

**Laporan Hasil Praktikum
Algoritma dan Struktur Data
Jobsheet 11 : Linked List**



Muhammad Firman Aditiasmara

244107020094

TI-1E

Program Studi Teknik Informatika

Jurusan Teknologi Informasi

Politeknik Negeri Malang

2025

1. Tujuan Praktikum

Setelah melakukan materi praktikum ini, mahasiswa mampu:

1. Membuat struktur data linked list
2. Membuat linked list pada program
3. Membedakan permasalahan apa yang dapat diselesaikan menggunakan linked list

2. Praktikum

2.1 Percobaan 1: Pembuatan Single Linked List

2.1.1 Langkah-langkah Percobaan

1. Membuat class Mahasiswa18

```
public class Mahasiswa18 {  
  
    String nim;  
    String nama;  
    String kelas;  
    double ipk;  
  
    public Mahasiswa18() {  
  
    }  
  
    public Mahasiswa18(String nm, String name, String kls, double ip) {  
        this.nim = nm;  
        this.nama = name;  
        this.kelas = kls;  
        this.ipk = ip;  
    }  
  
    public void tampilInformasi(){  
        System.out.println(nama+ "\t\t" + nim + "\t\t" + kelas +  
        "\t\t" + ipk );  
    }  
}
```

2. Membuat class NodeMahasiswa18 dan memberikan isinya

```
public class NodeMahasiswa18 {  
    Mahasiswa18 data;  
    NodeMahasiswa18 next;  
  
    public NodeMahasiswa18 (Mahasiswa18 data,  
    NodeMahasiswa18 next){  
        this.data = data;  
        this.next = next;  
    }  
}
```

3. Membuat class SingleLinkedList18 dan memberikan method

```
package TUGAS;

public class AntrianMahasiswa {
    int size = 0;
    int max = 10;
    NodeMahasiswa front;
    NodeMahasiswa rear;

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

    boolean isFull() {
        return (size >= max);
    }

    public void clear() {
        if (!isEmpty()) {
            front = rear = null;
            size = 0;
            System.out.println("Antrian berhasil dikosongkan");
        } else {
            System.out.println("Antrian masih kosong");
        }
    }

    public void tambahAntrian(Mahasiswa mhs) {
        if (isFull()) {
            System.out.println("Antrian sudah penuh, tidak dapat menambah data");
            return;
        }
        NodeMahasiswa add = new NodeMahasiswa(mhs);
        if (isEmpty()) {
            front = rear = add;
        } else {
            rear.next = add;
            rear = add;
        }
        size++;
        System.out.println("Mahasiswa " + mhs.nama + " telah ditambahkan ke antrian.");
    }
}
```

```

        public void insertAfter(String key, Mahasiswa18 input) {
            NodeMahasiswa18 ndInput = new NodeMahasiswa18(input,
null);
            NodeMahasiswa18 temp = head;
            do {
                if (temp.data.nama.equalsIgnoreCase(key)) {
                    ndInput.next = temp.next;
                    temp.next = ndInput;
                    if (ndInput.next == null) {
                        tail = ndInput;
                    }
                    break;
                }
                temp = temp.next;
            } while (temp != null);
        }

        public void insertAt(int index, Mahasiswa18 input) {
            if (index < 0) {
                System.out.println("indeks salah");
            } else if (index == 0) {
                addFirst(input);
            } else {
                NodeMahasiswa18 temp = head;
                for (int i = 0; i < index - 1; i++) {
                    temp = temp.next;
                }
                temp.next = new NodeMahasiswa18(input, temp.next);
                if (temp.next.next == null) {
                    tail = temp.next;
                }
            }
        }
    }
}

```

4. Membuat class SLLMain18, membuat fungsi main dan membuat objek dari class SingleLinkedList18. Buat empat object mahasiswa dengan nama mhs1, mhs2, mhs3, mhs4 kemudian isi data setiap object melalui konstruktor.

```

public class SLLMain18 {
    public static void main(String[] args) {

        SingleLinkedList18 sll = new SingleLinkedList18();

        Mahasiswa18 mhs1 = new Mahasiswa18("24212200", "Alvaro",
"1A", 4.0);
        Mahasiswa18 mhs2 = new Mahasiswa18("23212201", "Bimon",
"2B", 3.8);
        Mahasiswa18 mhs3 = new Mahasiswa18("22212202", "Cintia",
"3C", 3.5);
        Mahasiswa18 mhs4 = new Mahasiswa18("21212203", "Dirga",
"4D", 3.6);

        sll.print();
        sll.addFirst(mhs4);
        sll.print();
        sll.addLast(mhs1);
        sll.print();
        sll.insertAfter("Dirga", mhs3);
        sll.insertAt(2, mhs2);
        sll.print();

    }
}

