LAPORAN PRAKTIKUM (4) STRUKTUR DATA



Oleh:

Nama : Arkan Ubaidillah Warman

NIM : 2411537001

Dosen Pengampu : Dr. Wahyudi MT .

STRUKTUR DATA

(QUEUE,REVERSE DATA)

DEPARTEMEN INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

UNIVERITAS ANDALAS

PADANG

1. **PENDAHULUAN**
2. **LATAR BELAKANG**

Dalam dunia pemrograman, pengelolaan data sangat penting agar informasi dapat diproses dengan efisien. Salah satu struktur data dasar yang umum digunakan adalah **Queue (antrian)**, yang bekerja berdasarkan prinsip FIFO (*First In, First Out*). Selain itu, teknik *reverse* atau pembalikan data sering digunakan dalam berbagai kasus, seperti algoritma pencarian, pemrosesan string, dan manipulasi stack. Dengan memahami dan mengimplementasikan struktur data ini melalui Java di IDE seperti Eclipse, mahasiswa dapat mengembangkan kemampuan analisis algoritma dan penerapan struktur data dalam aplikasi nyata.

1. **TUJUAN PERCOBAAN**

* Mempelajari dan memahami konsep dasar serta implementasi struktur data **Queue** di Java.
* Mengimplementasikan proses **reverse data** dalam bentuk program Java.
* Melatih kemampuan penggunaan IDE Eclipse dalam membuat dan men-debug program berbasis struktur data.

1. **LANDASAN TEORI**

**3.1 Queue**

Queue adalah struktur data linier yang mengikuti aturan FIFO (First In, First Out). Elemen pertama yang masuk ke dalam queue akan menjadi elemen pertama yang keluar. Dalam Java, struktur ini dapat diimplementasikan menggunakan antarmuka Queue dan kelas seperti LinkedList.

**Operasi dasar Queue:**

* enqueue (menambahkan elemen ke belakang queue)
* dequeue (menghapus elemen dari depan queue)
* peek (melihat elemen depan tanpa menghapus)

