yang berperan penting di langkah tersebut akan diberi warna berbeda agar mudah dikenali. Fungsi highlightPivot digunakan untuk memberi warna kuning pada elemen pivot, yaitu elemen yang menjadi acuan dalam proses partisi. Warna kuning ini konsisten digunakan sepanjang program untuk menunjukkan posisi pivot, sehingga bisa langsung tahu mana elemen yang sedang menjadi titik pembanding.

Sementara itu, untuk proses perbandingan antara elemen dengan pivot, digunakan fungsi highlightCompare. Fungsi ini akan memberi warna cyan pada elemen j yang sedang dibandingkan, dan tetap mempertahankan warna kuning pada pivot. Kombinasi warna ini memudahkan membedakan antara elemen pembanding (j) dan pivot dalam satu langkah visual. Warna-warna tersebut tidak hanya memperindah tampilan, tetapi benar-benar punya makna logis dalam membantu memahami logika di balik langkah yang sedang berlangsung. Setelah satu langkah selesai dijalankan, semua warna akan direset ke putih melalui fungsi resetHighlights untuk menjaga agar tampilan tetap bersih dan fokus hanya pada elemen yang relevan di langkah berikutnya.

A computer screen shot of code

AI-generated content may be incorrect.

Bagian penukaran elemen dan pembaruan tampilan array ditangani oleh dua fungsi utama, yaitu swap dan updateLabels. Fungsi swap digunakan ketika dua elemen dalam array perlu ditukar posisinya, yang merupakan bagian penting dalam proses partisi Quick Sort. Proses penukaran dilakukan secara sederhana dengan menyimpan salah satu nilai dalam variabel sementara, lalu menyalin nilai dari indeks lainnya, dan akhirnya memasukkan kembali nilai yang disimpan ke posisi pasangannya. Setelah nilai dalam array diperbarui, fungsi updateLabels langsung dipanggil agar perubahan tersebut tercermin secara visual di tampilan GUI.

Fungsi updateLabels bekerja dengan cara mengganti teks di setiap label dalam labelArray agar sesuai dengan isi terbaru dari array. Jadi, jika elemen ke-2 dan ke-5 ditukar, maka label yang menampilkan angka-angka tersebut juga akan ikut berubah, memberikan kesan visual bahwa elemen benar-benar bertukar tempat. Selain memperbarui isi label, fungsi ini juga mencetak nama dan NIM mahasiswa di konsol sebagai bentuk identifikasi pembuat program. Kolaborasi antara swap dan updateLabels membuat program tidak hanya menjalankan proses sorting di balik layar, tapi juga benar-benar menampilkan dampaknya secara nyata di layar, sehingga bisa melihat perubahan setiap langkah dengan sangat jelas.

A computer screen shot of text

AI-generated content may be incorrect.

Fungsi reset dalam program ini berfungsi untuk mengembalikan seluruh komponen aplikasi ke kondisi awal, sehingga bisa memulai ulang proses Quick Sort dengan data baru tanpa harus menutup dan membuka ulang aplikasi. Saat tombol “Reset” ditekan, program akan mengosongkan kolom input (inputField) agar siap menerima angka baru dari . Panel visual array (panelArray) juga akan dihapus seluruh isinya, lalu direfresh agar tampilannya bersih.

Selain tampilan, variabel-variabel logika internal juga di-reset. Stack yang berisi daftar rentang indeks untuk partisi akan dikosongkan, lalu variabel sorting dan partitioning di-set menjadi false, menandakan bahwa tidak ada proses sorting yang sedang berjalan. Tombol “Langkah Selanjutnya” juga dinonaktifkan agar tidak bisa menjalankan langkah baru sebelum memasukkan data lagi. Area log (stepArea) yang sebelumnya berisi catatan langkah-langkah Quick Sort juga dibersihkan agar tidak bercampur dengan sesi baru. Akhirnya, variabel penghitung langkah (stepCount) di-reset kembali ke angka 1, siap digunakan untuk mencatat langkah-langkah baru.

