

POINTER PADA PASCAL

PEMBAHASAN :

1. Mengetahui tipe data Pointer.
2. Manipulasi memori lewat Pointer bertipe dan tak bertipe.
3. Linked List; meliputi operasi inisialisasi, menambah node baru, menyisipkan node baru, menghapus node yang berisi data, membaca data dari node.

Pointer merupakan suatu tipe data dalam Pascal yang berfungsi untuk menunjuk dan menyimpan alamat memori. Dalam penulisan pointer biasa digambar dengan panah, sedangkan bagian memori yang ditunjuk digambar dengan kotak, dan isinya ditulis di dalam kotak.

Bentuk umum dari deklarasi tipe pointer:

- Untuk pointer **bertipe**:
`<nama_var> : ^<tipe_data>;`
- Untuk pointer **tidak bertipe**:
`<nama_var> : pointer;`

Suatu pointer dapat menunjuk ke data **bertipe** *elementer*, *terstruktur*, *pointer yang lain*, atau *tidak bertipe*. Jika suatu pointer *tidak menunjuk ke mana-mana*, pointer itu dinamakan *dangling*, sedangkan bagian memori yang tidak dapat diakses karena tidak ada pointer yang menunjuk dinamakan *garbage* (sampah).

Dalam Pascal, pointer dapat diisi dengan nilai yang berasal dari:\

1. NIL
2. Fungsi Ptr
3. Operator @
4. Prosedur New dan GetMem
5. Pointer yang lain

Reserved word NIL

NIL merupakan reserved word dalam Pascal, di mana pointer yang bernilai NIL dianggap tidak menunjuk alamat memori manapun. NIL biasa digambarkan dengan lambang ground.

Fungsi Ptr

Sintaks:

```
Function Ptr(Seg, Ofs : word) : pointer;
```

dengan *Seg* : segmen memori.

Ofs : offset memori.

Fungsi Ptr mengembalikan pointer dari segmen dan offset yang dimasukkan.

Operator @

Sintaks:

```
<nama_var>:=@<variabel_yang_alamatnya_diambil>;
```

Operator ini digunakan untuk mengambil alamat variabel yang akan ditunjuk.

Prosedur New dan GetMem

Sintaks:

```
New(var P : pointer);
```

```
GetMem(var P : pointer, size : word);
```

Dengan *P* : pointer yang akan diisi.

Size : ukuran yang dipesan.

Prosedur **New** digunakan untuk memesan memori untuk *pointer bertipe*, sedangkan prosedur **GetMem** untuk *pointer tidak bertipe*. Kedua prosedur ini akan membentuk suatu variabel dinamik yang diletakkan dalam *Heap*. **Heap** adalah memori-memori di komputer yang belum dialokasikan, yaitu memori yang tidak digunakan oleh DOS, oleh program-program resident, oleh program Turbo Pascal, internal stack yang digunakan oleh Turbo Pascal dan variabel-variabel di data segmen.

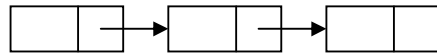
Pointer yang belum digunakan sebaiknya diisi dengan NIL, dan untuk pointer yang telah menunjuk sebuah alamat yang sudah dipesan memorinya, isinya dapat dimanipulasi melalui pointer.

Linked List

Linked List (daftar berkait) adalah suatu node (simpul) yang menunjuk ke node (simpul) berikutnya dalam suatu urutan. Suatu node minimal harus mempunyai 2 komponen, yaitu:

1. Satu atau lebih field yang berisi data di daftar berkait.

2. Satu atau lebih field berupa pointer yang menunjuk ke node lainnya. Field yang berupa pointer ini disebut dengan link.



