**LAPORAN PRAKTIKUM STRUKTUR DATA**

Queue



Oleh :

FAYI AMATULLAH AZHARA

NIM 2311537001

MATA KULIAH

STRUKTUR DATA

DOSEN PENGAMPU : Dr.WAHYUDI ST, MT.

**FAKULTAS TEKNOLOGI INFORMASI**

**DEPARTEMEN INFORMATIKA**

**UNIVERSITAS ANDALAS**

1. **PENDAHULUAN**

Queue adalah struktur data yang mengikuti prinsip FIFO (First In, First Out), yaitu elemen yang pertama kali dimasukkan akan menjadi elemen yang pertama kali dikeluarkan. Queue biasanya diimplementasikan dalam bentuk antrian yang serupa dengan antrian di kehidupan nyata, seperti antrian di kasir supermarket.

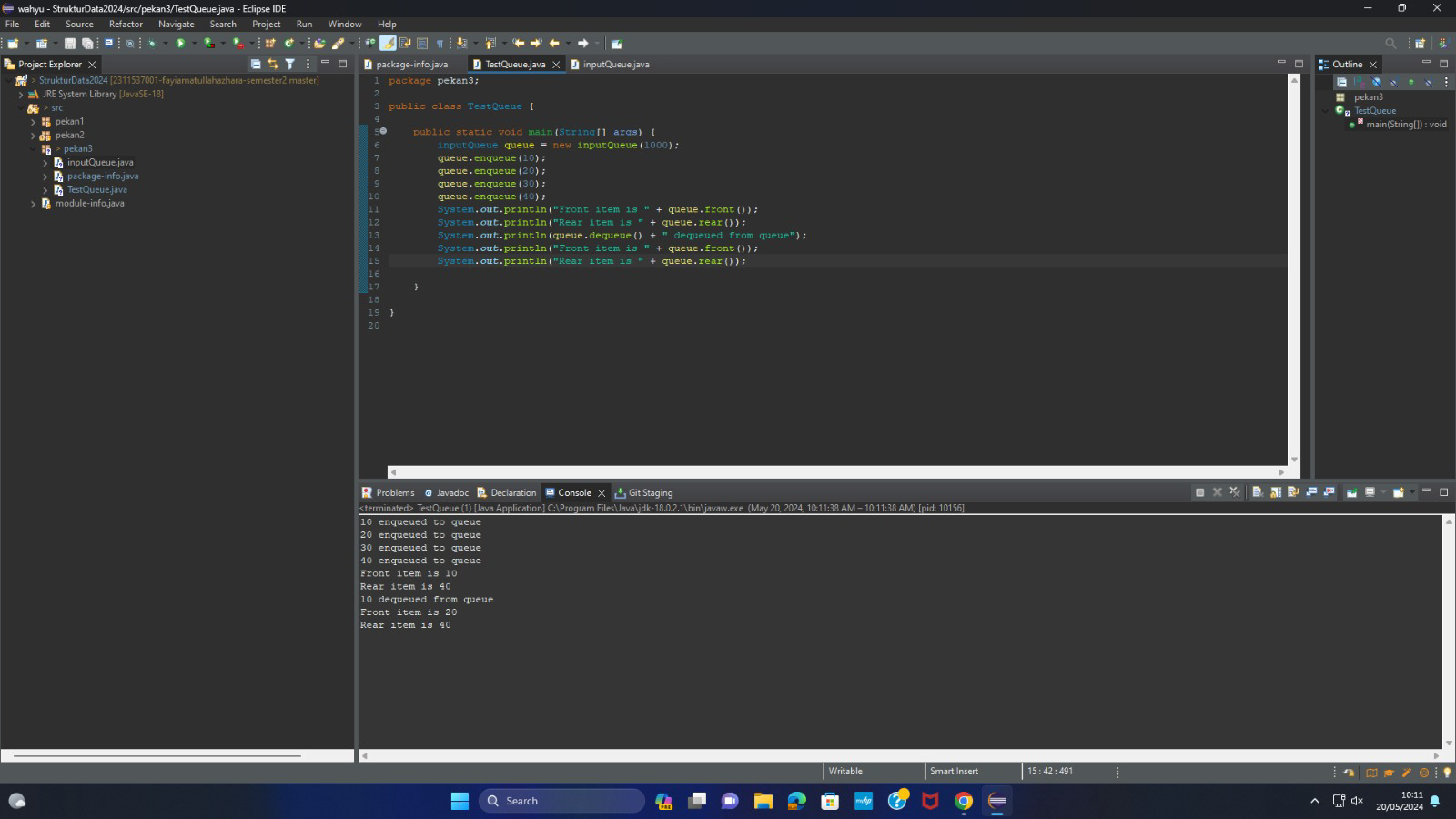
Queue dapat diimplementasikan menggunakan array atau linked list. Implementasi menggunakan array lebih sederhana tetapi memiliki keterbatasan dalam hal ukuran yang tetap. Implementasi menggunakan linked list lebih fleksibel karena ukuran antrian dapat berubah secara dinamis sesuai kebutuhan. Contoh penggunaan Queue di dunia nyata termasuk manajemen tugas di printer, penjadwalan proses di sistem operasi, dan pengelolaan buffer dalam komunikasi data.

