

Nama : Zaskia Maulidina Mutiara Hati

NIM : 244107060056

Kelas : SIB 1A

## JOBSHEET 11 PRAKTIKUM ALGORITMA STRUKTUR DATA

### Perbacaan 1

- **Class Mahasiswa24**

```
jobsheet_11 > Mahasiswa24.java > Mahasiswa24
1  package jobsheet_11;
2
3  public class Mahasiswa24 {
4      String nim, nama, kelas;
5      double ipk;
6
7      Mahasiswa24() {
8
9      }
10
11     Mahasiswa24(String nm, String name, String kls, double ip) {
12         nim = nm;
13         nama = name;
14         kelas = kls;
15         ipk = ip;
16     }
17
18     public void tampilInformasi(){
19         System.out.printf(format:"%-10s %-15s %-5s %-4.2f\n", nama, nim, kelas, ipk);
20     }
21 }
```

- **Class NodeMahasiswa24**

```
jobsheet_11 > NodeMahasiswa24.java > NodeMahasiswa24
1  package jobsheet_11;
2
3  public class NodeMahasiswa24 {
4      Mahasiswa24 data;
5      NodeMahasiswa24 next;
6
7      public NodeMahasiswa24(Mahasiswa24 data, NodeMahasiswa24 next) {
8          this.data = data;
9          this.next = next;
10     }
11 }
```

- **Class SingleLinkedList24**

jobsheet\_11 > SingleLinkedList24.java > SingleLinkedList24

```
1  package jobsheet_11;
2
3  public class SingleLinkedList24 {
4      NodeMahasiswa24 head, tail;
5
6      boolean isEmpty() {
7          return head == null;
8      }
9
10     public void print() {
11         if (!isEmpty()) {
12             NodeMahasiswa24 tmp = head;
13             System.out.println(x:"Isi Linked List:\t");
14             while (tmp != null) {
15                 tmp.data.tampilInformasi();
16                 tmp = tmp.next;
17             }
18             System.out.println(x:"");
19         } else {
20             System.out.println(x:"Linked List Kosong");
21         }
22     }
23
24     public void addFirst(Mahasiswa24 input) {
25         NodeMahasiswa24 ndInput = new NodeMahasiswa24(input, next:null);
26         if (isEmpty()) {
27             head = ndInput;
28             tail = ndInput;
29         } else {
30             ndInput.next = head;
31             head = ndInput;
32         }
33     }
34
35     public void addLast(Mahasiswa24 input) {
36         NodeMahasiswa24 ndInput = new NodeMahasiswa24(input, next:null);
37         if (isEmpty()) {
38             head = ndInput;
39             tail = ndInput;
40         } else {
41             tail.next = ndInput;
42             tail = ndInput;
43         }
44     }
45
46     public void insertAfter(String key, Mahasiswa24 input) {
47         NodeMahasiswa24 ndInput = new NodeMahasiswa24(input, next:null);
48         NodeMahasiswa24 temp = head;
49         do {
50             if (temp.data.nama.equalsIgnoreCase(key)) {
51                 ndInput.next = temp.next;
52                 temp.next = ndInput;
53                 if (ndInput.next == null) {
54                     tail = ndInput;
55                 }
56                 break;
57             }
58             temp = temp.next;
59         } while (temp != null);
60     }
```

```

62  ✓ public void insertArt(int index, Mahasiswa24 input) {
63  ✓     if (index < 0) {
64      ✓         System.out.println(x:"indeks salah");
65  ✓     } else if (index == 0) {
66      ✓         addFirst(input);
67  ✓     } else {
68      ✓         NodeMahasiswa24 temp = head;
69  ✓         for (int i = 0; i < index -1; i++){
70      ✓             temp = temp.next;
71      ✓         }
72      ✓         temp.next = new NodeMahasiswa24(input, temp.next);
73  ✓         if (temp.next.next == null) {
74      ✓             tail = temp.next;
75      ✓         }
76      ✓     }
77  ✓ }
78  ✓ }