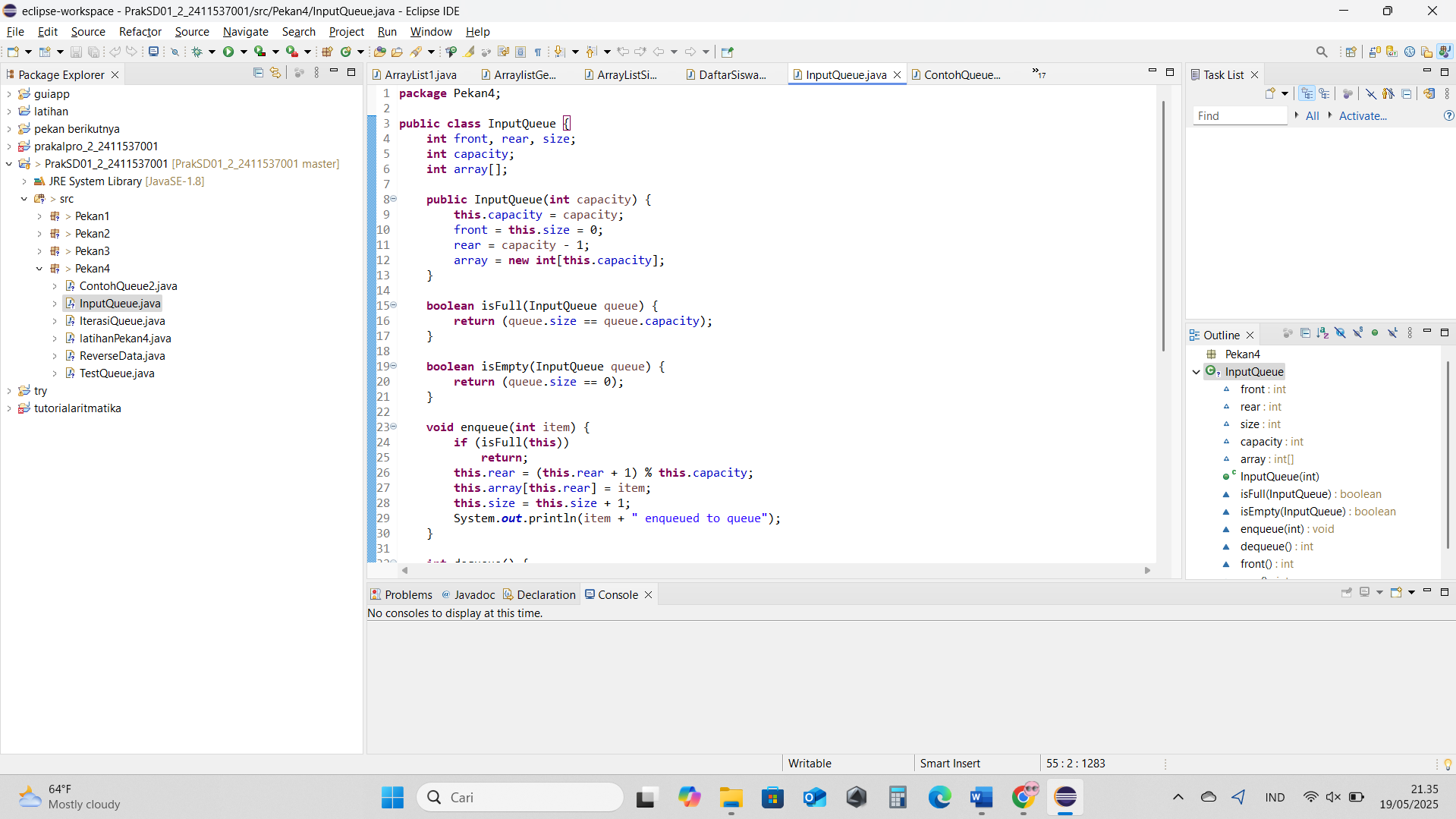
**3.2 Reverse Data**

*Reverse* berarti membalik urutan elemen dalam suatu data. Teknik ini umum digunakan untuk memproses data dalam urutan terbalik. Dalam Java, membalik urutan bisa dilakukan menggunakan Stack, karena Stack bekerja dengan prinsip LIFO (*Last In, First Out*), atau dengan perulangan sederhana yang memproses elemen dari akhir ke awal.

