

# LINKED LIST

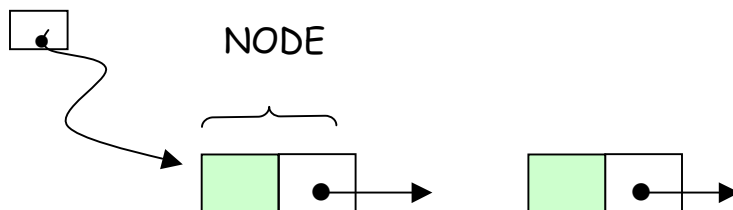
## I. Single Linked List

Atau disebut juga One Way List yaitu koleksi linier dari element data yang disebut simpul atau NODE.

Cara melinierkan urutan adalah dengan menggunakan pointer.

Bentuknya adalah sebagai berikut :

Start



Start = Variabel Pointer List

Data = Isi data/Item

Link = bagian yang berisi alamat node berikut

## II. Penyajian Single Link List dalam Memori

Misal List adalah sebuah Linked List terdiri dari dua field yaitu Data dan Link. Penyajiannya membutuhkan dua array Linier yang paralel yaitu DATA[K] dan LINK[K].

Contoh :

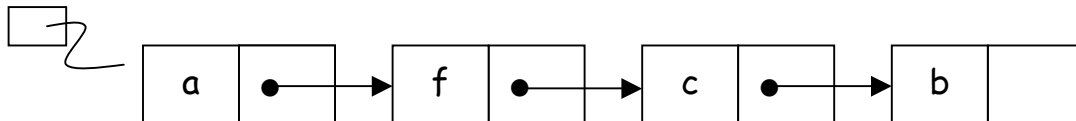
	DATA	LINK	START=5		
Start	1	O	3	Link[5] = 1	Data[5] = N
5	2	T	0	Link[1] = 3	Data[1] = O
	3		7	Link[3] = 7	Data[3] = Blank
	4	X	6	[7] = 4	[7] = E
	5	N	1	[4] = 6	[4] = X
	6	I	2	[6] = 2	[6] = I
	7	E	4	[2] = 0	[2] = T

### III. OPERASI DALAM SINGLE LINK LIST

#### 1. Searching

a. Searching untuk data yang belum berurut

PTR = Start

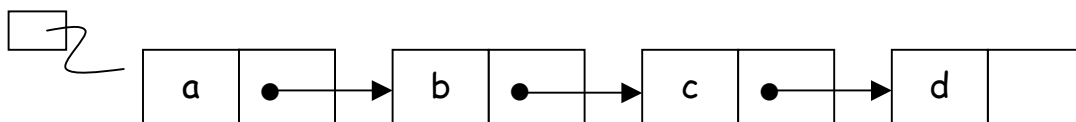


Algoritma :

1. Set PTR := start
2. Read Item
3. While PTR ≠ null
4. If data[PTR] = item then
  - Loc := PTR, Exit (item ada dalam list)
- Else
  - PTR:= Link[PTR] (membaca link pointer berikutnya)
5. Loc:= null (unsuccessful).

b. Searching untuk data yang sudah terurut.

PTR = Start



Algoritma :

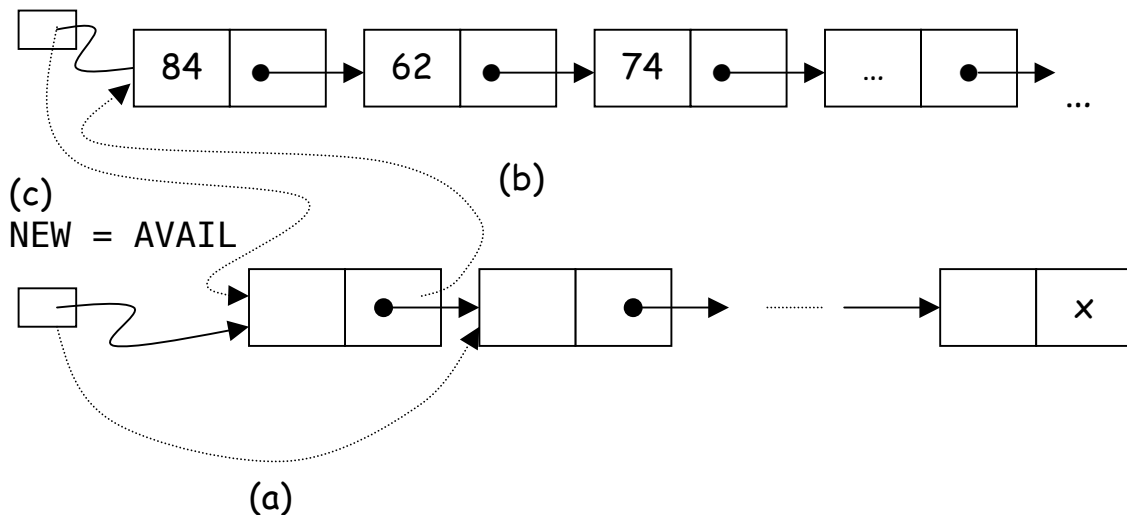
1. PTR := start
2. Read Item
3. While PTR  $\neq$  null
4. If item > data[PTR] then  
PTR:= Link [PTR]  
Else  
If item = Data [PTR] then  
Loc:=PTR, exit  
Else
5. Loc:= null

## 2. INSERT pada linked list

Ada beberapa cara penempatan dalam insert

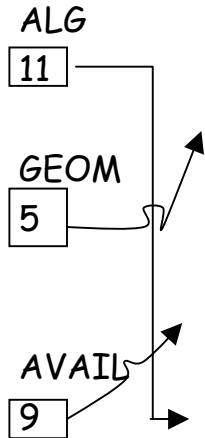
a. Penyisipan diawal list.