A computer screen shot of text

AI-generated content may be incorrect.

Fungsi arrayToString dalam program ini digunakan untuk mengubah isi array integer menjadi bentuk string yang rapi dan mudah dibaca oleh , terutama saat ditampilkan di area log (stepArea). Fungsi ini memanfaatkan StringBuilder untuk menyusun string secara efisien. Dengan melakukan iterasi terhadap setiap elemen dalam array, nilai-nilai tersebut ditambahkan satu per satu ke dalam StringBuilder. Setelah menambahkan setiap angka, jika elemen tersebut bukan elemen terakhir, maka akan ditambahkan tanda koma dan spasi sebagai pemisah.

Hasil akhirnya adalah representasi array seperti “5, 2, 8, 1, 4” yang dapat langsung ditampilkan di log teks agar bisa melihat bentuk array terkini dalam format yang familiar dan mudah dipahami. Fungsi ini sering digunakan setelah langkah sorting selesai atau ketika program ingin menunjukkan kondisi array pada titik tertentu dalam proses.

A computer screen shot of a program code

AI-generated content may be incorrect.**4. Class ShellSortGUI**

Pada bagian inisialisasi dan struktur umum program ShellSortGUI, terlihat bahwa program ini dibangun sebagai aplikasi desktop berbasis Java Swing, dengan class utama ShellSortGUI yang merupakan turunan dari JFrame. Hal ini memungkinkan aplikasi menampilkan jendela interaktif dengan elemen-elemen grafis seperti tombol, panel, dan teks area. Program dimulai dari method main, yang menjalankan pembuatan objek GUI di dalam EventQueue.invokeLater, sebuah cara standar untuk memastikan elemen GUI berjalan di thread yang aman.

Di dalam class, terdapat sejumlah variabel yang disiapkan untuk menjalankan logika Shell Sort dan kebutuhan tampilan. Variabel array menyimpan data yang akan diurutkan, sementara labelArray menampung representasi visual dari elemen-elemen array dalam bentuk label. Komponen antarmuka seperti inputField, stepButton, resetButton, dan stepArea digunakan untuk mengatur interaksi : mulai dari memasukkan angka, memulai proses sorting, hingga menampilkan hasil dan log tiap langkah. Selain itu, variabel-variabel seperti i, j, gap, temp, sorting, dan isMapping digunakan untuk mengatur proses internal algoritma Shell Sort langkah demi langkah.

Di bagian atas jendela (BorderLayout.NORTH), terdapat inputPanel yang berisi label instruksi, kolom input teks (inputField) untuk memasukkan angka-angka array yang ingin diurutkan, serta tombol “Set Array” yang akan memproses input tersebut. Komponen-komponen ini disusun secara horizontal menggunakan FlowLayout, membuatnya terlihat simpel dan langsung.

Pada bagian tengah jendela (BorderLayout.CENTER), ada panelArray, yaitu tempat di mana elemen-elemen array akan ditampilkan dalam bentuk JLabel. Setiap label divisualisasikan seperti kotak angka: punya border, ukuran tetap, dan warna latar putih. Di sinilah nanti perubahan posisi elemen akan tampak saat proses sorting berlangsung. Panel ini juga menggunakan FlowLayout agar elemen ditampilkan sejajar secara horizontal.

Kemudian di bagian bawah jendela (BorderLayout.SOUTH), ada controlPanel yang berisi dua tombol: “Langkah Selanjutnya” untuk memproses langkah demi langkah dari algoritma Shell Sort, dan “Reset” untuk menghapus semuanya dan memulai dari awal. Terakhir, pada sisi kanan jendela (BorderLayout.EAST), terdapat stepArea, yaitu sebuah area teks yang dibungkus dengan JScrollPane. Area ini digunakan untuk menampilkan log langkah-langkah yang dilakukan selama proses sorting, lengkap dengan penomoran dan deskripsi yang memudahkan mengikuti apa yang sedang terjadi.

A screen shot of a computer code

AI-generated content may be incorrect.