```

2.1.2 Verifikasi Hasil Percobaan

```

C:\Users\user> javac java\juc_18\002\src\18_002\src\Main18.java
C:\Users\user> java -cp .\bin Main18
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

```

2.1.3 Pertanyaan

1. Mengapa hasil compile kode program di baris pertama menghasilkan “Linked List Kosong”?
Jawab :
Pada compile baris pertama menghasilkan kosong karena sll.print yang pertama belum ada isinya, belum memanggil perintah add.

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

Jawab :

Variabel temp digunakan sebagai pointer sementara yang digunakan untuk traversal (penelusuran). Temp digunakan dominan pada penelusuran dan pencarian posisi, atau sebelum perubahan, ini digunakan agar data asli pada node tidak berubah maka digunakan temp untuk melakukan penelusuran.

3. Lakukan modifikasi agar data dapat ditambahkan dari keyboard!

Jawab :

```
// Modifikasi pertanyaan percobaan 1
Scanner input = new Scanner(System.in);
System.out.print("Masukkan jumlah
mahasiswa: ");
int jml = input.nextInt();
input.nextLine();

for (int i = 0; i < jml; i++) {
    System.out.println("Masukkan
mahasiswa ke-" + (i+1));
    System.out.print("NIM: ");
    String nim = input.nextLine();
    System.out.print("Nama: ");
    String nama = input.nextLine();
    System.out.print("Kelas: ");
    String kelas = input.nextLine();
    System.out.print("IPK: ");
    double ipk = input.nextDouble();
    input.nextLine();
    System.out.println();

    Mahasiswa18 mhs = new
    Mahasiswa18(nim, nama, kelas, ipk);
```

Modifikasi pada class SLLMain18.java agar dapat diinputkan oleh pengguna, dan berikut contoh outputnya

```
Masukkan jumlah mahasiswa: 2
Masukkan mahasiswa ke-1
NIM: 12345
Nama: Aditiasmar
Kelas: 1E
IPK: 4,9

Masukkan mahasiswa ke-2
NIM: 1234567
Nama: Adiiiiit
Kelas: 1U
IPK: 3,5
```

2.2 Percobaan 2 : Modifikasi Elemen Pada Single Linked List

2.2.1 Langkah-langkah Percobaan

1. Menambahkan method getData() pada class SingleLinkedList18

```
public void getData (int index) {
    NodeMahasiswa18 tmp = head;
    for (int i = 0; i < index; i++) {
        tmp = tmp.next;
    }
    tmp.data.tampilInformasi();
}
```

2. Membuat method indexOf()

```
public int indexOf(String key) {
    NodeMahasiswa18 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;
    }
}
```

3. Menambahkan method removeFirst() dan removeLast()

```
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 {
        NodeMahasiswa18 temp = head;
        while (temp.next != tail) {
            temp = temp.next;
        }
        temp.next = null;
        tail = temp;
    }
}
```

4. Menambahkan method remove() dan removeAt()

```
public void remove(String key) {
    if (isEmpty()) {
        System.out.println("Linked List masih kosong,
tidak dapat dihapus!");
    } else {
        NodeMahasiswa18 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 {
        NodeMahasiswa18 temp = head;
        for (int i = 0; i < index -1; i++) {
            temp = temp.next;
        }
        temp.next = temp.next.next;
        if (temp.next == null) {
            tail = temp;
        }
    }
}
```

5. Melakukan pengaksesan dan penghapusan data di method main pada class SLLMain

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

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

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


2.2.2 Verifikasi Hasil Percobaan

```
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
```


2.2.3 Pertanyaan

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

Jawab :

Break digunakan pada fungsi remove() yaitu untuk menghentikan loop setelah penghapusan telah selesai dilakukan

2. Jelaskan kegunaan kode dibawah pada method remove



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

Jawab :

- Untuk temp.next = temp.next.next digunakan untuk menentukan yang dihapus dengan melakukan next agar melewati yang tidak dihapus
- Pada kondisi jika temp.next == null artinya setelah menghapus temp.next.next yang berada pada indeks terakhir, maka tail akan berpindah pada node sebelumnya yaitu temp.next

