LAPORAN PRAKTIKUM ALGORITMA DAN STRUKTUR DATA
DOUBLE LINKED LIST

AHMAD SOFYAN BADAWI

244107020073

KELAS TI-1B

PRODI D-IV TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

2025

Percobaan 1

Langkah-langkah:

1. Perhatikan diagram class Mahasiswa04, Node04 dan class DoublelinkedLists di bawah ini! Diagram class ini yang selanjutnya akan dibuat sebagai acuan dalam membuat kode program DoubleLinkedLists.
2. Pada Project yang sudah dibuat pada Minggu sebelumnya, buat folder atau package baru bernama Jobsheet12 di dalam repository Praktikum ASD.
3. Buat class di dalam paket tersebut dengan nama Mahasiswa01. Di dalam class tersebut, deklarasikan atribut sesuai dengan diagram class di atas. Tambahkan juga konstruktor dan method sesuai diagram di atas

```
public class Mahasiswa04 {
    String nim, nama, kelas;
    double ipk;

    Mahasiswa04(String nim, String nama, String kelas, double ipk) {
        this.nim = nim;
        this.nama = nama;
        this.kelas = kelas;
        this.ipk = ipk;
    }

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

4. Buat class di dalam paket tersebut dengan nama Node01. Di dalam class tersebut, deklarasikan atribut sesuai dengan diagram class di atas. Selanjutnya tambahkan konstruktor sesuai diagram di atas

```
public class Node04 {
    Mahasiswa04 data;
    Node04 prev;
    Node04 next;

    Node04(Mahasiswa04 data) {
        this.data = data;
        this.next = null;
        this.prev = null;
    }
}
```

5. Buatlah sebuah class baru bernama DoubleLinkedLists pada package yang sama dengan Node01. Pada class DoubleLinkedLists tersebut, deklarasikan atribut sesuai dengan diagram class di atas.

```
public class DoubleLinkedList {
    Node04 head;
    Node04 tail;
```

6. Selajuntnya, buat konstruktor pada class DoubleLinkedLists sesuai gambar berikut.

```
    DoubleLinkedList() {
        head = null;
        tail = null;
    }
```

7. Buat method isEmpty(). Method ini digunakan untuk memastikan kondisi linked list kosong.

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

8. Kemudian, buat method addFirst(). Method ini akan menjalankan penambahan data di bagian depan linked list.

```
public void addFirst(Mahasiswa04 data) {
    Node04 newNode = new Node04(data);
    if (isEmpty()) {
        head = tail = newNode;
    } else {
        newNode.next = head;
        head.prev = newNode;
        head = newNode;
    }
}
```

9. Selain itu pembuatan method addLast() akan menambahkan data pada bagian belakang linked list.

```
public void addLast(Mahasiswa04 data) {
    Node04 newNode = new Node04(data);
    if (isEmpty()) {
        head = tail = newNode;
    } else {
        tail.next = newNode;
        newNode.prev = tail;
        tail = newNode;
    }
}

public void print() {
    Node04 current = head;
    if (isEmpty()) {
        System.out.println(x:"Warning! Linked List Masih Kosong");
    }
    while (current != null) {
        current.data.tampil();
        current = current.next;
    }
}
```

10. Untuk menambahkan data pada posisi setelah node yang menyimpan data key, dapat dibuat dengan cara sebagai berikut.

```java
public void insertAfter(String keyNIM, Mahasiswa04 data) {
    Node04 current = head;

    while (current != null && !current.data.nim.equals(keyNIM)) {
        current = current.next;
    }

    if (current == null) {
        System.out.println("Node dengan NIM " + keyNIM + " tidak ditemukan.");
        return;
    }

    Node04 newNode = new Node04(data);

    if (current == tail) {
        current.next = newNode;
        newNode.prev = current;
        tail = newNode;
    } else {
        newNode.next = current.next;
        newNode.prev = current;
        current.next.prev = newNode;
        current.next = newNode;
    }

    System.out.println("node berhasil disisipkan setelah NIM " + keyNIM);
}
```

11. Untuk mencetak isi dari linked lists dibuat method print(). Method ini akan mencetak isi linked lists berapapun size-nya.

