**LAPORAN**

**PRAKTIKUM STRUKTUR DATA**

**Pertemuan ke – 7**

****

Disusun Oleh :

Rendra Eka Herlambang

175410028

**LABORATORIUM TERPADU**

**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER AKAKOM YOGYAKARTA**

**2018**

**ANTRIAN/QUEUE**

1. **TUJUAN**

-Mahasiswa dapat membuat program dengan antrian

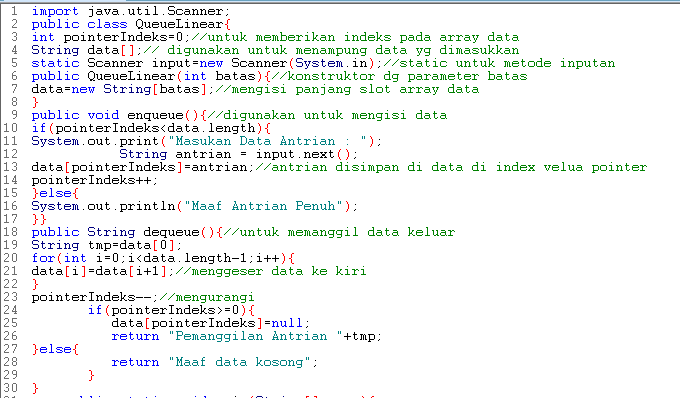
- Mahasiswa mampu mengimplementasikan antrian

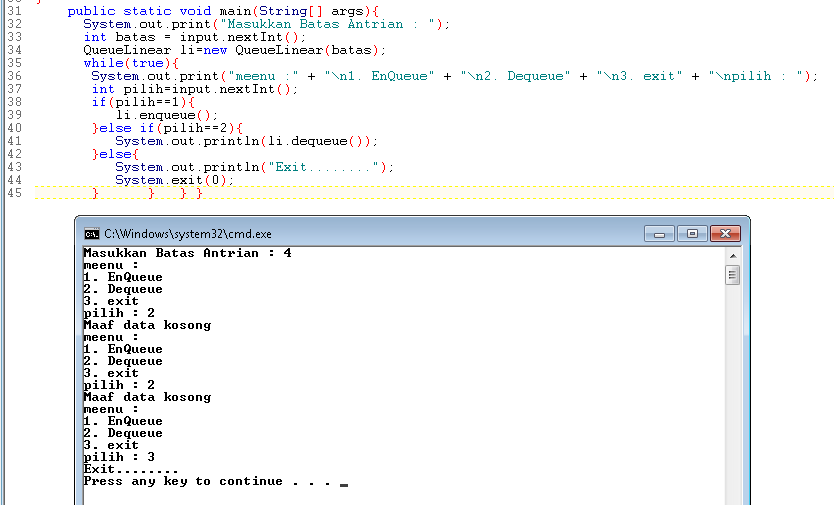
- Mahasiswa mampu mengimplemetasikan bentuk data dengan menggunakan java.

-Mahasiswa mampu mengiplementasikan dengan queque dan dequeue

1. **PEMBAHASAN PRAKTIK**

Praktik





Pembahasan

import java.util.Scanner; // yaitu untuk membuat program java.

public class QueueLinear{ yaitu nama class QueueLinear.

int pointerIndeks=0;//untuk memberikan indeks pada array data

String data[];// digunakan untuk menampung data yg dimasukkan

static Scanner input=new Scanner(System.in);//static untuk metode inputan

public QueueLinear(int batas){//konstruktor dg parameter batas

data=new String[batas];//mengisi panjang slot array data

}

public void enqueue(){//digunakan untuk mengisi data

if(pointerIndeks<data.length){// yaitu jika pointer < banyak datanya.

System.out.print("Masukan Data Antrian : "); //yaitu keluaran masukan data antrian.

String antrian = input.next();

data[pointerIndeks]=antrian;//antrian disimpan di data di index velua pointer

pointerIndeks++; //yaitu maka pointer indeks akan nambah 1 secara otomatis.

}else{

System.out.println("Maaf Antrian Penuh");

}}

//yaitu jika data penuh atau melebihi pointer datanya.

public String dequeue(){//untuk memanggil data keluar

String tmp=data[0];

for(int i=0;i<data.length-1;i++){ //yaitu untuk perulangan jika data i < banyak datanya atau melebihi tempatnya -1 maka akan otomatis nambah 1.

data[i]=data[i+1];//menggeser data ke kiri

}

pointerIndeks--;//mengurangi

if(pointerIndeks>=0){

data[pointerIndeks]=null;

return "Pemanggilan Antrian "+tmp;

//yaitu untuk mengurangi jika data index>=0 maka data yang di ambil sesuai tempatnya dan yang paling atas sendiri.

}else{

return "Maaf data kosong";// yaitu jika sampai data kosong

}

}

public static void main(String[] args){ yaitu untuk method untuk memasukan batas antrian dan

System.out.print("Masukkan Batas Antrian : ");

int batas = input.nextInt(); yaitu untuk inputkan data.