```

- **Class SLLMain24**

```

jobsheet_11 > SLLMain24.java > SLLMain24
1  package jobsheet_11;
2
3  public class SLLMain24 {
4
5      Run | Debug
6      public static void main(String[] args) {
7          SingleLinkedList24 sll = new SingleLinkedList24();
8          Mahasiswa24 mhs1 = new Mahasiswa24(nm:"24212200", name:"Alvaro", kls:"1A", ip:4.0);
9          Mahasiswa24 mhs2 = new Mahasiswa24(nm:"23212201", name:"Bimon", kls:"2B", ip:3.8);
10         Mahasiswa24 mhs3 = new Mahasiswa24(nm:"22212202", name:"Cintia", kls:"3C", ip:3.5);
11         Mahasiswa24 mhs4 = new Mahasiswa24(nm:"21212203", name:"Dirga", kls:"4D", ip:3.6);
12
13         sll.print();
14         sll.addFirst(mhs4);
15         sll.print();
16         sll.addLast(mhs1);
17         sll.print();
18         sll.insertAfter(key:"Dirga", mhs3);
19         sll.insertArt(index:2, mhs2);
20         sll.print();
21     }

```

- **Output**

```

Linked List Kosong
Isi Linked List:
Dirga      21212203      4D      3,60

Isi Linked List:
Dirga      21212203      4D      3,60
Alvaro     24212200      1A      4,00

Isi Linked List:
Dirga      21212203      4D      3,60
Cintia     22212202      3C      3,50
Bimon      23212201      2B      3,80
Alvaro     24212200      1A      4,00

```

### Pertanyaan 1

1. Mengapa hasil compile kode program di baris pertama menghasilkan “Linked List Kosong”?
  - Karena pada saat `sll.print()`; pertama dipanggil, kita belum menambahkan data apa pun ke dalam linked list. Objek `SingleLinkedList24` masih kosong, sehingga kondisi `head == null`. Oleh karena itu, method `print()` akan menampilkan pesan “Linked List Kosong”.
2. Jelaskan kegunaan variable temp secara umum pada setiap method!
  - Variabel temp digunakan sebagai pointer sementara untuk menelusuri node-node dalam linked list. Tujuannya agar kita bisa membaca, mencari, atau memodifikasi node tanpa mengubah posisi head dari linked list.
3. Lakukan modifikasi agar data dapat ditambahkan dari keyboard!

```
jobsheet_11 > SLLMain24.java > SLLMain24
1  package jobsheet_11;
2  import java.util.Scanner;
3  public class SLLMain24 {
4
5      Run | Debug
6      public static void main(String[] args) {
7          Scanner input24 = new Scanner(System.in);
8
9          SingleLinkedList24 sll = new SingleLinkedList24();
10         int pilihan;
11         do {
12             System.out.println(x:"=== Program Single Linked List ===");
13             System.out.println(x:"1. Tambah Data Mahasiswa di awal");
14             System.out.println(x:"2. Tambah Data Mahasiswa di akhir");
15             System.out.println(x:"3. Tambah Data Mahasiswa setelah data tertentu");
16             System.out.println(x:"4. Tambah Data Mahasiswa di posisi tertentu");
17             System.out.println(x:"5. Tampilkan Data Mahasiswa");
18             System.out.println(x:"0. Keluar");
19             System.out.print(s:"Masukkan pilihan (0-5): ");
20             pilihan = input24.nextInt();
21             input24.nextLine();
22         } while (pilihan != 0);
23     }
24 }
```

```

22     switch (pilihan) {
23         case 1:
24             System.out.println(x:"== Tambah Data Mahasiswa di Awal ==");
25             sll.addFirst(inputMhs(input24));
26             break;
27         case 2:
28             System.out.println(x:"== Tambah Data Mahasiswa di Akhir ==");
29             sll.addLast(inputMhs(input24));
30             break;
31         case 3:
32             System.out.println(x:"== Tambah Data Mahasiswa Setelah Data Tertentu ==");
33             System.out.print(s:"Masukkan nama mahasiswa yang diinginkan: ");
34             String key = input24.nextLine();
35             sll.insertAfter(key, inputMhs(input24));
36             break;
37         case 4:
38             System.out.println(x:"== Tambah Data Mahasiswa di Posisi Tertentu ==");
39             System.out.print(s:"Masukkan posisi yang diinginkan: ");
40             int pos = input24.nextInt();
41             input24.nextLine();
42             sll.insertArt(pos, inputMhs(input24));
43             break;
44         case 5:
45             System.out.println(x:"== Tampilkan Data Mahasiswa ==");
46             sll.print();
47             break;
48         case 0:
49             System.out.println(x:"Terima kasih!");
50             break;

