LAPORAN PRAKTIKUM
ALGORITMA DAN STRUKTUR DATA
JOBSHEET 12

NAMA = RAKAGALI RESDA KRISANDI PUTRA

KELAS = TI -1B / 19

NIM = 244107020136

Percobaan 1

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
0. Exit
Pilih menu : 1
Masukkan NIM: 20304050
Masukkan Nama: hermione
Masukkan Kelas: Gryfindoor
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
0. Exit
Pilih menu : 5
NIM : 20304050, Nama: hermione, Kelas: Gryfindoor, IPK: 4.0
```

Peertanyaan

12.2.3 Pertanyaan Percobaan

1. Jelaskan perbedaan antara single linked list dengan double linked lists!

Double linked list memiliki variable prev sedangkan linked list tidak

2. Perhatikan class Node01, di dalamnya terdapat atribut next dan prev. Untuk apakah atribut tersebut?

- **next**: untuk menunjuk ke node berikutnya dalam linked list.
- **prev**: untuk menunjuk ke node sebelumnya dalam linked list.

3. Perhatikan konstruktor pada class DoubleLinkedLists. Apa kegunaan dari konstruktor tersebut?

untuk menginisialisasi linked list dalam kondisi kosong, di mana **head** dan **tail** sama-sama null. Artinya, pada saat pembuatan objek baru dari class ini, list belum memiliki elemen apa pun.

4. Pada method addFirst(), apa maksud dari kode berikut?

- Kode tersebut mengecek apakah list masih kosong (menggunakan method isEmpty()).
- Jika kosong, maka node baru yang ditambahkan akan menjadi **head** sekaligus **tail** (karena hanya ada satu node di list).

5. Perhatikan pada method addFirst(). Apakah arti statement head.prev = newNode ?

Setelah menambahkan node baru di depan, node yang tadinya menjadi **head** sekarang memiliki node baru di depannya.

Maka, atribut prev pada node lama (**head** sebelum update) di-set agar menunjuk ke **newNode** yang baru ditambahkan.

6. Modifikasi code pada fungsi print() agar dapat menampilkan warning/ pesan bahwa

linked lists masih dalam kondisi.

```
public void print() {
if (isEmpty()) {
    System.out.println(x:"Warning: Linked list masih kosong.");
} else {
    Node19 current = head;
    System.out.println(x:"Daftar Mahasiswa:");
    while (current != null) {
        current.data.tampil();
        current = current.next;
    }
}
```

7. Pada insertAfter(), apa maksud dari kode berikut ?

current.next.prev = newNode;

baris current.next.prev = newNode; **menghubungkan node setelah current agar backward-link-nya (prev) menunjuk ke node baru** yang baru saja disisipkan di antara mereka.

8. Modifikasi menu pilihan dan switch-case agar fungsi insertAfter() masuk ke dalam menu

pilihan dan dapat berjalan dengan baik.

```java
case 7: {
    System.out.print(s:"Masukkan NIM yang akan disisipkan setelahnya: ");
    String keyNim = scan.nextLine();
    Mahasiswa19 mhs = inputMahasiswa(scan);
    list.insertAfter(keyNim, mhs);
    break;
}
```

Percobaan 2

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. Sisipkan data setelah NIM tertentu
0. Exit
Pilih menu : 2
Masukkan NIM: 20304050
Masukkan Nama: herman
Masukkan Kelas: gryfindoor
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. Sisipkan data setelah NIM tertentu
0. Exit
Pilih menu : 3
```

Pertanyaan

1. Apakah maksud statement berikut pada method removeFirst()?
   head = head.next;
   head.prev = null;
   - head = head.next;
     Artinya, node pertama (head lama) dihapus dari list. Pointer head sekarang menunjuk ke node berikutnya, yaitu node kedua menjadi node pertama yang baru.
   - head.prev = null;
     Setelah head pindah ke node kedua, pointer prev dari node baru ini harus di-set ke null agar tidak lagi menunjuk ke node yang sudah dihapus.

2. Modifikasi kode program untuk menampilkan pesan “Data sudah berhasil dihapus. Data yang terhapus adalah … “

```
public void removeFirst() {
    if (isEmpty()) {
        System.out.println(x:"List kosong, tidak bisa dihapus.");
        return;
    }
    Mahasiswa19 dataTerhapus = head.data; // simpan data sebelum dihapus
    if (head == tail) {
        head = tail = null;
    } else {
        head = head.next;
        head.prev = null;
    }
    System.out.print(s:"Data sudah berhasil dihapus. Data yang terhapus adalah: ");
    dataTerhapus.tampil(); // asumsi ada method tampil() di Mahasiswa19
}

public void removeLast() {
    if (isEmpty()) {
        System.out.println(x:"List kosong, tidak bisa dihapus.");
        return;
    }
    Mahasiswa19 dataTerhapus = tail.data;
    if (head == tail) {
        head = tail = null;
    } else {
        tail = tail.prev;
        tail.next = null;
    }
    System.out.print(s:"Data sudah berhasil dihapus. Data yang terhapus adalah: ");
    dataTerhapus.tampil();
}
```