POLITEKNIK NEGERI MALANG

TEKNOLOGI INFORMASI

TEKNIK INFORMATIKA

Mohammad Ariq Baihaqi

244107020161

TI – 1A

16

2. Praktikum

2.1.1 Verifikasi Hasil Percobaan

Class Mahasiswa16

```
package Minggu12;


public class Mahasiswa16 {
    String nim;
    String nama;
    String kelas;
    double ipk;

    Mahasiswa16() {


    }

    Mahasiswa16(String nm, String name, String kls, double ip) {
        this.nim = nm;
        this.nama = name;
        this.ipk = ip;
        this.kelas = kls;
    }

    public void tampilInformasi() {
        System.out.println("Nama: " + nama);
        System.out.println("NIM: " + nim);
        System.out.println("Kelas: " + kelas);
        System.out.println("IPK: " + ipk);
    }
}
```

Class Node16

```
package Minggu12;
public class Node16 {
    Mahasiswa16 data;
    Node16 next;

    public Node16(Mahasiswa16 data, Node16 next) {
        this.data = data;
        this.next = next;
    }
}
```

Class SingleLinkList16

```
package Minggu12;
public class SingleLinkedList16 {
    Node16 head;
    Node16 tail;

    boolean isEmpty() {
        return head == null;
    }

    public void print() {
        if (!isEmpty()) {
            Node16 tmp = head;
            System.out.print("Isi Linked List:\t");
            while (tmp != null) {
                tmp.data.tampilInformasi();
                tmp = tmp.next;
            }
            System.out.println(" ");
        } else {
            System.out.println("Linked List Kosong");
        }
    .
```

```
public void addFirst(Mahasiswa16 data) {
        Node16 ndInput = new Node16(data, null);
        if (isEmpty()) {
            head = ndInput;
            tail = ndInput;
        } else {
            ndInput.next = head;
            head = ndInput;
        }
    }

    public void addLast(Mahasiswa16 data) {
        Node16 ndInput = new Node16(data, null);
        if (isEmpty()) {
            head = ndInput;
            tail = ndInput;
        } else {
            tail.next = ndInput;
            tail = ndInput;
        }
    }

    public void insertAfter(String key, Mahasiswa16 input) {
        Node16 ndInput = new Node16(input, null);
        Node16 temp = head;
        do {
            if (temp.data.nama.equalsIgnoreCase(key)) {
                ndInput.next = temp.next;
                temp.next = ndInput;
                if (temp == tail) {
                    tail = ndInput;
                }
                break;
            }
            temp = temp.next;
        } while (temp != null);
    }
```

```
public void InsertAt(int index, Mahasiswa16 input) {
        if (index < 0) {
            System.out.println("Index salah");
        } else if(index == 0) {
            addFirst(input);
        } else {
            Node16 temp = head;
            for (int i = 0; i < index -1; i++) {
                temp = temp.next;
            }
            temp.next = new Node16(input, temp.next);
            if (temp.next == null) {
                tail = temp.next;
            }
        }
    }
}
```

Class SLLMain16

```
public void InsertAt(int index, Mahasiswa16 input) {
        if (index < 0) {
            System.out.println("Index salah");
        } else if(index == 0) {
            addFirst(input);
        } else {
            Node16 temp = head;
            for (int i = 0; i < index -1; i++) {
                temp = temp.next;
            }
            temp.next = new Node16(input, temp.next);
            if (temp.next == null) {
                tail = temp.next;
            }
        }
    }
}
```

OUTPUT

```
Linked List Kosong

== Tambah di Awal ==
Isi Linked List:        Nama: Dirga
NIM: 21212203
Kelas: 4D
IPK: 3.6


== Tambah di Akhir ==
Isi Linked List:        Nama: Dirga
NIM: 21212203
Kelas: 4D
IPK: 3.6
Nama: Alvaro
NIM: 24212200
Kelas: 1A
IPK: 4.0


== Tambah di Index ke-1 ==
Isi Linked List:        Nama: Dirga
NIM: 21212203
Kelas: 4D
IPK: 3.6
Nama: Cintia
NIM: 22212202
Kelas: 3C
IPK: 3.5
Nama: Alvaro
NIM: 24212200
Kelas: 1A
IPK: 4.0


== Tambah di Index ke-2 ==
Isi Linked List:        Nama: Dirga
NIM: 21212203
Kelas: 4D
IPK: 3.6
Nama: Cintia
```

