

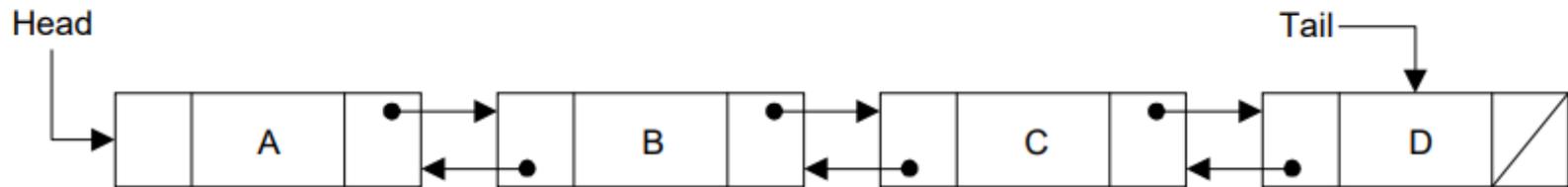
# Praktikum Struktur Data

## Pertemuan Ke-6

# Double Linked List

- ❑ Double linked list adalah linked list yang memiliki dua buah pointer yang menunjuk ke simpul sebelumnya (Prev) dan yang menunjuk ke simpul sesudahnya (Next).
- ❑ Data yang saling terhubung antar 1 data dengan data berikutnya.

# Bentuk Linked List



Gambar 1 Double Linked list dengan Empat Simpul

# Deklarasi Linked List

Deklarasi Double Linked List di dalam Pascal :

Type

PSimpul = ^Simpul

Simpul = Record

Info : Tipe Data;

Prev : PSimpul;

Next : PSimpul;

End;

Var

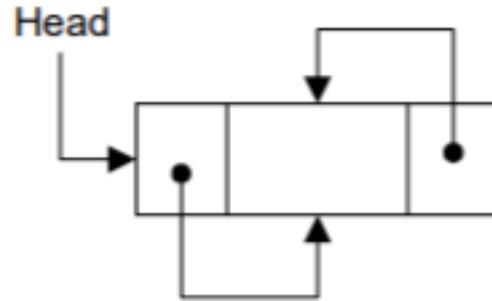
Head, Tail : PSimpul;

# Operasi Pada Double Linked List

Operasi yang dapat dilakukan pada Double Linked list diantaranya adalah :

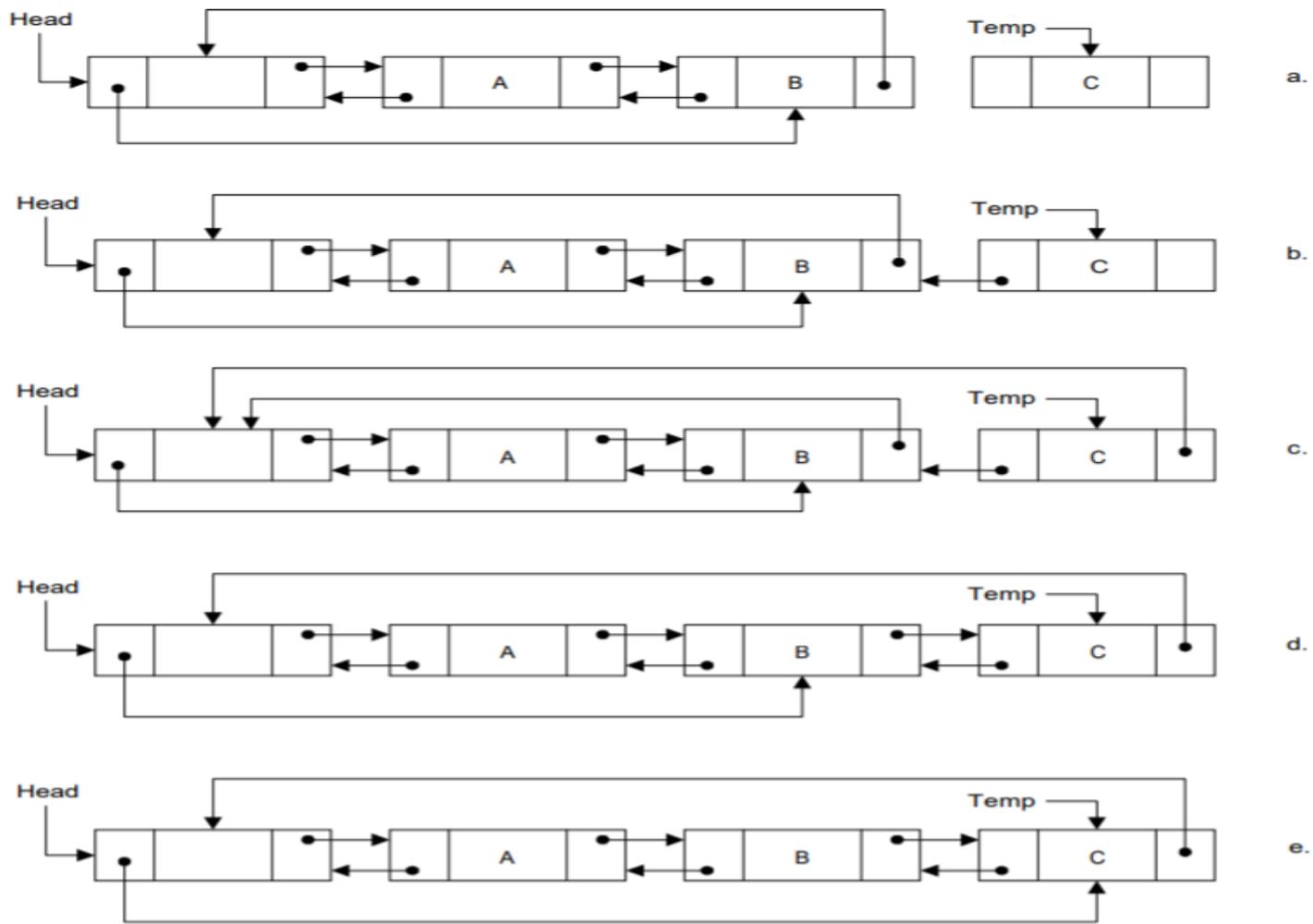
- a. Inisialisasi
- b. Menambah Simpul (di Depan, Belakang dan Tengah)
- c. Menghapus Simpul (di Depan, Belakang dan Tengah)
- d. Membaca isi linked list (Membaca maju dan mundur).

