

STRUKTUR DATA DAN ALGORITMA

“List dan Linked List”



Indra Hermawan, S.Kom, M.Kom

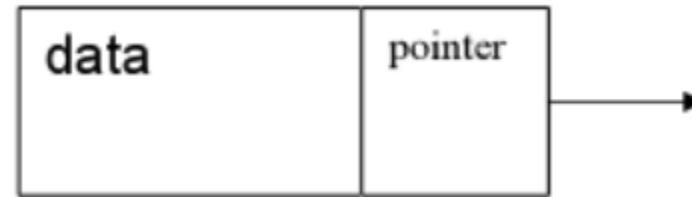
indrah13@gmail.com / indra@nurulfikri.ac.id

- Linked List
 - Pengertian
 - Deklarasi
 - Penggunaan
- Single Linked List
- Double Linked List
- Circular List
 - Single Linked List
 - Double Linked List

- Linked List adalah salah satu bentuk struktur data, berisi **kumpulan data (node)** yang **tersusun secara sekuensial, saling sambung-menyambung, dinamis dan terbatas.**
- Linked List sering disebut juga Senarai Berantai
- Linked List saling terhubung dengan bantuan variabel pointer
- Masing-masing data dalam Linked List disebut dengan node (simpul) yang menempati alokasi memori secara dinamis dan biasanya berupa struct yang terdiri dari beberapa field.

➤ Deklarasi Node

```
typedef struct TNode{  
    int data;  
    TNode *next;  
};
```



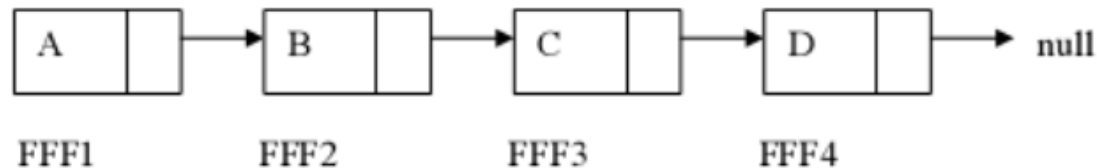
Menempati alamat memori tertentu

➤ Penjelasan:

- Pembuatan struct bernama TNode yang berisi 2 field, yaitu field data bertipe integer dan field **next yang bertipe pointer dari TNode**
- Setelah pembuatan struct, buat variabel head yang bertipe pointer dari Tnode yang berguna sebagai kepala linked list.

Pengertian:

- Single : artinya field pointer-nya hanya satu buah saja dan satu arah serta pada akhir node, pointernya menunjuk NULL
- Linked List : artinya node-node tersebut saling terhubung satu sama lain.

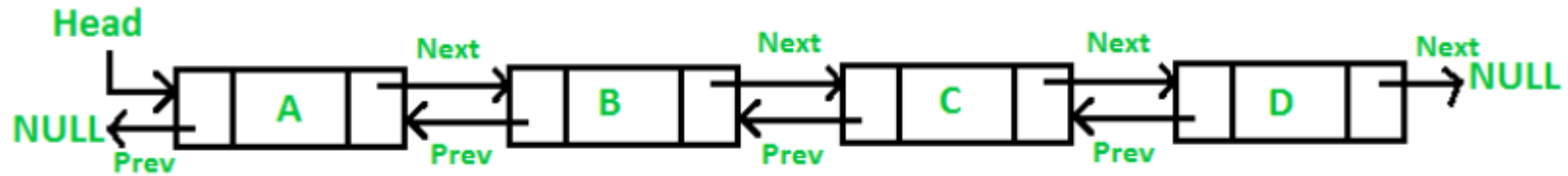


Ilustrasi Linked List

- Setiap node pada linked list mempunyai field yang berisi pointer ke node berikutnya, dan juga memiliki field yang berisi data.
- Node terakhir akan menunjuk ke NULL yang akan digunakan sebagai kondisi berhenti pada saat pembacaan isi linked list.

Pengertian:

- Double : artinya field pointer-nya berjumlah dua, satu “menunjuk” node berikutnya dan satu lagi “menunjuk” node sebelumnya dan pada akhir node pointernya menunjuk NULL
- Linked List : artinya node-node tersebut saling terhubung satu sama lain.