2.1.2 Pertanyaan

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

- Karena saat list.print(); dipanggil belumada data yang dimasukkan ke dalam linekd list

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

- Membaca dan mencetak isi list
- Mencari node tertentu berdasarkan kondisi tertentu
- Menemukan lokasi yang tepat untuk meyisipkan data baru

3. Lakukan modifikasi agar data dapat ditambahkan dari keyboard!

```
Kelas  : 1F
IPK    : 2
Linked list setelah penambahan:
Isi Linked List:        Nama: Ariq
NIM: 243
Kelas: 1B
IPK: 3.0
Nama: Hanif
NIM: 342
Kelas: 1A
IPK: 4.0
Nama: Nuril
NIM: 2131
Kelas: 1F
IPK: 2.0


Mahasiswa ke-4
Nama   : Ilham
NIM    : 312
Kelas  : 1H
IPK    : 4
Linked list setelah penambahan:
Isi Linked List:        Nama: Ariq
NIM: 243
Kelas: 1B
IPK: 3.0
Nama: Hanif
NIM: 342
Kelas: 1A
IPK: 4.0
Nama: Nuril
NIM: 2131
Kelas: 1F
IPK: 2.0
Nama: Ilham
NIM: 312
Kelas: 1H
IPK: 4.0
```

2.2 Modifikasi Elemen pada Single Linked List

2.2.2 Verifikasi Hasil Percobaan

```
public void getData(int index) {
        Node16 tmp = head;
        for (int i = 0; i < index; i++) {
            tmp = tmp.next;
        }
        tmp.data.tampilInformasi();
    }

    public int indexOf(String key) {
        Node16 tmp = head;
        int index = 0;
        while (tmp != null && !tmp.data.nama.equalsIgnoreCase(key)) {
            tmp = tmp.next;
            index++;
        }
        if (tmp == null) {
            return -1;
        } else {
            return index;
        }
    }

    public void removeFirst() {
        if (isEmpty()) {
            System.out.println("Linked List Masih Kosong, tidak dapat dihapus");
        } else if (head == tail) {
            head = tail = null;
        } else {
            head = head.next;
        }
    }
```

```
public void removeLast() {
    if (isEmpty()) {
        System.out.println("Linked List masih Kosong, tidak dapat dihapus!");
    } else if (head == tail) {
        head = tail = null;
    } else {
        Node16 temp = head;
        while (temp.next != tail) {
            temp = temp.next;
        }
        temp.next = null;
        tail = temp;
    }
}
    public void remove(String key) {
    if (isEmpty()) {
        System.out.println("Linked List masih Kosong, tidak dapat dihapus!");
    } else {
        Node16 temp = head;
        while (temp != null) {
            if ((temp.data.nama.equalsIgnoreCase(key)) && (temp == head)) {
                this.removeFirst();
                break;
            } else if (temp.data.nama.equalsIgnoreCase(key)) {
                temp.next = temp.next.next;
                if (temp.next == null) {
                    tail = temp;
                }
                break;
            }
            temp = temp.next;
        }
    }
}
```

```
public void removeAt(int index) {
    if (index == 0) {
        removeFirst();
    } else {
        Node16 temp = head;
        for (int i = 0; i < index - 1; i++) {
            temp = temp.next;
        }

        temp.next = temp.next.next;
        if (temp.next == null) {
            tail = temp;
        }
    }
}
```

Class SLLMain16

```
System.out.println("data index 1 :");
    list.getData(1);

    System.out.println("data mahasiswa an Bimon berada pada index :" +
list.indexOf("Bimon"));
    System.out.println();

    list.removeFirst();
    list.removeLast();
    list.print();
    list.removeAt(0);
    list.print();
```

OUTPUT

```
Data index ke-1:
Nama: Cintia
NIM: 22212202
Kelas: 3C
IPK: 3.5

Data mahasiswa bernama Bimon berada pada index:
2

== Setelah removeFirst ==
Isi Linked List:        Nama: Cintia
NIM: 22212202
Kelas: 3C
IPK: 3.5
Nama: Bimon
NIM: 23212201
Kelas: 2B
IPK: 3.8
Nama: Alvaro
NIM: 24212200
Kelas: 1A
IPK: 4.0


== Setelah removeLast ==
Isi Linked List:        Nama: Cintia
NIM: 22212202
Kelas: 3C
IPK: 3.5
Nama: Bimon
NIM: 23212201
Kelas: 2B
IPK: 3.8


== Setelah removeAt index 0 ==
Isi Linked List:        Nama: Bimon
NIM: 23212201
Kelas: 2B
IPK: 3.8
```