```

```

51         default:
52             System.out.println(x:"Pilihan tidak valid. Silakan coba lagi.");
53     }
54     } while (pilihan != 0);
55 }
56
57 static Mahasiswa24 inputMhs(Scanner input24) {
58     System.out.print(s:"Masukkan NIM: ");
59     String nim = input24.nextLine();
60     System.out.print(s:"Masukkan Nama: ");
61     String nama = input24.nextLine();
62     System.out.print(s:"Masukkan Kelas: ");
63     String kelas = input24.nextLine();
64     System.out.print(s:"Masukkan IPK: ");
65     double ipk = input24.nextDouble();
66     input24.nextLine();
67     return new Mahasiswa24(nim, nama, kelas, ipk);
68 }
69

```

```

==== Program Single Linked List ====
1. Tambah Data Mahasiswa di awal
2. Tambah Data Mahasiswa di akhir
3. Tambah Data Mahasiswa setelah data tertentu
4. Tambah Data Mahasiswa di posisi tertentu
5. Tampilkan Data Mahasiswa
0. Keluar
Masukkan pilihan (0-5): 1
=== Tambah Data Mahasiswa di Awal ===
Masukkan NIM: 244107060056
Masukkan Nama: Zaskia
Masukkan Kelas: 1A
Masukkan IPK: 3,9
==== Program Single Linked List ====
1. Tambah Data Mahasiswa di awal
2. Tambah Data Mahasiswa di akhir
3. Tambah Data Mahasiswa setelah data tertentu
4. Tambah Data Mahasiswa di posisi tertentu
5. Tampilkan Data Mahasiswa
0. Keluar
Masukkan pilihan (0-5): 3
=== Tambah Data Mahasiswa Setelah Data Tertentu ===
Masukkan nama mahasiswa yang diinginkan: Zaskia
Masukkan NIM: 244107060057
Masukkan Nama: Iak
Masukkan Kelas: 1D
Masukkan IPK: 3,8

```

```

==== Program Single Linked List ====
1. Tambah Data Mahasiswa di awal
2. Tambah Data Mahasiswa di akhir
3. Tambah Data Mahasiswa setelah data tertentu
4. Tambah Data Mahasiswa di posisi tertentu
5. Tampilkan Data Mahasiswa
0. Keluar
Masukkan pilihan (0-5): 4
=== Tambah Data Mahasiswa di Posisi Tertentu ===
Masukkan posisi yang diinginkan: 2
Masukkan NIM: 244107060058
Masukkan Nama: Rara
Masukkan Kelas: 1E
Masukkan IPK: 3,8
==== Program Single Linked List ====
1. Tambah Data Mahasiswa di awal
2. Tambah Data Mahasiswa di akhir
3. Tambah Data Mahasiswa setelah data tertentu
4. Tambah Data Mahasiswa di posisi tertentu
5. Tampilkan Data Mahasiswa
0. Keluar
Masukkan pilihan (0-5): 2
=== Tambah Data Mahasiswa di Akhir ===
Masukkan NIM: 244107060059
Masukkan Nama: Tia
Masukkan Kelas: 1F
Masukkan IPK: 3,7

```

```

=== Tambah Data Mahasiswa di Akhir ===
Masukkan NIM: 244107060059
Masukkan Nama: Tia
Masukkan Kelas: 1F
Masukkan IPK: 3,7
==== Program Single Linked List ====
1. Tambah Data Mahasiswa di awal
2. Tambah Data Mahasiswa di akhir
3. Tambah Data Mahasiswa setelah data tertentu
4. Tambah Data Mahasiswa di posisi tertentu
5. Tampilkan Data Mahasiswa
0. Keluar
Masukkan pilihan (0-5): 5
=== Tampilkan Data Mahasiswa ===
Isi Linked List:
Zaskia      244107060056    1A    3,90
Iak         244107060057    1D    3,80
Rara        244107060058    1E    3,80
Tia         244107060059    1F    3,70

==== Program Single Linked List ====
1. Tambah Data Mahasiswa di awal
2. Tambah Data Mahasiswa di akhir
3. Tambah Data Mahasiswa setelah data tertentu
4. Tambah Data Mahasiswa di posisi tertentu
5. Tampilkan Data Mahasiswa
0. Keluar
Masukkan pilihan (0-5): 0
Terima kasih!

