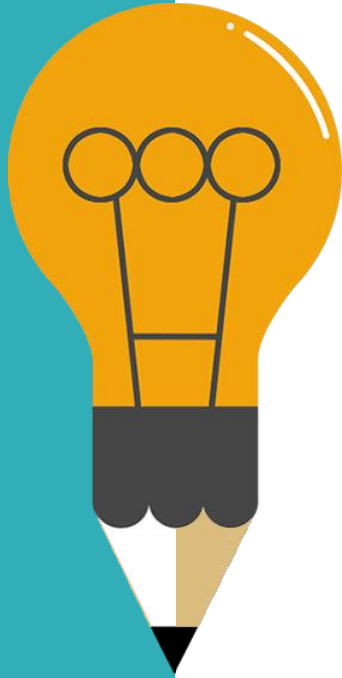


ALGORITMA & STRUKTUR DATA (IFUWP3337)

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



LINKED LIST



01

SLLNC dengan Menggunakan Head dan Tail

02

Menghapus Data / Node pada SLLNC

03

Operasi Dasar pada Linked List

04

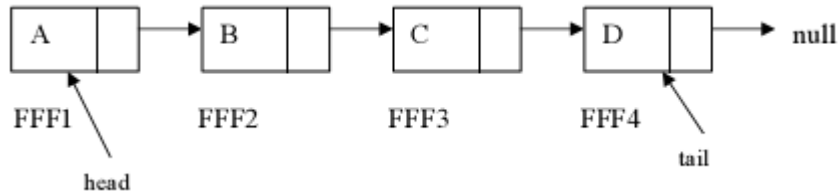
Tugas 4



SLLNC DENGAN MENGUNAKAN HEAD DAN TAIL

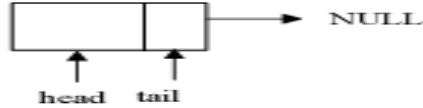
SLLNC dengan Menggunakan Head & Tail

- ❑ Dibutuhkan dua buah variabel pointer: **head** dan **tail**
- ❑ **Head** akan selalu menunjuk pada **node pertama**, sedangkan **tail** akan selalu menunjuk pada **node terakhir**.



Ilustrasi SLLNC dengan Menggunakan Head & Tail

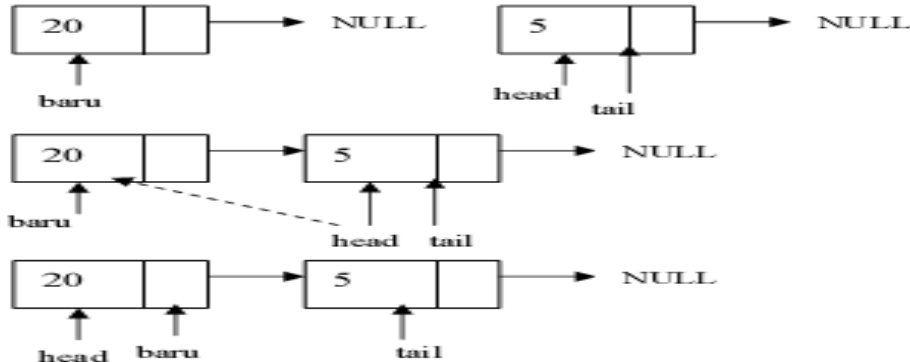
1. List masih kosong (head=tail=NULL)



2. Masuk data baru, misalnya 5



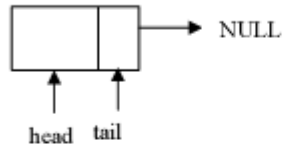
3. Datang data baru, misalnya 20



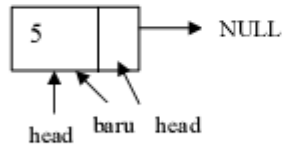
Ilustrasi SLLNC dengan Menggunakan Head & Tail

Ilustrasi:

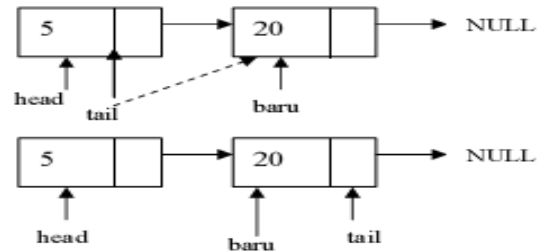
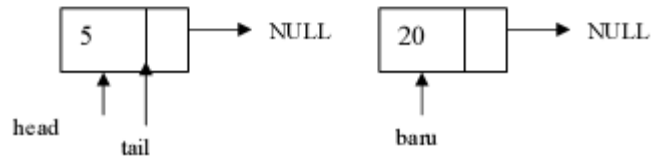
1. List masih kosong ($\text{head}=\text{tail}=\text{NULL}$)



