Nama : Farid Aziz Wicaksono

Kelas : TI/1C

Absen : 14

1. PRAKTIKUM

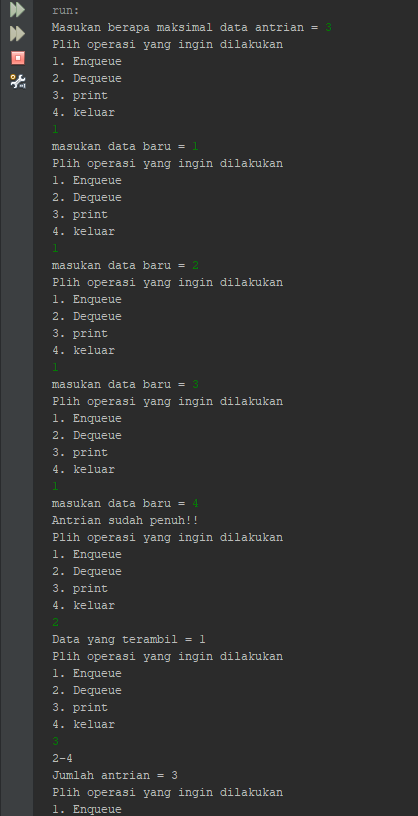
Praktikum1

|  |  |
| --- | --- |
| No | Queue.java |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85 | package minggu8;  public class Queue {  int max, size, front, rear;  int[] q;    Queue(int m){  max = m;  create();  }  void create(){  q = new int[max];  size = 0;  front = rear = -1;  }  boolean isEmpty(){  if(size == 0){  return true;  }  else  return false;  }  boolean isFull(){  if(size == max){  return true;  }  else  return false;  }  void equeue(int data){  if(isFull()){  System.out.println("Antrian sudah penuh!!");  }  else{  if(isEmpty()){  front = rear = 0;  }  else{  if(rear == max-1){  rear = 0;  }  else{  rear++;  }  }  }  q[rear] = data;  size++;  }  int dequeue(){  int data = 0;  if(isEmpty()){  System.out.println("Antrian kosng!!");  }  else{  data = q[front];  size--;  if(isEmpty()){  front = rear = -1;  }  else{  if(front == max-1){  front = 0;  }  else{  front++;  }  }  }  return data;  }  void print(){  if(isEmpty()){  System.out.print("Antrian kosong!!");  }  else{  int i = front;  while(i!= rear){  System.out.print(q[i]+"-");  i = (i+1)%max;  }  System.out.println(q[i]+" ");  System.out.println("Jumlah antrian = "+size);  }  }  } |

Praktikum 2

|  |  |
| --- | --- |
| No | QueueMain.java |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37 | package minggu8;  import java.util.Scanner;  public class QueueMain {  static void menu(){  System.out.println("Plih operasi yang ingin dilakukan");  System.out.println("1. Enqueue");  System.out.println("2. Dequeue");  System.out.println("3. print");  System.out.println("4. keluar");  }    public static void main(String[] args){  Scanner sc = new Scanner(System.in);  int pil = 0;  System.out.print("Masukan berapa maksimal data antrian = ");  int m = sc.nextInt();  Queue qobj = new Queue(m);  do{  menu();  pil = sc.nextInt();  switch(pil){  case 1: System.out.print("masukan data baru = ");  int dataIn = sc.nextInt();  qobj.equeue(dataIn);  break;  case 2: int dataOut = qobj.dequeue();  if(dataOut != 0)  System.out.println("Data yang terambil = "+dataOut);  break;  case 3: qobj.print();  break;  }  }  while(pil != 4);  }  } |

Output :



1. PERTANYAAN
2. Jelaskan pengertian dari Queue!
3. Perhatikan class Queue, di dalamnya terdapat atribut q. Untuk apakah atribut tersebut?
4. Jelaskan apa kegunaan dari atribut max, size, front dan rear yang ada di dalam class Queue!
5. Perhatikan konstruktor class Queue, di dalam konstruktor tersebut terdapat statement untuk memanggil method create(). Apa tujuan pemanggilan tersebut method tersebut?
6. Perhatikan isi method create(), kenapa atribut front dan rear diinisialisasi ke -1 dan tidak ke 0?
7. Perhatikan isi dari method isEmpty() dan isFull(), kapan suatu queue dinyatakan kosong? Dan kapan pula queue dinyatakan penuh?
8. Perhatikan kembali isi dari method isFull(), jika kondisi di dalam IF diubah menjadi size==max-1, menurut Anda pengaruh apa yang akan terjadi?
9. Perhatikan method enqueue(), di dalamnya terdapat statement sbb:

if(rear==max-1){

rear = 0;

}

Untuk apakah proses tersebut?

