

ALGORITMA DAN STRUKTUR DATA

“Laporan hasil Praktikum pada Jobsheet 11 “Linked list” ”

Oleh:

Hafiz Rizqi Hernanda

NIM (244107020154)



Jurusan Teknologi informasi

Teknik Informatika

Politeknik Negeri Malang

2. Praktikum

2.1 Pembuatan Single Linked List

Didalam praktikum ini, kita akan mempraktikkan bagaimana membuat Single Linked List dengan representasi data berupa Node, pengaksesan linked list dan metode penambahan data.

1. Pada Project yang sudah dibuat pada Minggu sebelumnya. Buat folder atau package baru bernama Jobsheet11 di dalam repository Praktikum ASD.
2. Tambahkan class-class berikut:
 - a. Mahasiswa11.java
 - b. Node11.java
 - c. SingleLinkedList11.java
 - d. SLLMain11.java
3. Implementasikan Class Mahasiswa00 sesuai dengan diagram class berikut ini :

```
1  package Jobsheet11;
2
3  public class Mahasiswa11 {
4      String nim;
5      String nama;
6      String kelas;
7      double ipk;
8
9      public Mahasiswa11() {}
10
11     public Mahasiswa11(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.println(nama + "\t" + nim + "\t" + kelas + "\t" + ipk);
20     }
21
22 }
```

4. Implementasi class Node seperti gambar berikut ini

```

Jobsheet11 > J NodeMahasiswa11.java > NodeMahasiswa11
1  package Jobsheet11;
2  public class NodeMahasiswa11 {
3      Mahasiswa11 data;
4      NodeMahasiswa11 next;
5
6      public NodeMahasiswa11(Mahasiswa11 data, NodeMahasiswa11 next) {
7          this.data = data;
8          this.next = next;
9      }
10
11 }
12

```

5. Tambahkan attribute head dan tail pada class SingleLinkedList

```

1  package Jobsheet11;
2
3  public class SingleLinkedList11 {
4      NodeMahasiswa11 head;
5      NodeMahasiswa11 tail;
6

```

6. Sebagai langkah berikutnya, akan diimplementasikan method-method yang terdapat pada SingleLinkedList.

7. Tambahkan method isEmpty().

```

7      boolean isEmpty() {
8          return head == null;
9      }

```

8. Implementasi method untuk mencetak dengan menggunakan proses traverse.

```

11      public void print() {
12          if (!isEmpty()) {
13              NodeMahasiswa11 tmp = head;
14              System.out.println(x:"Isi Linked List:\t");
15              while (tmp != null) {
16                  tmp.data.tampilInformasi();
17                  tmp = tmp.next;
18              }
19              System.out.println(x:"");
20          } else {
21              System.out.println(x:"Linked List Kosong");
22          }
23      }

```

9. Implementasikan method addFirst()

```

25     public void addFirst(Mahasiswa11 input) {
26         NodeMahasiswa11 ndInput = new NodeMahasiswa11(input, next:null);
27         if (isEmpty()) {
28             head = ndInput;
29             tail = ndInput;
30         } else {
31             ndInput.next = head;
32             head = ndInput;
33         }
34     }

```

10. Implementasikan method addLast()

```

36     public void addLast(Mahasiswa11 input) {
37         NodeMahasiswa11 ndInput = new NodeMahasiswa11(input, next:null);
38         if (isEmpty()) {
39             head = ndInput;
40             tail = ndInput;
41         } else {
42             tail.next = ndInput;
43             tail = ndInput;
44         }
45     }

```

11. Implementasikan method insertAfter, untuk memasukkan node yang memiliki data

```

47     public void insertAfter(String key, Mahasiswa11 input) {
48         NodeMahasiswa11 ndInput = new NodeMahasiswa11(input, next:null);
49         NodeMahasiswa11 temp = head;
50         do {
51             if (temp.data.nama.equalsIgnoreCase(key)) {
52                 ndInput.next = temp.next;
53                 temp.next = ndInput;
54                 if (ndInput.next == null) {
55                     tail = ndInput;
56                 }
57                 break;
58             }
59             temp = temp.next;
60         } while (temp != null);
61     }

