

Nama : Abdullah Azzam

Kelas : 1F

Absen : 01

### Pertanyaan

#### 2.1.2 Pertanyaan

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

**Jawab :**

Karena eksekusi awal langsung mencetak isi dari linked list yang mana linked listnya berada di dalam kondisi antrian atau listnya sedang kosong atau tidak memiliki data.

2. Pada step 10, jelaskan kegunaan kode berikut

```
ndInput.next = temp.next;  
temp.next = ndInput;
```

**Jawab :**

- Baris pertama merupakan Node baru yang mana merujuk pada alamat memori dari node setelah key
- Baris kedua ialah pointer temp menyimpan alamat memori ndInput

3. Perhatikan class **\*\*SingleLinkedList\*\***, pada method **\*\*insertAt\*\*** Jelaskan kegunaan kode berikut

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

**Jawab :**

Digunakan untuk kondisi jika temp.next.next (temp dari setelah temp) sama dengan null, maka tail sama dengan temp.next

#### 2.2.3 Pertanyaan

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

**Jawab :**

Karena pada code tersebut merupakan sebuah perulangan yang mana pada kondisi tersebut jika salah satu dari dua kondisi yang telah ditentukan tersebut terpenuhi, maka sudah tidak perlu dilakukan perulangan lagi, maka dari itu diberikan keyword break agar tidak berulang.

2. Jelaskan kegunaan kode dibawah pada method remove

```
else if (temp.next.data == key) {  
    temp.next = temp.next.next;
```

**Jawab :**

Kondisi tersebut dieksekusi jika kondisi yang pertama tidak cocok. Jika data temp setelahnya (temp.next.data) sama dengan key (data yang dirujuk), maka temp.next berubah menjadi temp.next.next atau terjadi perubahan posisi pada node sebelumnya ke node setelahnya.

3. Apa saja nilai kembalian yang dapat dikembalikan pada method indexOf? Jelaskan maksud masing-masing kembalian tersebut!

**Jawab :**

- Jika `tmp == null`, maka akan mengembalikan nilai -1 atau tidak terdapat dalam index (data kosong).
- jika tidak, maka akan mengembalikan nilai dari index yang diinputkan