```

## Percobaan 2

- Class SingleLinkedList24

```
78     public void getData(int index) {
79         NodeMahasiswa24 tmp = head;
80         for (int i = 0; i < index; i++) {
81             tmp = tmp.next;
82         }
83         tmp.data.tampilInformasi();
84     }
85
86     public int indexOf(String key) {
87         NodeMahasiswa24 tmp = head;
88         int index = 0;
89         while (tmp != null && !tmp.data.nama.equalsIgnoreCase(key)) {
90             tmp = tmp.next;
91             index++;
92         }
93
94         if (tmp == null) {
95             return -1;
96         } else {
97             return index;
98         }
99     }
100
101     public void removeFirst() {
102         if (isEmpty()) {
103             System.out.println(x:"Linked List masih Kosong, tidak dapat dihapus!");
104         } else if (head == tail) {
105             head = tail = null;
106         } else {
107             head = head.next;
108         }
109     }
110
111     public void removeLast() {
112         if (isEmpty()) {
113             System.out.println(x:"Linked List masih Kosong, tidak dapat dihapus!");
114         } else if (head == tail) {
115             head = tail = null;
116         } else {
117             NodeMahasiswa24 temp = head;
118             while (temp.next != tail) {
119                 temp = temp.next;
120             }
121             temp.next = null;
122             tail = temp;
123         }
124     }
```

```

126 public void remove(String key) {
127     if (isEmpty()) {
128         System.out.println(x:"Linked List masih Kosong, tidak dapat dihapus!");
129     } else {
130         NodeMahasiswa24 temp = head;
131         while (temp != null) {
132             if ((temp.data.nama.equalsIgnoreCase(key)) && (temp == head)) {
133                 this.removeFirst();
134                 break;
135             } else if (temp.data.nama.equalsIgnoreCase(key)) {
136                 temp.next = temp.next.next;
137                 if (temp.next == null) {
138                     tail = temp;
139                 }
140                 break;
141             }
142             temp = temp.next;
143         }
144     }
145 }

147 public void removeAt(int index) {
148     if (index == 0) {
149         removeFirst();
150     } else {
151         NodeMahasiswa24 temp = head;
152         for (int i = 0; i < index - 1; i++) {
153             temp = temp.next;
154         }
155         temp.next = temp.next.next;
156         if (temp.next == null) {
157             tail = temp;
158         }
159     }
160 }
161 }
162 }
163 }

```

- **Class SLLMain24**

```

jobsheet_11 > SLLMain24.java > SLLMain24 > main(String[])
1 package jobsheet_11;
2 public class SLLMain24 {
3     Run | Debug
4     public static void main(String[] args) {
5         SingleLinkedList24 sll = new SingleLinkedList24();
6         Mahasiswa24 mhs1 = new Mahasiswa24(nm:"24212200", name:"Alvaro", kls:"1A", ip:4.0);
7         Mahasiswa24 mhs2 = new Mahasiswa24(nm:"23212201", name:"Bimon", kls:"2B", ip:3.8);
8         Mahasiswa24 mhs3 = new Mahasiswa24(nm:"22212202", name:"Cintia", kls:"3C", ip:3.5);
9         Mahasiswa24 mhs4 = new Mahasiswa24(nm:"21212203", name:"Dirga", kls:"4D", ip:3.6);
10
11         sll.print();
12         sll.addFirst(mhs4);
13         sll.print();
14         sll.addLast(mhs1);
15         sll.print();
16         sll.insertAfter(key:"Dirga", mhs3);
17         sll.insertAt(index:2, mhs2);
18         sll.print();
19
20         System.out.println(x:"data index 1 : ");
21         sll.getData(index:1);
22
23         System.out.println("data mahasiswa an Bimon berada pada index : " + sll.indexOf(key:"bimon"));
24         System.out.println();
25
26         sll.removeFirst();
27         sll.removeLast();
28         sll.print();
29         sll.removeAt(index:0);
30         sll.print();
31     }

```



- **Output**

```
Linked List Kosong
Isi Linked List:
Dirga      21212203      4D      3,60

Isi Linked List:
Dirga      21212203      4D      3,60
Alvaro     24212200      1A      4,00

Isi Linked List:
Dirga      21212203      4D      3,60
Cintia     22212202      3C      3,50
Bimon      23212201      2B      3,80
Alvaro     24212200      1A      4,00

data index 1 :
Cintia     22212202      3C      3,50
data mahasiswa an Bimon berada pada index : 2

Isi Linked List:
Cintia     22212202      3C      3,50
Bimon      23212201      2B      3,80

Isi Linked List:
Bimon      23212201      2B      3,80
```

## TUGAS

Buatlah implementasi program antrian layanan unit kemahasiswaan sesuai dengan berikut ini :

- Implementasi antrian menggunakan Queue berbasis Linked List!
- Program merupakan proyek baru bukan modifikasi dari percobaan.
- Ketika seorang mahasiswa akan mengantri, maka dia harus mendaftarkan datanya.
- Cek antrian kosong, Cek antrian penuh, Mengosongkan antrian.
- Menambahkan antrian.
- Memanggil antrian.
- Menampilkan antrian terdepan dan antrian paling akhir.
- Menampilkan jumlah mahasiswa yang masih mengantre.