**3.3 Java dan Eclipse**

Java adalah bahasa pemrograman berorientasi objek yang banyak digunakan untuk pengembangan aplikasi. Eclipse adalah salah satu IDE (Integrated Development Environment) yang mendukung pengembangan aplikasi Java secara efisien dengan fitur-fitur seperti auto-complete, debugging, dan manajemen proyek.

1. **LANGKAH LANGKAH**

* INPUTQUEUE.JAVA****

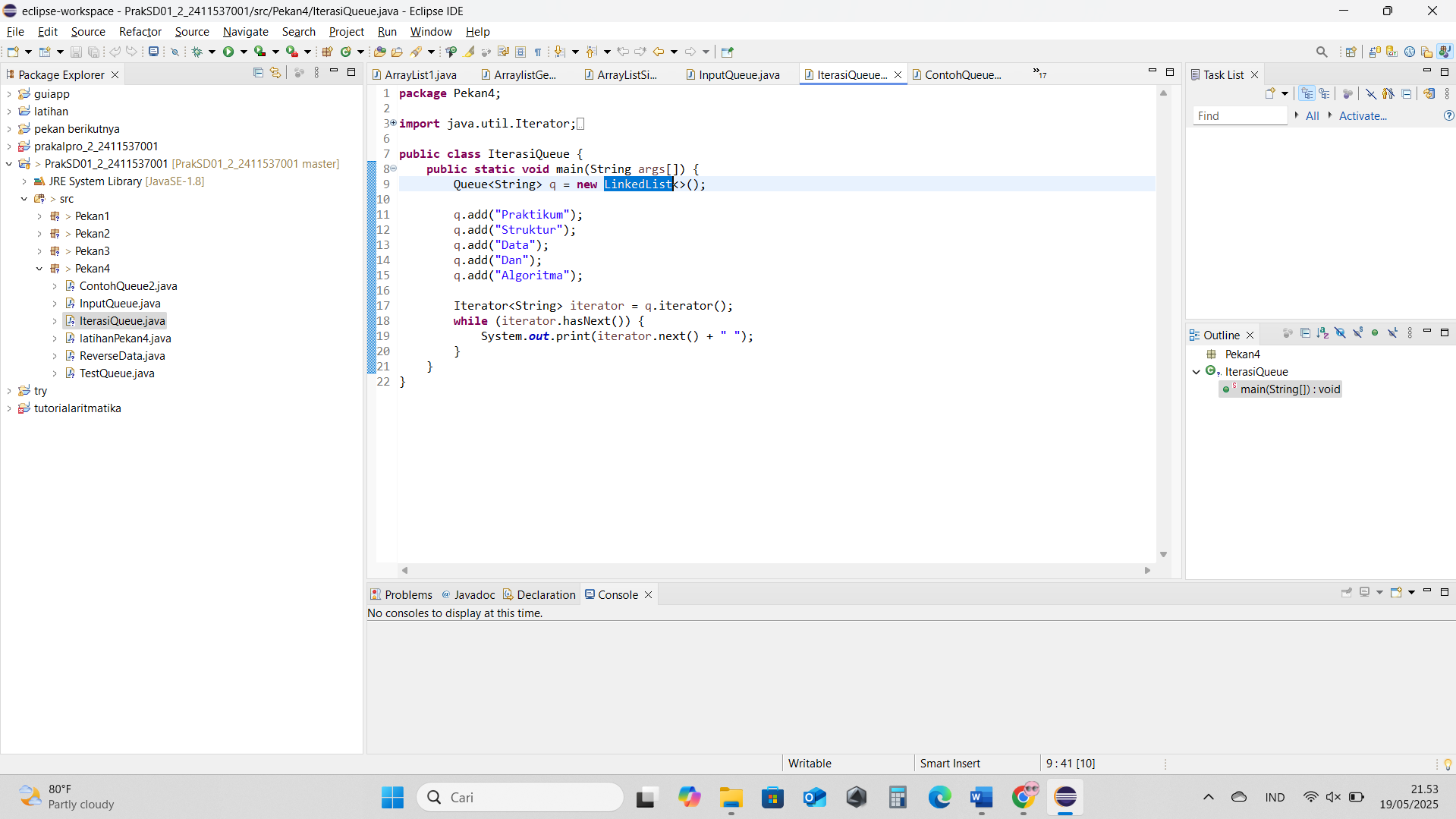
Keterangan :