# Inisialisasi



```
Procedure Inisialisasi(Var Head : PSimpul);  
Begin  
    New(Head);  
    Head^.Prev := Head;  
    Head^.Next := Head;  
End;
```

# Menambah Simpul



# Menambah Simpul

```
Procedure Tambah(Var Head : PSimpul; Elemen : Char);  
Var Temp : PSimpul;  
Begin  
    New(Temp); Temp^.Info :=  
    Elemen; Temp^.Prev :=  
    Head^.Prev; Temp^.Next :=  
    Head; Head^.Prev^.Next :=  
    Temp; Head^.Prev := Temp;  
  
End;
```

# Menghapus Simpul

```
Procedure Hapus(Var Head : PSimpul; Elemen : Char);
Var Temp : PSimpul;
Begin
  If Head^.Next = Head Then { Jika Linked list Masih Kosong }
    Writeln('Double Linked list Masih Kosong')
  Else {Menghapus simpul tengah / }
    Begin {Akhir }
      Temp := Head
      { Memulai proses pencarian elemen yang akan dihapus }
      Repeat
        Temp := Temp^.Next;
      Until (Temp^.Info=Elemen) or (Bantu = Head);
      If Temp^.Info = Elemen Then { Jika simpul ketemu }
        Begin
          Temp^.Prev^.Next := Temp^.Next;
          Temp^.Next^.Prev := Temp^.Prev;
          Dispose(Temp);
          Temp := Head;
        End
      Else
        { Simpul yang akan dihapus tidak ketemu }
        Writeln('Simpul Tidak di ketemukan !');
    End;
End;
```

End;

# Contoh Program Double Linked List

```
1 program LinkedListDoubel;
2 {menggunakan double pointer, dengan menyisipkan data secara terurut}
3
4 uses crt;
5
6 type
7     PList = ^TList;
8     TList = record
9         info : integer;
10        prev : PList;
11        next : PList;
12    end;
13
14 procedure insert(var L:PList; x:integer);
15 var
16     baru,bantu : PList;
17 begin
18     new(baru);
19     baru^.info:=x;
20     baru^.next:=nil;
21     baru^.prev:=nil;
22
```

# Contoh Program Double Linked List

```
23 {insert data pertama}
24 if (L=nil) then
25 begin
26     L:=baru;
27 end else
28 {insert data didepan,jika node awal lebih besar atau node baru paling kecil}
29 if (L^.info>x) then
30 begin
31     baru^.next:=L;
32     L^.prev:=baru;
33     L:=baru;
34 end else
35 begin
36     bantu:=L;
37     while (bantu^.next<>nil) and (bantu^.next^.info<x) do
38         bantu := bantu^.next;
39
40 {insert tengah}
41 if bantu^.next<>nil then
42 begin
43 {mengkaitkan dengan node di belakang node}
44 baru^.next:=bantu^.next;
45 baru^.next^.prev:=baru;
46 end;
47
```

