Laporan Praktikum Struktur Data Pekan 5

DOSEN PENGAMPU:

Dr. Wahyudi S.T, M.T

OLEH:

Wahyu Khairi

2311531009

UNIVERSITAS ANDALAS

FAKULTAS TEKNOLOGI INFORMASI

JURUSAN INFORMATIKA

2023/2024

1. **TUJUAN**
2. Mengenal apa itu sorting algorithm dalam bahasa pemrograman java
3. Memahami langkah langkah menjalankan program menggunakan sorting algorithm
4. Memahami kapan sorting algorithm dapat digunakan dalam sebuah program
5. **KAJIAN TEORI**
6. **Sorting Algorithm**

Algoritma pengurutan mengatur elemen daftar dalam urutan tertentu. Urutan numerik dan leksikografis, baik menaik maupun menurun, adalah urutan yang paling sering digunakan. Pengurutan yang efisien sangat penting untuk mengoptimalkan efisiensi algoritma lain, seperti algoritma pencarian dan penggabungan, yang memerlukan data dalam bentuk daftar yang terurut. Selain itu, pengurutan sering digunakan untuk mengkanonisasi data dan menghasilkan output yang mudah dibaca oleh manusia.

Dua kondisi harus dipenuhi oleh output algoritma pengurutan secara formal: Input adalah permutasi (urutan ulang, tetapi tetap mempertahankan semua elemen asli) dari input, dan output berada dalam urutan monotonik (setiap elemen tidak lebih kecil atau lebih besar dari elemen sebelumnya, sesuai dengan urutan yang diinginkan).

Pengurutan adalah topik yang menarik dalam ilmu komputer, dan banyak algoritma pengurutan yang dapat digunakan. Beberapa algoritma pengurutan yang umum digunakan meliputi:

1. Insertion Sort

Insert Sort adalah algoritma pengurutan sederhana yang menggunakan perbandingan untuk membuat array terurut akhir (atau daftar) satu elemen pada satu waktu. Namun, ia kurang efisien untuk daftar besar dibandingkan dengan algoritma pengurutan yang lebih canggih seperti quicksort, heapsort, atau merge sort. Namun, Insertion Sort memiliki beberapa keunggulan:

1. Stabilitas: Algoritma ini menghasilkan urutan relatif elemen yang memiliki nilai yang sama dalam output yang terurut.
2. In-Place: Sortasi penambahan bekerja secara "in-place", artinya hanya membutuhkan array asli sebagai ruang memori.
3. Selection Sort

Algoritma pengurutan sederhana memilih elemen terkecil atau terbesar dari bagian daftar yang belum terurut dan menukarnya dengan elemen pertama dari bagian yang belum terurut. Prosedur ini diulang hingga seluruh daftar terurut.

Mari kita lihat bagaimana algoritma Sort Selection bekerja:

1. Pembagian (Divide): Algoritma membagi daftar menjadi dua bagian. Bagian Terurut terletak di ujung kiri daftar dari kiri ke kanan, dan Bagian Belum Terurut mengandung elemen yang belum terurut.
2. Penyortiran (Conquer): Algoritma memilih elemen terkecil dari bagian yang belum terurut secara berulang kali dan menukarnya dengan elemen pertama dalam bagian yang belum terurut. Ini menunjukkan bahwa elemen terkecil akan berada di tempat yang tepat dalam daftar terurut setelah setiap iterasi.
3. Penggabungan (Merge): Proses ini diulang sampai seluruh daftar terurut. Ini dilakukan setelah semua elemen dipindahkan ke posisi yang benar.
4. Bubble Sort

Algoritma pengurutan Bubble Sort berfungsi dengan membandingkan elemen dalam daftar yang berdekatan berulang kali dan mengubah posisi mereka jika diperlukan. Proses ini diulang hingga tidak ada lagi pertukaran yang perlu dilakukan selama satu iterasi, yang menandakan bahwa daftar sudah terurut.

