



## **LAPORAN PRAKTIKUM ALGORITMA DAN STRUKTUR DATA JOBSHEET 11**



**NAMA       = RAKAGALI RESDA KRISANDI PUTRA**  
**KELAS     = TI -1B / 19**  
**NIM        = 244107020136**



## JOBSHEET XI LINKED LIST

### 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 Pembuatan Single Linked List

Waktu percobaan : 30 menit

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 ASO**.

2. Tambahkan class-class berikut:

- a. Mahasiswa00.java
- b. Node00.java
- c. SingleLinkedList00.java
- d. SLLMain00.java

**Ganti 00 dengan nomer Absen Anda**

3. Implementasikan Class Mahasiswa00 sesuai dengan diagram class berikut ini :

Mahasiswa
nim: String nama: String kelas: String ipk: double
Mahasiswa() Mahasiswa(nm: String, name: String, kls: String, ip: double) tampilInformasi(): void

4. Implementasi class Node seperti gambar berikut ini



```

1  public class NodeMahasiswa00 {
2      Mahasiswa00 data;
3      NodeMahasiswa00 next;
4
5      public NodeMahasiswa00(Mahasiswa00 data, NodeMahasiswa00 next) {
6          this.data = data;
7          this.next = next;
8      }
9  }

```

5. Tambahkan attribute head dan tail pada class SingleLinked List

```

NodeMahasiswa00 head;
NodeMahasiswa00 tail;

```

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

7. Tambahkan method isEmpty().

```

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

```

8. Implementasi method untuk mencetak dengan menggunakan proses traverse.

```

public void print() {
    if (!isEmpty()) {
        NodeMahasiswa00 tmp = head;
        System.out.print(s:"Isi Linked List:\t");
        while (tmp != null) {
            tmp.data.tampilInformasi();
            tmp = tmp.next;
        }
        System.out.println(x:"");
    } else {
        System.out.println(x:"Linked list kosong");
    }
}

```

9. Implementasikan method addFirst().

```

public void addFirst(Mahasiswa00 input) {
    NodeMahasiswa00 ndInput = new NodeMahasiswa00(input, next:null);
    if (isEmpty()) {
        head = ndInput;
        tail = ndInput;
    } else {
        ndInput.next = head;
        head = ndInput;
    }
}

```

10. Implementasikan method addLast().



```
public void addLast(Mahasiswa00 input) {
    NodeMahasiswa00 ndInput = new NodeMahasiswa00(input, next:null);
    if (isEmpty()) {
        head = ndInput;
        tail = ndInput;
    } else {
        tail.next = ndInput;
        tail = ndInput;
    }
}
```

11. Implementasikan method `insexLACTex`, untuk memasukkan node yang memiliki data input setelah node yang memiliki data key.

```
public void insertAfter(String key, Mahasiswa00 input) {
    NodeMahasiswa00 ndInput = new NodeMahasiswa00(input, next:null);
    NodeMahasiswa00 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);
}
```

12. Tambahkan method penambahan node pada indeks tertentu.

```
public void insertAt(int index, Mahasiswa00 input) {
    if (index < 0) {
        System.out.println(x:"indeks salah");
    } else if (index == 0) {
        addFirst(input);
    } else {
        NodeMahasiswa00 temp = head;
        for (int i = 0; i < index - 1; i++) {
            temp = temp.next;
        }
        temp.next = new NodeMahasiswa00(input, temp.next);
        if (temp.next.next == null) {
            tail = temp.next;
        }
    }
}
```

13. Pada class `SLLMainOO`, buatlah fungsi main, kemudian buat object dari class `SingleLinked List`.
14. Buat empat object mahasiswa dengan nama `mhs1`, `mhs2`, `mhs3`, `mhs4` kemudian isi data setiap object melalui konstruktor.
15. Tambahkan Method penambahan data dan pencetakan data di setiap penambahannya agar terlihat perubahannya.



```
sll.print();
sll.addFirst(mhs4);
sll.print();
sll.addLast(mhs1);
sll.print();
sll.insertAfter(key:"Dirga", mhs3);
sll.insertAt(index:2, mhs2);
sll.print();|
```

### 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:

NIM : 234004

Nama : Ani

Kelas : TI-1D

IPK : 3.8

Isi Linked List:

NIM : 234004

Nama : Ani

Kelas : TI-1D

IPK : 3.8

NIM : 234001

Nama : Budi

Kelas : TI-1A

IPK : 3.5

Isi Linked List:

NIM : 234004

Nama : Ani

Kelas : TI-1D

IPK : 3.8

NIM : 234001

Nama : Budi

Kelas : TI-1A

Nama : Budi

Kelas : TI-1A

Kelas : TI-1A

IPK : 3.5

NIM : 234002

Nama : Citra

Kelas : TI-1B

IPK : 3.6



---

2.1.2 Pertanyaan

1. Mengapa hasil compile kode program di baris pertama menghasilkan “Linked List Kosong”?  
dipanggil **sebelum ada data mahasiswa yang dimasukkan ke dalam linked list**. Pada saat itu, objek SingleLinkedList19 sll baru saja dibuat
2. Jelaskan kegunaan variable temp secara umum pada setiap method!  
Variabel temp digunakan menyimpan data dari node awal yang akan ditelusuri karna kalo pakai head akan mengubah node yang awal dicari sehingga akan hilang atau tidak terbaca
3. Lakukan modifikasi agar data dapat ditambahkan dari keyboard!