Start



Contoh :

	TES	LINK
1		16
2	74	14
3		1
4	82	0
5	84	
6	78	0
7	71	
8	100	
9		10
10		3
11	88	2
12	62	
13	74	
14	93	4
15		0
16		15



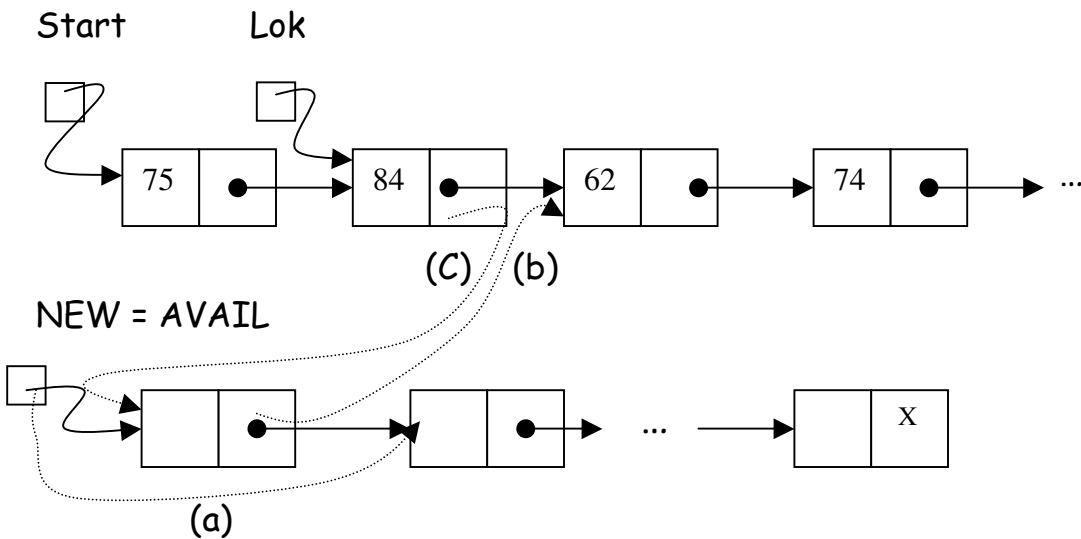
Misalkan angka 75 akan disisipkan  
Pada awal dari list *GEOM*.

NEW := 9  
AVAIL := 10  
DATA[q] = 75  
LINK[q] = 5  
START := 9

Urutan *GEOM* :  
84, 62, 71, 100, 74, 78

*ALG* : 88, 74, 93, 82

b. Penyisipan ditengah List. ( setelah Lok)



Misal disisipkan angka 55 diantara 84 dan 62  
Algoritma :

1. new := avail	NEW := 10
2. AVAIL :=LINK[AVAIL] ...(a)	AVAIL := 3
3. DATA[NEW] := ITEM	DATA[10]:=55
4. LINK [NEW] := LINK[LOK] ... (b)	LINK[10]:= LINK[5]=12
5. LINK[Lok] := NEW ... (C)	LINK[5] = 10
6. EXIT	

c. Penyisipan diakhir list algoritma :

Misal disispkan 80 diakhir List GEOM  
Algoritma :

1. NEW := AVAIL	NEW := 3
2. AVAIL := LINK[AVAIL]	AVAIL := 1
3. DATA[NEW] := ITEM	DATA[3] := 80
4. LINK[NEW] := NULL	LINK[3] := 0
5. LIKN[Lok] := NEW	LLINK[6] := 3
6. EXIT	

### 3. DELETE dalam LINKED LIST

1. If Lok P = NULL and Lok  $\neq$  NULL then  
START := LINK[START]

Menghapus elemen di awal

2. Menghapus elemen di tengah  
if Lok P  $\neq$  NULL and LINK[Lok]  $\neq$  NULL then  
Link[LokP] := LINK[Lok]

3. menghapus elemen di akhir  
if Lok P  $\neq$  NULL and LINK[Lok] = NULL then  
Link[LokP] := NULL

4. LINK [Lok] = AVAIL

5. AVAIL = Lok

6. EXIT

Ket Lok : adalah alamat dari simpul yang akan dihapus  
LokP : adalah lokasi dari simpul yang mendahului simpul yang akan dihapus

Contoh : menghapus elemen 93 ditengah-tengah list