1. Perhatikan kembali method enqueue(), stetment mana yang menunjukan bahwa data baru disimpan di dalam posisi terakhir queue?
2. Perhatikan method dequeue(), mengapa method tersebut tidak dibuat ber-tipe data void?
3. Pada method dequeue(), statement mana yang menunjukan bahwa pada proses pengambilan data , data paling depan yang terambil?
4. Pada method dequeue(),di dalamnya terdapat statement sbb:

if(front==max-1){

front = 0;

}

Untuk apakah proses tersebut?

1. Statement mana yang menunjukan bahwa ketika data terambil dari suatua queue, maka jumlah data yang ada di dalam berkurang satu?
2. Perhatikan method print(), mengapa pada awal proses perulangan, variabel i dimulai dari front (int i=front, bukan int i=0)?
3. Perhatikan method print(), untuk apakah proses i=(i+1)%max ?
4. Perhatikan method print(), kapan proses perulangan di dalam method print() akan berhenti?
5. Perhatikan method main(), statement mana saja yang menunjukan:

a. Proses pembuatan objek dari class Queue

b. Proses pemanggilan method enqueue()

c. Proses pemanggilan method dequeue

d. Proses pemanggilan method print()

Jawab :

1. **Queue (antrian) a**dalah sekumpulan elemen/data dimana proses **memasukkan/menambah** elemen/data dilakukan pada posisi **belakang** (rear) dan proses **mengeluarkan/mengambil** elemen/data dilakukan pada elemen/data di posisi **depan** (front).
2. Atribut **Q** digunakan untuk menyimpan data queue
3. **Max :** atribut/variabel yang akan digunakan untuk menyimpan banyak data

maksimal yang bisa disimpan di dalam queue

**Size :** atribut/variabel yang akan digunakan untuk menyimpan berapa

banyak data yang ada dalam antrian

**Front :** atribut/variabel yang akan digunakan untuk menyimpan nilai

indeks array, dimana data terdepan dari antrian berada

**Rear :** atribut/variabel yang akan digunakan untuk menyimpan nilai indeks

array, dimana data paling belakang dari antrian berada

1. Digunakan untuk membuat objek queue
2. Agar queue dalam keadaan kosong, jika di set menjadi 0 maka queue tersebut sudah ada isinya yaitu terdapat di index ke 0
3. Queue dinyatakan kosong ketika size atau queue == 0, dinyatakan penuh ketika size sudah dalam kondisi max (ukuran maksimal queue telah diset oleh inputan di method main)
4. Jika size == max -1 maka isi dari maksimal yang telah diinputkan user akan dikurang -1, misalkan input 3 makan maksimalnnya adalah 3-1 =2
5. Untuk mengurangi isi dari queue
6. Ada pada statemen q[rear] =data; size++;
7. Karena dequeue mempunyai nilai kembalian
8. else if (front == max - 1) {

front = 0;

} else {

front++;

}

1. Untuk proses pengurangan nilai yang paling atas
2. if(isEmpty())

front = rear = -1;

1. dimulai dari nilai yang paling awal (rear)
2. Sebagai parameter dari i yang terus bertambah 1, selama nilai i tidak sama dengan nilai rear maka proses tersebut akan berjalan
3. Ketika queue sedang dalam keadaan kosong maupun sedang terisi
   1. System.out.print(“Masukkan berapa maksimal data antrian : “)

Int m =sc.nextInt(); Queue qobj = new Queue(m);

* 1. Case 1:

System.out.print(“Masukkan data baru :”);

Int dataIn = sc.nextInt();

Qobj.enqueue(dataIn); break;

