Linked List

Brigida Arie Minartiningtyas, M.Kom

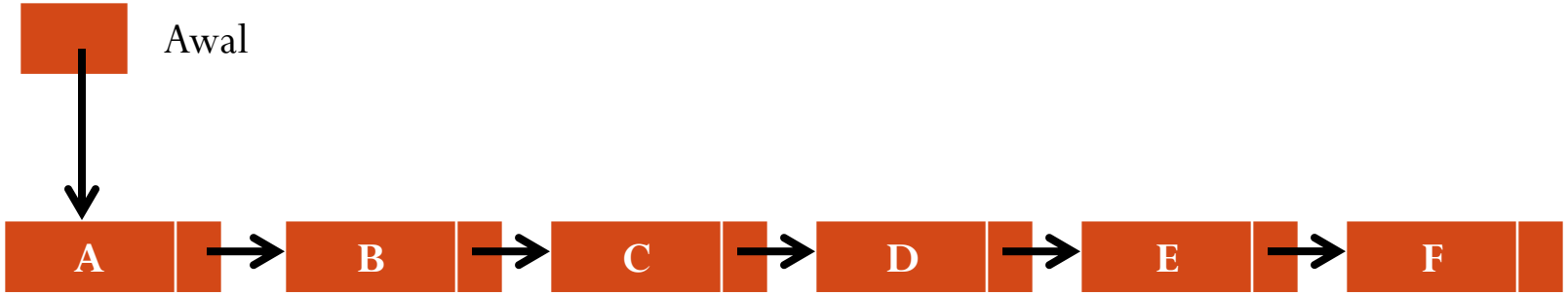
Review Pointer

- Penggunaan pointer sangat mendukung dalam pembentukan struktur data dinamis
- Salah satu struktur data dinamis adalah Linked List (senarai berantai)

Linked List

- Linked list adalah kumpulan komponen yang disusun secara berurutan dengan menggunakan pointer.
- Masing-masing komponen dinamakan dengan simpul (node)
- Setiap simpul dalam suatu linked list terbagi menjadi dua:
  - Bagian pertama : medan informasi
  - Bagian kedua : medan penyambung

Ilustrasi Linked List



Contoh 1

- Pada bangsal sebuah rumah sakit terdapat 12 tempat tidur. Sembilan di antaranya telah ditempati Pasien.
- Kita hendak membuat list nama para pasien tersebut secara alfabetik.

START 5

Bed Number	Patient	Next
1	Kirk	7
2		
3	Dean	11
4	Maxwell	12
5	Adams	3
6		
7	Lane	4
8	Green	1
9	Samuels	0
10		
11	Fields	8
12	Nelson	9

Contoh 2

- Menggambarkan suatu linked list dalam memori.
- Data dalam Array INFO(K) adalah sebuah karakter tunggal.