Masukkan jumlah mahasiswa yang ingin ditambahkan: 2

Data Mahasiswa ke-1

NIM : 123

Nama : raka

Kelas : ti1b

IPK : 4

Mahasiswa berhasil ditambahkan.

Linked List setelah penambahan:

Isi Linked List:

NIM : 123

Nama : raka

Kelas : ti1b

IPK : 4.0

Data Mahasiswa ke-2

NIM : sucahyo

Nama : 123

Kelas : ti1b

IPK : 3

Mahasiswa berhasil ditambahkan.

Linked List setelah penambahan:

Isi Linked List:

NIM : 123

Nama : raka

Kelas : ti1b

IPK : 4.0

NIM : sucahyo

Nama : 123

Kelas : ti1b

IPK : 3.0

Data akhir dalam Linked List:

Isi Linked List:

NIM : 123

Nama : raka

Kelas : ti1b

IPK : 4.0

NIM : sucahyo

Nama : 123

Kelas : ti1b

IPK : 3.0

## 2.2 Modifikasi Elemen pada Single Linked List

Waktu percobaan : 30 menit

Tim /jar Algoritma dan Struktur Data 2024-2025 D4 Teknik Informatika

Jurusan Teknologi Informasi -Politeknik Negeri Malang





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

```
public void getData(int index) {
    NodeMahasiswa00 tmp = head;
    for (int i = 0; i < index; i++) {
        tmp = tmp.next;
    }
    tmp.data.tampilInformasi();
}
```



3. Implementasikan method indexOC.

```
public int indexOf(String key) {
    NodeMahasiswa00 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;
    }
}
```

4. Tambahkan method removeFirst pada class SingleLinked List

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

5. Tambahkan method untuk menghapus data pada bagian belakang pada class SingleLinked List

```
public void removeLast() {
    if (isEmpty()) {
        System.out.println(x:"Linked List masih Kosong, tidak dapat dihapus!");
    } else if (head == tail) {
        head = tail = null;
    } else {
        NodeMahasiswa00 temp = head;
        while (temp.next != tail) {
            temp = temp.next;
        }
        temp.next = null;
        tail = temp;
    }
}
```

6. Sebagai langkah berikutnya, akan diimplementasikan method remove



```
public void remove(String key) {
    if (isEmpty()) {
        System.out.println("Linked List masih Kosong, tidak dapat dihapus!");
    } else {
        NodeMahasiswa00 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;
        }
    }
}
```

7. Implementasi method untuk menghapus node dengan menggunakan index.

```
public void removeAt(int index) {
    if (index == 0) {
        removeFirst();
    } else {
        NodeMahasiswa00 temp = head;
        for (int i = 0; i < index - 1; i++) {
            temp = temp.next;
        }
        temp.next = temp.next.next;
        if (temp.next == null) {
            tail = temp;
        }
    }
}
```

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

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

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

sll.removeFirst();
sll.removeLast();
sll.print();
sll.removeAt(index:0);
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 akhir dalam Linked List:
Isi Linked List:
NIM      : 123
Nama     : Rizki
Kelas   : 3C
IPK      : 4.0
NIM      : 124
Nama     : Bimon
Kelas   : 3C
IPK      : 4.0

data index 1 :
NIM      : 124
Nama     : Bimon
Kelas   : 3C
IPK      : 4.0
data mahasiswa an Bimon berada pada index : 1

Linked List Kosong
Linked List Masih Kosong, tidak dapat menghapus
Linked List Kosong
```

### 2.2.3 Pertanyaan

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

Keyword break digunakan dalam fungsi remove(String key) untuk menghentikan perulangan (looping) setelah data yang dicari ditemukan dan berhasil dihapus dari linked list.

2. Jelaskan kegunaan kode dibawah pada method remove



```

    temp->next = temp->next->next
    ( temp->next )
    tail = temp;
    
```

berfungsi untuk menghapus node tertentu dari linked list dan menjaga konsistensi struktur data setelah penghapusan. Baris pertama memutuskan koneksi node yang ingin dihapus dengan langsung menghubungkan node sebelumnya (temp) ke node setelahnya, sehingga node target dilewati dan otomatis dihapus oleh garbage collector. Baris kedua memastikan bahwa jika node yang dihapus adalah node terakhir (tail), maka pointer tail diperbarui agar menunjuk ke node sebelumnya (temp), sehingga linked list tetap valid dan tidak berujung pada node yang sudah dihapus.

### 3. Tugas

Waktu pengerjaan : 50 menit

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.