**LAPORAN PRAKTIKUM 4**

**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**

1. A screenshot of a computer

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

   AI-generated content may be incorrect.**Class TestQueue**

Program di atas merupakan implementasi dari struktur data queue menggunakan array dalam bahasa Java. Kode ini terdiri atas dua kelas, yaitu TestQueue sebagai kelas utama untuk menguji fungsi-fungsi queue, dan InputQueue yang merepresentasikan struktur queue itu sendiri. Pada awalnya, objek InputQueue dibuat dengan kapasitas maksimum 1000 elemen. Selanjutnya, dilakukan operasi enqueue untuk menambahkan empat elemen ke dalam queue secara berurutan, yaitu 10, 20, 30, dan 40. Setiap kali elemen berhasil ditambahkan, program mencetak pesan bahwa elemen tersebut telah dimasukkan ke dalam queue.

Setelah proses pengisian, program mencetak elemen paling depan (front) dan paling belakang (rear) dari queue menggunakan metode front() dan rear(). Hasil yang ditampilkan adalah Front item is 10 dan Rear item is 40, yang menunjukkan bahwa elemen 10 berada di depan, sementara elemen 40 berada di belakang. Kemudian, dilakukan operasi dequeue untuk menghapus satu elemen dari depan queue. Hasilnya, elemen 10 dikeluarkan, ditampilkan dalam output sebagai 10 dequeued from queue. Setelah penghapusan ini, program kembali mencetak elemen depan dan belakang. Kali ini, elemen depan berubah menjadi 20, sedangkan elemen belakang tetap 40.

Program juga dilengkapi dengan metode pengecekan apakah queue dalam keadaan penuh (isFull()) atau kosong (isEmpty()), dan menangani kondisi ekstrem dengan mengembalikan nilai Integer.MIN\_VALUE bila operasi dilakukan pada queue yang kosong. Dengan demikian, program ini menunjukkan bagaimana operasi dasar queue seperti enqueue, dequeue, dan pemeriksaan posisi front dan rear dapat diterapkan secara efisien menggunakan array.

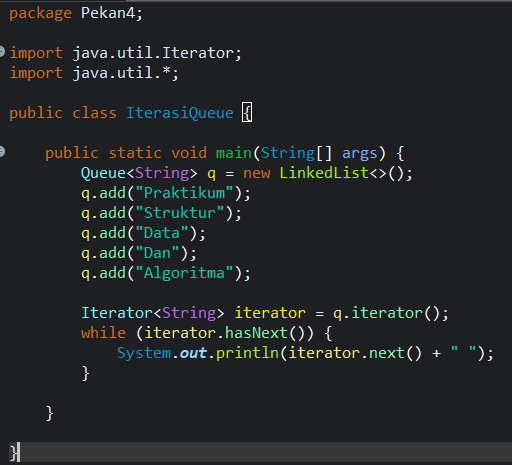
1. **A screenshot of a computer

   AI-generated content may be incorrect.Class ReverseData**

Program ini menunjukkan cara membalik urutan data (reverse) dari struktur queue dengan bantuan struktur stack. Pada awal program, objek Queue bertipe Integer dibuat menggunakan implementasi LinkedList. Tiga elemen yaitu 1, 2, dan 3 ditambahkan ke dalam queue menggunakan metode add(). Setelah itu, isi queue ditampilkan ke layar dengan hasil: sebelum reverse [1, 2, 3], yang menunjukkan bahwa data dalam queue masih berurutan dari elemen paling awal ditambahkan hingga paling akhir.

Program membuat objek Stack bertipe Integer, lalu menggunakan perulangan while untuk memindahkan seluruh elemen dari queue ke stack satu per satu menggunakan metode remove() dan push(). Proses ini menyebabkan elemen yang paling awal masuk ke queue menjadi yang paling akhir berada di stack.

Setelah seluruh elemen berada di dalam stack, dilakukan perulangan kedua untuk mengeluarkan elemen dari stack dan memasukkannya kembali ke queue menggunakan metode pop() dan add(). Karena stack mengeluarkan elemen dari yang terakhir masuk, maka saat data dimasukkan kembali ke queue, urutannya sudah menjadi terbalik. Hasil akhir ditampilkan dengan output: sesudah reverse = [3, 2, 1].

1. **Class IterasiQueue**

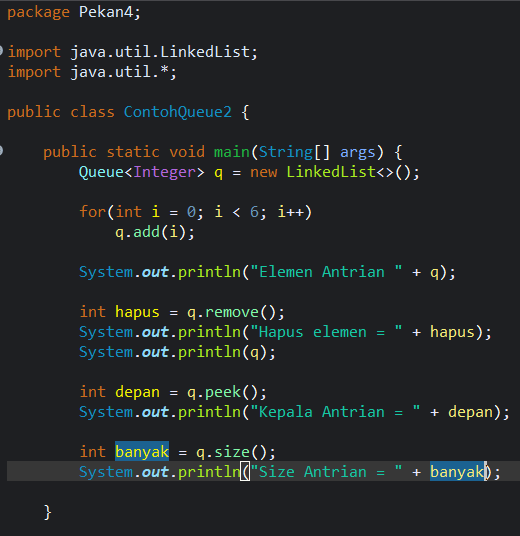
A screen shot of a computer

AI-generated content may be incorrect.

Program ini merupakan contoh implementasi iterasi pada struktur data queue menggunakan iterator. Di dalam metode main, program mendeklarasikan dan menginisialisasi objek Queue bertipe String dengan implementasi LinkedList. Queue tersebut kemudian diisi dengan lima elemen string, yaitu "Praktikum", "Struktur", "Data", "Dan", dan "Algoritma" menggunakan metode add().

Setelah data dimasukkan, program menggunakan objek Iterator<String> untuk melakukan iterasi terhadap elemen-elemen dalam queue. Iterator diperoleh melalui metode iterator() yang tersedia pada interface Queue. Proses iterasi dilakukan dengan menggunakan perulangan while yang akan terus berjalan selama masih ada elemen yang bisa diakses (hasNext() bernilai true). Di setiap iterasi, elemen saat ini diambil dengan next() dan langsung dicetak ke layar.

1. A screenshot of a computer

   AI-generated content may be incorrect.**Class ContohQueue2**

Program ini merupakan ilustrasi dasar penggunaan struktur data queue di Java dengan memanfaatkan implementasi LinkedList. Pada awal program, objek Queue bertipe Integer dibuat, kemudian diisi dengan enam buah elemen bilangan bulat dari 0 hingga 5 melalui perulangan for dan metode add(). Setelah semua elemen ditambahkan, isi queue ditampilkan ke layar melalui perintah System.out.println("Elemen Antrian " + q);, menghasilkan output: Elemen Antrian [0, 1, 2, 3, 4, 5].

Langkah berikutnya adalah menghapus satu elemen dari antrian menggunakan metode remove(), yang secara otomatis menghapus elemen paling depan dari queue. Dalam hal ini, elemen 0 dihapus, dan hasil penghapusan ditampilkan sebagai: Hapus elemen = 0. Isi queue setelah penghapusan ditampilkan kembali, menunjukkan hasil: [1, 2, 3, 4, 5].

Selanjutnya, program menggunakan metode peek() untuk mengakses elemen yang berada di posisi paling depan (tanpa menghapusnya), yang dalam hal ini adalah angka 1. Hasilnya ditampilkan sebagai Kepala Antrian = 1. Terakhir, metode size() digunakan untuk mengetahui jumlah elemen saat ini dalam antrian, yang setelah satu elemen dihapus, menghasilkan Size Antrian = 5.