- **Class MahasiswaTGS**

```
jobsheet_11 > TugasJB11 > MahasiswaTGS.java > MahasiswaTGS
1  package jobsheet_11.TugasJB11;
2
3  public class MahasiswaTGS {
4      String nim, nama, kelas;
5      double ipk;
6
7      MahasiswaTGS(String nim, String nama, String kelas, double ipk) {
8          this.nim = nim;
9          this.nama = nama;
10         this.kelas = kelas;
11         this.ipk = ipk;
12     }
13
14     public void tampilInformasi(){
15         System.out.printf(format:"%-10s %-15s %-5s %-4.2f\n", nama, nim, kelas, ipk);
16     }
17 }
```

- **Class NodeMhs**

```
jobsheet_11 > TugasJB11 > NodeMhs.java > ...
1  package jobsheet_11.TugasJB11;
2
3  public class NodeMhs {
4      MahasiswaTGS data;
5      NodeMhs next;
6
7      public NodeMhs(MahasiswaTGS data) {
8          this.data = data;
9          this.next = null;
10     }
11 }
```

- **Class QueueLinkedList**

```
jobsheet_11 > TugasJB11 > QueueLinkedList.java > QueueLinkedList
1  package jobsheet_11.TugasJB11;
2
3  public class QueueLinkedList {
4      NodeMhs front;
5      NodeMhs rear;
6      int size;
7      int capacity;
8
9      public QueueLinkedList(int capacity) {
10         this.front = null;
11         this.rear = null;
12         this.size = 0;
13         this.capacity = capacity;
14     }
15
16     // Cek apakah antrian kosong
17     public boolean isEmpty() {
18         return front == null;
19     }
20
21     // Cek apakah antrian penuh
22     public boolean isFull() {
23         return size == capacity;
24     }
25
26     // Mengosongkan antrian
27     public void clear() {
28         front = null;
29         rear = null;
30         size = 0;
31         System.out.println(x:"Antrian telah dikosongkan!");
32     }
33
34
35     public void enqueue(MahasiswaTGS mahasiswa) {
36         if (isFull()) {
37             System.out.println(x:"Antrian penuh! Tidak bisa menambahkan mahasiswa baru.");
38             return;
39         }
40
41         NodeMhs newNode = new NodeMhs(mahasiswa);
42
43         if (isEmpty()) {
44             front = newNode;
45         } else {
46             rear.next = newNode;
47         }
48
49         rear = newNode;
50         size++;
51         System.out.println("Mahasiswa " + mahasiswa.nama + " berhasil ditambahkan ke antrian.");
52     }
```

```

54 // Memanggil mahasiswa dari antrian
55 public MahasiswaTGS dequeue() {
56     if (isEmpty()) {
57         System.out.println(x:"Antrian kosong! Tidak ada mahasiswa untuk dipanggil.");
58         return null;
59     }
60
61     MahasiswaTGS mahasiswa = front.data;
62     front = front.next;
63     size--;
64
65     if (front == null) {
66         rear = null;
67     }
68
69     return mahasiswa;
70 }
71
72 // Menampilkan mahasiswa terdepan
73 public MahasiswaTGS getFront() {
74     if (isEmpty()) {
75         System.out.println(x:"Antrian kosong! Tidak ada mahasiswa terdepan.");
76         return null;
77     }
78     return front.data;
79 }

```

```

81 // Menampilkan mahasiswa paling akhir
82 public MahasiswaTGS getRear() {
83     if (isEmpty()) {
84         System.out.println(x:"Antrian kosong! Tidak ada mahasiswa paling akhir.");
85         return null;
86     }
87     return rear.data;
88 }
89
90 // Menampilkan jumlah mahasiswa yang masih mengantri
91 public int getSize() {
92     return size;
93 }
94
95 // Menampilkan semua mahasiswa dalam antrian
96 public void displayAll() {
97     if (isEmpty()) {
98         System.out.println(x:"Antrian kosong!");
99         return;
100     }
101
102     System.out.println(x:"\n=== DAFTAR MAHASISWA DALAM ANTRIAN ===");
103     NodeMhs current = front;
104     int index = 1;
105     while (current != null) {
106         System.out.println(index + ". " + current.data);
107         current = current.next;
108         index++;
109     }
110     System.out.println(x:"=====");
111
112