```

12. Tambahkan method penambahan node pada indeks tertentu.

```

63     public void insertAt (int index, Mahasiswa11 input) {
64         if (index < 0) {
65             System.out.println(x:"Index salah");
66         } else if (index == 0) {
67             addFirst(input);
68         } else {
69             NodeMahasiswa11 temp = head;
70             for (int i = 0; i < index - 1; i++) {
71                 temp = temp.next;
72             }
73             temp.next = new NodeMahasiswa11(input, temp.next);
74             if (temp.next.next == null) {
75                 tail = temp.next;
76             }
77         }
78     }
79 }

```

13. Pada class SLLMain00, buatlah fungsi main, kemudian buat object dari class SingleLinkedList.

```

Jobsheet11 > J SLLMain11.java > SLLMain11 > main(String[])
1   package Jobsheet11;
2
3   public class SLLMain11 {
4       Run | Debug
5       public static void main(String[] args) {
6           SingleLinkedList11 sll = new SingleLinkedList11();

```

14. Buat empat object mahasiswa dengan nama mhs1, mhs2, mhs3, mhs4 kemudian isi data setiap object melalui konstruktor.

```

7   Mahasiswa11 mhs1 = new Mahasiswa11(nm:"24212200", name:"Alvaro", kls:"1A", ip:4.0);
8   Mahasiswa11 mhs2 = new Mahasiswa11(nm:"24212201", name:"Bimon", kls:"2B", ip:3.8);
9   Mahasiswa11 mhs3 = new Mahasiswa11(nm:"24212202", name:"Cintia", kls:"3C", ip:3.5);
10  Mahasiswa11 mhs4 = new Mahasiswa11(nm:"24212203", name:"Dirga", kls:"4D", ip:3.6);

```

15. Tambahkan Method penambahan data dan pencetakan data di setiap penambahannya agar terlihat perubahannya.

```

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.print();
20     sll.insertAt(index:2, mhs2);
21     sll.print();
22 }
23 }

```

2.1.1 Verifikasi Hasil Percobaan

Cocokkan hasil compile kode program anda dengan gambar berikut ini

```
Linked List Kosong
Isi Linked List:
Dirga      21212203      4D      3.6

Isi Linked List:
Dirga      21212203      4D      3.6
Alvaro     24212200      1A      4.0

Isi Linked List:
Dirga      21212203      4D      3.6
Cintia     22212202      3C      3.5
Bimon      23212201      2B      3.8
Alvaro     24212200      1A      4.0
```

```
Linked List Kosong
Isi Linked List:
Dirga      24212203      4D      3.6

Isi Linked List:
Dirga      24212203      4D      3.6
Alvaro     24212200      1A      4.0

Isi Linked List:
Dirga      24212203      4D      3.6
Cintia     24212202      3C      3.5
Alvaro     24212200      1A      4.0

Isi Linked List:
Dirga      24212203      4D      3.6
Cintia     24212202      3C      3.5
Bimon      24212201      2B      3.8
Alvaro     24212200      1A      4.0

PS C:\Regulus\Praktikum-ASD> 
```

2.1.2 Pertanyaan

1. Mengapa hasil compile kode program di baris pertama menghasilkan “Linked List Kosong”?

Karena tidak ada node yang ditambahkan

2. Jelaskan kegunaan variable temp secara umum pada setiap method!

Temp itu berguna sebagai pengganti sementara head sebagai pointer berjalan agar head tidak berubah

3. Lakukan modifikasi agar data dapat ditambahkan dari keyboard!

```
3 public class SLLMain11 {
    Run | Debug
4     public static void main(String[] args) {
5         SingleLinkedList11 sll = new SingleLinkedList11();
6         Scanner input = new Scanner(System.in);
7         System.out.println(x:"=== Data Mahasiswa ===");
8         System.out.print(s:"Masukkan jumlah mahasiswa: ");
9         int jumlah = input.nextInt();
10        input.nextLine();
11        for (int i = 0; i < jumlah; i++) {
12            System.out.print(s:"NIM: ");
13            String nim = input.nextLine();
14            System.out.print(s:"Nama: ");
15            String nama = input.nextLine();
16            System.out.print(s:"Kelas: ");
17            String kelas = input.nextLine();
18            System.out.print(s:"IPK: ");
19            double ipk = input.nextDouble();
20            input.nextLine();
21            Mahasiswa11 mhs = new Mahasiswa11(nim, nama, kelas, ipk);
22            sll.addLast(mhs);
23        }
24    }
```

```

=== Data Mahasiswa ===
Masukkan jumlah mahasiswa: 2
NIM: 123
Nama: Arif
Kelas: 1B
IPK: 4,0
NIM: 124
Nama: Rizky
Kelas: 1G
IPK: 3,7
Isi Linked List:
Arif    123    1B    4.0
Rizky   124    1G    3.7