2.2.3 Pertanyaan

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

- Digunakan untuk menghentikan perulangan secaa paksa setelah data yang ingin dihapus telah ditemukan dan diproses

2. Jelaskan kegunaan kode dibawah pada method remove

Temp.next = temp.next.next

```
If (temp.next == null) {
     Tail = temp;
}
```

- Untuk menghapus node yang dituju dari Linked list

3. Tugas

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

a. Implementasi antrian menggunakan Queue berbasis Linked List!

b. Program merupakan proyek baru bukan modifikasi dari percobaan

c. Ketika seorang mahasiswa akan mengantri, maka dia harus mendaftarkan datanya

d. Cek antrian kosong, Cek antrian penuh, Mengosongkan antrian.

e. Menambahkan antrian

f. Memanggil antrian

g. Menampilkan antrian terdepan dan antrian paling akhir

h. Menampilkan jumlah mahasiswa yang masih mengantre.

Class Mahasiswa16

```
package Minggu12.Tugas;
class Mahasiswa {
    String nama;
    String nim;
    Mahasiswa next;

    public Mahasiswa(String nama, String nim) {
        this.nama = nama;
        this.nim = nim;
        this.next = null;
    }
}

class AntrianMahasiswa {
    private Mahasiswa head, tail;
    private int size;

    public AntrianMahasiswa() {
        this.head = this.tail = null;
        this.size = 0;
    }

    // Cek antrian kosong
    public boolean isEmpty() {
        return head == null;
    }

    // Cek antrian penuh
    public boolean isFull() {
        return false;
    }
```

```
// Tambah antrian
    public void enqueue(String nama, String nim) {
        Mahasiswa baru = new Mahasiswa(nama, nim);
        if (isEmpty()) {
            head = tail = baru;
        } else {
            tail.next = baru;
            tail = baru;
        }
        size++;
        System.out.println("Mahasiswa berhasil ditambahkan ke antrian.");
    }

    // Panggil antrian
    public void dequeue() {
        if (isEmpty()) {
            System.out.println("Antrian kosong. Tidak ada mahasiswa yang bisa
dipanggil.");
        } else {
            System.out.println("Memanggil: " + head.nama + " (NIM: " + head.nim +
")");
            head = head.next;
            if (head == null) {
                tail = null;
            }
            size--;
        }
    }

    //  mahasiswa di antrian terdepan
    public void tampilDepan() {
        if (isEmpty()) {
            System.out.println("Antrian kosong.");
        } else {
            System.out.println("Antrian terdepan: " + head.nama + " (NIM: " +
head.nim + ")");
        }
    }
```

```
//  mahasiswa di antrian terakhir
    public void tampilBelakang() {
        if (isEmpty()) {
            System.out.println("Antrian kosong.");
        } else {
            System.out.println("Antrian terakhir: " + tail.nama + " (NIM: " +
tail.nim + ")");
        }
    }

    //  jumlah mahasiswa dalam antrian
    public void tampilJumlah() {
        System.out.println("Jumlah mahasiswa dalam antrian: " + size);
    }

    // Kosongkan antrian
    public void kosongkan() {
        head = tail = null;
        size = 0;
        System.out.println("Antrian telah dikosongkan.");
    }

    //  seluruh isi antrian
    public void tampilkanAntrian() {
        if (isEmpty()) {
            System.out.println("Antrian kosong.");
        } else {
            System.out.println("Daftar Mahasiswa dalam Antrian:");
            Mahasiswa current = head;
            int nomor = 1;
            while (current != null) {
                System.out.println(nomor++ + ". " + current.nama + " (NIM: " +
current.nim + ")");
                current = current.next;
            }
        }
    }
}
```

