

Nama : Bayu Setiawan

NIM : 2109106026

A1-21

### **RANGKUMAN MODUL 6-DOUBLE LINKED LIST**

1. Double Linked List adalah elemen-elemen yang dihubungkan dengan dua pointer dalam satu elemen, yaitu prev dan next, sehingga pembacaan data dapat melintas baik maju ke depan, maupun mundur ke belakang.
2. Head adalah sebagai penunjuk(Pointer) data awal, sedangkan Tail adalah penunjuk data terakhir.
3. kelebihan :
  - insert, delete data dapat di lakukan setelah tail linked list.
  - dapat digunakan dalam sistem navigasi (maju-mundur)
  - dapat mengimplementasikan tree
  - dapat melakukan transversal kedepan dan belakang
  - read dan delete banyak node mudah
  - dapat melakukan akses mundur
  - insert data lebih efisien.
4. kekurangan :
  - perlu banyak memori dan waktu karena punya banyak pointer.
  - tidak bisa akses elemen random.
5. Implementasi: undo-redo, backward-forward pada web, solitaire.
6. Alogaritma untuk membentuk node dimulai dengan inisiasi struktur node dan inisiasi variabel struct berupa variabel global.
7. Pada Transversal, diperlukan variabel bantu dan menggunakan perulangan saat variabel bantu memiliki nilai dan juga transversal tidak sampai pada index di node yang ingin di operasikan.

8. Untuk penambahan Node, nilai pada node akan di letakkan pada head dan tail node dengan syarat jika sudah memiliki data pada linked list, perlu dilakukan transversal dan variabel bantu akan di isi dengan head atau tail.
9. Untuk Insert First, untuk mengubah prev jadi nilai kosong yang mana pada next menjadi nilai awal pada head lalu ubah prev dengan alamat node sehingga node awal menjadi alamat node baru.
10. Insert last, langkah pertama, prev pada node baru akan menjadi node awal di tail, lalu pada next dengan nilai yang kosong akan di ubah pada node akhir dengan alamat node yang baru sehingga node akhir pada tail akan menjadi node baru.
11. Insert specific, dimulai dengan melakukan transversal ke indeks yang ingin di masukkan data, lalu ubah next dari node sebelumnya dan ubah prev dari node setelahnya, sehingga next pada node sebelumnya menjadi alamat node baru dan akhirnya prev pada node setelahnya jadi alamat node baru dengan lokasi yang di inginkan.
12. Untuk mengubah nilai node, diperlukan transversal pada node yang ingin di ubah.
13. Delete first, dengan node del sebagai head (node pertama) yang dimulai dengan mengubah isi head dengan nilai kosong sehingga node del sebagai perantara di hapus dengan delete.
14. Delete last, di mulai dengan memberi nilai node del(perantara) dengan alamat node awal pada tail, lalu mengubah nilai tail dengan alamat node sebelum tail sehingga nilai pada tail selanjutnya di isi dengan nilai kosong dan node tersebut di hapus dengan delete.
15. Delete specific, di mulai bantuan variabel lain untuk melakukan transversal ke node yang ingin di hapus dengan pilihan jika node awal yang di hapus maka harus mengubah isi head dan jika pada akhir node perlu mengubah isi tail. Dan jika data yang ingin di hapus berada pada node sebelum ataupun sesudah, maka kita perlu melakukan perubahan pada next ataupun prev dari node yang ingin di hapus, dan menggunakan fungsi free untuk menghapus node.