Untuk mendefinisikan linked list biasa digunakan record, dengan sintaks:

```

Type <nama_pointer> = ^<nama_rec>;
  <nama_rec> = record
      data1 : <tipe_data1>;
      data2 : <tipe_data2>;
      . . . .
      next : <nama_pointer>;
  end;
  
```

Masing-masing kotak pada linked list disebut node (simpul), node paling depan disebut head, node belakang disebut tail. Untuk mengisi linked list kosong maupun mengakhiri linked list biasanya digunakan nilai NIL.

Operasi-operasi yang terdapat pada linked list antara lain:

1. Inisialisasi; yaitu mengisi variabel list dengan NIL.
2. Menambah node.
3. Menyisipkan node.
4. Menghapus node yang berisi data.
5. Membaca data dari node.
6. Menghapus seluruh isi list.

CONTOH PENGGUNAAN POINTER

❖ Deklarasi tipe data pointer:

```

➤ program deklarasi;
  uses crt;
  var p : ^integer;
      nilai : integer;
  begin
    clrscr;
    nilai:=12;
    p:=@nilai;
    writeln(p^);
    p^:=100;
  end;
  
```

```
writeln(p^);
writeln(nilai);
readln;
end.
```

➤ Output: 12
 100
 100

➤ Pembahasan:

Pada contoh program deklarasi ini, pertama-tama dideklarasikan variabel *p* sebagai pointer yang bertipe integer. Dibuat sebuah variabel lagi yang diberi nama *nilai* dan bertipe integer.

Variabel *nilai* diisi dengan nilai 12. Kemudian variabel *p* menunjuk alamat dari variabel *nilai* dengan operator *@*, sehingga variabel *p* berisi nilai 12, dan ditampilkan outputnya di layar. Kemudian variabel *p* diberi nilai 100, dan secara otomatis variabel *nilai* juga bernilai 100 karena sudah ditunjuk oleh variabel *p*. Kemudian isi dari variabel *p* yang baru dan variabel *nilai* ditampilkan di layar.

❖ Manipulasi memori lewat Pointer bertipe:

```
uses wincrt;
var p,q,r,s : ^string;
    nilai : string;
begin
  clrscr;
  nilai:='ini string nilai!!!';
  p:=@nilai;
  new(q);
  q^:='ini isi pointer q!!!';
  r:=q;
  s^:=q^;
  writeln('alamat pointer p = ',seg(p^):4,':',ofs(p^):4,';
isi pointer p = ',p^);
  writeln('alamat pointer q = ',seg(q^):4,':',ofs(q^):4,';
isi pointer q = ',q^);
  writeln('alamat pointer r = ',seg(r^):4,':',ofs(r^):4,';
isi pointer r = ',r^);
```

```
writeln('alamat pointer s = ',seg(s^):4,':',ofs(s^):4,');
isi pointer s = ',s^);
readln;
end.
```

Output program:

```
alamat pointer p = 4446: 98; isi pointer p = ini string nilai!!!
alamat pointer q = 5559: 0; isi pointer q = ini isi pointer q!!!
alamat pointer q = 5559: 0; isi pointer r = ini isi pointer q!!!
alamat pointer q = 0: 0; isi pointer s = ini isi pointer q!!!
```

Pembahasan:

Pada program ini dibuat 4 variabel pointer yang bertipe string, yaitu p, q, r, s. Dan sebuah variabel dengan nama nilai yang bertipe juga string (bukan pointer).

Pertama-tama variabel nilai diisi dengan nilai 'ini string nilai!!!'. Kemudian variabel p (pointer) menunjuk alamat dari variabel nilai dengan operator @, sehingga variabel p juga berisi 'ini string nilai!!!'. Kemudian dipesan tempat pada heap dengan prosedur New untuk variabel q. Kemudian variabel q diisi 'ini isi pointer q!!!'. Setelah itu variabel r menunjuk alamat dari variabel q, sehingga alamat maupun isinya sama. Kemudian variabel s diisi dengan nilai dari variabel q, tapi dalam hal ini hanya nilainya saja, sedangkan alamatnya tidak.