1. **Deklarasi Variabel:**

* + front, rear → penunjuk indeks data pertama dan terakhir.
  + size → jumlah data saat ini dalam antrian.
  + capacity → kapasitas maksimal antrian.
  + array[] → array tempat menyimpan data.

1. **Konstruktor InputQueue(int capacity)**
   * Inisialisasi queue berdasarkan kapasitas.
   * Nilai awal: front = 0, rear = capacity - 1, size = 0.
2. **Method isFull() dan isEmpty()**
   * Mengecek apakah antrian sudah penuh (size == capacity) atau kosong (size == 0).
3. **Method enqueue(int item)**
   * Menambahkan data ke antrian.
   * rear dihitung menggunakan modulus untuk circular queue.
4. **Method dequeue()**
   * Menghapus data dari depan antrian.
   * front digeser maju (circular).
   * size dikurangi 1.
   * Jika antrian kosong, kembalikan Integer.MIN\_VALUE.
5. **Method front() dan rear()**
   * Mengembalikan nilai elemen paling depan dan paling belakang dari antrian.

* ITERASIQUEUE.JAVA



Keterangan :

1. **Import Library:**

* Queue, LinkedList, dan Iterator digunakan dari Java Collection Framework.

1. **Deklarasi Queue:**

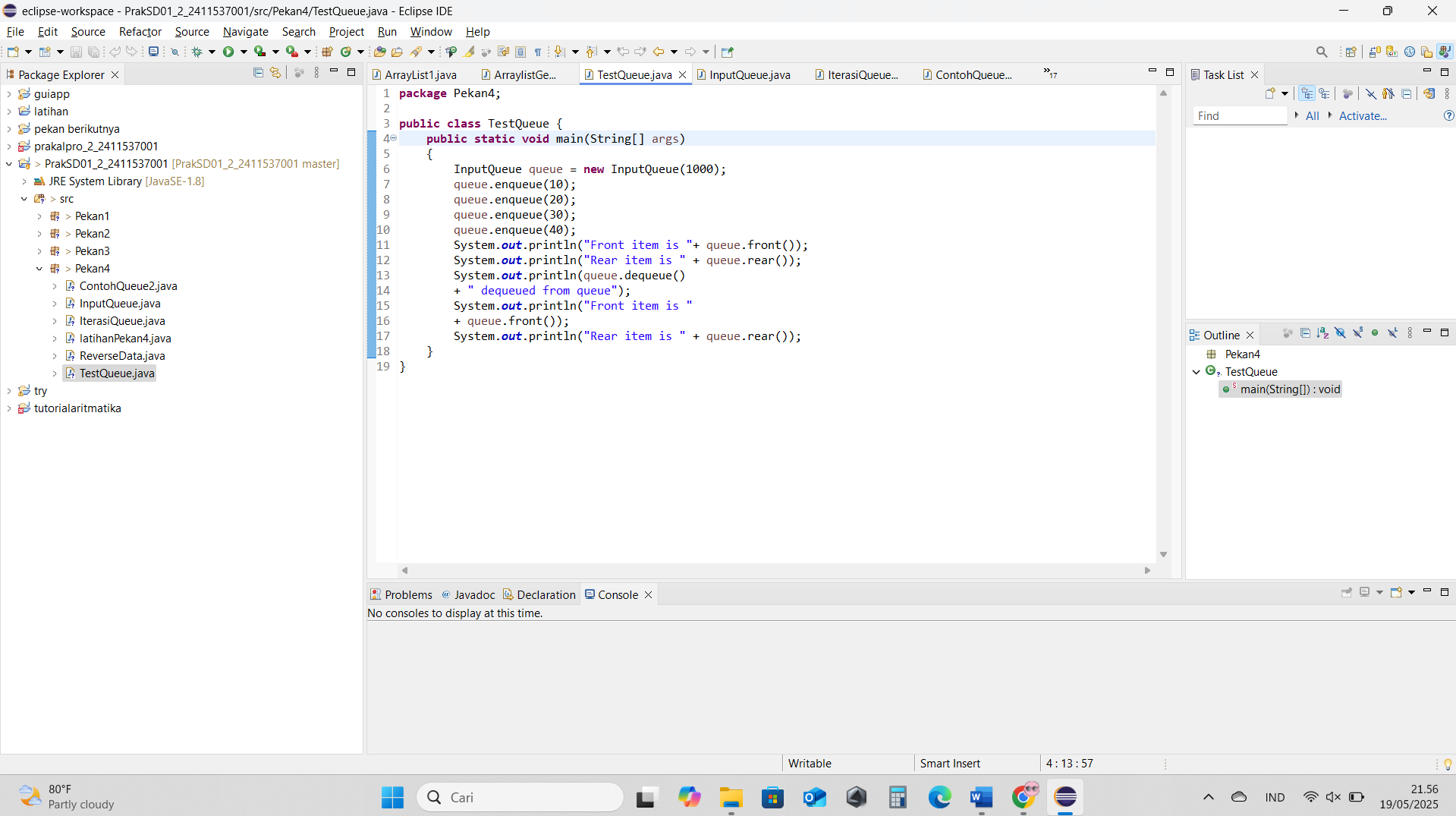
* Queue<String> q = new LinkedList<>(); → membuat objek Queue bertipe String.

1. **Penambahan Elemen ke Antrian:**

* Menggunakan q.add() untuk menambahkan lima kata ke antrian secara berurutan.

