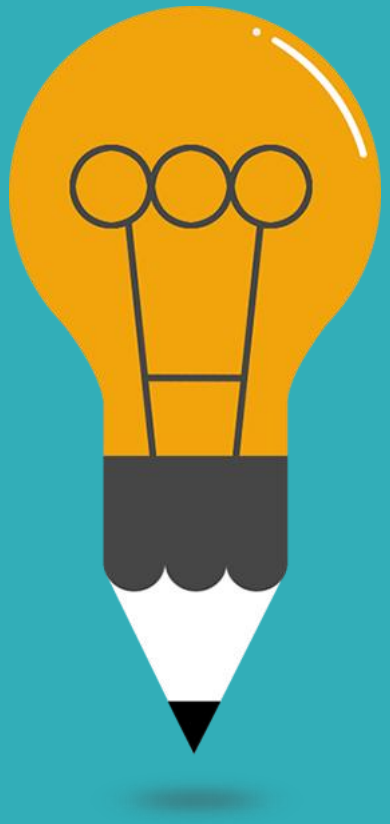
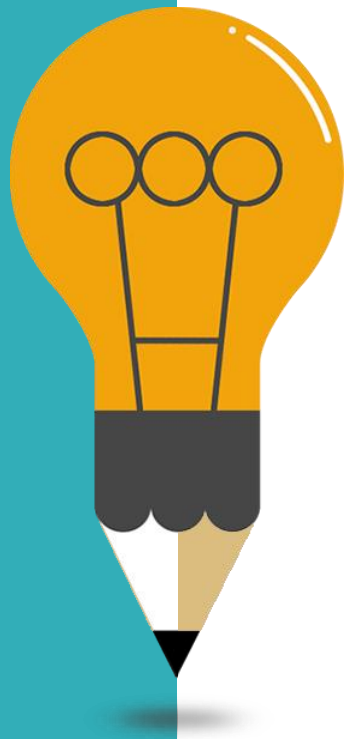


ALGORITMA & STRUKTUR DATA (IFUWP3337)

Dosen Pengampu:
Yosep Septiana, S.Kom., M.Kom.

Pertemuan 4

LINKED LIST



01 Linked List (List Berkait)

02 Single Linked List Non Circular (SLLNC)

03 SLLNC dengan Menggunakan Head

04 Tugas 3

LINKED LIST
(LIST BERKAIT)

Sejarah Linked List

- Dikembangkan tahun 1955-1956 oleh Allen Newell, Cliff Shaw dan Herbert Simon di RAND Corporation sebagai struktur data utama untuk bahasa Information Processing Language (IPL).
- IPL dibuat untuk mengembangkan program artificial intelligence, seperti pembuatan Chess Solver.
- Victor Yngve di Massachusetts Institute of Technology (MIT) juga menggunakan linked list pada natural language processing dan machine transitions pada bahasa pemrograman COMMIT.

Konsep Dasar Linked List

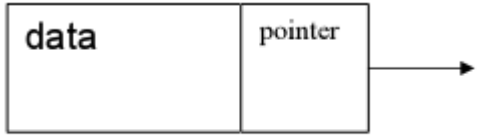
- Linked List adalah salah satu bentuk struktur data, berisi kumpulan data (node) yang tersusun secara sekuensial, saling sambung-menyambung, dinamis dan terbatas.
- Linked List sering disebut juga Senarai Berantai.
- Linked List saling terhubung dengan bantuan variabel pointer.
- Masing-masing data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

Perbedaan Linked List dengan Array

ARRAY	LINKED LIST
Statis	Dinamis
Penambahan / penghapusan data terbatas	Penambahan / penghapusan data tidak terbatas
Random access	Sequential access
Penghapusan array tidak mungkin	Penghapusan linked list mudah

SINGLE LINKED LIST NON CIRCULAR (SLLNC)

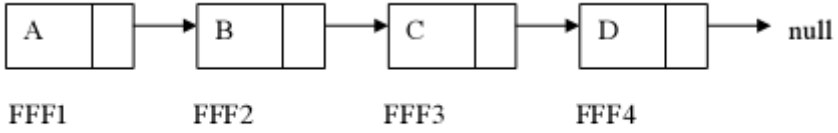
Bentuk Node SLLNC



Menempati alamat memori tertentu

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.