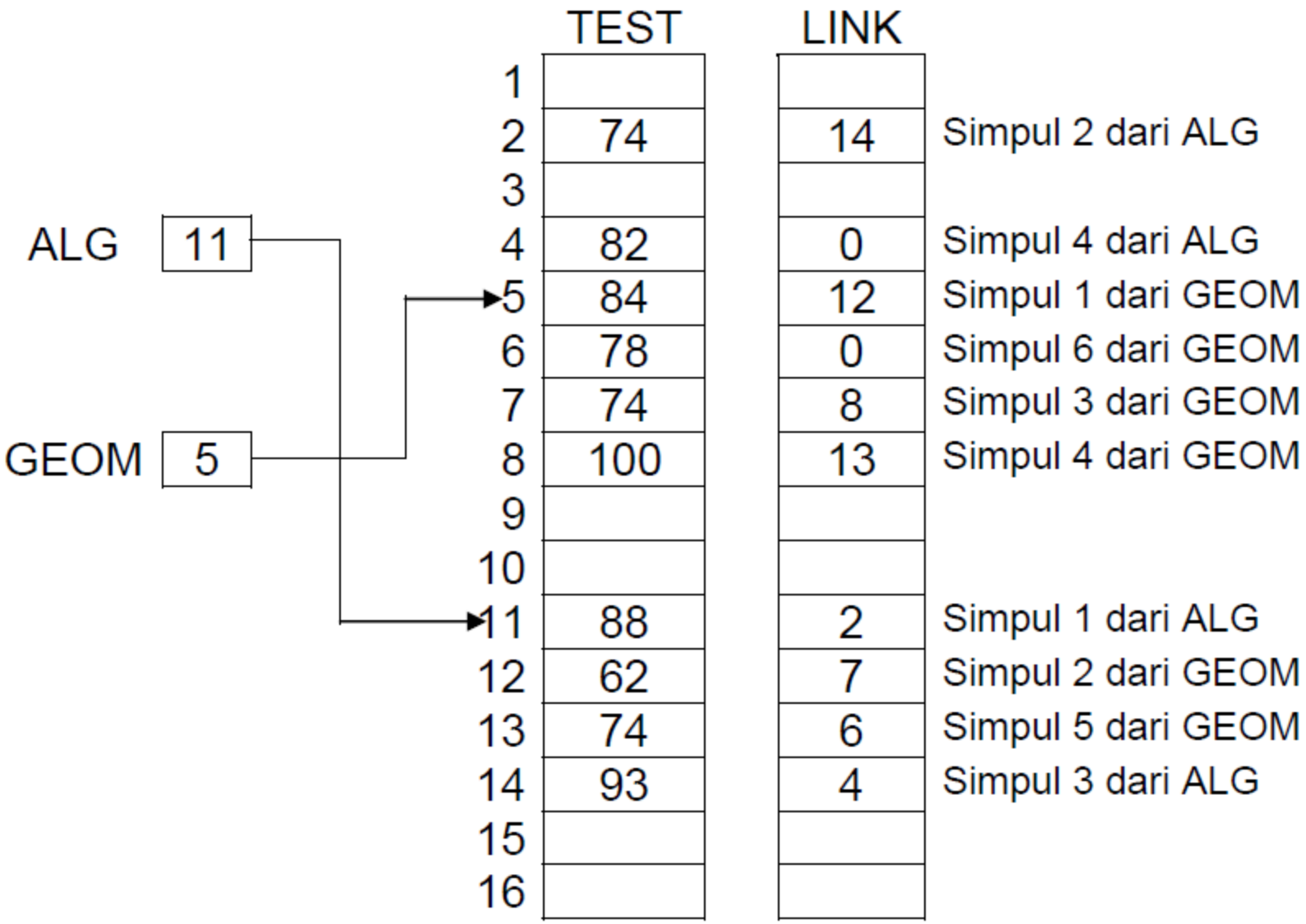
START 9

	INFO	LINK
1		
2		
3	O	6
4	T	0
5		
6	□	11
7	X	10
8		
9	N	3
10	I	4
11	E	7
12		

- START = 9 INFO[9] = N, elemen pertama adalah N
- LINK(9) = 3 INFO[3] = O, elemen kedua adalah O
- LINK(3) = 6 INFO[6] = Blank, elemen ketiga adalah blank
- LINK(6) = 11 INFO[11] = E, elemen keempat adalah E
- LINK(11) = 7 INFO[7] = X, elemen kelima adalah X
- LINK(7) = 10 INFO(10] = I, elemen keenam adalah I
- LINK(10) = 4 INFO[4] = T, elemen ketujuh adalah T
- LINK(4) = 0 Null, list terakhir

Contoh 3

- Memperlihatkan dua buah linked list, ALG dan GEOM, yang berturut-turut berisi nilai matakuliah Algoritma dan Geometri, dapat tersimpan bersama dalam array TEST dan LINK yang sama.
- Pointer ALG berisi nilai 11, yakni lokasi dari simpul pertama list ALG,
- Pointer GEOM berisi nilai 5, yakni lokasi simpul pertama dari list GEOM.


TEST
LINK
ALG
11
GEOM
5
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
74
82
84
78
74
100
88
62
74
93
14
0
12
0
8
13
2
7
6
4
Simpul 2 dari ALG
Simpul 4 dari ALG
Simpul 1 dari GEOM
Simpul 6 dari GEOM
Simpul 3 dari GEOM
Simpul 4 dari GEOM
Simpul 1 dari ALG
Simpul 2 dari GEOM
Simpul 5 dari GEOM
Simpul 3 dari ALG

Operasi Linked List

1. **Menambah simpul**
   - Tambah belakang
   - Tambah depan
   - Tambah Tengah

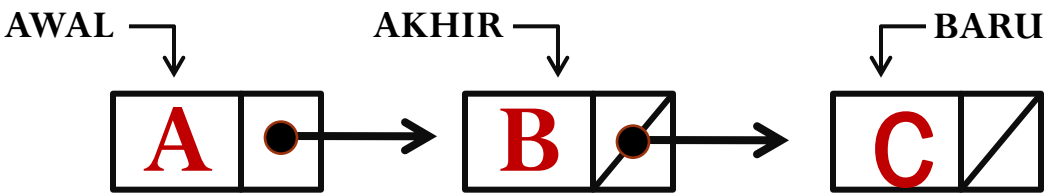
2. **Menghapus simpul**
   - Hapus awal
   - Hapus tengah
   - Hapus akhir