A screen shot of a computer code

AI-generated content may be incorrect.Fungsi setArrayFromInput adalah titik awal interaksi dengan program. Saat memasukkan deretan angka yang dipisahkan koma ke dalam inputField dan menekan tombol “Set Array”, fungsi ini akan langsung dijalankan. Pertama-tama, isi dari inputField akan diambil dan dipotong berdasarkan tanda koma. Setiap elemen hasil pemotongan kemudian dikonversi ke dalam bentuk integer dan disimpan dalam array utama. Jika ada karakter non-angka atau input kosong, maka program akan menampilkan pesan error menggunakan JOptionPane dan menghentikan proses.

Setelah input berhasil diproses menjadi array, variabel-variabel penting seperti i, j, dan stepCount diinisialisasi ulang ke nilai awal. Khusus untuk algoritma Shell Sort, nilai awal gap ditentukan sebagai setengah dari panjang array. Variabel sorting akan di-set ke true sebagai tanda bahwa proses sorting sudah siap dimulai, dan tombol “Langkah Selanjutnya” pun diaktifkan. Selain itu, area log (stepArea) dikosongkan agar siap mencatat langkah-langkah baru, dan panelArray juga dibersihkan untuk memuat visualisasi array yang baru.

Setiap elemen array kemudian ditampilkan dalam bentuk label (JLabel) yang menyerupai kotak angka. Label ini diberi tampilan seragam: ukuran tetap, border hitam, latar putih, serta teks angka di tengahnya. Semua label ini ditambahkan ke panelArray dan ditampilkan ulang menggunakan revalidate dan repaint. Terakhir, program akan mencetak log awal di area teks untuk memberi tahu bahwa proses Shell Sort telah dimulai, termasuk mencantumkan nilai awal dari gap.

A screen shot of a computer program

AI-generated content may be incorrect.

A computer screen with text

AI-generated content may be incorrect.Fungsi performStep adalah bagian utama dari program ini yang menangani proses sorting Shell Sort secara bertahap, satu klik satu langkah. Ketika tombol “Langkah Selanjutnya” ditekan, fungsi ini akan dijalankan untuk memproses satu langkah dari algoritma. Langkah pertama yang dilakukan adalah memanggil resetHighlights() untuk menghapus warna pada elemen-elemen array, agar tampilan bersih sebelum melanjutkan ke langkah berikutnya. Jika gap bernilai nol atau sorting bernilai false, artinya proses sudah selesai. Maka, program akan mencetak pesan bahwa sorting selesai, menampilkan hasil akhir array dalam bentuk string, menonaktifkan tombol langkah, dan memunculkan notifikasi ke . Namun jika sorting masih berlangsung, proses akan dilanjutkan.

Selama nilai i masih berada dalam batas array, program akan menentukan apakah sedang dalam proses “mapping” atau belum. Jika belum (isMapping == false), artinya elemen ke-i baru akan diproses: nilainya disimpan dalam variabel temp, nilai j di-set sama dengan i, isMapping diubah menjadi true, dan elemen yang dipilih diberi highlight warna kuning. Kemudian dicetak log bahwa elemen pada posisi tersebut sedang dipilih.

Pada langkah berikutnya, jika j masih lebih besar atau sama dengan gap dan elemen di posisi j - gap lebih besar dari temp, maka nilai di posisi tersebut digeser ke kanan, posisi visualnya di-highlight, dan label diperbarui dengan nilai baru. Ini menunjukkan bahwa elemen yang lebih besar sedang digeser untuk memberi tempat bagi elemen temp. Setelah itu, j dikurangi sebesar gap, dan fungsi berhenti sejenak untuk menunggu klik berikutnya. Jika kondisi penggeseran tidak lagi terpenuhi, maka nilai temp akan diletakkan di posisi akhir yang sesuai (j). Label-label akan diperbarui untuk mencerminkan susunan array terbaru, dan proses pindah ke elemen berikutnya dengan menambah i serta mengatur ulang isMapping ke false. Begitu i melewati batas array, artinya proses pada gap saat ini telah selesai. Maka, nilai gap akan dibagi dua, i di-set ke nilai gap baru, dan proses dilanjutkan lagi. Pesan log akan dicetak untuk memberi tahu bahwa gap telah berkurang. Seluruh proses ini berlangsung langkah demi langkah hingga gap menjadi nol, dan array sudah sepenuhnya terurut.