Pendeklarasian SLLNC

Deklarasi Node :

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

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.

Pembuatan SLLNC

Digunakan keyword new yang berarti mempersiapkan sebuah node baru berserta alokasi memorinya, kemudian node tersebut diisi data dan pointer nextnya ditunjuk ke NULL.

```
TNode *baru;
baru = new TNode;
baru->data = databaru;
baru->next = NULL;
```

Alokasi Pointer pada SLLNC

- Menggunakan alokasi memori secara manual
- Menggunakan header **stdlib.h** atau **malloc.h**
- Menggunakan fungsi:

```
<pointer type> *malloc(int size);
```

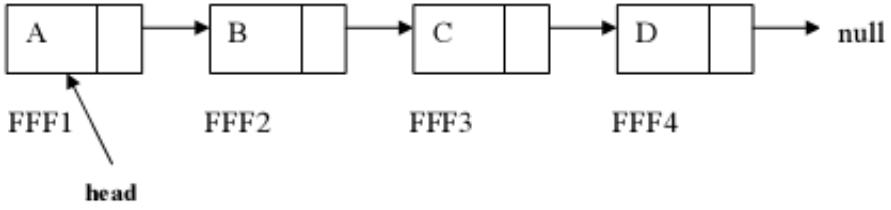
Contoh SLLNC

```c
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>

typedef struct Mahasiswa{
                int nim;
                Mahasiswa *next;
        };

void init(Mahasiswa **p){
     *p = NULL;
}

Mahasiswa *alokasi(int nim){
     Mahasiswa *P;
     P =(Mahasiswa*) malloc(sizeof(Mahasiswa));
     if(P!=NULL){
          P->next=NULL;
          P->nim = nim;
     }
     return (P);
}

void Add(Mahasiswa **p,int nim){
      *p=alokasi(nim);
      printf("%d",(*p)->nim);
}

int main(){
     Mahasiswa *head;
     init(&head);
     Add(&head,20);
     getch();
}
```

SLLNC
MENGGUNAKAN HEAD

SLLNC MENGGUNAKAN HEAD

- Dibutuhkan satu buah variabel pointer: head
- Head akan selalu menunjuk pada node pertama



- Deklarasi Pointer Penunjuk Kepala Single Linked List
- Manipulasi linked list tidak bisa dilakukan langsung ke node yang dituju, melainkan harus menggunakan suatu pointer penunjuk ke node pertama dalam linked list (dalam hal ini adalah head). Deklarasinya sebagai berikut:

```
TNode *head;
```

SLLNC MENGGUNAKAN HEAD

- Fungsi Inisialisasi Single LinkedList

```
void init() {
head = NULL;
}
```

- Function untuk mengetahui kosong tidaknya Single Linked List
- Jika pointer head tidak menunjuk pada suatu node maka kosong

```
int isEmpty() {
if(head == NULL) return 1;
else return 0;
```

SLLNC MENGGUNAKAN HEAD

Penambahan data di depan :

- Penambahan node baru akan dikaitan di node paling depan, namun pada saat pertama kali (data masih kosong), maka penambahan data dilakukan dengan cara: node head ditunjukkan ke node baru tersebut.

- Pada prinsipnya adalah mengkaitkan node baru dengan head, kemudian head akan menunjuk pada data baru tersebut sehingga head akan tetap selalu menjadi data terdepan.

SLLNC MENGGUNAKAN HEAD

Penambahan data di depan :

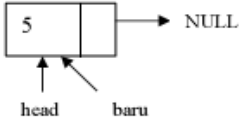
```
void insertDepan(int databaru) {
  TNode *baru;
  baru = new TNode;
  baru->data = databaru;
  baru->next = NULL;
  if(isEmpty()==1) {
    head=baru;
    head->next = NULL;
  }
  else {
    baru->next = head;
    head = baru;
  }
  printf(”Data masuk\n”);
}
```

SLLNC MENGGUNAKAN HEAD

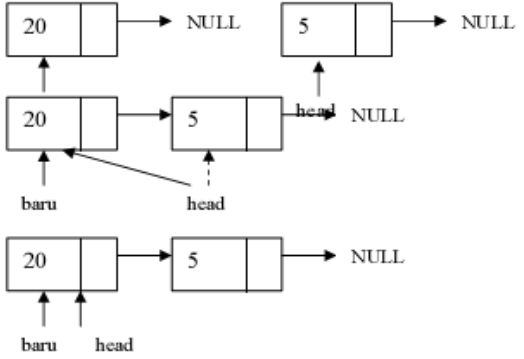
Ilustrasi :