Class LayananKemahasiswaan16

```
package Minggu12.Tugas;
import java.util.Scanner;
public class LayananKemahasiswaan16 {
    public static void main(String[] args) {
        Scanner sc = new Scanner (System.in);
        AntrianMahasiswa antrian = new AntrianMahasiswa();
        int pilihan;

        do {
            System.out.println("\n=== MENU LAYANAN UNIT KEMAHASISWAAN ===");
            System.out.println("1. Tambah Antrian Mahasiswa");
            System.out.println("2. Panggil Antrian");
            System.out.println("3. Tampilkan Antrian Terdepan");
            System.out.println("4. Tampilkan Antrian Terakhir");
            System.out.println("5. Tampilkan Jumlah Mahasiswa dalam Antrian");
            System.out.println("6. Tampilkan Seluruh Antrian");
            System.out.println("7. Kosongkan Antrian");
            System.out.println("0. Keluar");
            System.out.print("Pilih menu: ");
            pilihan = sc.nextInt();
            sc.nextLine();

            switch (pilihan) {
                case 1:
                    System.out.print("Masukkan nama mahasiswa: ");
                    String nama = sc.nextLine();
                    System.out.print("Masukkan NIM mahasiswa: ");
                    String nim = sc.nextLine();
                    antrian.enqueue(nama, nim);
                    break;
                case 2:
                    antrian.dequeue();
                    break;
```

```
case 3:
                    antrian.tampilDepan();
                    break;
                case 4:
                    antrian.tampilBelakang();
                    break;
                case 5:
                    antrian.tampilJumlah();
                    break;
                case 6:
                    antrian.tampilkanAntrian();
                    break;
                case 7:
                    antrian.kosongkan();
                    break;
                case 0:
                    System.out.println("Terima kasih. Program selesai.");
                    break;
                default:
                    System.out.println("Pilihan tidak valid.");
            }
        } while (pilihan != 0);


    }
}
```

OUTPUT

```
=== MENU LAYANAN UNIT KEMAHASISWAAN ===
1. Tambah Antrian Mahasiswa
2. Panggil Antrian
3. Tampilkan Antrian Terdepan
4. Tampilkan Antrian Terakhir
5. Tampilkan Jumlah Mahasiswa dalam Antrian
6. Tampilkan Seluruh Antrian
7. Kosongkan Antrian
0. Keluar
Pilih menu: 1
Masukkan nama mahasiswa: Ariq
Masukkan NIM mahasiswa: 20
Mahasiswa berhasil ditambahkan ke antrian.

=== MENU LAYANAN UNIT KEMAHASISWAAN ===
1. Tambah Antrian Mahasiswa
2. Panggil Antrian
3. Tampilkan Antrian Terdepan
4. Tampilkan Antrian Terakhir
5. Tampilkan Jumlah Mahasiswa dalam Antrian
6. Tampilkan Seluruh Antrian
7. Kosongkan Antrian
0. Keluar
Pilih menu: 1
Masukkan nama mahasiswa: Hanif
Masukkan NIM mahasiswa: 21
Mahasiswa berhasil ditambahkan ke antrian.

=== MENU LAYANAN UNIT KEMAHASISWAAN ===
1. Tambah Antrian Mahasiswa
2. Panggil Antrian
3. Tampilkan Antrian Terdepan
4. Tampilkan Antrian Terakhir
5. Tampilkan Jumlah Mahasiswa dalam Antrian
6. Tampilkan Seluruh Antrian
7. Kosongkan Antrian
0. Keluar
Pilih menu: 1
Masukkan nama mahasiswa: Nuril
Masukkan NIM mahasiswa: 23
```