1. **TUJUAN**

Memahami Konsep dan Implementasi Queue

1. **LANGKAH-LANGKAH**
2. Membuka Eclipse IDE dan import proyek github dengan nama “STRUKTURDATA2024”
3. Membuat Package baru dengan nama “pekan3”
4. Membuat Kelas baru dengan nama “TestQueue”

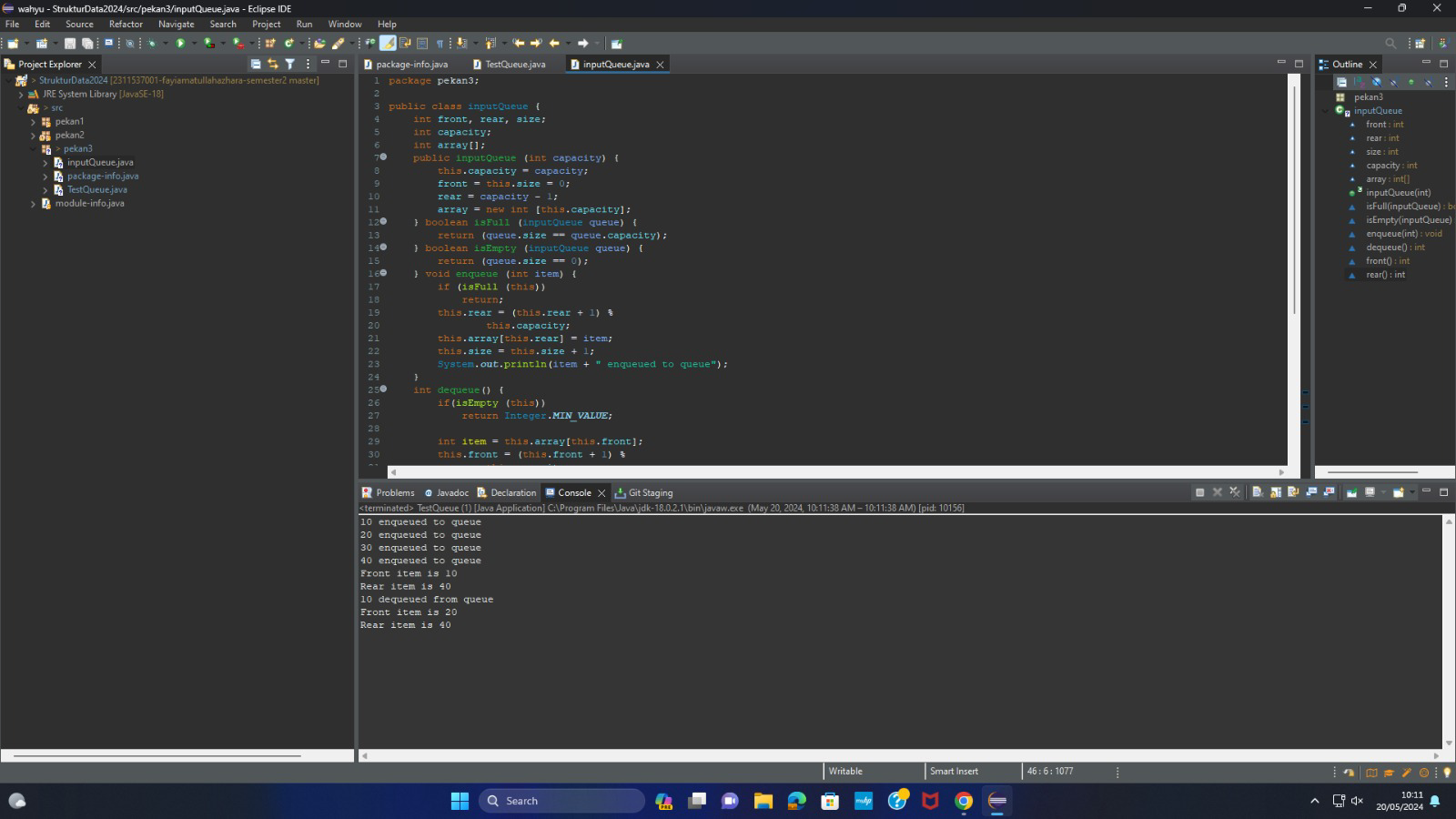
Pada kelas “TestQueue”, masukkan kode seperti gambar di bawah