2.3 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 4**
- h. Menampilkan jumlah mahasiswa yang masih mengantre.**

- Kode Program class Mahasiswa.java

```
package TUGAS;

public class Mahasiswa {
    String nama;
    String nim;
    String kelas;

    public Mahasiswa () {

    }

    public Mahasiswa (String nama, String nim, String kelas) {
        this.nama = nama;
        this.nim = nim;
        this.kelas = kelas;
    }
}
```

- Kode Program class NodeMahasiswa.java

```
package TUGAS;

public class NodeMahasiswa {
    Mahasiswa data;
    NodeMahasiswa next;

    public NodeMahasiswa(Mahasiswa data) {
        this.data = data;
        this.next = null;
    }
}
```

- Kode Program class AntrianMahasiswa

```
public boolean isFull() {
    if (size == max) {
        return true;
    } else {
        return false;
    }
}

public void clear() {
    if (!IsEmpty()) {
        front = rear = -1;
        size = 0;
        System.out.println("Queue berhasil dikosongkan");
    } else {
        System.out.println("Queue masih kosong");
    }
}

public void tambahAntrian(Mahasiswa mhs) {
    if (isFull()) {
        System.out.println("Antrian penuh, tidak dapat menambah mahasiswa.");
        return;
    }
    rear = (rear + 1) % max;
    data[rear] = mhs;
    size++;
    System.out.println(mhs.nama + " berhasil masuk ke antrian.");
}

public void memanggilAtrian() {
    if (size < 2) {
        System.out.println("Antrian kurang dari 2. Minimal 2 mahasiswa untuk proses KRS.");
        return;
    }

    System.out.println("Memproses KRS untuk:");
    for (int i = 0; i < 2; i++) {
        Mahasiswa mhs = data[front];
        mhs.tampil();
        front = (front + 1) % max;
        size--;
        proses++;
    }
}

public void tampilSemua() {
    if (IsEmpty()) {
        System.out.println("Antrian kosong.");
        return;
    }
}
```

```

public void panggilAntrian() {
    if (isEmpty()) {
        System.out.println("Antrian Kosong");
    } else {
        System.out.println("Memanggil antrian: ");
        System.out.println("Nama: " + front.data.nama);
        System.out.println("NIM: " + front.data.nim);
        System.out.println("Kelas: " + front.data.kelas);

        front = front.next;
        size--;
    }
}

public void antrianTerdepan() {
    if (isEmpty()) {
        System.out.println("Antrian kosong.");
    } else {
        System.out.println("Mahasiswa terdepan: " +
front.data.nama + " (" + front.data.nim + ")");
    }
}

public void antrianTerakhir() {
    if (isEmpty()) {
        System.out.println("Antrian kosong.");
    } else {
        System.out.println("Mahasiswa terakhir: " + rear.data.nama
+ " (" + rear.data.nim + ")");
    }
}

public void jmlAntrian () {
    System.out.println("Jumlah dalam antrian: " + size);
}
}