Deklarasi Pointer

```
Type Simpul = ^Data;
 Data = record
   Isi      :   char;
   Next     :   Simpul;
 end;
var  Elemen   :   char;
 Awal, Akhir, Baru  :  Simpul;
```

MENAMBAH SIMPUL

1. TAMBAH BELAKANG

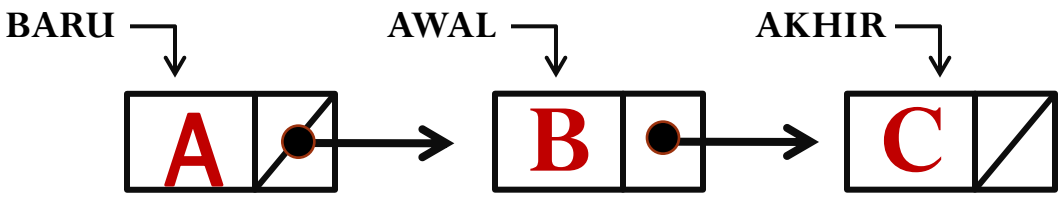



```
Type Simpul = ^Data;
     Data = record
        Isi     :  char;
        Next    :  Simpul;
     end;
var  Elemen  :  char;
     Awal, Akhir, Baru  :  Simpul;
```

```
Procedure TAMBAH(var Awal, Akhir : Simpul; Elemen : char) ;

var Baru : Simpul;
begin
   new(Baru);
   Baru^.Isi:= Elemen;
   Akhir^.next := Baru;
   Akhir := Baru;
   Akhir^.next := nil
end;
```

MENAMBAH SIMPUL

2. TAMBAH DEPAN

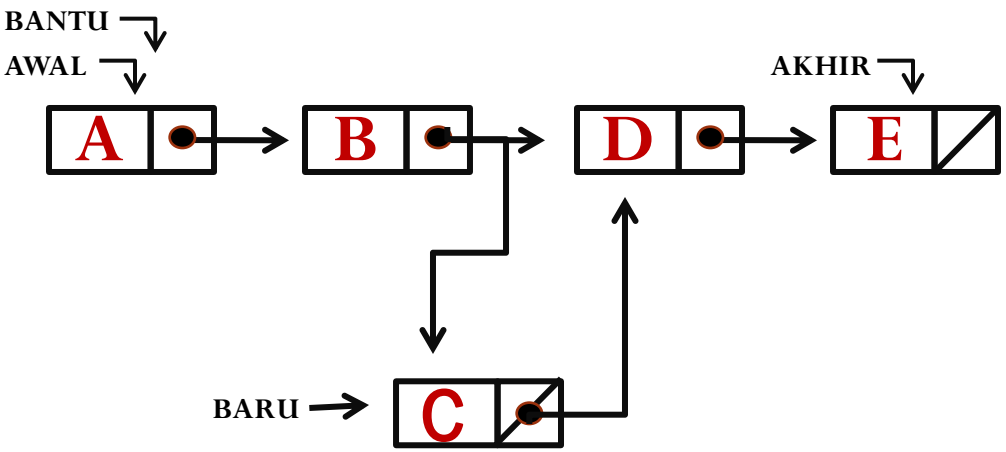



```
Type Simpul = ^Data;
     Data = record
        Isi     :  char;
        Next    :  Simpul;
     end;
var  Elemen  :  char;
     Awal, Akhir, Baru  :  Simpul;
```

```
Procedure TAMBAH(var Awal, Akhir :
Simpul; Elemen : char) ;

var Baru : Simpul;
begin
   new(Baru);
   Baru^.Isi:= Elemen;
   Baru^.next := Awal;
   Awal := Baru;
end;
```

MENAMBAH SIMPUL

3. TAMBAH TENGAH

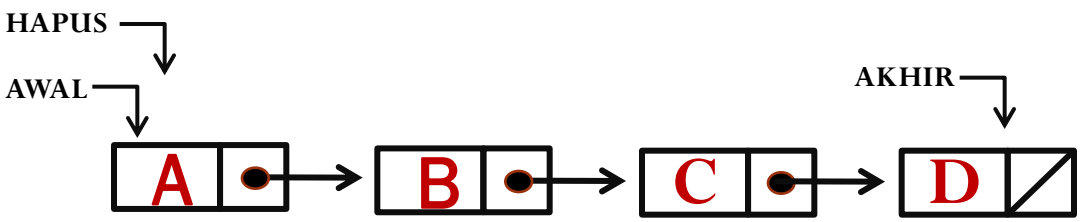



```
Type Simpul = ^Data;
     Data = record
        Isi     :  char;
        Next    :  Simpul;
     end;
var  Elemen  :  char;
     Awal, Akhir, Baru  :  Simpul;
```

```
Procedure TAMBAH(var Awal, Akhir : Simpul; Elemen : char) ;

var Bantu, Baru : Simpul;
begin
   new(Baru);
   Baru^.Isi:= Elemen;

   Bantu := Awal;
   While Elemen > Bantu^.next^.Isi do
      Bantu := Bantu^.Next;

   Baru^.next := Bantu^.Next;
   Bantu^.Next := Baru;
end;
```