1. **LANGKAH PENGERJAAN**
2. **InsertionSort**
3. Metode sort(int arr[]):

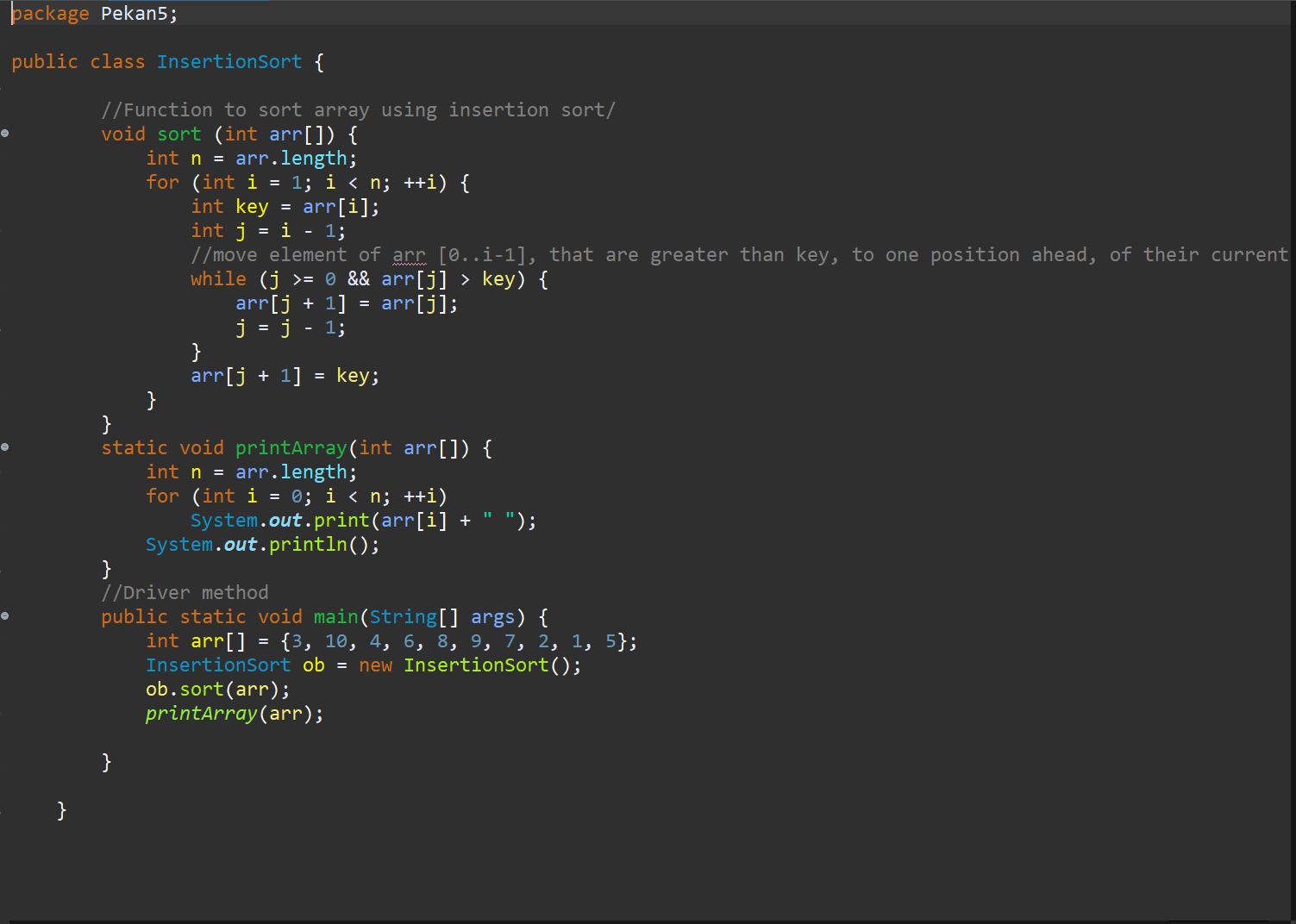
Metode ini menerima parameter berupa array integer arr.Algoritma Insertion Sort diimplementasikan di dalam metode ini. Pertama, kita mengambil panjang array (n). Kemudian, kita melakukan iterasi dari indeks kedua hingga akhir array. Pada setiap iterasi, kita membandingkan elemen saat ini dengan elemen-elemen sebelumnya. Jika elemen sebelumnya lebih besar dari elemen saat ini, kita tukar posisi keduanya. Proses ini diulang hingga elemen saat ini berada di posisi yang benar dalam array terurut.

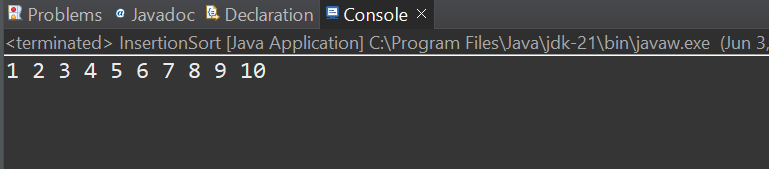
1. Metode printArray(int arr[]):

Metode ini menerima parameter berupa array integer arr. Metode ini mencetak isi array ke layar.