```

- **Class AntrianMahasiswa**

```
jobsheet_11 > TugasJB11 > AntrianMahasiswa.java > AntrianMahasiswa > main(String[])
1  package jobsheet_11.TugasJB11;
2
3  import java.util.Scanner;
4  public class AntrianMahasiswa {
5      static Scanner scanner = new Scanner(System.in);
6      static QueueLinkedList queue;
7
8      Run | Debug
9      public static void main(String[] args) {
10         System.out.println(x:"=== PROGRAM ANTRIAN LAYANAN UNIT KEMAHASISWAAN ===");
11         System.out.print(s:"Masukkan kapasitas maksimal antrian: ");
12         int capacity = scanner.nextInt();
13         scanner.nextLine();
14
15         queue = new QueueLinkedList(capacity);
16
17         int pilihan;
18         do {
19             tampilkanMenu();
20             System.out.print(s:"Pilihan Anda: ");
21             pilihan = scanner.nextInt();
22             scanner.nextLine();
23
24             switch (pilihan) {
25                 case 1:
26                     if (queue.isFull()) {
27                         System.out.println(x:"Antrian penuh! Tidak bisa menambahkan mahasiswa baru.");
28                         return;
29                     }
30
31                     System.out.println(x:"\n=== PENDAFTARAN MAHASISWA ===");
32                     System.out.print(s:"NIM: ");
33                     String nim = scanner.nextLine();
34                     System.out.print(s:"Nama: ");
35                     String nama = scanner.nextLine();
36                     System.out.print(s:"Kelas: ");
37                     String kelas = scanner.nextLine();
38                     System.out.print(s:"IPK: ");
39                     double ipk = scanner.nextDouble();
40                     scanner.nextLine();
41
42                     MahasiswaTGS mahasiswaBaru = new MahasiswaTGS(nim, nama, kelas, ipk);
43                     queue.enqueue(mahasiswaBaru);
44                     break;
45                 case 2:
46                     MahasiswaTGS mahasiswaDequeued = queue.dequeue();
47                     if (mahasiswaDequeued != null) {
48                         System.out.println(x:"\n=== MEMANGGIL MAHASISWA ===");
49                         System.out.println("Mahasiswa yang dipanggil: " + mahasiswaDequeued.nama);
50                     }
51                     break;
52             }
53         } while (pilihan != 0);
54     }
55 }
```

```

51         case 3:
52             MahasiswaTGS mahasiswaFront = queue.getFront();
53             if (mahasiswaFront != null) {
54                 System.out.println(x:"\n=== MAHASISWA TERDEPAN ===");
55                 System.out.println("Mahasiswa terdepan: " + mahasiswaFront.nama);
56             }
57             break;
58         case 4:
59             MahasiswaTGS mahasiswaRear = queue.getRear();
60             if (mahasiswaRear != null) {
61                 System.out.println(x:"\n=== MAHASISWA TERAKHIR ===");
62                 System.out.println("Mahasiswa terakhir: " + mahasiswaRear.nama);
63             }
64             break;
65         case 5:
66             System.out.println(x:"\n=== JUMLAH MAHASISWA DALAM ANTRIAN ===");
67             System.out.println("Jumlah mahasiswa yang masih mengantri: " + queue.getSize());
68             break;
69         case 6:
70             queue.displayAll();
71             break;
72         case 7:
73             System.out.println(x:"\n=== CEK ANTRIAN KOSONG ===");
74             if (queue.isEmpty()) {
75                 System.out.println(x:"Antrian saat ini kosong.");
76             } else {
77                 System.out.println(x:"Antrian tidak kosong.");
78             }
79             break;
80         case 8:
81             System.out.println(x:"\n=== CEK ANTRIAN PENUH ===");
82             if (queue.isFull()) {
83                 System.out.println(x:"Antrian saat ini penuh.");
84             } else {
85                 System.out.println(x:"Antrian belum penuh.");
86             }
87             break;
88         case 9:
89             System.out.println(x:"\n=== MENGOSONGKAN ANTRIAN ===");
90             queue.clear();
91             break;
92         case 0:
93             System.out.println(x:"Terima kasih telah menggunakan program antrian!");
94             break;
95         default:
96             System.out.println(x:"Pilihan tidak valid. Silakan coba lagi.");
97     }
98     System.out.println();
99 } while (pilihan != 0);
100
101 scanner.close();
102