2. Masuk data baru, misalnya 5



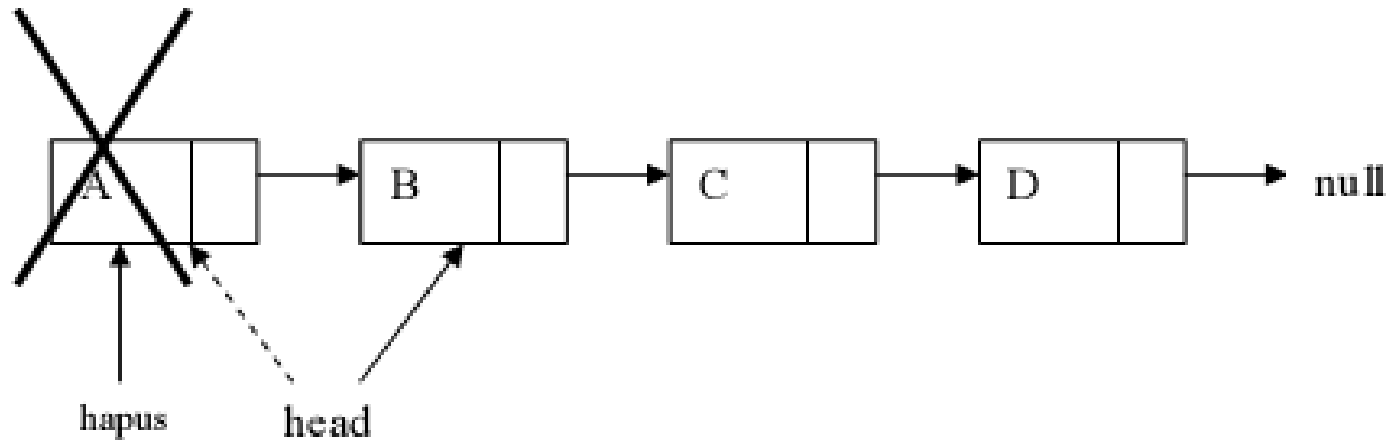
3. Datang data baru, misalnya 20



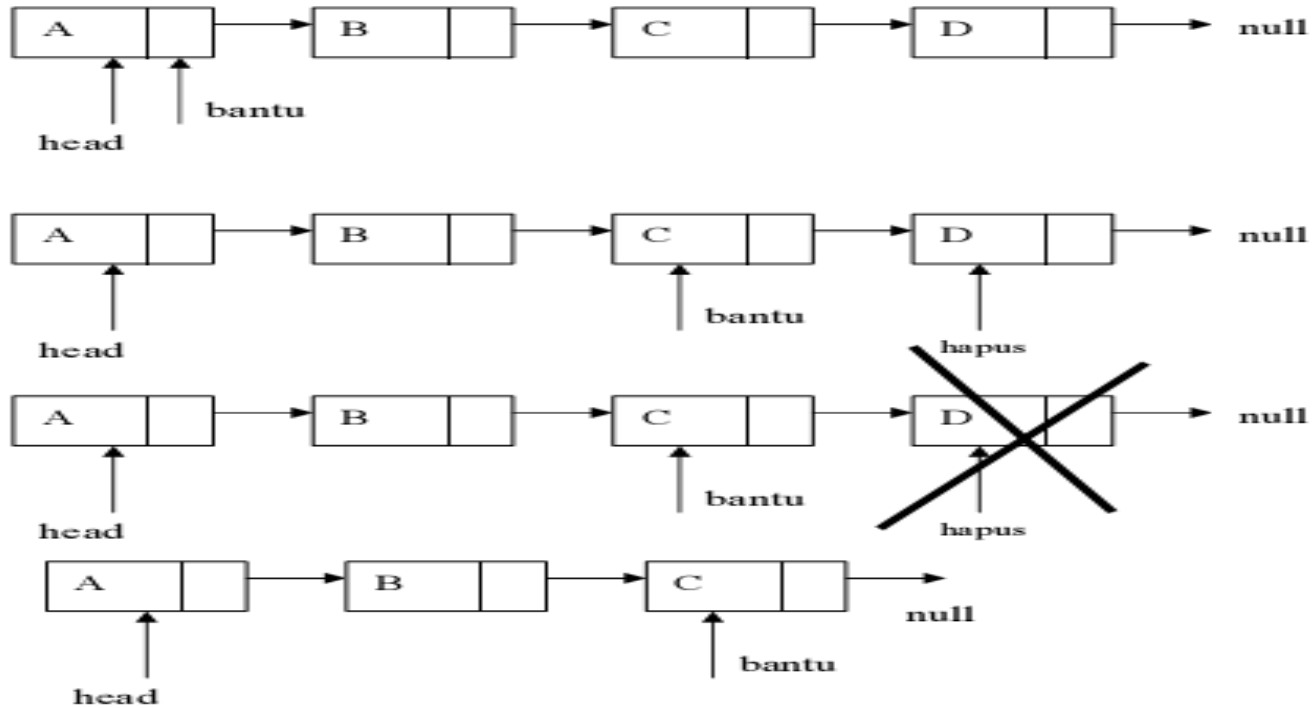


MENGHAPUS DATA / NODE PADA SLLNC

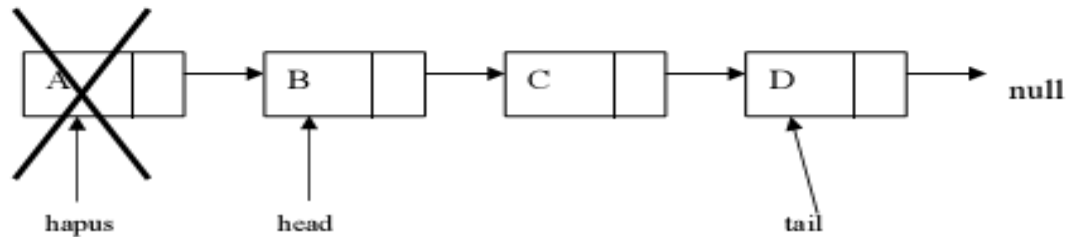
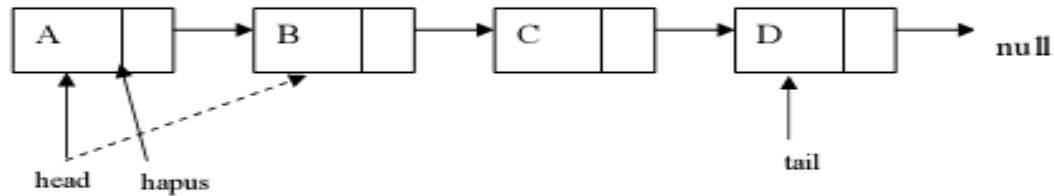
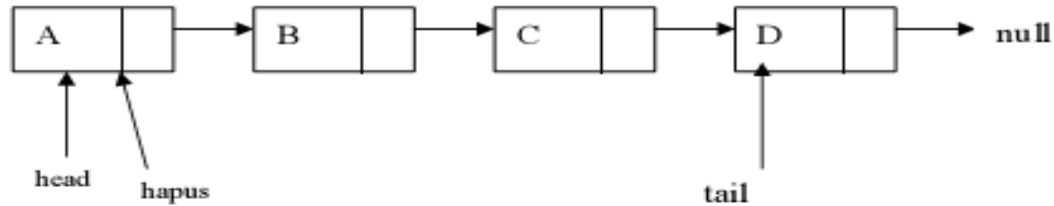
Menghapus Data / Node pada SLLNC



Menghapus Data / Node pada SLLNC

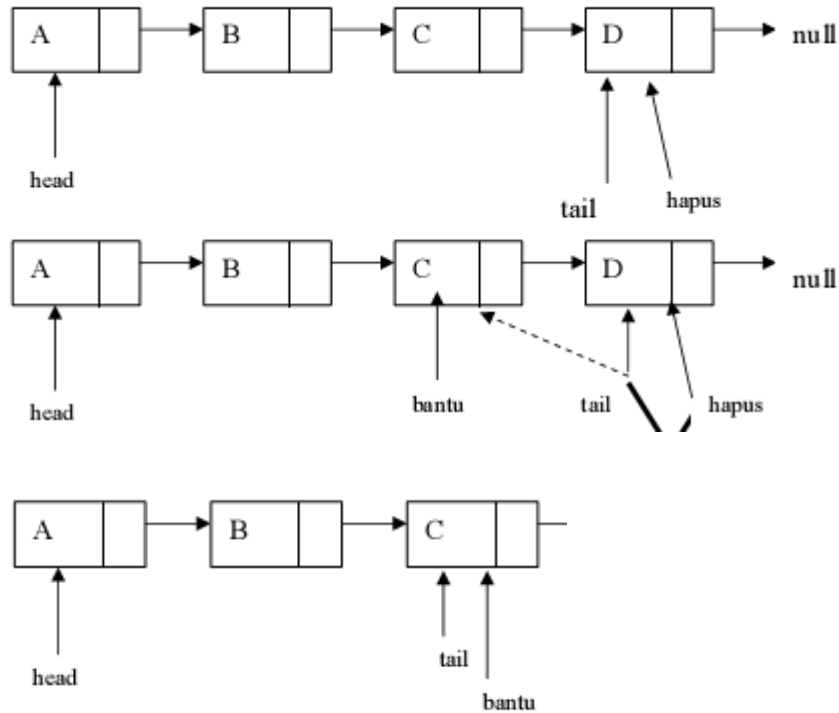


Menghapus Data / Node pada SLLNC



Menghapus Data / Node pada SLLNC

Ilustrasi:





OPERASI DASAR PADA LINKED LIST

Operasi Dasar pada Linked List

- ❑ **NODE(P)**, artinya node yang ditunjuk oleh pointer P.
- ❑ **INFO(P)**, artinya nilai INFO dari node yang ditunjuk pointer P.
- ❑ **NEXT(P)**, artinya hubungan (link) selanjutnya dari node yang ditunjuk oleh pointer P.
- ❑ **GETNODE(NEW)**, artinya menyisipkan node baru pada linked list.
- ❑ **FREENODE(P)**, artinya menghapus node yang ditunjuk pointer P.



Sekian dan Terima Kasih



TUGAS 4