1. Metode main(String[] args):

Metode ini merupakan metode utama yang akan dijalankan ketika program dijalankan. Di dalam metode ini, kita membuat array dengan elemen [3, 10, 4, 6, 8, 9, 7, 2, 1, 5]. Kemudian, kita membuat objek dari kelas InsertionSort. Selanjutnya, kita memanggil metode sort(arr) untuk mengurutkan array. Terakhir, kita mencetak array yang sudah terurut menggunakan metode printArray(arr).

****

**OUTPUT**

1. **InsertionSort2**
2. Metode insertionSort(int[] arr):

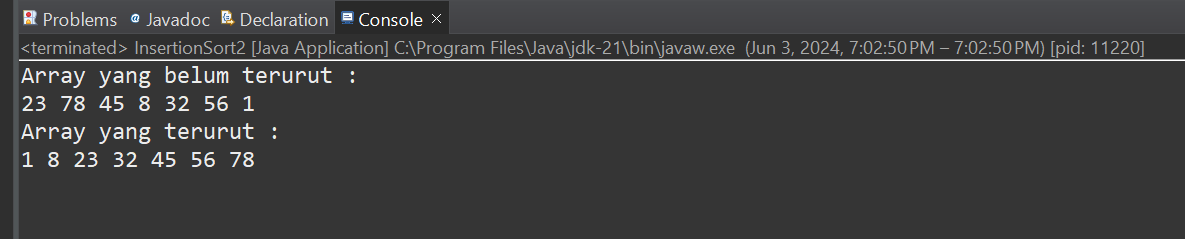
Metode ini menerima parameter berupa array integer arr. Algoritma Insertion Sort diimplementasikan di dalam metode ini. Pertama, kita mengambil panjang array (n). Kemudian, kita melakukan iterasi dari indeks pertama hingga akhir array. Pada setiap iterasi, kita membandingkan elemen saat ini dengan elemen-elemen sebelumnya. Jika elemen sebelumnya lebih besar dari elemen saat ini, kita tukar posisi keduanya. Proses ini diulang hingga elemen saat ini berada di posisi yang benar dalam array terurut.

1. Metode main(String[] args):

Metode ini merupakan metode utama yang akan dijalankan ketika program dijalankan. Di dalam metode ini, kita membuat array dengan elemen [23, 78, 45, 8, 32, 56, 1]. Kemudian, kita memanggil metode insertionSort(arr) untuk mengurutkan array. Terakhir, kita mencetak array yang sudah terurut.