```java
public void print() {
    Node04 current = head;
    if (isEmpty()) {
        System.out.println(x:"Warning! Linked List Masih Kosong");
    }
    while (current != null) {
        current.data.tampil();
        current = current.next;
    }
}
```

12. Selanjutya dibuat class Main DoubleLinkedListsMain untuk mengeksekusi semua method yang ada pada class DoubleLinkedLists.

```java
Run | Debug
public static void main(String[] args) {
    DoubleLinkedList list = new DoubleLinkedList();
    Scanner sc = new Scanner(System.in);
    int pilihan;
```

13. Buatlah menu pilihan pada class main.

```
do {
    System.out.println(x:"\nMenu Double Linked List Mahasiswa");
    System.out.println(x:"1. Tambah di awal");
    System.out.println(x:"2. Tambah di akhir");
    System.out.println(x:"3. Hapus di awal");
    System.out.println(x:"4. Hapus di akhir");
    System.out.println(x:"5. Tampilkan data");
    System.out.println(x:"6. Cari Mahasiswa Berdasarkan NIM");
    System.out.println(x:"7. Tambah data setelah NIM");
    System.out.println(x:"0. Keluar");
    System.out.print(s:"Pilih menu: ");
    pilihan = sc.nextInt();
    sc.nextLine();
```

14. Tambahkan switch case untuk menjalankan menu pilihan di atas.

```
switch (pilihan) {
    case 1 -> {
        Mahasiswa04 mhs = inputMahasiswa(sc);
        list.addFirst(mhs);
    }
    case 2 -> {
        Mahasiswa04 mhs = inputMahasiswa(sc);
        list.addLast(mhs);
    }
    case 3 -> list.removeFirst();
    case 4 -> list.removeLast();
    case 5 -> list.print();
    case 6 -> {
        System.out.print(s:"Masukkan NIM yang dicari: ");
        String nim = sc.nextLine();
        Node04 found = list.search(nim);
        if (found != null) {
            System.out.println(x:"Data ditemukan:");
            found.data.tampil();
        } else {
            System.out.println(x:"Data tidak ditemukan.");
        }
    }
    case 0 -> System.out.println(x:"Keluar dari program.");
    default -> System.out.println(x:"Pilihan tidak valid!");

    }
```

15. Jangan lupa tambahkan while di bawah switch case dan close untuk menutup object scanner.

```
} while (pilihan != 0);
sc.close();
```

16. Ada satu karakter yang perlu ditambahkan agar code bisa berjalan. Silakan dianalisis kekurangannya dan ditambahkan sendiri.

Verifikasi Hasil Percobaan

```
Menu Double Linked List Mahasiswa
1. Tambah di awal
2. Tambah di akhir
3. Hapus di awal
4. Hapus di akhir
5. Tampilkan data
7. Cari Mahasiswa berdasarkan NIM
0. Keluar
Pilih menu: 1
Masukkan NIM: 20304050
Masukkan Nama: Hermione
Masukkan Kelas: Gryffindor
Masukkan IPK: 4.0

Menu Double Linked List Mahasiswa
1. Tambah di awal
2. Tambah di akhir
3. Hapus di awal
4. Hapus di akhir
5. Tampilkan data
7. Cari Mahasiswa berdasarkan NIM
0. Keluar
Pilih menu: 5
NIM: Hermione, Nama: 20304050, Kelas: Gryffindor, IPK: 4.0
```

Pertanyaan

1. Jelaskan perbedaan antara single linked list dengan double linked lists!
   Jawab:
   Single linked list hanya menggunakan pointer nexr
   Double linked list menggunakan pointer next dan prev dimana bisa mengakses data sebelum dan sesudah (depan/belakang atau kanan/kiri)
2. Perhatikan class Node04, di dalamnya terdapat atribut next dan prev. Untuk apakah atribut tersebut?
   Jawab:
   Next ditujukkan menghubungkan data yang ada di sebelah kanan serta bisa mengakses data yang ada di sebelah kanan
   Prev ditujukkan menghubungkan data yang ada di sebelah kiri serta bisa mengakses data yang ada di sebelah kiri
3. Perhatikan konstruktor pada class DoubleLinkedLists. Apa kegunaan dari konstruktor tersebut?
   Jawab:
   Untuk set value head dan tail menjadi null
4. Pada method addFirst(), apa maksud dari kode berikut?
   Jawab:

Ditujukkan untuk menambahkan data dipaling depan/awal

5. Perhatikan pada method addFirst(). Apakah arti statement head.prev = newNode ?
   Jawab:
   Statement tersebut ditujukkan untuk menghubungkan head dengan newNode dengan menggunakan head.prev diisi dengan newNode
6. Modifikasi code pada fungsi print() agar dapat menampilkan warning/ pesan bahwa linked lists masih dalam kondisi.