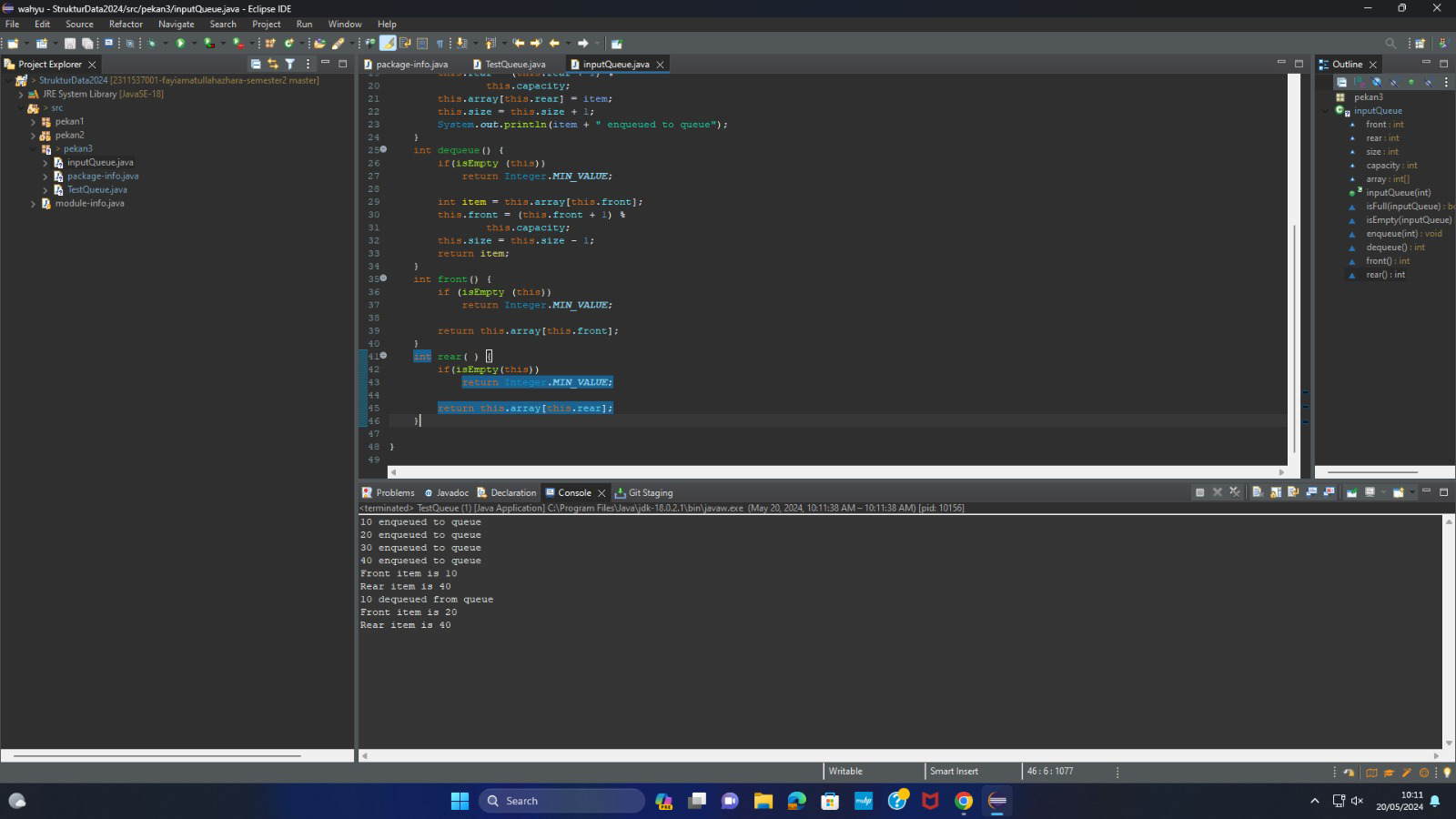


kelas TestQueue digunakan untuk menguji operasi-operasi pada struktur data antrian. Melalui metode main(), kelas ini membuat objek inputQueue, menambahkan elemen-elemen ke antrian menggunakan metode enqueue, dan menguji operasi-operasi seperti front(), rear(), dan dequeue() pada antrian tersebut.