```

```

Isi Linked List:
Dirga   24212203    4D    3.6
Arif    123    1B    4.0
Rizky   124    1G    3.7

```

```

Isi Linked List:
Dirga   24212203    4D    3.6
Arif    123    1B    4.0
Rizky   124    1G    3.7
Alvaro  24212200    1A    4.0

```

```

Isi Linked List:
Dirga   24212203    4D    3.6
Cintia  24212202    3C    3.5
Arif    123    1B    4.0
Rizky   124    1G    3.7
Alvaro  24212200    1A    4.0

```

```

Isi Linked List:
Dirga   24212203    4D    3.6
Cintia  24212202    3C    3.5
Bimon   24212201    2B    3.8
Arif    123    1B    4.0
Rizky   124    1G    3.7
Alvaro  24212200    1A    4.0

```

2.2 Modifikasi Elemen pada Single Linked List

Didalam praktikum ini, kita akan mempraktekkan bagaimana mengakses elemen, mendapatkan indeks dan melakukan penghapusan data pada Single Linked List.:

2.2.1 Langkah-langkah Percobaan

1. Implementasikan method untuk mengakses data dan indeks pada linked list
2. Tambahkan method untuk mendapatkan data pada indeks tertentu pada class Single Linked List


```

79     public void getData(int index) {
80         NodeMahasiswa11 tmp = head;
81         for (int i = 0; i < index; i++) {
82             tmp = tmp.next;
83         }
84         tmp.data.tampilInformasi();
85     }
86 }

```

3. Implementasikan method indexOf.

```

87     public int indexOf(String key) {
88         NodeMahasiswa11 tmp = head;
89         int index = 0;
90         while (tmp != null && !tmp.data.nama.equalsIgnoreCase(key)) {
91             tmp = tmp.next;
92             index++;
93         }
94         if (tmp == null) {
95             return -1;
96         } else {
97             return index;
98         }
99     }

```

4. Tambahkan method removeFirst pada class SingleLinkedList

```

100    public void removeFirst() {
101        if (isEmpty()) {
102            System.out.println(x:"Linked List masih kosong, tidak dapat dihapus");
103        } else if (head == tail) {
104            head = tail = null;
105        } else {
106            head = head.next;
107        }
108    }

```

5. Tambahkan method untuk menghapus data pada bagian belakang pada class SingleLinkedList

```

109    public void removeLast() {
110        if (isEmpty()) {
111            System.out.println(x:"Linked List masih kosong, tidak dapat dihapus");
112        } else if (head == tail) {
113            head = tail = null;
114        } else {
115            NodeMahasiswa11 temp = head;
116            while (temp.next != tail) {
117                temp = temp.next;
118            }
119            temp.next = null;
120            tail = temp;
121        }
122    }

```

6. Sebagai langkah berikutnya, akan diimplementasikan method remove

```

123 public void remove(String key) {
124     if (isEmpty()) {
125         System.out.println(x:"Linked List masih kosong, tidak dapat dihapus");
126     } else {
127         NodeMahasiswa11 temp = head;
128         while (temp != null) {
129             if ((temp.data.nama.equalsIgnoreCase(key)) && (temp == head)) {
130                 this.removeFirst();
131                 break;
132             } else if (temp.data.nama.equalsIgnoreCase(key)) {
133                 temp.next = temp.next.next;
134                 if (temp.next == null) {
135                     tail = temp;
136                 }
137             }
138             temp = temp.next;
139         }
140     }
141 }

```

7. Implementasi method untuk menghapus node dengan menggunakan index

```

142 public void removeAt(int index) {
143     if (index == 0) {
144         removeFirst();
145     } else {
146         NodeMahasiswa11 temp = head;
147         for (int i = 0; i < index - 1; i++) {
148             temp = temp.next;
149         }
150         temp.next = temp.next.next;
151         if (temp.next == null) {
152             tail = temp;
153         }
154     }
155 }