```

```
package TUGAS;

import java.util.Scanner;

public class LayananMahasiswa {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        AntrianMahasiswa list = new AntrianMahasiswa();
        int pilihan;

        do {
            System.out.println("\nMENU LAYANAN UNIT KEMAHASISWAAN");
            System.out.println("1. Cek Antrian");
            System.out.println("2. Tambah Antrian");
            System.out.println("3. Panggil Antrian");
            System.out.println("4. Tampilkan Antrian Terdepan");
            System.out.println("5. Tampilkan Antrian Terakhir");
            System.out.println("6. Tampilkan Jumlah Antrian");
            System.out.println("7. Kosongkan Antrian");
            System.out.println("0. Keluar");
            System.out.print("Pilih Menu: ");
            pilihan = input.nextInt();
            input.nextLine();

            switch (pilihan) {
                case 1:
                    if (!list.isFull()) {
                        System.out.println("Antrian Masih bisa ditambahkan");
                    } else {
                        System.out.println("Antrian Sudah penuh");
                    }
                    break;
                case 2:
                    if (!list.isFull()) {
                        System.out.print("Masukkan nama: ");
                        String nama = input.nextLine();
                        System.out.print("Masukkan NIM: ");
                        String nim = input.nextLine();
                        System.out.print("Masukkan kelas: ");
                        String kelas = input.nextLine();

                        Mahasiswa mhs = new Mahasiswa(nama, nim, kelas);
                        list.tambahAntrian(mhs);
                    } else {
                        System.out.println("Antrian sudah penuh");
                    }
                    break;
                case 3:
                    list.panggilAntrian();
                    break;
                case 4:
                    list.antrianTerdepan();
                    break;
                case 5:
                    list.antrianTerakhir();
                    break;
            }
        } while (pilihan != 0);
    }
}
```

```
package TUGAS;

import java.util.Scanner;

public class LayananMahasiswa {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        AntrianMahasiswa list = new AntrianMahasiswa();
        int pilihan;

        do {
            System.out.println("\nMENU LAYANAN UNIT KEMAHASISWAAN");
            System.out.println("1. Cek Antrian");
            System.out.println("2. Tambah Antrian");
            System.out.println("3. Panggil Antrian");
            System.out.println("4. Tampilkan Antrian Terdepan");
            System.out.println("5. Tampilkan Antrian Terakhir");
            System.out.println("6. Tampilkan Jumlah Antrian");
            System.out.println("7. Kosongkan Antrian");
            System.out.println("0. Keluar");
            System.out.print("Pilih Menu: ");
            pilihan = input.nextInt();
            input.nextLine();

            switch (pilihan) {
                case 1:
                    if (!list.isFull()) {
                        System.out.println("Antrian Masih bisa ditambahkan");
                    } else {
                        System.out.println("Antrian Sudah penuh");
                    }
                    break;
                case 2:
                    if (!list.isFull()) {
                        System.out.print("Masukkan nama: ");
                        String nama = input.nextLine();
                        System.out.print("Masukkan NIM: ");
                        String nim = input.nextLine();
                        System.out.print("Masukkan kelas: ");
                        String kelas = input.nextLine();

                        Mahasiswa mhs = new Mahasiswa(nama, nim, kelas);
                        list.tambahAntrian(mhs);
                    } else {
                        System.out.println("Antrian sudah penuh");
                    }
                    break;
                case 3:
                    list.panggilAntrian();
                    break;
                case 4:
                    list.antrianTerdepan();
                    break;
                case 5:
                    list.antrianTerakhir();
                    break;
            }
        } while (pilihan != 0);
    }
}
```

```
case 6:
    list.jmlAntrian();
    break;
case 7:
    list.clear();
    break;
case 0:
    System.out.println("Program Selesai, Terima
Kasih");
    break;
default:
    System.out.println("Pilihan menu tidak valid");
}
} while (pilihan != 0);
}
```

Hasil Compile kode Program

```
MENU LAYANAN UNIT KEMAHASISWAAN
1. Cek Antrian
2. Tambah Antrian
3. Panggil Antrian
4. Tampilkan Antrian Terdepan
5. Tampilkan Antrian Terakhir
6. Tampilkan Jumlah Antrian
7. Kosongkan Antrian
0. Keluar
Pilih Menu: 2
Masukkan nama: Aditiasmara
Masukkan NIM: 23144
Masukkan kelas: 1E
Mahasiswa Aditiasmara telah ditambahkan ke antrian

MENU LAYANAN UNIT KEMAHASISWAAN
1. Cek Antrian
2. Tambah Antrian
3. Panggil Antrian
4. Tampilkan Antrian Terdepan
5. Tampilkan Antrian Terakhir
6. Tampilkan Jumlah Antrian
7. Kosongkan Antrian
0. Keluar
Pilih Menu: 2
Masukkan nama: Maulina
Masukkan NIM: 24456
Masukkan kelas: 1D
Mahasiswa Maulina telah ditambahkan ke antrian.

MENU LAYANAN UNIT KEMAHASISWAAN
1. Cek Antrian
2. Tambah Antrian
3. Panggil Antrian
4. Tampilkan Antrian Terdepan
5. Tampilkan Antrian Terakhir
6. Tampilkan Jumlah Antrian
7. Kosongkan Antrian
0. Keluar
Pilih Menu: 2
Masukkan nama: Bagus
Masukkan NIM: 21121
Masukkan kelas: 1E
Mahasiswa Bagus telah ditambahkan ke antrian.
```

MENU LAYANAN UNIT KEMAHASISWAAN

1. Cek Antrian
2. Tambah Antrian
3. Panggil Antrian
4. Tampilkan Antrian Terdepan
5. Tampilkan Antrian Terakhir
6. Tampilkan Jumlah Antrian
7. Kosongkan Antrian
0. Keluar