****

**OUTPUT :**

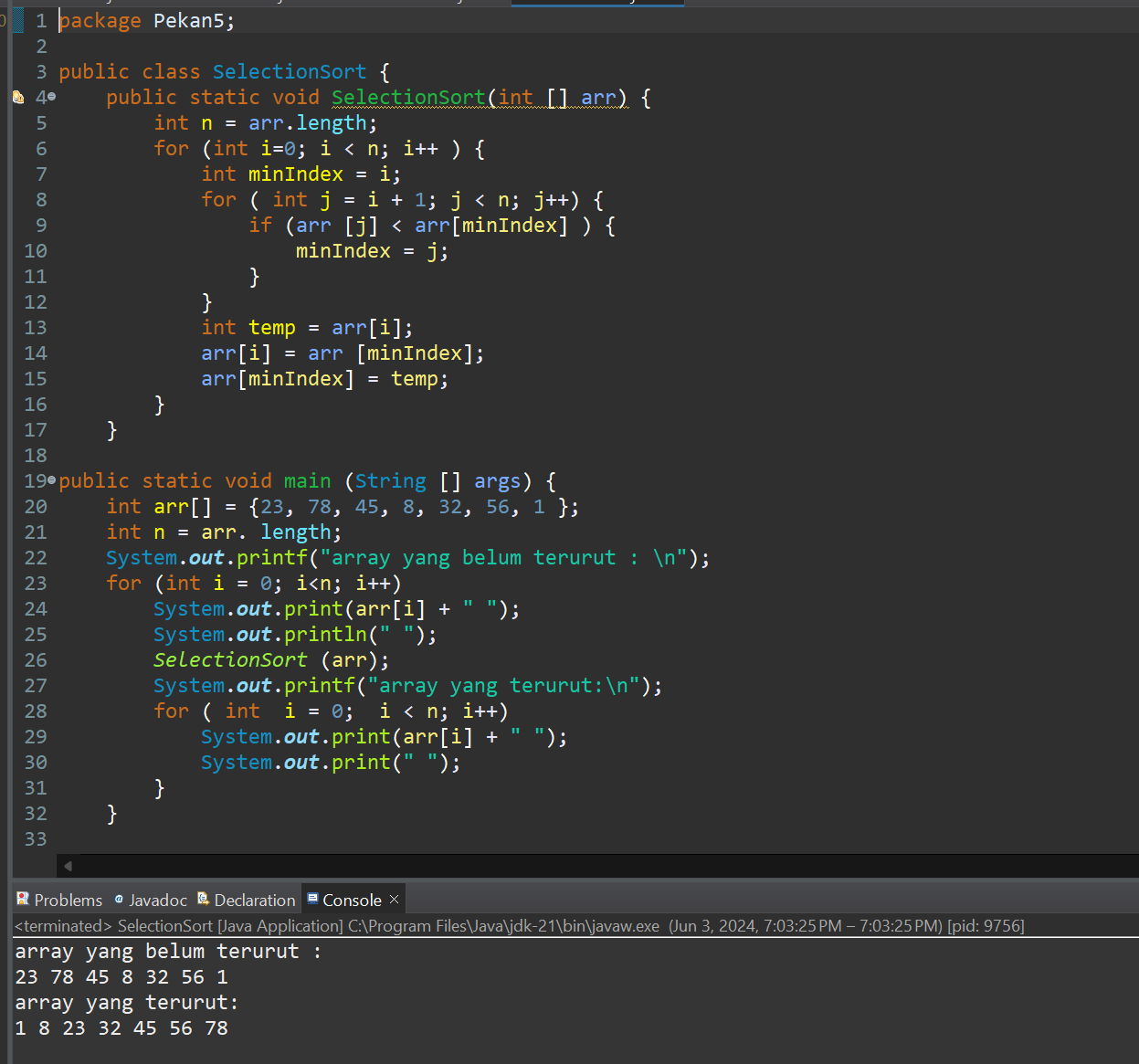
****

1. **selectionSort**
2. Metode selectionSort(int[] arr):

Metode ini menerima parameter berupa array integer arr. Algoritma Selection Sort diimplementasikan di dalam metode ini. Pertama, kita mengambil panjang array (n). Kemudian, kita melakukan iterasi dari indeks pertama hingga akhir array. Pada setiap iterasi, kita mencari elemen terkecil dalam bagian yang belum terurut. Setelah menemukan elemen terkecil, kita menukarnya dengan elemen pertama dalam bagian yang belum terurut. Proses ini diulang hingga seluruh array terurut.

1. Metode main(String[] args):

Metode ini merupakan metode utama yang akan dijalankan ketika program dijalankan. Di dalam metode ini, kita membuat array dengan elemen [23, 78, 45, 8, 32, 56, 1].Kemudian, kita memanggil metode selectionSort(arr) untuk mengurutkan array. Terakhir, kita mencetak array yang sudah terurut.

****

1. **bubbleSort**
2. Metode bubbleSort(int[] arr):

Metode ini menerima parameter berupa array integer arr. Algoritma Bubble Sort diimplementasikan di dalam metode ini. Pertama, kita mengambil panjang array (n). Kemudian, kita melakukan dua iterasi: Iterasi pertama (indeks i) berjalan dari 0 hingga n−1. Iterasi kedua (indeks j) berjalan dari 0 hingga n−1 (kecuali indeks terakhir karena kita membandingkan elemen dengan elemen berikutnya). Pada setiap iterasi kedua, kita membandingkan elemen saat ini dengan elemen berikutnya. Jika elemen saat ini lebih besar dari elemen berikutnya, kita tukar posisi keduanya. Proses ini diulang hingga seluruh array terurut.

1. Metode main(String[] args):

Metode ini merupakan metode utama yang akan dijalankan ketika program dijalankan. Di dalam metode ini, kita membuat array dengan elemen [23, 78, 45, 8, 32, 56, 1]. Kemudian, kita memanggil metode bubbleSort(arr) untuk mengurutkan array. Terakhir, kita mencetak array yang sudah terurut.