```

8. Kemudian, coba lakukan pengaksesan dan penghapusan data di method main pada class SLLMain dengan menambahkan kode berikut

```

22 System.out.println(x:"data index 1 : ");
23 sll.getData(index:1);
24
25 System.out.println("data mahasiswa an Bimon berada pada index : " + sll.indexOf(key:"Bimon"));
26 System.out.println();
27
28 sll.removeFirst();
29 sll.removeLast();
30 sll.print();
31 sll.removeAt(index:0);
32 sll.print();

```

9. Jalankan class SLLMain

2.2.2 Verifikasi Hasil Percobaan

Cocokkan hasil compile kode program anda dengan gambar berikut ini

```
data index 1 :
Cintia      22212202      3C      3.5
data mahasiswa an Bimon berada pada index : 2

Isi Linked List:
Cintia      22212202      3C      3.5
Bimon      23212201      2B      3.8

Isi Linked List:
Bimon      23212201      2B      3.8
```

```
data index 1 :
Cintia  24212202      3C      3.5
data mahasiswa an Bimon berada pada index : 2

Isi Linked List:
Cintia  24212202      3C      3.5
Bimon  24212201      2B      3.8

Isi Linked List:
Bimon  24212201      2B      3.8
```

2.2.3 Pertanyaan

1. Mengapa digunakan keyword break pada fungsi remove? Jelaskan!

Jika key yang di temukan itu pada node pertama, fungsi break itu menghentikan paksa pencarian karena key sudah di temukan pada node pertama

2. Jelaskan kegunaan kode dibawah pada method remove

```
1 temp.next = temp.next.next;
2 if (temp.next == null) {
3     tail = temp;
4 }
```

Temp.next = temp.next.next;

Pointer pada node temp menuju ke node yang di tuju tetapi melompati 1 node

Jika pointer node temp itu null berapa node temp itu merupakan tail

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

- g. Menampilkan antrian terdepan dan antrian paling akhir
- h. Menampilkan jumlah mahasiswa yang masih mengantre.

```
1 package Jobsheet11;
2
3 public class Mahasiswa {
4     String nim;
5     String nama;
6     String prodi;
7     String kelas;
8
9     public Mahasiswa(String nim, String nama, String prodi, String kelas) {
10         this.nim = nim;
11         this.nama = nama;
12         this.prodi = prodi;
13         this.kelas = kelas;
14     }
15
16     public void tampilkanData() {
17         System.out.println(nim + " - " + nama + " - " + prodi + " - " + kelas);
18     }
19
20 }
```

```
1 package Jobsheet11;
2
3 public class SLLAntrianLayanan {
4     NodeMahasiswa head;
5     NodeMahasiswa tail;
6
7     int size = 0;
8     int kapasitas = 10;
9
10    public SLLAntrianLayanan() {
11        head = null;
12        tail = null;
13    }
14
15    public void daftarMahasiswa(Mahasiswa input) {
16        if (size < kapasitas) {
17            addLast(input);
18            System.out.println("Mahasiswa " + input.nama + " telah ditambahkan ke antrian.");
19        } else {
20            System.out.println(x:"Antrian sudah penuh, tidak dapat menambahkan mahasiswa baru.");
21        }
22    }
23
24
25
26    public boolean isEmpty() {
27        return head == null;
28    }
29
30    public void isFull() {
31        if (size >= kapasitas) {
32            System.out.println(x:"Antrian sudah penuh.");
33        } else {
34            System.out.println(x:"Antrian masih tersedia.");
35        }
36    }
37
38    public void clear() {
39        head = null;
40        tail = null;
41        size = 0;
42    }
43 }
```

```

43 public void addLast(Mahasiswa input) {
44     NodeMahasiswa newNode = new NodeMahasiswa(input, next:null);
45     if (isEmpty()) {
46         head = newNode;
47         tail = newNode;
48     } else {
49         tail.next = newNode;
50         tail = newNode;
51     }
52     size++;
53 }
54
55 public void removeFirst() {
56     if (isEmpty()) {
57         System.out.println(x:"Linked List masih kosong, tidak dapat dihapus");
58     } else if (head == tail) {
59         head = tail = null;
60     } else {
61         head = head.next;
62     }
63     size--;
64 }
65
66 public void lihatTerdepan() {
67     if (isEmpty()) {
68         System.out.println(x:"Antrian kosong");
69     } else {
70         System.out.print(s:"Mahasiswa terdepan: ");
71         head.data.tampilkanData();
72     }
73 }
74
75 public void lihatBelakang() {
76     if (isEmpty()) {
77         System.out.println(x:"Antrian kosong");
78     } else {
79         System.out.print(s:"Mahasiswa di belakang: ");
80         tail.data.tampilkanData();
81     }
82 }
83
84 public int getJumlahAntrian() {
85     return size;
86 }