# Contoh Program Double Linked List

```
48 {untuk mengkaitkan node baru dengan node sebelumnya}
49 bantu^.next:=baru;
50 baru^.prev:=bantu;
51 end;
52 writeln('Insert ', x, ' sukses!');
53 writeln;
54 end;
55
56 procedure display(L:Plist);
57 begin
58     if L<>nil then
59     begin
60         while(L<>nil) do
61         begin
62             write(L^.info:5);
63             L:=L^.next;
64         end;
65     end else
66 write('Linked List kosong');
67 end;
68
```

# Contoh Program Double Linked List

```
69 procedure display2(akhir:PList);
70 begin
71     if akhir<>nil then
72     begin
73         while(akhir^.next <> nil) do
74             akhir:=akhir^.next;
75
76         while(akhir<>nil) do
77         begin
78             write(akhir^.info:5);
79             akhir:=akhir^.prev;
80         end;
81     end else
82     write('Linked List kosong');
83 end;
84
85 function count(P:PList):integer;
86 var
87     jum:integer;
88 begin
89     jum := 0;
90     while (P <> nil) do
91     begin
92         jum := jum+1;
93         P := P^.next;
94     end;
95 count := jum;
96 end;
97
```

# Contoh Program Double Linked List

```
98 procedure delete(var P:PList;x:integer);
99 var
100     hapus:PList;
101 begin
102     hapus:=P;
103     if P<>nil then
104     begin
105         while (hapus<>nil) and (hapus^.info<>x) do
106             hapus := hapus^.next;
107             if (hapus=nil) then
108                 writeln('Data tidak ditemukan')
109                 {delete data awal}
110             else if (hapus=P) then
111                 begin
112                     if(hapus^.next=nil) then
113                         begin
114                             hapus:=P;
115                             P:=nil;
116                         end else
117                         begin
118                             P:=P^.next;
119                             P^.prev:=nil;
120                         end;
121                 {delete data di akhir}
122             end else if (hapus^.next=nil) then
123                 begin
124                     hapus^.prev^.next:=nil;
125                 end else
126                 begin
127                     hapus^.prev^.next:=hapus^.next;
128                     hapus^.next^.prev:=hapus^.prev;
129                 end;
130
```

# Contoh Program Double Linked List

```
131 if(hapus<>nil) then
132 begin
133     dispose(hapus);
134     writeln('Delete ', x, ' sukses...');
135 end;
136 end
137 else
138 writeln('Linked List Kosong');
139
140 end;
141
142 procedure clear(var P:PList);
143 var
144     hapus:PList;
145 begin
146     hapus:=P;
147     if(P<>nil) then
148     begin
149         while hapus<>nil do
150         begin
151             P:=P^.next;
152             dispose(hapus);
153             hapus:=P
154         end;
155         writeln('Linked List Clear');
156     end else
157     writeln('Linked List sudah dalam keadaan kosong');
158 end;
159
```

# Contoh Program Double Linked List

```
160 var
161 P:PList;
162 x:integer;
163 jawab:char;
164 begin
165 P:=nil;
166
167 repeat
168 writeln('—————');
169 writeln('#*****MENU*****#');
170 writeln('|—————|');
171 writeln('|1. Insert |');
172 writeln('|2. Delete |');
173 writeln('|3. Display Ascending |');
174 writeln('|4. Display Descending |');
175 writeln('|5. Count |');
176 writeln('|6. Clear |');
177 writeln('|—————|');
178 writeln('|0. Exit |');
179 writeln('#—————#');
180 writeln;
181 write('Pilih yang mana? ');readln(jawab);
182 writeln;
183
```

# Contoh Program Double Linked List

```
184 case jawab of
185   '1':
186   begin
187     write('Input   : '); readln(x);
188     insert(P,x);
189   end;
190   '2':
191   begin
192     write('Delete  : '); readln(x);
193     delete(P,x);
194   end;
195   '3':
196   begin
197     write('Linked List Asc.      : '); display(P);
198   end;
199   '4':
200   begin
201     write('Linked List Desc.     : '); display2(P);
202   end;
203   '5': write('Total node : ', count(P));
204   '6': clear(P);
205   '0': writeln('Terimakasih');
206 else
207   writeln('Input anda salah, silahkan ulangi...');
208 end;
209   writeln;
210   until (jawab='0');
211 end.
```

# Tugas

Jalankan contoh program diatas, apa outputnya dan sebutkan metode apa saja yang ada pada program tersebut!

Thank you!