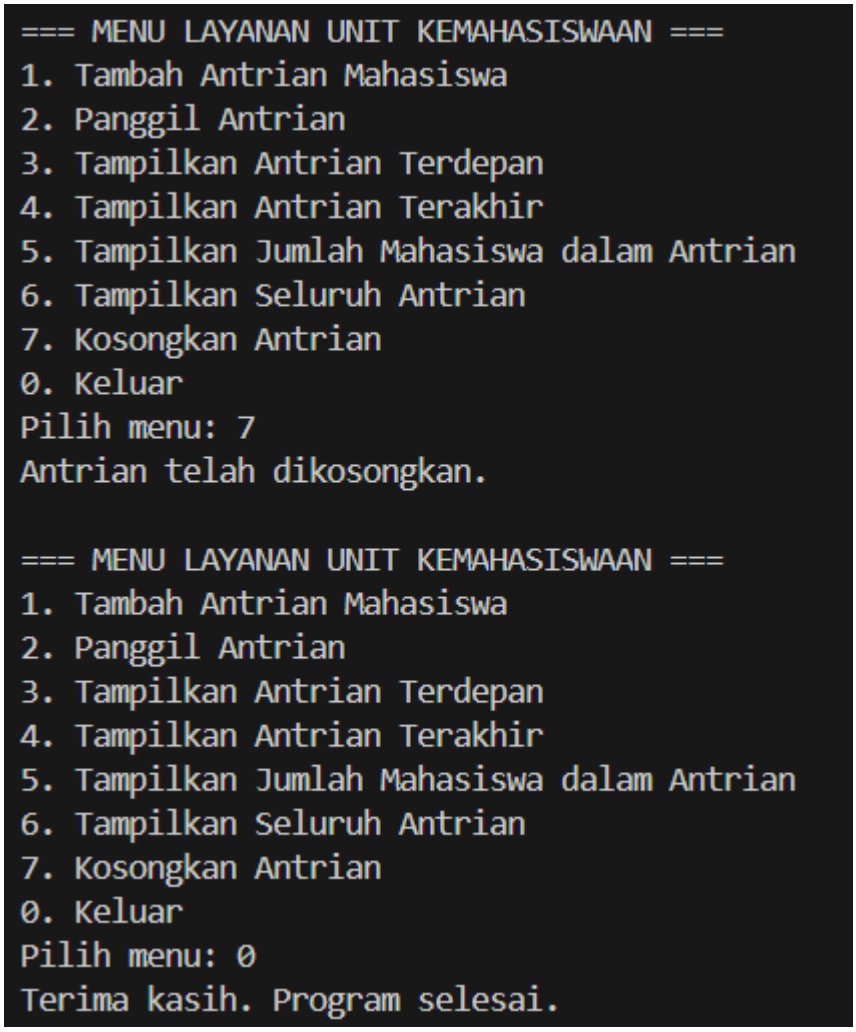
```
=== MENU LAYANAN UNIT KEMAHASISWAAN ===
1. Tambah Antrian Mahasiswa
2. Panggil Antrian
3. Tampilkan Antrian Terdepan
4. Tampilkan Antrian Terakhir
5. Tampilkan Jumlah Mahasiswa dalam Antrian
6. Tampilkan Seluruh Antrian
7. Kosongkan Antrian
0. Keluar
Pilih menu: 6
Daftar Mahasiswa dalam Antrian:
1. Ariq (NIM: 20)
2. Hanif (NIM: 21)
3. Nuril (NIM: 23)
4. Ilham (NIM: 24)
```

```
=== MENU LAYANAN UNIT KEMAHASISWAAN ===
1. Tambah Antrian Mahasiswa
2. Panggil Antrian
3. Tampilkan Antrian Terdepan
4. Tampilkan Antrian Terakhir
5. Tampilkan Jumlah Mahasiswa dalam Antrian
6. Tampilkan Seluruh Antrian
7. Kosongkan Antrian
0. Keluar
Pilih menu: 5
Jumlah mahasiswa dalam antrian: 4

=== MENU LAYANAN UNIT KEMAHASISWAAN ===
1. Tambah Antrian Mahasiswa
2. Panggil Antrian
3. Tampilkan Antrian Terdepan
4. Tampilkan Antrian Terakhir
5. Tampilkan Jumlah Mahasiswa dalam Antrian
6. Tampilkan Seluruh Antrian
7. Kosongkan Antrian
0. Keluar
Pilih menu: 3
Antrian terdepan: Ariq (NIM: 20)
```

```
=== MENU LAYANAN UNIT KEMAHASISWAAN ===
1. Tambah Antrian Mahasiswa
2. Panggil Antrian
3. Tampilkan Antrian Terdepan
4. Tampilkan Antrian Terakhir
5. Tampilkan Jumlah Mahasiswa dalam Antrian
6. Tampilkan Seluruh Antrian
7. Kosongkan Antrian
0. Keluar
Pilih menu: 4
Antrian terakhir: Ilham (NIM: 24)

=== MENU LAYANAN UNIT KEMAHASISWAAN ===
1. Tambah Antrian Mahasiswa
2. Panggil Antrian
3. Tampilkan Antrian Terdepan
4. Tampilkan Antrian Terakhir
5. Tampilkan Jumlah Mahasiswa dalam Antrian
6. Tampilkan Seluruh Antrian
7. Kosongkan Antrian
0. Keluar
Pilih menu: 2
Memanggil: Ariq (NIM: 20)
```



Link Github : https://github.com/Ariqq16?tab=repositories