* 1. Case 2:

int dataOut = qobj.dequeue(); if(dataOut!=0)

System.out.println(“Data yang terambil : ”+dataOut);

break;

* 1. if(isEmpty()){

System.out.println("Antrian kosong!!");

}else{

int i = front;

while(i!=rear){

System.out.print(q[i]+"\_");

i = (i+1)%max;

1. TUGAS

1. Tambahkan method-method di bawah ini di dalam class Queue di atas.

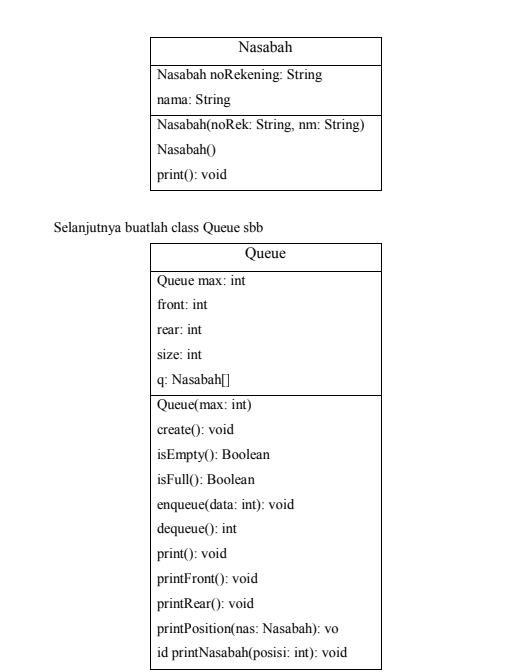
a. Method printFront() : void > untuk menampilkan data/nilai yang ada di posisi antrian paling depan

b. Method printRear() : void > untuk menampilkan data/nilai yang ada di posisi antrian paling belakang

c. Method printPosition(data: int) : void > untuk menampilkan di posisi ke berapa dalam antrian, suatu data/nilai berada

d. Method printDataByPos(position: int) : void > untuk menampilkan data/nilai yang ada di suatu posisi antrian tertentu. Kemudian lakukan penyesuaian di dalam class QueueMain untuk mensimulasikan pemanggilan method-method yang baru ditambahkan tersebut!

2. Buatlah program antrian nasabah di suatu bank. Ketika seorang nasabah akan antri, maka ia harus menuliskan terlebih dulu no. rekening, dan nama. Jadi, antrian yang akan dibuat, berisi data-data nasabah berupa no. rekening dan nama. Sehingga pertama kali yang harus dibuat adalah class Nasabah sbb:



Catatan:

* Method create(), isEmpty(), isFull(), enqueue(), dequeue() dan print(), kegunaannya sama seperti yang telah dibuat pada sesi praktikum
* Method printFront(): digunakan untuk menampilkan data Nasabah yang ada di posisi antrian paling depan
* Method printRear(): digunakan untuk menampilkan data Nasabah yang ada di posisi antrian paling belakang
* Method printPosition(): digunakan untuk menampilkan posisi antrian ke berapa, seorang Nasabah berada
* Method printNasabah(): digunakan untuk menampilkan data nasabah pada suatu posisi tertentu dalam antrian

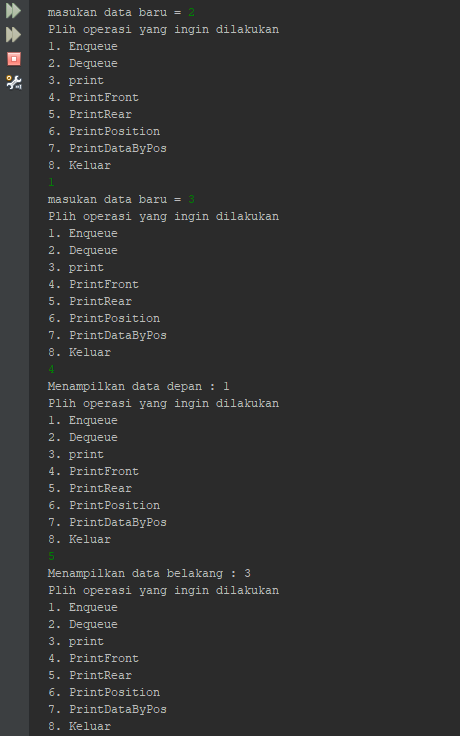
Jawab :