```
public void print() {
    Node04 current = head;
    if (isEmpty()) {
        System.out.println(x:"Warning! Linked List Masih Kosong");
    }
```

7. Pada insertAfter(), apa maksud dari kode berikut ? current.next.prev = newNode;
   Jawab:
   itu maksudnya menghubungkan prev data disebelah current dengan newNode
8. Modifikasi menu pilihan dan switch-case agar fungsi insertAfter() masuk ke dalam menu pilihan dan dapat berjalan dengan baik.

```
case 7 -> {
    System.out.print(s:"Masukkan NIM yang ingin diinsert setelahnya: ");
    String keyNIM = sc.nextLine();
    Mahasiswa04 mhs = inputMahasiswa(sc);
    list.insertAfter(keyNIM, mhs);
}
```

Percobaan 2

Langkah-langkah:

Pada praktikum 2 ini akan dibuat beberapa method untuk menghapus isi LinkedLists pada class DoubleLinkedLists. Penghapusan dilakukan dalam tiga cara di bagian paling depan, paling belakang, dan sesuai indeks yang ditentukan pada linkedLists.

1. Buatlah method removeFirst() di dalam class DoubleLinkedLists.

```
public void removeFirst() {
    if (isEmpty()) {
        System.out.println(x:"List Kosong, tidak ada yang dihapus.");
        return;
    }
    if (head == tail) {
        head = tail= null;
    } else {
        head = head.next;
        head.prev = null;
    }
}
```

2. Tambahkan method removeLast() di dalam class DoubleLinkedLists.

```java
public void removeLast() {
    if (isEmpty()) {
        System.out.println(x:"List Kosong, tidak ada yang dihapus.");
        return;
    }
    if (head == tail) {
        head = tail = null;
    } else {
        tail = tail.prev;
        tail.next = null;
    }
}
```

Verifikasi Hasil Percobaan

```
Menu Double Linked List Mahasiswa
1. Tambah di awal
2. Tambah di akhir
3. Hapus di awal
4. Hapus di akhir
5. Tampilkan data
6. Cari Mahasiswa Berdasarkan NIM
7. Cari Mahasiswa berdasarkan NIM
0. Keluar
Pilih menu: 2
Masukkan NIM: 20304050
Masukkan Nama: Hermione
Masukkan Kelas: Gryffindor
Masukkan IPK: 4.0

Menu Double Linked List Mahasiswa
1. Tambah di awal
2. Tambah di akhir
3. Hapus di awal
4. Hapus di akhir
5. Tampilkan data
6. Cari Mahasiswa Berdasarkan NIM
7. Cari Mahasiswa berdasarkan NIM
0. Keluar
Pilih menu: 3

Menu Double Linked List Mahasiswa
1. Tambah di awal
2. Tambah di akhir
3. Hapus di awal
4. Hapus di akhir
5. Tampilkan data
6. Cari Mahasiswa Berdasarkan NIM
7. Cari Mahasiswa berdasarkan NIM
0. Keluar
Pilih menu: 5
Warning! Linked List Masih Kosong
```

Pertanyaan

1. Apakah maksud statement berikut pada method removeFirst()? head = head.next;
   Jawab:
   head.prev = null;
   head = head.next ditujukan untuk memindahkan posisi head kepada node posisi setelah head yang awal. Maka dari itu head yang sekarang sudah berubah.
   head.prev ditujukkan untuk melepas hubungan head yang baru dengan head yang lama.
2. Modifikasi kode program untuk menampilkan pesan “Data sudah berhasil dihapus. Data yang terhapus adalah … “
   Pada removeLast

