|  |  |
| --- | --- |
| **C:\Users\USER\AppData\Local\Microsoft\Windows\INetCache\Content.Word\logohitamputih.jpg** | **FAKULTAS TEKNIK**  PROGRAM STUDI TEKNIK INFORMATIKA  UNIVERSITAS PGRI YOGYAKARTA |

|  |  |
| --- | --- |
| Mata Ujian : Struktur Data  Waktu : 25-Mei-2017 23:59  Sifat : Non Tatap Muka / Tugas T08  Dosen : Setia Wardani., M.Kom | Nama : Gagas Novregi Nandiwardhana  Kelas : TI – 16A2  NPM : 16 1111 000 62 |

1. Perbedaan antara Stack(tumpukan) dengan Queue(antrian)

Stack memakai sistem LIFO (last in first out) dengan kosep yang pertama masuk akan keluar terakhir, begitu pula yang terakhir masuk akan keluar pertama kali, sedangkan Queue memakai sistem FIFO (first in first out) dengan konsep yang pertama masuk akan keluar pertama, begitu pula yang masuk terakhir akan keluar terakhir

1. Operasi yang digunakan pada antrian
2. Create()

Untuk menciptakan dan menginisialisasi Queue dengan cara membuat Head dan Tail = -1

1. IsEmpty()

Untuk memeriksa apakah Antrian sudah penuh atau belum dengan cara memeriksa nilai Tail, jika Tail = -1 maka empty. Kita tidak memeriksa Head, karena Head adalah tanda untuk kepala antrian (elemen pertama dalam antrian) yang tidak akan berubah-ubah. Pergerakan pada Antrian terjadi dengan penambahan elemen Antrian kebelakang, yaitu menggunakan nilai Tail.

1. IsFull

Untuk mengecek apakah Antrian sudah penuh atau belum dengan cara mengecek nilai Tail, jika Tail >= MAX-1 (karena MAX-1 adalah batas elemen array pada C) berarti sudah penuh

1. Enqueue

Untuk menambahkan elemen ke dalam Antrian, penambahan elemen selalu ditambahkan di elemen paling belakang. Penambahan elemen selalu menggerakan variabel Tail dengan cara increment counter Tail terlebih dahulu

1. Dequeue()

Digunakan untuk menghapus elemen terdepan/pertama (head) dari Antrian dengan cara menggeser semua elemen antrian kedepan dan mengurangi Tail dgn 1. Penggeseran dilakukan dengan menggunakan looping.

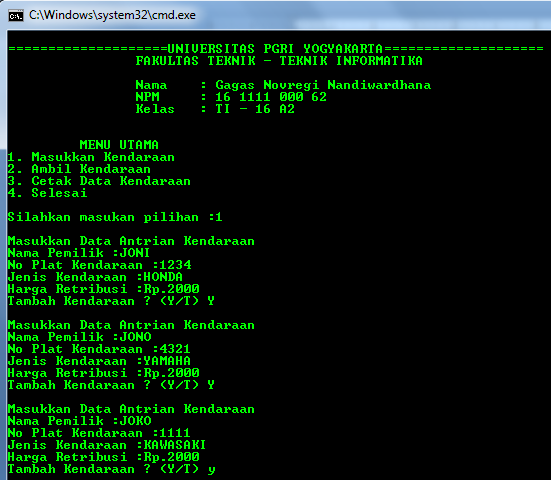
1. Clear()

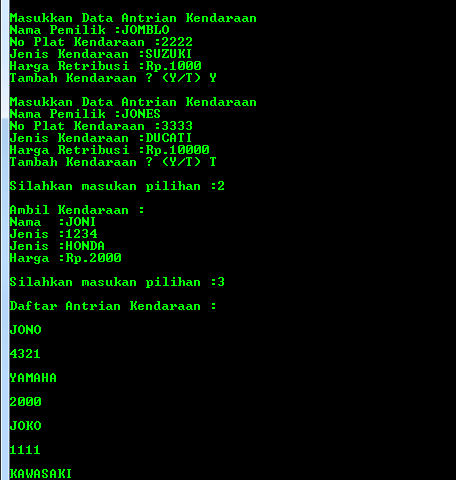
Untuk menghapus elemen-elemen Antrian dengan cara membuat Tail dan Head = -1. Penghapusan elemen-elemen Antrian sebenarnya tidak menghapus arraynya, namun hanya mengeset indeks pengaksesan-nya ke nilai -1 sehingga elemen-elemen Antrian tidak lagi terbaca