A computer screen shot of text

AI-generated content may be incorrect.Highlight dan perpindahan elemen dalam program ini dirancang untuk membantu memahami secara visual bagaimana elemen-elemen array berpindah selama proses Shell Sort berlangsung. Ketika elemen tertentu sedang dipilih untuk diproses, label elemen tersebut diberi warna kuning menggunakan method setBackground, menandakan bahwa nilai tersebut sedang “dipegang” oleh algoritma dan akan dipindahkan ke posisi yang lebih tepat. Ini terjadi saat program masuk ke fase isMapping == false, dan nilai elemen tersebut disimpan ke dalam variabel temp.

Saat algoritma menemukan bahwa elemen di posisi sebelumnya (berjarak gap) lebih besar dari temp, elemen tersebut digeser ke kanan. Untuk memperjelas perpindahan ini, elemen yang sedang digeser diberi warna hijau, sementara elemen asalnya yang lebih besar ditandai dengan warna cyan. Warna-warna ini membuat langsung paham: yang hijau sedang “ditiban”, dan yang cyan adalah nilai yang digeser. Setelah selesai mencari posisi yang tepat, nilai temp akan ditempatkan ke posisi j yang baru, dan tampilan array diperbarui melalui pemanggilan fungsi updateLabels().

A computer screen shot of code

AI-generated content may be incorrect.

Fungsi reset dalam program ShellSortGUI digunakan untuk mengembalikan seluruh komponen aplikasi ke kondisi awal, sehingga bisa memulai proses Shell Sort dari awal tanpa perlu menutup program. Saat fungsi ini dipanggil—baik karena menekan tombol “Reset” atau ingin mencoba array baru—program akan langsung menghapus isi kolom input (inputField) dan menghapus seluruh elemen array visual dari panelArray, lalu menyegarkan tampilannya dengan revalidate() dan repaint() agar benar-benar kosong.

Selain tampilan, fungsi ini juga mengosongkan area teks log (stepArea) agar seluruh catatan langkah-langkah sebelumnya hilang dan tidak membingungkan . Tombol “Langkah Selanjutnya” juga akan dinonaktifkan kembali karena tidak ada proses sorting yang aktif. Kemudian, seluruh variabel penting seperti sorting, isMapping, stepCount, serta indeks i, j, dan gap di-reset ke nilai awal. Ini penting supaya logika program tidak menyimpan status dari percobaan sebelumnya yang bisa mengacaukan proses baru.

A computer code with colorful text

AI-generated content may be incorrect.Fungsi arrayToString berperan untuk mengubah isi array menjadi bentuk string yang mudah dibaca, terutama saat ditampilkan di area log (stepArea). Fungsi ini digunakan ketika program ingin menunjukkan hasil akhir dari proses sorting, atau sekadar mencetak kondisi array saat ini dengan format yang rapi. Cara kerjanya cukup sederhana: setiap elemen dalam array dibaca satu per satu dan dimasukkan ke dalam StringBuilder. Setelah setiap angka ditambahkan, jika angka tersebut bukan elemen terakhir, maka ditambahkan tanda koma dan spasi sebagai pemisah.

Sebagai contoh, jika array berisi {5, 3, 8}, maka hasil dari fungsi ini akan menjadi "5, 3, 8", yang jauh lebih enak dibaca dibandingkan hanya menampilkan array secara mentah. Meskipun kelihatannya sepele, fungsi ini sangat membantu menjaga log program tetap bersih, jelas, dan nyaman dipahami, terutama saat ingin melihat perkembangan array setelah beberapa langkah sorting.