```

```

87 public void tampilkanSemua() {
88     if (isEmpty()) {
89         System.out.println(x:"Antrian kosong");
90         return;
91     }
92     System.out.println(x:"Daftar Mahasiswa dalam antrian:");
93     NodeMahasiswa temp = head;
94     int index = 1;
95     while (temp != null) {
96         System.out.print(index + ". ");
97         temp.data.tampilkanData();
98         temp = temp.next;
99         index++;
100     }
101 }

```

```

1 package Jobsheet11;
2 import java.util.Scanner;
3 public class MainAntrianLayanan {
4     Run | Debug
5     public static void main(String[] args) {
6         Scanner input = new Scanner(System.in);
7         SLLAntrianLayanan antrian = new SLLAntrianLayanan();
8         int pilihan;
9
10        do {
11            System.out.println(x:"Menu Antrian Layanan:");
12            System.out.println(x:"1. daftar Mahasiswa");
13            System.out.println(x:"2. Kosongkan Antrian");
14            System.out.println(x:"3. Memanggil Antrian");
15            System.out.println(x:"4. Lihat Antrian Terdepan");
16            System.out.println(x:"5. Lihat Antrian Akhir");
17            System.out.println(x:"6. Jumlah Mahasiswa masih dalam antrian");
18            System.out.println(x:"7. Lihat Semua Antrian");
19            System.out.println(x:"0. Keluar");
20            System.out.print(s:"Pilih menu: ");
21            pilihan = input.nextInt();

```

```

22        switch (pilihan) {
23            case 1:
24                System.out.print(s:"Masukkan NIM: ");
25                String nim = input.next();
26                System.out.print(s:"Masukkan Nama: ");
27                String nama = input.next();
28                System.out.print(s:"Masukkan Prodi: ");
29                String prodi = input.next();
30                System.out.print(s:"Masukkan Kelas: ");
31                String kelas = input.next();
32                Mahasiswa mhs = new Mahasiswa(nim, nama, prodi, kelas);
33                antrian.daftarMahasiswa(mhs);
34                break;
35            case 2:
36                antrian.clear();
37                System.out.println(x:"Antrian telah dikosongkan.");
38                break;
39            case 3:
40                antrian.removeFirst();
41                System.out.println(x:"Antrian terdepan telah dipanggil.");
42                break;
43            case 4:
44                antrian.lihatTerdepan();
45                break;
46            case 5:
47                antrian.lihatBelakang();
48                break;
49            case 6:
50                System.out.println("Jumlah mahasiswa dalam antrian: " + antrian.getJumlahAntrian());
51                break;
52            case 7:
53                System.out.println(x:"Daftar Antrian:");
54                antrian.tampilkanSemua();
55                break;
56            case 0:
57                System.out.println(x:"Keluar dari program.");
58                break;
59            default:
60                System.out.println(x:"Pilihan tidak valid.");
61        }
62    } while (pilihan != 0);
63 }

```

Menu Antrian Layanan:

1. daftar Mahasiswa
2. Kosongkan Antrian
3. Memanggil Antrian
4. Lihat Antrian Terdepan
5. Lihat Antrian Akhir
6. Jumlah Mahasiswa masih dalam antrian
7. Lihat Semua Antrian
0. Keluar

Pilih menu: 1

Masukkan NIM: 123

Masukkan Nama: Aldi

Masukkan Prodi: TI

Masukkan Kelas: 1A

Mahasiswa Aldi telah ditambahkan ke antrian.