1. List masih kosong (`head=NULL`)

2. Masuk data baru, misalnya 5



3. Datang data baru, misalnya 20 (penambahan di depan)



SLLNC MENGGUNAKAN HEAD

Penambahan data di belakang:

- Penambahan data dilakukan di belakang, namun pada saat pertama kali, node langsung ditunjuk oleh head.

- Penambahan di belakang lebih sulit karena kita membutuhkan pointer bantu untuk mengetahui node terbelakang, kemudian setelah itu, dikaitkan dengan node baru. Untuk mengetahui data terbelakang perlu digunakan perulangan.

SLLNC MENGGUNAKAN HEAD

Penambahan data di belakang :

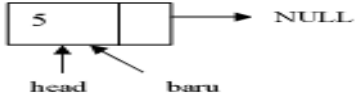
```
void insertBelakang (int databaru) {
  TNode *baru,*bantu;
  baru = new TNode;
  baru->data = databaru;
  baru->next = NULL;
  if(isEmpty()==1) {
    head=baru;
    head->next = NULL;
  }
  else {
    bantu=head;
    while(bantu->next!=NULL) {
      bantu=bantu->next;
    }
    bantu->next = baru;
  }
  printf("Data masuk\n“);
}
```

SLLNC MENGGUNAKAN HEAD

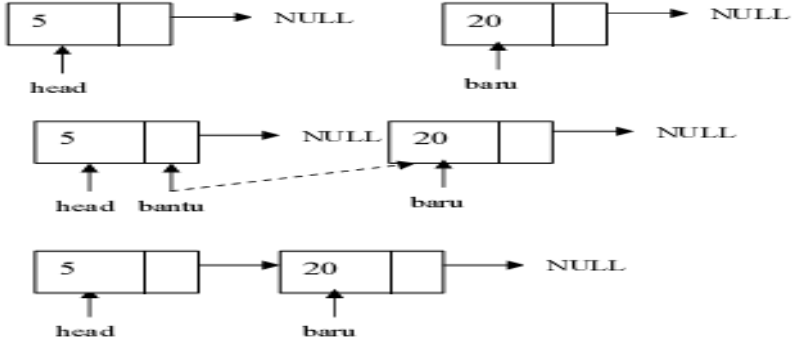
Ilustrasi :

1. List masih kosong (`head=NULL`)

2. Masuk data baru, misalnya 5



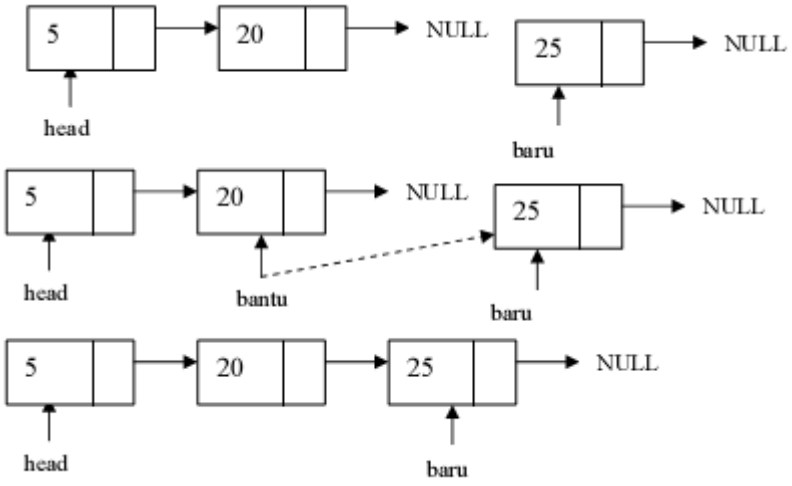
3. Datang data baru, misalnya 20 (penambahan di belakang)



SLLNC MENGGUNAKAN HEAD

Ilustrasi :

4. Datang data baru, misal 25 (penambahan di belakang)



Sekian dan Terima Kasih

TUGAS 3