QueueLinear li=new QueueLinear(batas);

while(true){ //yaitu untuk melakukan perulangan bila true

System.out.print("meenu :" + "\n1. EnQueue" + "\n2. Dequeue" + "\n3. exit" + "\npilih : ");

int pilih=input.nextInt(); yaitu untuk pilihan.

if(pilih==1){

li.enqueue();

}else if(pilih==2){

System.out.println(li.dequeue());

}else{

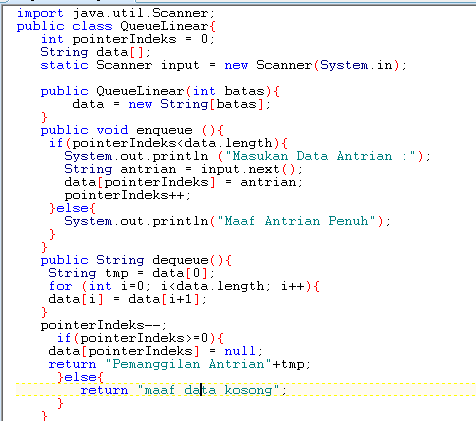
System.out.println("Exit........");

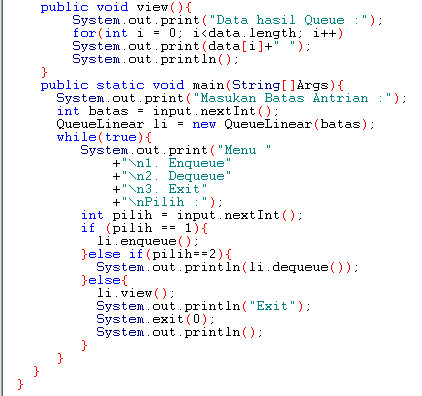
System.exit(0);

} } } }

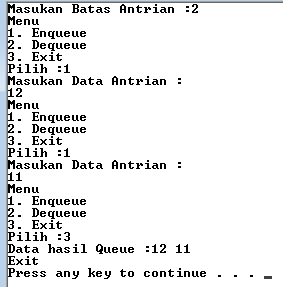
//yaitu untuk piliah 1. Enqueue 2. Dequeue dan exit maka keluar.

memodifikasi



1. 

Output :



Pembahasan

import java.util.Scanner;

public class QueueLinear{

// yaitu untuk membuat Scanner dengan nama class QueueLinear

int pointerIndeks = 0;

String data[];

//yaitu untuk pointerindex awal =0

static Scanner input = new Scanner(System.in);

public QueueLinear(int batas){ // yaitu method untuk batas

data = new String[batas];

}

public void enqueue (){ //yaitu method untuk enqueue masukan data.

if(pointerIndeks<data.length){ // yaitu jika data pointer indeks < banyak datanya.

System.out.println ("Masukan Data Antrian :");

String antrian = input.next();

//yaitu untuk memasukan data yang dari keyboard.

data[pointerIndeks] = antrian;

pointerIndeks++;

//yaitu untuk pointer dataindex,maka akan nambah 1 secara otomatis.

}else{

System.out.println("Maaf Antrian Penuh");

}

//yaitu jika data penuh maka memberi tahu bahwa data penuh.

}

public String dequeue(){// yaitu untuk method keluaran dequeue atau keluaran

String tmp = data[0];

for (int i=0; i<data.length; i++){

data[i] = data[i+1];

//yaitu mulai dari index 0 maka jika data i< banyak data maka i nambah 1 secara otomatis.

}

pointerIndeks--;

if(pointerIndeks>=0){

data[pointerIndeks] = null;

return "Pemanggilan Antrian"+tmp;

}else{

return "maaf data kosong";

// yaitu jika pointer index >=0 maka melakukan pemanggilan dan hasil data kosong.

}

}

public void view(){

System.out.print("Data hasil Queue :");

for(int i = 0; i<data.length; i++)

System.out.print(data[i]+" ");

System.out.println();

//yaitu method untuk view yaitu hasil data yang di masukan maka akan menampilkan data yang telah kita masukan.

}

public static void main(String[]Args){

System.out.print("Masukan Batas Antrian :");

int batas = input.nextInt();

QueueLinear li = new QueueLinear(batas);

//yaitu method untuk memasukan batas antrian

while(true){

System.out.print("Menu "

+"\n1. Enqueue"

+"\n2. Dequeue"

+"\n3. Exit"

+"\nPilih :");

int pilih = input.nextInt();

// maka hasil keluaran pada menu pilihan

if (pilih == 1){

li.enqueue();

//maka jika pilih 1 maka enqueue

}else if(pilih==2){

System.out.println(li.dequeue());

//yaitu maka jika pilih 2 maka dequeue

}else{

li.view();

System.out.println("Exit");

System.exit(0);

System.out.println();

//yaitu jika tidak maka data view dan akan melakukan Exit

}

}

}

}

Kesimpulan  
kita dapat menyimpulkan bahwa queue atau antrian yaitu sebuah jalur tunggu, queue yaitu data yang di masukan sedangkan dequeue,bersifat FIFO(First In Firs Out).queue sebuah list di mana semua penambahan elemen di buat di ujung belakang dan penghapusan elemen list di buat di ujung depan.