Menu Antrian Layanan:

1. daftar Mahasiswa
2. Kosongkan Antrian
3. Memanggil Antrian
4. Lihat Antrian Terdepan
5. Lihat Antrian Akhir
6. Jumlah Mahasiswa masih dalam antrian
7. Lihat Semua Antrian
0. Keluar

Pilih menu: 1

Masukkan NIM: 124

Masukkan Nama: Bobi

Masukkan Prodi: TI

Masukkan Kelas: 1G

Mahasiswa Bobi telah ditambahkan ke antrian.

Menu Antrian Layanan:

1. daftar Mahasiswa
2. Kosongkan Antrian
3. Memanggil Antrian
4. Lihat Antrian Terdepan
5. Lihat Antrian Akhir
6. Jumlah Mahasiswa masih dalam antrian
7. Lihat Semua Antrian
0. Keluar

Pilih menu: 7

Daftar Antrian:

Daftar Mahasiswa dalam antrian:

1. 123 - Aldi - TI - 1A
2. 124 - Bobi - TI - 1G

Menu Antrian Layanan:

1. daftar Mahasiswa
2. Kosongkan Antrian
3. Memanggil Antrian
4. Lihat Antrian Terdepan
5. Lihat Antrian Akhir
6. Jumlah Mahasiswa masih dalam antrian
7. Lihat Semua Antrian
0. Keluar

Pilih menu: 6

Jumlah mahasiswa dalam antrian: 2

Menu Antrian Layanan:

1. daftar Mahasiswa
2. Kosongkan Antrian
3. Memanggil Antrian
4. Lihat Antrian Terdepan
5. Lihat Antrian Akhir
6. Jumlah Mahasiswa masih dalam antrian
7. Lihat Semua Antrian
0. Keluar

Pilih menu: 5

Mahasiswa di belakang: 124 - Bobi - TI - 1G

Menu Antrian Layanan:

1. daftar Mahasiswa
2. Kosongkan Antrian
3. Memanggil Antrian
4. Lihat Antrian Terdepan
5. Lihat Antrian Akhir
6. Jumlah Mahasiswa masih dalam antrian
7. Lihat Semua Antrian
0. Keluar

Pilih menu: 4

Mahasiswa terdepan: 123 - Aldi - TI - 1A

Menu Antrian Layanan:

1. daftar Mahasiswa
2. Kosongkan Antrian
3. Memanggil Antrian
4. Lihat Antrian Terdepan
5. Lihat Antrian Akhir
6. Jumlah Mahasiswa masih dalam antrian
7. Lihat Semua Antrian
0. Keluar

Pilih menu: 3

Antrian terdepan telah dipanggil.

Menu Antrian Layanan:

1. daftar Mahasiswa
2. Kosongkan Antrian
3. Memanggil Antrian
4. Lihat Antrian Terdepan
5. Lihat Antrian Akhir
6. Jumlah Mahasiswa masih dalam antrian
7. Lihat Semua Antrian
0. Keluar

Pilih menu: 6

Jumlah mahasiswa dalam antrian: 1

Menu Antrian Layanan:

1. daftar Mahasiswa
2. Kosongkan Antrian
3. Memanggil Antrian
4. Lihat Antrian Terdepan
5. Lihat Antrian Akhir
6. Jumlah Mahasiswa masih dalam antrian
7. Lihat Semua Antrian
0. Keluar

Pilih menu: 2

Antrian telah dikosongkan.

Menu Antrian Layanan:

1. daftar Mahasiswa
2. Kosongkan Antrian
3. Memanggil Antrian
4. Lihat Antrian Terdepan
5. Lihat Antrian Akhir
6. Jumlah Mahasiswa masih dalam antrian
7. Lihat Semua Antrian
0. Keluar

Pilih menu: 6

Jumlah mahasiswa dalam antrian: 0

Menu Antrian Layanan:

1. daftar Mahasiswa
2. Kosongkan Antrian
3. Memanggil Antrian
4. Lihat Antrian Terdepan
5. Lihat Antrian Akhir
6. Jumlah Mahasiswa masih dalam antrian
7. Lihat Semua Antrian
0. Keluar

Pilih menu: 7

Daftar Antrian:

Antrian kosong