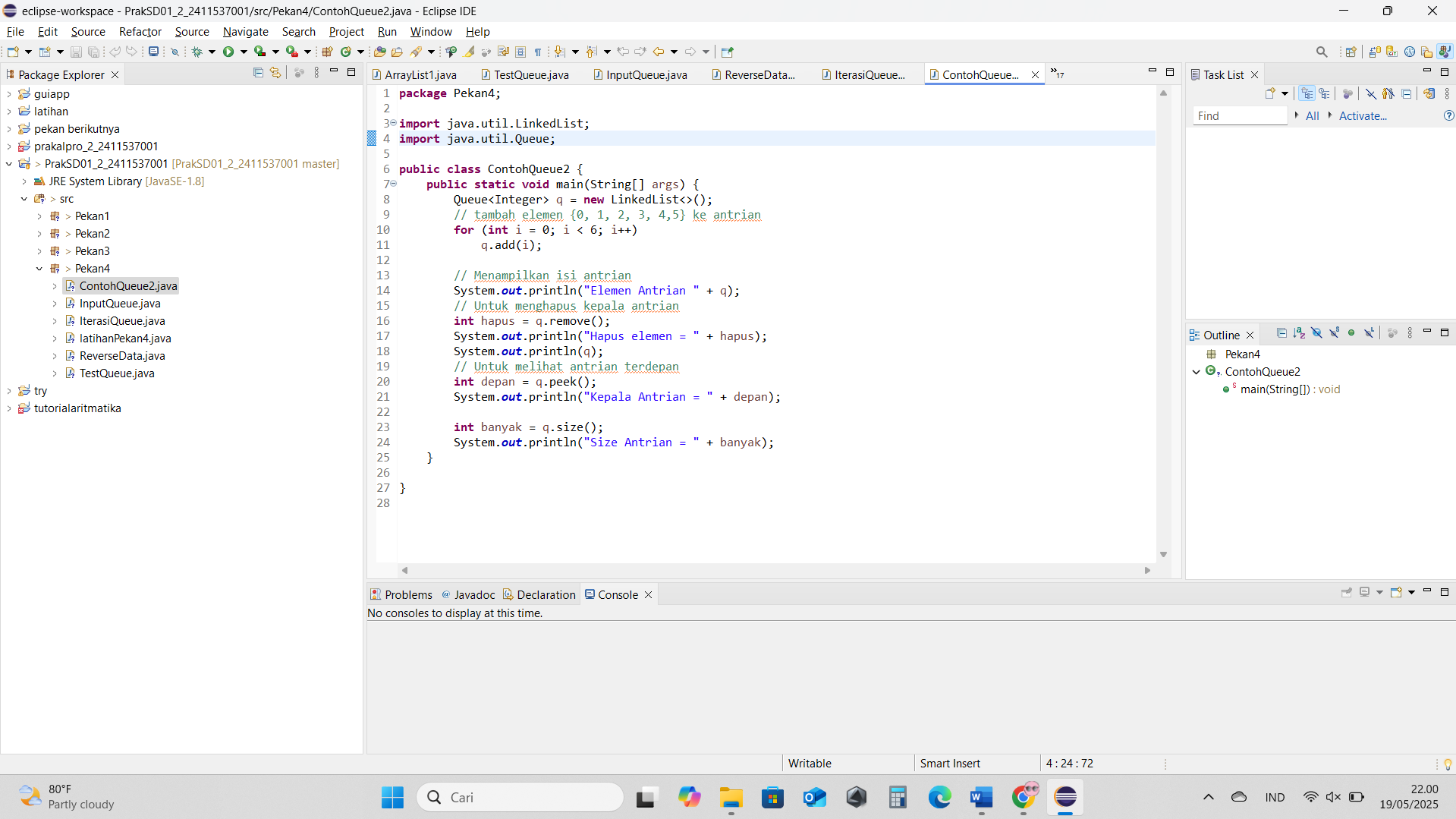
1. **Iterasi Queue:**

* Menggunakan Iterator<String> untuk membaca semua elemen dari antrian.
* while (iterator.hasNext()) → perulangan akan terus berjalan selama masih ada elemen.
* TESTQUEUE.JAVA



