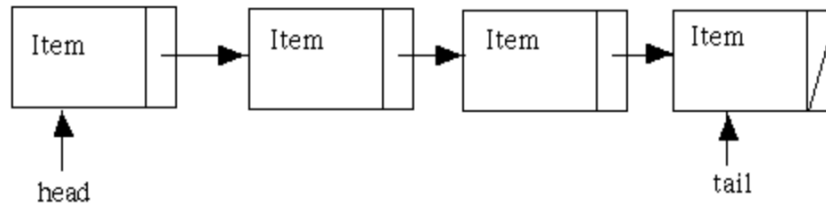


A. Singly Linked List with Tail

Singly Linked List dengan Tail mirip seperti latihan pada modul sebelumnya, namun kali ini diberi tambahan Tail sebagai penunjuk terakhir.



```
class SinglyLL
{
    int nilai;
    SinglyLL next;

    public SinglyLL() {
    }

    public SinglyLL(int nilai) {
        this.nilai = nilai;
        this.next = null;
    }
}
```

Kemudian di Main, kita dapat melakukan deklarasi global **head** dan **tail**, isi awalnya adalah **null**. Berikutnya diberikan contoh function **AddFirst()** untuk menambah sebuah data baru di depan :

```
class Main {
    SinglyLL head = null;
    SinglyLL tail = null;

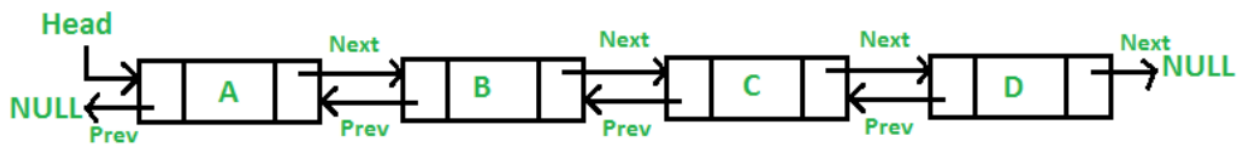
    public void addFirst(int nilai) {
        if (head == null) {
            head = new SinglyLL(nilai);
            tail = head;
        } else {
            SinglyLL temp = new SinglyLL(nilai);
            temp.next = head;
            head = temp;
        }
    }
}
```

Buatlah sebuah menu berisi pemanggilan function - function berikut dengan parameter yang sesuai :

1. **printData()** untuk menampilkan seluruh data nilai dalam linked list secara berurut dari Head.
2. **addLast(int nilai)** untuk menambahkan data baru di urutan belakang. Anda dapat memanfaatkan pointer Tail untuk menambah data dengan mudah.
3. **deleteLast()** untuk menghapus suatu data dari node terakhir (Tail). Anda tidak perlu me-null-kan node yang ingin dihapus, cukup dengan menghilangkan referensi next/prev/head/tail terhadap node tersebut, maka otomatis akan dihapus oleh Java Garbage Collector (mekanisme penghapusan data yang sudah tidak di-reference atau tidak digunakan lagi secara otomatis oleh Java).

B. Doubly Linked List

Doubly (Reverse) Linked List adalah bentuk LL yang memiliki Head sebagai penunjuk Node pertama, serta 2 buah pointer yang menunjuk ke node selanjutnya (Next) dan Node sebelumnya (Prev).



```
class DoublyLL
{
    int nilai;
    DoublyLL next, prev;

    public DoublyLL() {
    }

    public DoublyLL(int nilai) {
        this.nilai = nilai;
        this.next = null;
        this.prev = null;
    }
}
```

Kemudian di Main, kita dapat melakukan deklarasi global **head**, isi awalnya adalah **null**. Berikutnya diberikan contoh function **AddFirst()** untuk menambah sebuah data baru di depan :

```
class Main {
    DoublyLL head = null;

    public void addFirst(int nilai) {
        if (head == null) {
            head = new DoublyLL(nilai);
        } else {
            DoublyLL temp = new DoublyLL(nilai);
            temp.next = head;
            head = temp;
            head.next.prev = head;
        }
    }
}
```

Buatlah sebuah menu berisi pemanggilan function - function berikut dengan parameter yang sesuai :

1. **printDataAscending()** untuk menampilkan seluruh data nilai dalam linked list secara berurut dari Head. Ini sama dengan function-function Print sebelumnya.
2. **printDataDescending()** untuk menampilkan seluruh data nilai dalam linked list secara berurut dari node paling terakhir. Anda perlu membuat looping untuk mencari node terakhir terlebih dahulu, lalu menggunakan pointer **prev** untuk mundur sambil print data.
3. **addOrdered(int nilai)** untuk menambahkan data baru, aturannya setiap data dalam linked list ini selalu terurut secara Ascending (kecil ke besar), sehingga terdapat 3 kemungkinan, data di-add di depan, di belakang, ataupun disisip di tengah.