```

```
104     static void tampilkanMenu() {
105         System.out.println(x:"\n=== MENU ANTRIAN LAYANAN KEMAHASISWAAN ===");
106         System.out.println(x:"1. Tambah Mahasiswa ke Antrian");
107         System.out.println(x:"2. Panggil Mahasiswa dari Antrian");
108         System.out.println(x:"3. Lihat Mahasiswa Terdepan");
109         System.out.println(x:"4. Lihat Mahasiswa Terakhir");
110         System.out.println(x:"5. Lihat Jumlah Mahasiswa dalam Antrian");
111         System.out.println(x:"6. Tampilkan Semua Mahasiswa dalam Antrian");
112         System.out.println(x:"7. Cek Antrian Kosong");
113         System.out.println(x:"8. Cek Antrian Penuh");
114         System.out.println(x:"9. Kosongkan Antrian");
115         System.out.println(x:"0. Keluar Program");
116         System.out.println(x:"=====");
117     }
118 }
```

- **Output**

```
=== PROGRAM ANTRIAN LAYANAN UNIT KEMAHASISWAAN ===
Masukkan kapasitas maksimal antrian: 3

=== MENU ANTRIAN LAYANAN KEMAHASISWAAN ===
1. Tambah Mahasiswa ke Antrian
2. Panggil Mahasiswa dari Antrian
3. Lihat Mahasiswa Terdepan
4. Lihat Mahasiswa Terakhir
5. Lihat Jumlah Mahasiswa dalam Antrian
6. Tampilkan Semua Mahasiswa dalam Antrian
7. Cek Antrian Kosong
8. Cek Antrian Penuh
9. Kosongkan Antrian
0. Keluar Program
=====
Pilihan Anda: 1

=== PENDAFTARAN MAHASISWA ===
NIM: 123
Nama: Iak
Kelas: 1A
IPK: 3,9
Mahasiswa Iak berhasil ditambahkan ke antrian.
```

```
=== MENU ANTRIAN LAYANAN KEMAHASISWAAN ===
1. Tambah Mahasiswa ke Antrian
2. Panggil Mahasiswa dari Antrian
3. Lihat Mahasiswa Terdepan
4. Lihat Mahasiswa Terakhir
5. Lihat Jumlah Mahasiswa dalam Antrian
6. Tampilkan Semua Mahasiswa dalam Antrian
7. Cek Antrian Kosong
8. Cek Antrian Penuh
9. Kosongkan Antrian
0. Keluar Program
=====
Pilihan Anda: 1

=== PENDAFTARAN MAHASISWA ===
NIM: 456
Nama: Zaskia
Kelas: 1B
IPK: 3,8
Mahasiswa Zaskia berhasil ditambahkan ke antrian.
```



```

=== MENU ANTRIAN LAYANAN KEMAHASISWAAN ===
1. Tambah Mahasiswa ke Antrian
2. Panggil Mahasiswa dari Antrian
3. Lihat Mahasiswa Terdepan
4. Lihat Mahasiswa Terakhir
5. Lihat Jumlah Mahasiswa dalam Antrian
6. Tampilkan Semua Mahasiswa dalam Antrian
7. Cek Antrian Kosong
8. Cek Antrian Penuh
9. Kosongkan Antrian
0. Keluar Program
=====
Pilihan Anda: 1

=== PENDAFTARAN MAHASISWA ===
NIM: 789
Nama: Rara
Kelas: 1C
IPK: 3,7
Mahasiswa Rara berhasil ditambahkan ke antrian.

=== MENU ANTRIAN LAYANAN KEMAHASISWAAN ===
1. Tambah Mahasiswa ke Antrian
2. Panggil Mahasiswa dari Antrian
3. Lihat Mahasiswa Terdepan
4. Lihat Mahasiswa Terakhir
5. Lihat Jumlah Mahasiswa dalam Antrian
6. Tampilkan Semua Mahasiswa dalam Antrian
7. Cek Antrian Kosong
8. Cek Antrian Penuh
9. Kosongkan Antrian
0. Keluar Program
=====
Pilihan Anda: 3

=== MAHASISWA TERDEPAN ===
Mahasiswa terdepan: Iak

=== MENU ANTRIAN LAYANAN KEMAHASISWAAN ===
1. Tambah Mahasiswa ke Antrian
2. Panggil Mahasiswa dari Antrian
3. Lihat Mahasiswa Terdepan
4. Lihat Mahasiswa Terakhir
5. Lihat Jumlah Mahasiswa dalam Antrian
6. Tampilkan Semua Mahasiswa dalam Antrian
7. Cek Antrian Kosong
8. Cek Antrian Penuh
9. Kosongkan Antrian
0. Keluar Program
=====
Pilihan Anda: 4

=== MAHASISWA TERAKHIR ===
Mahasiswa terakhir: Rara