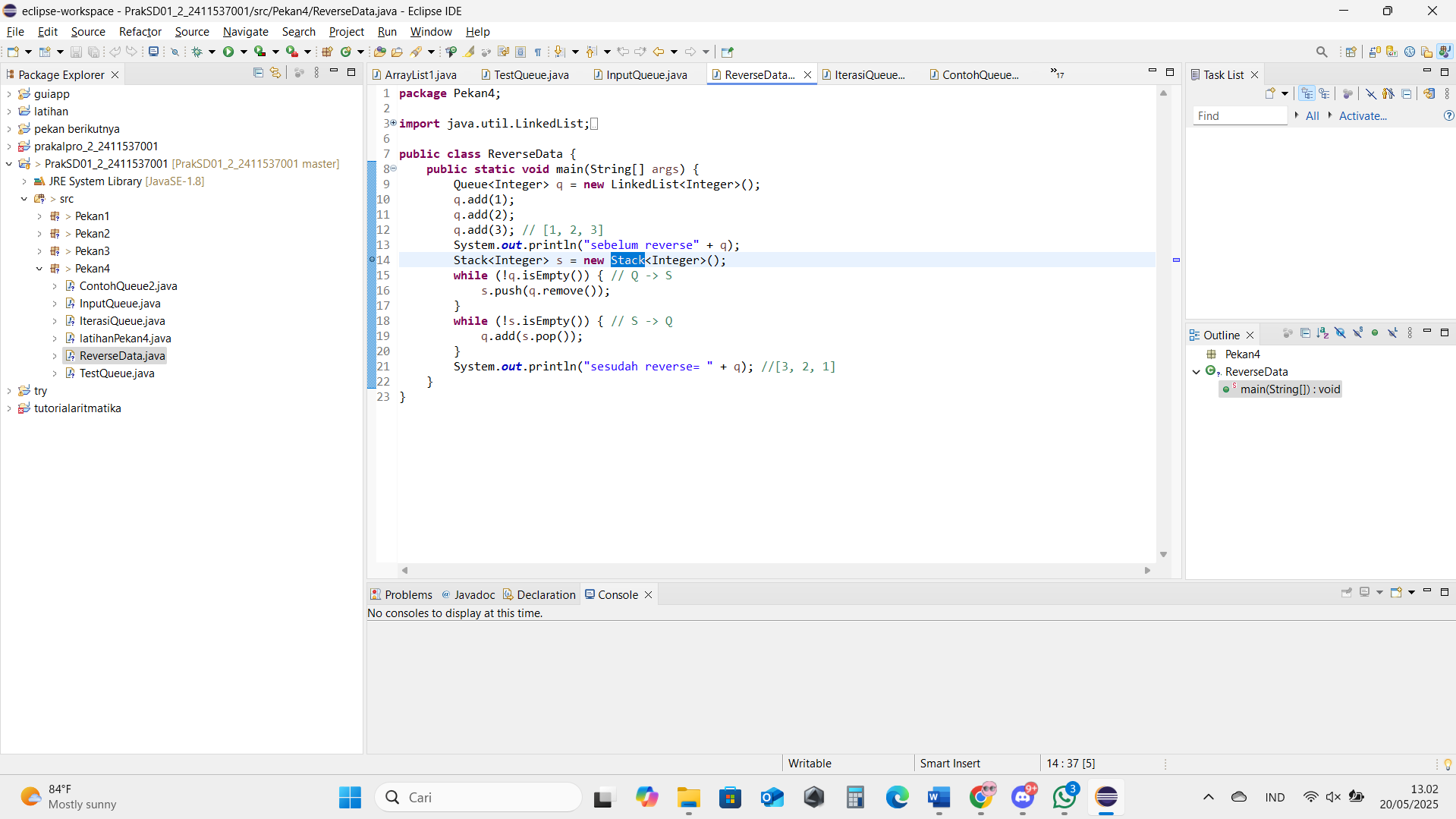
Keterangan :

* enqueue(x): Menambahkan elemen x ke belakang antrian
* dequeue(): Menghapus dan mengembalikan elemen dari depan antrian
* front(): Melihat elemen depan tanpa menghapus
* rear(): Melihat elemen belakang tanpa menghapus
* contohqueue2.java



Keterangan:

* **Operasi Antrian:**
  + **Menampilkan Elemen Antrian:** Menggunakan **System.out.println** untuk menampilkan elemen-elemen yang ada di antrian.
  + **Menghapus Elemen Terdepan:** Menghapus elemen teratas dari antrian dan menampilkannya.
  + **Melihat Kepala Antrian:** Mengambil (tanpa menghapus) elemen terdepan dan menampilkannya.
  + **Menghitung Size Antrian:** Menggunakan method **size()** untuk mendapatkan jumlah elemen dalam antrian dan menampilkannya.
* **Output yang Dihasilkan:**
  + Elemen antrian saat ini.
  + Elemen yang dihapus.
  + Kepala antrian.
  + Jumlah elemen dalam antrian.
* REVERSEDATA.JAVA



Keterangan:

1. **Input**: Antrian dengan elemen [1, 2, 3].
2. **Proses**: Menggunakan stack untuk membalik urutan elemen.
3. **Output**: Antrian terbalik menjadi [3, 2, 1].
4. **KESIMPULAN**

Melalui praktikum ini, mahasiswa dapat memahami cara kerja dan implementasi **struktur data Queue dan proses Reverse Data** menggunakan bahasa Java di Eclipse. Queue memungkinkan pemrosesan data secara berurutan sesuai urutan masuk, sedangkan reverse memungkinkan pembalikan data untuk kebutuhan tertentu. Java menyediakan kemudahan dalam pengelolaan struktur data melalui kelas-kelas bawaan seperti Queue, LinkedList, dan Stack. Eclipse sebagai IDE membantu mempercepat pengembangan dan pengujian program secara efisien. Dengan menguasai materi ini, mahasiswa memiliki dasar kuat untuk membangun aplikasi yang membutuhkan pengolahan data secara sistematis dan terstruktur.