

Praktikum Struktur Data

Pertemuan Ke-5

LINKED LIST

- ❑ Linked list atau biasa disebut senarai berantai adalah suatu kumpulan
- ❑ Data yang saling terhubung antar 1 data dengan data berikutnya.

Contoh Deklarasi Linked List Pada Pointer

type

```
tipeinfo = record  
    npm : string;  
    nilai : integer;
```

end;

```
tipeptr = ^tipenode;  
tipelist = tipeptr;  
tipenode = record  
    info : tipeinfo;  
    next : tipeptr;
```

end;

```
var list : tipelist;
```

Operasi Pada Linked List

Membuat List

```
1 procedure inisialisasi (var list : tipelist);  
2 begin  
3     new(list);  
4     list := nil;  
5 end;
```

Prosedur ini untuk membuat list pertama kali, yaitu mengalokasikan pointer untuk head. Nilai awal dari list adalah kosong (nil).

Operasi Pada Linked List

Mengetahui Panjang List (Jumlah Elemen)

```
1  function size (list : tipelist) : integer;
2  var i : integer;
3  begin
4      i := 0;
5      while list <> nil do
6          begin
7              i := i + 1;
8              list := list^.next;
9          end;
10     size := i;
11 end;
```

Mengetahui panjang list dilakukan dengan menghitung seluruh node. Caranya adalah mengunjungi setiap node dan menaikkan nilai counter sehingga dijumpai node terakhir.

Operasi Pada Linked List

Menyisipkan node baru

```
1 Procedure sisipnodebaru (var list : tipelist; sisipbaru : tipeinfo);
2 Var
3     Nodebaru, ptr : tipeptr;
4     Ketemu : boolean;
5 Begin
6     New(Nodebaru);
7     Nodebaru^.info := sisipbaru;
8     Nodebaru^.next := nil;
9     if list = nil then list := nodebaru
10    else
11        if sisipbaru.NPM <=list^.info.NPM then
12            Begin
13                Nodebaru^.next := list;
14                List := nodebaru;
15            End
16        Else
17            Begin
18                ketemu := false;
19                ptr := list;
20                while (ptr^.next <> nil) and not ketemu do
21                    Begin
22                        If ptr^.next^.info.NPM>= nodebaru.NPM
23                            Then ketemu := true
24                        Else
25                            ptr := ptr^.next
26                    End;
27                Nodebaru^.next := ptr^.next;
28                Ptr^.next := nodebaru
29            End
30 End;
```

Menyisipkan node baru pada list dilakukan dengan cara mencari lokasi tempat node baru akan disisipkan, kemudian menyisipkan node baru tersebut. Hal ini dapat dilakukan menggunakan bantuan sebuah pointer untuk mencari sebuah node yang akan tersambung langsung dengan node baru. Kemudian, node baru dapat disisipkan pada lokasi sebelum atau sesudah node tersebut.

Operasi Pada Linked List

Menghapus node dari suatu list

```
1 Procedure hapusnode (var list : tipelist; kuncihapus : string);
2   Var ptr1, ptr2 : tipeptr;
3 Begin
4   New(ptr1);
5   New(ptr2);
6   Ptr1 := nil;
7   Ptr2 := nil;
8   While (ptr2^.info.NPM <> kuncihapus) do
9   Begin
10    ..... Ptr1 := ptr2;
11    ..... Ptr2 := ptr2^.next;
12  End;
13  If ptr1 = nil then
14  ..... List := list^.next
15  Else
16  Ptr1^.next := ptr2^.next
17  Dispose (ptr2)
18 End;
```

Menghapus node dapat dilakukan dengan mencari/menandai node yang akan dihapus, yaitu node yang memuat nilai seperti yang akan dihapus, kemudian mengarahkan pointer pada node sebelumnya ke arah node sesudah node yang akan dihapus, dan kemudian menghapus node yang dimaksud

Operasi Pada Linked List

Mengganti Nilai Informasi pada Suatu Node dalam List

Mengganti nilai informasi hanya mengganti info pada suatu node tanpa menghapus node tersebut.

Hal ini dapat dilakukan dengan mencari node yang sesuai dengan nilai yang akan diganti, selanjutnya mengganti nilai lama dengan nilai yang baru.

Contoh : prosedur untuk mengganti nilai pada suatu list

- a. Mengganti nilai mahasiswa berdasarkan nomor mahasiswa,
- b. Mengganti semua node yang mempunyai nilai tertentu (nilai lama) dengan nilai yang baru.

Contoh prosedur untuk mengganti nilai pada suatu list

```
1 procedure gantinodel (var list : tipelist; kunciGanti : string; nilaibaru : integer);
2 var ptr : tipeptr;
3 begin
4     new(ptr);
5     ptr := list;
6     while (ptr <> nil) and (ptr^.info.nim <> kunciganti) do
7         ptr := ptr^.next;
8     if ptr <> nil then
9         ptr^.info.nilai := nilai baru
10 end;
```

```
1 procedure gantinode2(var list : tipelist; nlama, nbaru : integer);
2 var ptr : tipeptr;
3 begin
4     new(ptr);
5     ptr := list;
6     while (ptr <> nil) do
7         begin
8             if ptr^.info.nilai := nlama then
9                 ptr^.info.nilai := nbaru;
10            ptr := ptr^.next
11        end;
12 end;
```

Contoh : LIFO pada Linked List

```
1 program lifo;
2 uses wincrt;
3 Type
4 Point      = ^RecPoint;
5 Recpoint   = Record
6           Isi : string;
7 Next,pre   : Point;
8 End;
9 Var
10 Head, tail, Now : Point;
11
12 procedure create;
13 begin
14   head := nil ; tail := nil;
15 end;
16
17 {Function Empty : Boolean;
18 Begin
19 If head = nil then
20   Empty:= true
21 else
22   empty:= false;
23 end;}
24
```

Contoh : LIFO pada Linked List

```
25 procedure find_first;
26 begin
27     now := head;
28     write (now^.isi);
29 end;
30
31 procedure find_next;
32 begin
33     if now^.next <> nil then
34         now := now^.next;
35         write(now^.isi);
36     end;
37
```

Contoh : LIFO pada Linked List

```
38 Procedure INSERT(elemen:string);
39 Var Now : Point;
40 Begin
41     New(Now);
42     Now^.Isi:=elemen;
43     If Head=Nil then
44         Now^.Next:=Nil
45     Else
46     Now^.Next:=Head;
47     Head:=Now;
48 End;
49
50 begin
51 clrscr;
52 insert(' tv ');
53 insert(' vcd ');
54 insert(' compo ');
55 writeln;
56 write('data pertama      :');
57 find_first;writeln;
58 write('data berikutnya :');
59 find_next;
60 readln;
61 end.
```

Thank you!