1. Tugas 1

|  |  |
| --- | --- |
| No | Queue.java |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  3940  41  42  43  44  45  46  47  48  49  5051  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  6869  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  9293  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145 | package minggu8;  public class Queue {  int max, size, front, rear;  int[] q;    Queue(int m){  max = m;  create();  }    void create(){  q = new int[max];  size = 0;  front = rear = -1;  }    boolean isEmpty(){  if(size == 0){  return true;  }  else  return false;  }    boolean isFull(){  if(size == max){  return true;  }  else  return false;  }    void equeue(int data){  if(isFull()){  System.out.println("Antrian sudah penuh!!");  }  else{  if(isEmpty()){  front = rear = 0;  }  else{  if(rear == max-1){  rear = 0;  }  else{  rear++;  }  }  }  q[rear] = data;  size++;  }    int dequeue(){  int data = 0;  if(isEmpty()){  System.out.println("Antrian kosng!!");  }  else{  data = q[front];  size--;  if(isEmpty()){  front = rear = -1;  }  else{  if(front == max-1){  front = 0;  }  else{  front++;  }  }  }  return data;  }    void print(){  if(isEmpty()){  System.out.print("Antrian kosong!!");  }  else{  int i = front;  while(i!= rear){  System.out.print(q[i]+"-");  i = (i+1)%max;  }  System.out.println(q[i]+" ");  System.out.println("Jumlah antrian = "+size);  }  }    void printFront() {  int elemenFront = q[front];  System.out.println("Menampilkan data depan : "+ elemenFront);  }    void printRear() {  int elemenRear = q[rear];  System.out.println("Menampilkan data belakang : " + elemenRear);  }    void printPosition(int data) {  int count = 0;  if (isEmpty()) {  System.out.println("\n Data tidak ditemukan ");  }  else {  int i = front;  while (i != rear) {  count++;  if (data == q[i]) {  System.out.println("\n Data ditemukan diIndex: " + count);  break;  }  i = (i + 1) % max;  }  count++;  if (data == q[i]) {  }  System.out.println("\n Data ditemukan di Index :" + count);  }  }    void printDataByPos(int position) {  int count = 0;  if (isEmpty()) {  System.out.println("\n Data tidak ditemukan ");  }  else{  int i = front;  while (i != rear) {  count++;  if (position == count) {  System.out.println("\n Indeks " + (i + 1)+ " : " + q[i]);  break;  }  i = (i + 1) % max;  }  count++;  System.out.println("\n Indeks " + (i + 1) + ":" + q[i]);  }  }  } |

|  |  |
| --- | --- |
| No | QueueMain.java |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52 | package minggu8;  import java.util.Scanner;  public class QueueMain {  static void menu(){  System.out.println("Plih operasi yang ingin dilakukan");  System.out.println("1. Enqueue");  System.out.println("2. Dequeue");  System.out.println("3. print");  System.out.println("4. PrintFront");  System.out.println("5. PrintRear");  System.out.println("6. PrintPosition");  System.out.println("7. PrintDataByPos");  System.out.println("8. Keluar");  }    public static void main(String[] args){  Scanner sc = new Scanner(System.in);  int pil = 0;  System.out.print("Masukan berapa maksimal data antrian = ");  int m = sc.nextInt();  Queue qobj = new Queue(m);  do{  menu();  pil = sc.nextInt();  switch(pil){  case 1: System.out.print("masukan data baru = ");  int dataIn = sc.nextInt();  qobj.equeue(dataIn);  break;  case 2: int dataOut = qobj.dequeue();  if(dataOut != 0)  System.out.println("Data yang terambil = "+dataOut);  break;  case 3: qobj.print();  break;  case 4: qobj.printFront();  break;  case 5: qobj.printRear();  break;  case 6: System.out.println("Masukan angka : ");  int data = sc.nextInt();  qobj.printPosition(data);  break;  case 7: System.out.println("Masukan Indeks : ");  int position = sc.nextInt();  qobj.printDataByPos(position);  break;  }  } while (pil != 8);  }  } |

Output :