1. Membuat Kelas baru dengan nama “inputQueue”

Pada kelas “inputQueue”, masukkan kode seperti gambar di bawah

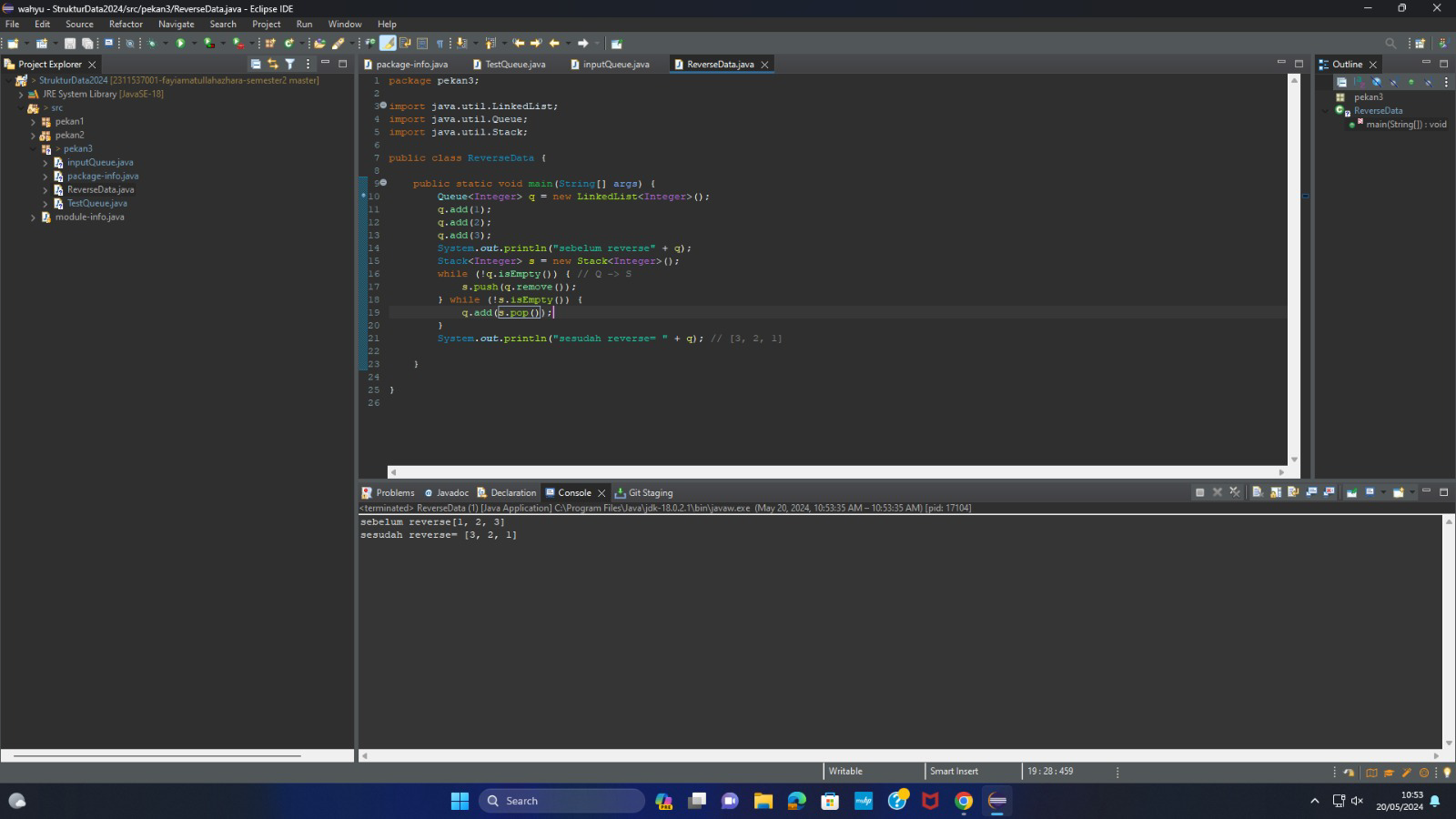




kelas inputQueue berperan sebagai struktur data utama yang mengatur elemen-elemen dalam antrian. Kelas ini memiliki variabel seperti front, rear, size, capacity, dan array[], serta metode-metode seperti isFull(inputQueue queue) untuk memeriksa keadaan penuhnya antrian, isEmpty(inputQueue queue) untuk memeriksa keadaan kosongnya antrian, enqueue(int item) untuk menambahkan elemen ke antrian, dequeue() untuk menghapus dan mengembalikan elemen dari depan antrian, front() untuk mendapatkan elemen di depan antrian tanpa menghapusnya, dan rear() untuk mendapatkan elemen di belakang antrian tanpa menghapusnya.

1. Membuat Kelas baru dengan nama “ReverseData”

Pada kelas “ReverseData”, masukkan kode seperti gambar di bawah



kelas ReverseData memperlihatkan aplikasi dari struktur data antrian dan tumpukan (stack). Dalam kelas ini, sebuah antrian (Queue<Integer> q) dibuat dan beberapa elemen ditambahkan ke dalamnya. Proses pemindahan elemen dari antrian ke tumpukan, dan kemudian kembali ke antrian setelah tumpukan kosong, menghasilkan efek pembalikan data dalam antrian. Dengan demikian, ketiga kelas ini saling terkait dalam mengilustrasikan konsep struktur data antrian, pengujian operasi antrian, dan aplikasi penggunaan tumpukan untuk memanipulasi data dalam antrian.

**D. PENUTUP**

Dalam laporan praktikum ini, saya telah mengimplementasikan struktur data antrian (queue) dan memahami operasi-operasi dasar yang terkait, seperti penambahan (enqueue), penghapusan (dequeue), dan akses ke elemen-elemen antrian (front dan rear). Saya juga telah melihat aplikasi dari struktur data tumpukan (stack) dalam membalikkan data dalam antrian. Melalui praktikum ini, saya dapat mengonfirmasi pemahaman konsep-konsep dasar dalam pemrograman terkait struktur data tersebut dan memperoleh pengalaman praktis dalam penggunaannya. Dengan demikian, praktikum ini telah memberikan wawasan yang berharga dalam memahami dan mengimplementasikan struktur data antrian serta mengaplikasikannya dalam pemrosesan data.