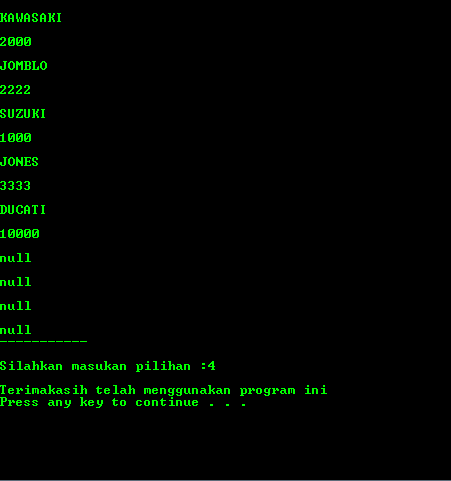
1. Tampil()

Untuk menampilkan nilai-nilai elemen Antrian menggunakan looping dari head s/d tail

1. Fungsi Private pada program
2. private int ukuran; adalah membuat variable dengan tipe data integer dengan nama ukuran dengan hak akses terhadap variable tersebut hanya dapat diakses oleh internal class queue..
3. private long[] antrian; adalah membuat variable array dengan tipe data long dengan nama antrian dengan hak akses terhadap variable tersebut hanya dapat diakses oleh internal class queue.
4. public Antrian(int s) adalah variable integer s didalam nama antrian dan dapat diakses oleh semua class dan sub class dalam queue.
5. Output Program







1. Program Java

import java.io.\*;

import java.util.\*;

class queue

{

private static int maxSize=30;

static private int pilih;

static private String nama;

static private String b;

static private String noplat;

static private String jenis;

static private String harga;

static private String[] Antrian;

private static int top;

static int jumItem;

private int belakang;

public static void Stack(int max)

{

maxSize=max;

Antrian = new String[maxSize];

top=-1;

}

public static void push(String j)

{

Antrian[++top]=j;

jumItem++;

}

public static String keluar()

{

String temp = Antrian[0];

for (int i=0;i<jumItem;i++)

Antrian[i]=Antrian [i+1];

jumItem--;

return temp;

}

public static String peek()

{

return Antrian[top];

}

public static boolean isEmpty()

{

return (top==-1);

}

static void baca ()

{

int i;

System.out.println("\nDaftar Antrian Kendaraan : ");

if(top>-1)

{

for(i=0;i<=top;i++)

System.out.println("\n"+Antrian[i]);

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

}

else System.out.println("Antrian Kendaraan Kosong!!");

}

public static void main (String[]args)

{

Scanner input = new Scanner(System.in);

BufferedReader read = new BufferedReader(new InputStreamReader(System.in));

BufferedReader data = new BufferedReader(new InputStreamReader(System.in));

Stack(30);

System.out.println("\n====================UNIVERSITAS PGRI YOGYAKARTA====================");

System.out.println(" FAKULTAS TEKNIK - TEKNIK INFORMATIKA");

System.out.println("\n Nama : Gagas Novregi Nandiwardhana");

System.out.println(" NPM : 16 1111 000 62");

System.out.println(" Kelas : TI - 16 A2");

System.out.println("\n");

System.out.println("\t MENU UTAMA");

System.out.println("1. Masukkan Kendaraan");

System.out.println("2. Ambil Kendaraan");

System.out.println("3. Cetak Data Kendaraan");

System.out.println("4. Selesai");

do{

System.out.print("\nSilahkan masukan pilihan :");

try

{

pilih=Integer.parseInt(read.readLine());

}

catch(IOException e){}

switch(pilih)

{

case 1:

for(String ulang = "y"; ulang.equals("y") || ulang.equals("Y");){

System.out.println("\nMasukkan Data Antrian Kendaraan");

System.out.print("Nama Pemilik :");

try

{

nama=data.readLine();

}

catch(IOException e){}

push(nama);

System.out.print("No Plat Kendaraan :");

try

{

noplat=data.readLine();

}

catch(IOException e){}

push(noplat);

System.out.print("Jenis Kendaraan :");

try

{

jenis=data.readLine();

}

catch(IOException e){}

push(jenis);

System.out.print("Harga Retribusi :Rp.");

try

{

harga=data.readLine();

}

catch(IOException e){}

push(harga);

System.out.print("Tambah Kendaraan ? (Y/T) ");

ulang = input.next();}

break;

case 2:

nama=keluar();

noplat=keluar();

jenis=keluar();

harga=keluar();

System.out.println("\nAmbil Kendaraan :");

System.out.println("Nama :"+nama);

System.out.println("Jenis :"+noplat);

System.out.println("Jenis :"+jenis);

System.out.println("Harga :Rp."+harga);

break;

case 3:

baca();

break;

case 4:

System.out.println("\nTerimakasih telah menggunakan program ini");System.exit(0);

break;

default : System.out.println("tidak ada pilihan , silahkan coba lagi");

break;

}

}

while (pilih!=0);

}

}