```
System.out.println(x:"Data berhasil dihapus. Data yang terhapus adalah: " );
tail.data.tampil();
```

Pada removeFirst

```
System.out.println(x:"Data berhasil dihapus. Data yang terhapus adalah: " );
head.data.tampil();
```

Tugas Praktikum

1. Tambahkan fungsi add() pada kelas DoubleLinkedList untuk menambahkan node pada indeks tertentu.

```
public void addAt(int index, Mahasiswa04 data) {
    if (index < 0) {
        System.out.println(x:"Index tidak valid.");
        return;
    }
    if (index == 0) {
        addFirst(data);
    } else {
        Node04 newNode = new Node04(data);
        Node04 current = head;
        for (int i = 0; i < index - 1 && current != null; i++) {
            current = current.next;
        }
        if (current == null) {
            System.out.println(x:"Index melebihi jumlah elemen, menambahkan di akhir.");
            addLast(data);
        } else {
            newNode.next = current.next;
            newNode.prev = current;
            if (current.next != null) {
                current.next.prev = newNode;
            } else {
                tail = newNode;
            }
            current.next = newNode;
        }
    }
}
```

2. Tambahkan removeAfter() pada kelas DoubleLinkedList untuk menghapus node setelah data key.

```java
public void removeAfter(String keyNim) {

    Node04 current = head;

    if (isEmpty()) {
    System.out.println(x:"List kosong. Tidak ada yang bisa dihapus.");
    return;
    }


    while (current != null && !current.data.nim.equals(keyNim)) {
        current = current.next;
    }
    if (current == null) {
        System.out.println("Node dengan NIM " + keyNim + " tidak ditemukan.");
        return;
    }

    if (current.next != null) {
        Node04 removeAfter = current.next;
        current.next = removeAfter.next;
        if (removeAfter.next != null) {
            removeAfter.next.prev = current;
        } else {
            tail = current;
        }
        System.out.println("Node setelah NIM " + keyNim + " berhasil dihapus.");
    } else {
        System.out.println("Tidak ada node setelah NIM " + keyNim + ".");
    }

}
```

3. Tambahkan fungsi remove() pada kelas DoubleLinkedList untuk menghapus node pada indeks tertentu.

```java
public void removeAt(int index) {
    if (index < 0) {
        System.out.println(x:"Index tidak valid.");
        return;
    }
    if (isEmpty()) {
        System.out.println(x:"List Kosong, tidak ada yang dihapus.");
        return;
    }
    if (index == 0) {
        removeFirst();
    } else {
        Node04 current = head;
        for (int i = 0; i < index && current != null; i++) {
            current = current.next;
        }
        if (current == null) {
            System.out.println(x:"Index melebihi jumlah data yang ada, tidak ada yang dihapus.");
            return;
        }
        if (current == tail) {
            removeLast();
        } else {
            current.prev.next = current.next;
            if (current.next != null) {
                current.next.prev = current.prev;
            }
        }
    }
}
```

4. Tambahkan fungsi getFirst(), getLast() dan getIndex() untuk menampilkan data pada node head, node tail dan node pada indeks tertentu.

```java
public void getLast() {
    if (isEmpty()) {
        System.out.println(x:"List Kosong, tidak ada yang ditampilkan.");
    } else {
        tail.data.tampil();
    }
}

public int getIdx(String nim) {
    Node04 current = head;
    int index = 0;
    while (current != null) {
        if (current.data.nama.equals(nim)) {
            return index;
        }
        current = current.next;
        index++;
    }
    return -1;
}
```

5. Tambahkan kode program dan fungsi agar dapat membaca size/ jumlah data pada Double Linked List

```java
public int getTotalSize() {
    Node04 current = head;
    int size = 0;
    while (current != null) {
        size++;
        current = current.next;
    }
    return size;
}
```