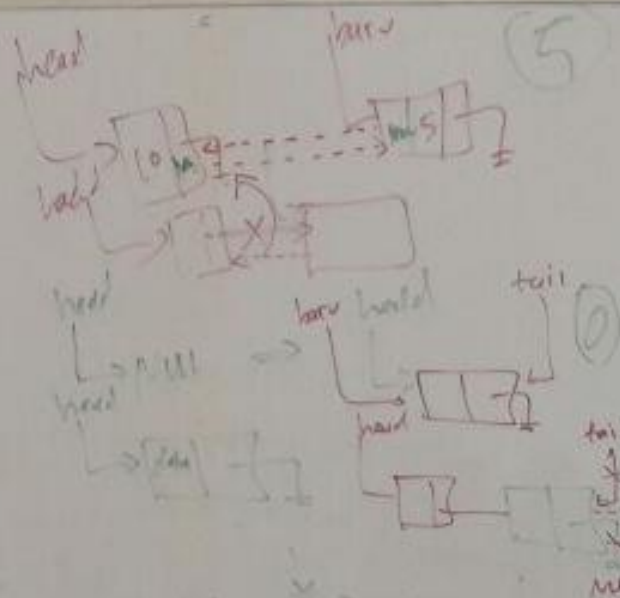
- Setiap node pada double linked list mempunyai field yang berisi pointer ke node berikutnya, pointer ke node sebelumnya, dan field yang berisi data.
- Node terakhir akan menunjuk ke NULL yang akan digunakan sebagai kondisi berhenti pada saat pembacaan isi linked list

- Deklarasi Node

```
typedef struct TNode{  
    int data;  
    TNode *next;  
    TNode *prev;  
};
```



- Penjelasan:
- Pembuatan struct bernama TNode yang berisi 3 field, yaitu field data bertipe integer dan field **next prev yang bertipe pointer dari Tnode.**
- Setelah pembuatan struct, buat variabel head yang bertipe pointer dari Tnode yang berguna sebagai kepala linked list.



struct list {

int data;

list *next;

list *prev;

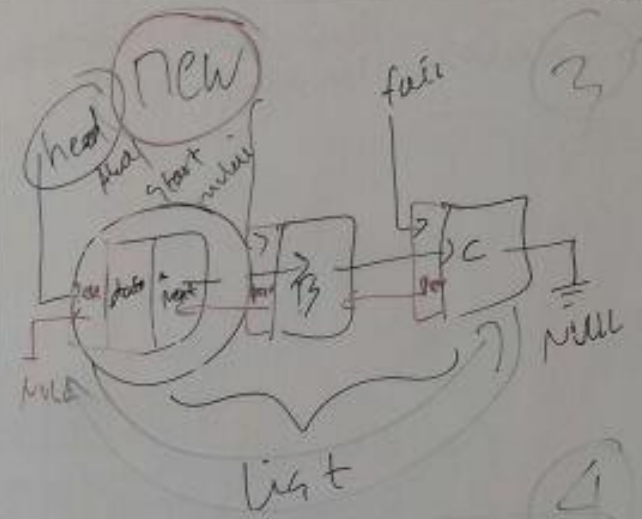
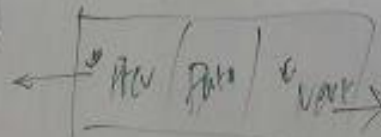


PA = &A
*PA = 10;

int main {

list data;

int data2;



list *head; => NULL

list *tail; => NULL

baru = new simpul;

head->data = 10;
head->next = NULL;

head->next = baru

Tambahkan belakang
Tambahkan depan
Tambahkan tengah

Butlah Program untuk :

1. Tambah data : Didepan, ditengah dan dibelakang
2. Hapus data : Didepan, ditengah dan dibelakang
3. Ubah data :Didepan ditengah dan dibelakang
4. Baca seluruh data

Membuat List

Membuat Linked List

Single Linked List

- Menambah data
 - Didepan, Ditengah dan dibelakang
- Mengubah data
 - Didepan, ditengah dan dibelakang
- Menghapus data
 - Didepan, ditengah dan dibelakang

Double Linked List

- Menambah data
 - Didepan, Ditengah dan dibelakang
- Mengubah data
 - Didepan, ditengah dan dibelakang
- Menghapus data
 - Didepan, ditengah dan dibelakang

Circular Linked List

- Menambah data
 - Didepan, Ditengah dan dibelakang
- Mengubah data
 - Didepan, ditengah dan dibelakang
- Menghapus data
 - Didepan, ditengah dan dibelakang