Pilih Menu: 1

Antrian Masih bisa ditambahkan

MENU LAYANAN UNIT KEMAHASISWAAN

1. Cek Antrian
2. Tambah Antrian
3. Panggil Antrian
4. Tampilkan Antrian Terdepan
5. Tampilkan Antrian Terakhir
6. Tampilkan Jumlah Antrian
7. Kosongkan Antrian
0. Keluar

Pilih Menu: 4

Mahasiswa terdepan: Aditiasmara (23144)

MENU LAYANAN UNIT KEMAHASISWAAN

1. Cek Antrian
2. Tambah Antrian
3. Panggil Antrian
4. Tampilkan Antrian Terdepan
5. Tampilkan Antrian Terakhir
6. Tampilkan Jumlah Antrian
7. Kosongkan Antrian
0. Keluar

Pilih Menu: 5

Mahasiswa terakhir: Bagus (21121)

MENU LAYANAN UNIT KEMAHASISWAAN

1. Cek Antrian
2. Tambah Antrian
3. Panggil Antrian
4. Tampilkan Antrian Terdepan
5. Tampilkan Antrian Terakhir
6. Tampilkan Jumlah Antrian
7. Kosongkan Antrian
0. Keluar

Pilih Menu: 6

Jumlah dalam antrian: 3

MENU LAYANAN UNIT KEMAHASISWAAN

1. Cek Antrian
2. Tambah Antrian
3. Panggil Antrian
4. Tampilkan Antrian Terdepan
5. Tampilkan Antrian Terakhir
6. Tampilkan Jumlah Antrian
7. Kosongkan Antrian
0. Keluar

Pilih Menu: 2

Masukkan nama: Fara

Masukkan NIM: 27811

Masukkan kelas: 1A

Mahasiswa Fara telah ditambahkan ke antrian.

MENU LAYANAN UNIT KEMAHASISWAAN

1. Cek Antrian
2. Tambah Antrian
3. Panggil Antrian
4. Tampilkan Antrian Terdepan
5. Tampilkan Antrian Terakhir
6. Tampilkan Jumlah Antrian
7. Kosongkan Antrian
0. Keluar

Pilih Menu: 5

Mahasiswa terakhir: Fara (27811)

MENU LAYANAN UNIT KEMAHASISWAAN

1. Cek Antrian
2. Tambah Antrian
3. Panggil Antrian
4. Tampilkan Antrian Terdepan
5. Tampilkan Antrian Terakhir
6. Tampilkan Jumlah Antrian
7. Kosongkan Antrian
0. Keluar

Pilih Menu: 3

Memanggil antrian:

Nama: Aditiasmara

NIM: 23144

Kelas: 1E

MENU LAYANAN UNIT KEMAHASISWAAN

1. Cek Antrian
2. Tambah Antrian
3. Panggil Antrian
4. Tampilkan Antrian Terdepan
5. Tampilkan Antrian Terakhir
6. Tampilkan Jumlah Antrian
7. Kosongkan Antrian
0. Keluar

Pilih Menu: 4

Mahasiswa terdepan: Maulina (24456)

MENU LAYANAN UNIT KEMAHASISWAAN

1. Cek Antrian
2. Tambah Antrian
3. Panggil Antrian
4. Tampilkan Antrian Terdepan
5. Tampilkan Antrian Terakhir
6. Tampilkan Jumlah Antrian
7. Kosongkan Antrian
0. Keluar

Pilih Menu: 7

Antrian berhasil dikosongkan

MENU LAYANAN UNIT KEMAHASISWAAN

1. Cek Antrian
2. Tambah Antrian
3. Panggil Antrian
4. Tampilkan Antrian Terdepan
5. Tampilkan Antrian Terakhir
6. Tampilkan Jumlah Antrian
7. Kosongkan Antrian
0. Keluar

Pilih Menu: 6

Jumlah dalam antrian: 0

MENU LAYANAN UNIT KEMAHASISWAAN

1. Cek Antrian
2. Tambah Antrian
3. Panggil Antrian
4. Tampilkan Antrian Terdepan
5. Tampilkan Antrian Terakhir
6. Tampilkan Jumlah Antrian
7. Kosongkan Antrian
0. Keluar

Pilih Menu: 0

Program Selesai, Terima Kasih