```

```

=== MENU ANTRIAN LAYANAN KEMAHASISWAAN ===
1. Tambah Mahasiswa ke Antrian
2. Panggil Mahasiswa dari Antrian
3. Lihat Mahasiswa Terdepan
4. Lihat Mahasiswa Terakhir
5. Lihat Jumlah Mahasiswa dalam Antrian
6. Tampilkan Semua Mahasiswa dalam Antrian
7. Cek Antrian Kosong
8. Cek Antrian Penuh
9. Kosongkan Antrian
0. Keluar Program
=====
Pilihan Anda: 5

=== JUMLAH MAHASISWA DALAM ANTRIAN ===
Jumlah mahasiswa yang masih mengantri: 3

=== MENU ANTRIAN LAYANAN KEMAHASISWAAN ===
1. Tambah Mahasiswa ke Antrian
2. Panggil Mahasiswa dari Antrian
3. Lihat Mahasiswa Terdepan
4. Lihat Mahasiswa Terakhir
5. Lihat Jumlah Mahasiswa dalam Antrian
6. Tampilkan Semua Mahasiswa dalam Antrian
7. Cek Antrian Kosong
8. Cek Antrian Penuh
9. Kosongkan Antrian
0. Keluar Program
=====
Pilihan Anda: 7

=== CEK ANTRIAN KOSONG ===
Antrian tidak kosong.

```

```
=== MENU ANTRIAN LAYANAN KEMAHASISWAAN ===
1. Tambah Mahasiswa ke Antrian
2. Panggil Mahasiswa dari Antrian
3. Lihat Mahasiswa Terdepan
4. Lihat Mahasiswa Terakhir
5. Lihat Jumlah Mahasiswa dalam Antrian
6. Tampilkan Semua Mahasiswa dalam Antrian
7. Cek Antrian Kosong
8. Cek Antrian Penuh
9. Kosongkan Antrian
0. Keluar Program
```

```
=====
Pilihan Anda: 8
```

```
=== CEK ANTRIAN PENUH ===
Antrian saat ini penuh.
```

```
=== MENU ANTRIAN LAYANAN KEMAHASISWAAN ===
1. Tambah Mahasiswa ke Antrian
2. Panggil Mahasiswa dari Antrian
3. Lihat Mahasiswa Terdepan
4. Lihat Mahasiswa Terakhir
5. Lihat Jumlah Mahasiswa dalam Antrian
6. Tampilkan Semua Mahasiswa dalam Antrian
7. Cek Antrian Kosong
8. Cek Antrian Penuh
9. Kosongkan Antrian
0. Keluar Program
```

```
=====
Pilihan Anda: 2
```

```
=== MEMANGGIL MAHASISWA ===
Mahasiswa yang dipanggil: Iak
```

```
=== MENU ANTRIAN LAYANAN KEMAHASISWAAN ===
1. Tambah Mahasiswa ke Antrian
2. Panggil Mahasiswa dari Antrian
3. Lihat Mahasiswa Terdepan
4. Lihat Mahasiswa Terakhir
5. Lihat Jumlah Mahasiswa dalam Antrian
6. Tampilkan Semua Mahasiswa dalam Antrian
7. Cek Antrian Kosong
8. Cek Antrian Penuh
9. Kosongkan Antrian
0. Keluar Program
```

```
=====
Pilihan Anda: 2
```

```
=== MEMANGGIL MAHASISWA ===
Mahasiswa yang dipanggil: Zaskia
```

```
=== MENU ANTRIAN LAYANAN KEMAHASISWAAN ===
1. Tambah Mahasiswa ke Antrian
2. Panggil Mahasiswa dari Antrian
3. Lihat Mahasiswa Terdepan
4. Lihat Mahasiswa Terakhir
5. Lihat Jumlah Mahasiswa dalam Antrian
6. Tampilkan Semua Mahasiswa dalam Antrian
7. Cek Antrian Kosong
8. Cek Antrian Penuh
9. Kosongkan Antrian
0. Keluar Program
```

```
=====
Pilihan Anda: 9
```

```
=== MENGOSONGKAN ANTRIAN ===
Antrian telah dikosongkan!
```