1. Tugas 2

|  |  |
| --- | --- |
| No | Nasabah.java |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | package minggu8;  public class Nasabah {  String NoRekening, Nama;    Nasabah(String nma, String nr) {  this.NoRekening = nr;  this.Nama = nma;  }    void printNasabah() {  System.out.println(Nama + " No Rekening : " + NoRekening);  }  } |

|  |  |
| --- | --- |
| No | NasabahQueue.java |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143 | package minggu8;  public class NasabahQueue {  int max, size, front, rear;  Nasabah[] nasabahh;  int idx;    NasabahQueue(int max) {  this.max = max;  create();  }    void create() {  nasabahh = new Nasabah[max];  size = 0;  front = rear = -1;  }    boolean isEmpty() {  if (size == 0) {  return true;  }  else{  return false;  }  }    boolean isFull() {  if (size == max) {  return true;  }  else {  return false;  }  }    void enqueue(Nasabah nasabah) {  if (isFull()) {  System.out.println("Queue sudah penuh! ");  }  else {  if (isEmpty()) {  front = rear = 0;  }  else if (rear == max - 1) {  rear = 0;  }  else {  rear++;  }  nasabahh[rear] = nasabah;  size++;  }  }    Nasabah dequeue() {  Nasabah data = null;  if (isEmpty()) {  System.out.println("Queue kosong ! ");  }  else {  data = nasabahh[front];  size--;  if (isEmpty()) {  front = rear = -1;  }  else if (front == max - 1) {  front = 0;  }  else {  front++;  }  }  return data;  }    void print() {  System.out.println();  if (isEmpty()) {  System.out.println("Queue kosong! ");  }  else {  int i = front;  while (i != rear) {  nasabahh[i].printNasabah();  System.out.println(" ===== ");  i = (i + 1) % max;  }  nasabahh[i].printNasabah();  System.out.println("Queue size : " + size+ "\n");  }  }    void printFront() {  System.out.println("Data Nasabah paling depan : ");  nasabahh[front].printNasabah();  }    void printRear() {  System.out.println("Data Nasabah paling terakhir: ");  nasabahh[rear].printNasabah();  }    void printPosition(String nama) {  int hitung = 0;  if (isEmpty()) {  System.out.println("\n Queue Kosong ");  }  else {  int i = front;  while (i != rear) {  hitung++;  if (nama.equalsIgnoreCase(nasabahh[i].Nama)) {  System.out.print("\n Nasabah yang ada dalamqueue:" + hitung);  break;  }  i = (i + 1) % max;  }  System.out.println("\n Nasabah yang ada dalam queue :" + hitung);  nasabahh[i].printNasabah();  }  }    void printDataByPosition(int position) {  int hitung = 0;  if (isEmpty()) {  System.out.println("\n Queue kosong ");  }  else {  int i = front;  while (i != rear) {  hitung++;  if (position == hitung) {  System.out.println("Yang ada dalam queue "+ hitung + " : ");  nasabahh[i].printNasabah();  break;  }  i = (i + 1) % max;  }  }  }  } |

|  |  |
| --- | --- |
| No | NasabahMain.java |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38 | package minggu8;  import java.util.\*;  public class NasabahMain {  static Scanner sc = new Scanner(System.in);    static Nasabah Informasi() {  System.out.print("Nama Nasabah : ");  String nama = sc.next();  System.out.print("Bank akun nasabah : ");  String NoRekening = sc.next();  Nasabah data = new Nasabah(nama, NoRekening);  return data;  }    public static void main(String[] args) {  System.out.println("Berapa Banyak Nasabah ? ");  System.out.print("Jumlah Nasabah : ");  int banyak = sc.nextInt();  NasabahQueue list = new NasabahQueue(banyak);    for (int i = 0; i < banyak; i++) {  list.nasabahh[i] = Informasi();  list.enqueue(list.nasabahh[i]);  }    list.print();  list.printFront();  list.printRear();  System.out.println();  System.out.print("Cari nama nasabah dalam queue : ");  String nama = sc.next();  list.printPosition(nama);  System.out.print("Indeks Queue : ");  int position = sc.nextInt();  list.printDataByPosition(position);  }  } |

Output :

