# RANGKUMAN TEORI LINKED LIST

## TEORI DASAR

### Pendahuluan

Linked List adalah struktur berupa rangkaian elemen saling berkait dimana tiap elemen dihubungkan ke elemen yang lain melalui pointer. Pointer adalah alamat elemen. Penggunaan pointer untuk mengacu elemen berakibat elemen-elemen bersebelahan secara logic walaupun tidak bersebelahan secara fisik di memori. Linked List merupakan kumpulan komponen yang saling berkaitan satu dengan yang lain melalui pointer. Masing-masing komponen sering disebut dengan simpul atau verteks.

### Singly Linked List

Merupakan Linked List yang paling sederhana. Setiap simpul dibagi menjadi dua bagian yaitu bagian isi dan bagian pointer. Bagian isi merupakan bagian yang berisi data yang disimpan oleh simpul, sedangkan bagian pointer merupakan bagian yang berisi alamat dari simpul berikutnya. Deklarasi Doubly Linked List:

typedef struct node \*simpul

struct node

{

type\_data Isi;

simpul Next;

};

### Doubly Linked List

Doubly Linked List merupakan Linked List dimana setiap simpul dibagi menjadi tiga bagian, yaitu bagian isi, bagian pointer kiri, dan bagian pointer kanan. Bagian isi merupakan bagian yang berisi data yang disimpan oleh simpul, bagian pointer kiri merupakan bagian yang berisi alamat dari simpul sebelumnya dan bagian pointer kanan merupakan bagian yang berisi alamat dari simpul berikutnya. Deklarasi Doubly Linked List:

typedef struct node \*simpul

struct node

{

char Isi;

simpul kanan;

simpul kiri;

};

# TUGAS PENDAHULUAN LINKED LIST

1. Apa yang dimaksud dengan Linked List!

Linked List merupakan kumpulan komponen yang saling berkaitan satu dengan yang lain melalui pointer. Masing-masing komponen sering disebut dengan simpul atau verteks.

1. Jelaskan perbedaan antara Singly Linked List, Doubly Linked List, dan Circular Linked List!

* Singly Linked List merupakan Linked List yang paling sederhana. Setiap simpul dibagi menjadi dua bagian yaitu bagian isi dan bagian pointer. Bagian isi merupakan bagian yang berisi data yang disimpan oleh simpul, sedangkan bagian pointer merupakan bagian yang berisi alamat dari simpul berikutnya.
* Doubly Linked List merupakan Linked List dimana setiap simpul dibagi menjadi tiga bagian, yaitu bagian isi, bagian pointer kiri, dan bagian pointer kanan. Bagian isi merupakan bagian yang berisi data yang disimpan oleh simpul, bagian pointer kiri merupakan bagian yang berisi alamat dari simpul sebelumnya dan bagian pointer kanan merupakan bagian yang berisi alamat dari simpul berikutnya.
* Circular Linked List adalah variasi dari Linked list di mana elemen pertama menunjuk ke elemen terakhir dan elemen terakhir menunjuk ke elemen pertama.

1. Jelaskan Operasi-Operasi pada Singly Linked List!

Insert: Menambahkan sebuah simpul baru ke dalam suatu linked list.

Konstruktor: Membuat sebuah linked list yang baru dan masih kosong.

IsEmpty: Menentukan apakah linked list kosong atau tidak.

Find First: Mencari elemen pertama dari linked list.

Find Next: Mencari elemen sesudah elemen yang ditunjuk now.

Retrieve: Mengambil elemen yang ditunjuk oleh now. Elemen tersebut lalu dikembalikan oleh fungsi.

Update: Mengubah elemen yang ditunjuk oleh now dengan isi dari sesuatu.

Delete Now: Menghapus elemen yang ditunjuk oleh now. Jika yang dihapus adalah elemen pertama dari linked list (head), head akan berpindah ke elemen berikut.

1. Jelaskan Operasi-Operasi pada Doubly Linked List!

Insert Tail: Menambah simpul di belakang (sebelah kanan) pada sebuah linked list.

Insert Head: Menambah simpul di depan (sebelah kiri) pada sebuah linked list.

Delete Tail: Menghapus simpul dari belakang (sebelah kanan). Fungsi Delete Tail akan mengarahkan Now kepada Tail dan kemudian memanggil fungsi Delete Now.

Delete Head: Menghapus simpul dari depan (sebelah kiri). Fungsi Delete Head akan mengarahkan Now kepada Head dan kemudian memanggil fungsi Delete Now.