MENGHAPUS SIMPUL

1. HAPUS DEPAN

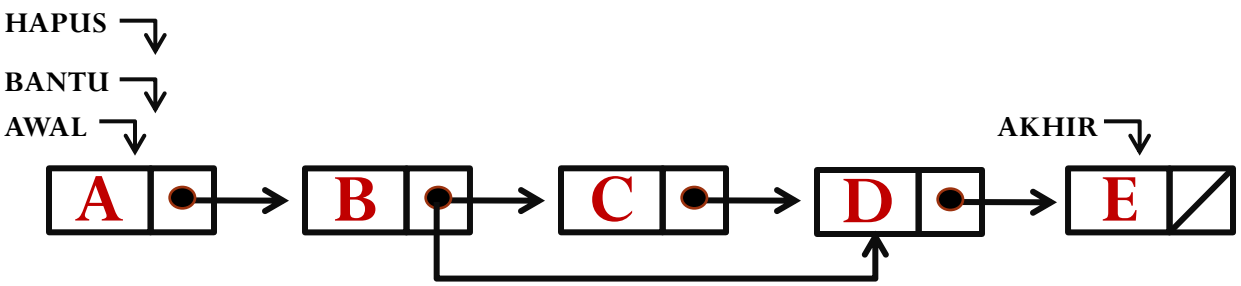



```
Type Simpul = ^Data;
     Data = record
        Isi     :  char;
        Next    :  Simpul;
     end;
var  Elemen  :  char;
     Awal, Akhir, Baru  :  Simpul;
```

```
Procedure HAPUS_SIMPUL(var Awal,
Akhir : Simpul; Elemen : char) ;

var Hapus : Simpul;
begin
   Hapus := Awal;
   Awal := Hapus^.Next;
   dispose(hapus);
end;
```

MENGHAPUS SIMPUL

2. HAPUS TENGAH

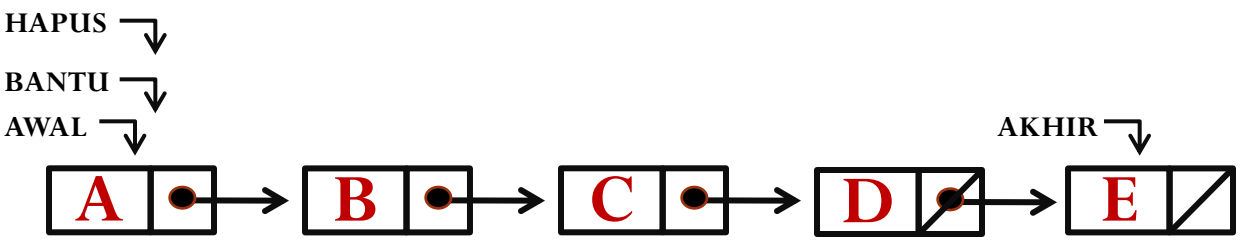



```
Type Simpul = ^Data;
     Data = record
        Isi     :  char;
        Next    :  Simpul;
     end;
var  Elemen  :  char;
     Awal, Akhir, Baru  :  Simpul;
```

```
Procedure HAPUS_SIMPUL(var Awal, Akhir : Simpul; Elemen :
char) ;
Var Bantu, Hapus : Simpul;
Begin
    Bantu := Awal;
    While (Elemen <> Bantu^.Isi) and (Bantu^.Next <> nil) do
        Bantu := Bantu^.Next;

    Hapus := Bantu^.Next
    Bantu^.next := Hapus^.Next;
    dispose(Hapus);
end;
```

MENGHAPUS SIMPUL

3. HAPUS BELAKANG




```
Type Simpul = ^Data;
     Data = record
        Isi     :   char;
        Next    :   Simpul;
     end;
var  Elemen  :  char;
     Awal, Akhir, Baru  :  Simpul;
```

```
Procedure HAPUS_SIMPUL(var Awal, Akhir  : Simpul; Elemen :
char) ;
Var Bantu, Hapus : Simpul;
Begin
    Bantu := Awal;
    While (Elemen <> Bantu^.Isi) and (Bantu^.Next <> nil) do
        Bantu := Bantu^.Next;

    Hapus := Bantu^.Next
    Akhir := Bantu;
    Akhir^.Next := nil;
    dispose(Hapus);
end;
```