LAPORAN PRAKTIKUM STRUKTUR DATA



OLEH :

ALIFFIA HUMAIRAH

NIM : 2311531004

DOSEN PENGAMPU :

DR. WAHYUDI,S.T,MT.

DEPARTEMEN INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS ANDALAS

PADANG

1. TUJUAN
2. Mahasiswa mampu menjelaskan pengertian queue dan dequeue
3. Mahasiwa mampu menjelaskan dan menunjukkan cara pembuatan queue, operasi push dan pop pada array.
4. Mahasiswa mampu menjelaskan dan menunjukkan program dengan ADT (Abstract Data Type) queue dan dequeue dengan array
5. KAJIAN TEORI

Queue atau antrian adalah suatu kumpulan data yang penambahan elemennya hanya bisa dilakukan pada suatu ujung(disebut dengan sisi belakang atau rear), dan penghapusan atau pengambilan elemen dilakukan lewat ujung yang lain (disebut dengan sisi depan atau front).

Kalau tumpukan (Stack) dikenal dengan menggunakan prinsip LIFO (Last In First Out), maka pada antrian prinsip yang digunakan adalah FIFO (First In First Out).

**Fungsi Queue**

Queue memiliki peran penting dalam berbagai aplikasi dan algoritma. Salah satu fungsi utamanya adalah mengatur dan mengelola antrean tugas atau operasi secara efisien. Dalam sistem komputasi, ia digunakan untuk menangani tugas-tugas seperti penjadwalan proses, antrean pesan, dan manajemen sumber daya.

**Jenis-Jenis Queue**

1. **Berdasarkan Implementasinya**

* Linear/Simple Queue : elemen-elemen data disusun dalam barisan linear dan penambahan serta penghapusan elemen hanya terjadi pada dua ujung barisan tersebut.
* Circular Queue : mirip dengan jenis linear, tetapi ujung-ujung barisan terhubung satu sama lain, menciptakan struktur antean yang berputar.

1. **Berdasarkan Penggunaan**

* Priority Queue : setiap elemen memiliki prioritas tersebut. Elemen dengan prioritas tertinggi akan diambil terlebih dahulu.
* Double-ended Queue (Dequeue) : elemen dapat ditambahkan atau dihapuskan dari kedua ujung antrean.

**Keuntungan dan Keterbatasan**

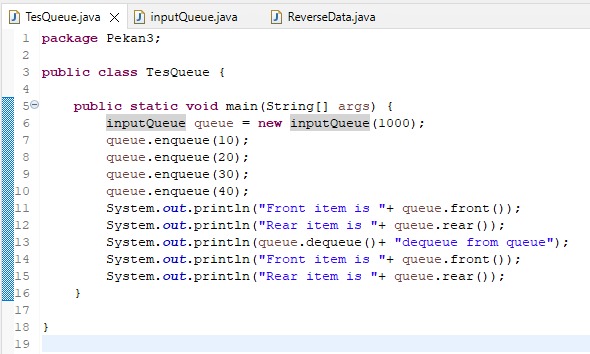
**Keuntungan:**

* data berjumlah besar dapat dikelola dengan mudah dan efisien.
* Proses insert dan delete data dapat dilakukan dengan mudah karena mengikuti aturan FIFO
* Efisien dalam menangani tugas berdasarkan ururtan kedatangan

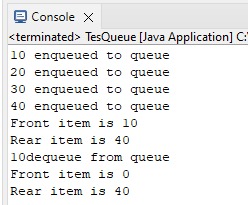
**Keterbatasan:**

* Tidak efisien untuk pencarian elemen tertentu dalam antrean
* Memerlukan alokasi memori yang cukup untuk menyimpan antrean

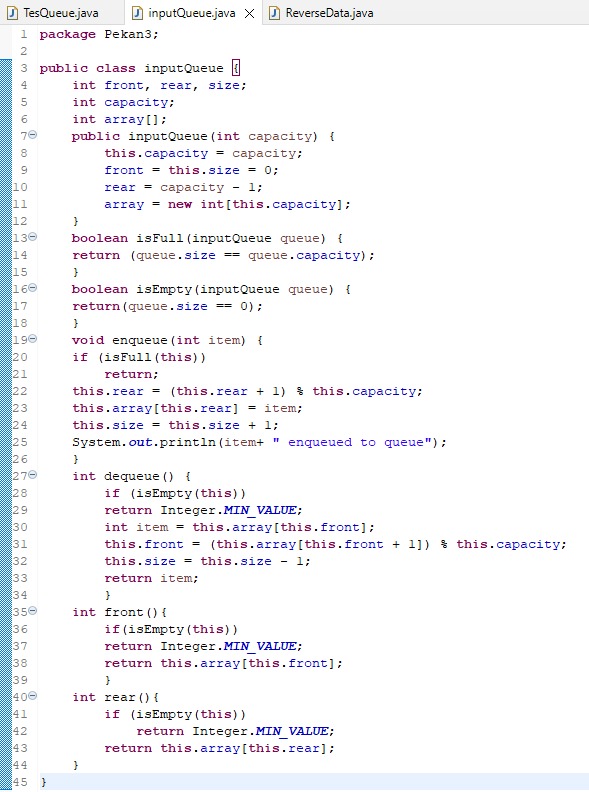
1. LANGKAH KERJA
2. Input untuk class TesQueue



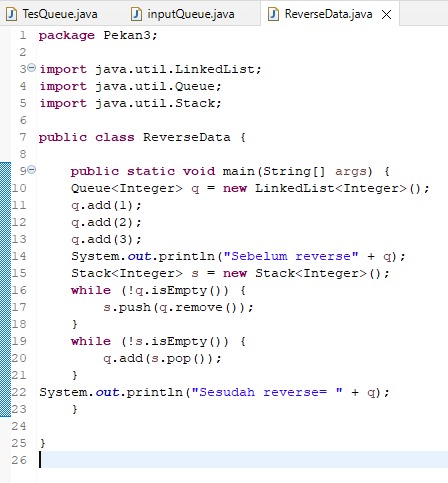
Output:



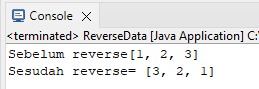
1. Input untuk class inputQueue



1. Input untuk class ReserveData



Outputnya:



1. KESIMPULAN

Pemahaman tentang struktur data queue menajdi kunci untuk mengoptimalkan efisien dan kinerja aplikasi. Struktur data yang mengatur elemen-elemen data dengan prinsip ”First In, First Out”, serupa dengan antrean di kehidupan sehari-hari.