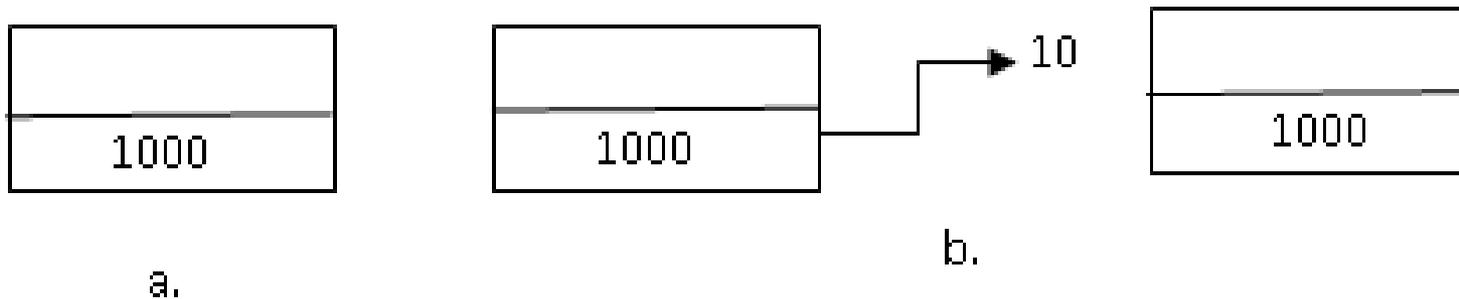


POINTER

Struktur Data

PENDAHULUAN

- Pada materi sebelumnya telah dijelaskan mengenai variable bertipe Array, suatu tipe data yang bersifat statis(ukuran dan urutannya sudah pasti). Selain itu ruang memori yang dipakai olehnya tidak dapat dihapus bila variable bertipe array tersebut sudah tidak digunakan lagi pada saat program dijalankan.
- Untuk memecahkan masalah diatas, kita dapat menggunakan variable pointer. Tipe data pointer bersifat dinamis, variable akan dialokasikan hanya pada saat dibutuhkan dan sesudah tidak dibutuhkan dapat dialokasikan kembali.



Gambar 1. ilustrasi perubah statis dan dinamis

- Gbr a : Perubah A adalah perubah statis. Dalam hal ini 1000 adalah nilai data yang sesungguhnya dan disimpan pada perubah (lokasi) A.
- Gbr b : Perubah A adalah perubah dinamis. Nilai perubah ini, misalnya adalah 10. Nilai ini bukan nilai data yang sesungguhnya, tetapi lokasi dimana data yang sesungguhnya berada. Jadi dalam hal ini nilai data yang sesungguhnya tersimpan pada lokasi 10

○ Array VS Pointer

Kriteria	Array	Pointer
Sifat	Statis	Dinamis
Ukuran	Pasti	Sesuai kebutuhan
Alokasi variable	Saat program dijalankan sampai selesai	Dapat diatur sesuai kebutuhan

PENGERTIAN POINTER

- Pointer merupakan suatu tipe data dalam Pascal yang berfungsi untuk menunjuk dan menyimpan alamat memori.
- Pengertian pointer (santoso, 1992) adalah suatu tipe data yang dapat digunakan untuk mengalokasikan dan mendealokasikan (mengambil/mengurangi) pengingat secara dinamis, yaitu sesuai dengan kebutuhan pada suatu program dieksekusi.

PENDEKLARASIAN VARIABEL POINTER

- Bentuk umum :

```
Type pengenal = ^simpul ;  
    Simpul     = tipe ;
```

- Dengan pengenal : nama pengenal yang menyatakan data bertipe pointer
- Simpul : nama simpul
- Tipe : tipe data dari simpul
- Pendeklarasian variable pointer tidak jauh berbeda dengan pendeklarasian variable biasa, hanya perlu **ditambahkan symbol topi (^)** biasa juga disebut ceret atau circumflex. Simbol topi tersebut menandakan bahwa variable tersebut menunjukan ke lokasi tertentu di memori.

- **Contoh :**

- `Typestr30 = string[30];`

- `Point = ^data;`

- `Data = record`

- `Nama_peg : str30;`

- `Alamat : str30;`

- `Pekerjaan : str30;`

- `End;`

- `Var p1, p2 : point;`

- `A, B, C : str30`

- **Note :** P1, p2 bertipe pointer.

- A, B, dan C perubah statis yang bertipe string.

STATEMEN NEW

- Untuk mengalokasikan simpul dalam pengingat, statement yang digunakan adalah statemen *new*, yang mempunyai bentuk umum :

```
new (perubah);
```

dengan perubah adalah nama perubah yang bertipe pointer

Contoh :

```
new(p1);
```

```
new(p2);
```

OPERASI PADA POINTER

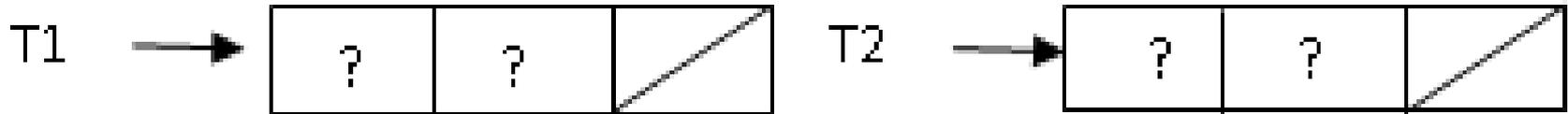
- Mengkopi pointer(sebuah simpul akan ditunjuk oleh lebih dari sebuah pointer)
- Mengkopi isi simpul(dua atau lebih simpul yang ditunjuk oleh pointer yang berbeda mempunyai isi yang sama)

Contoh :

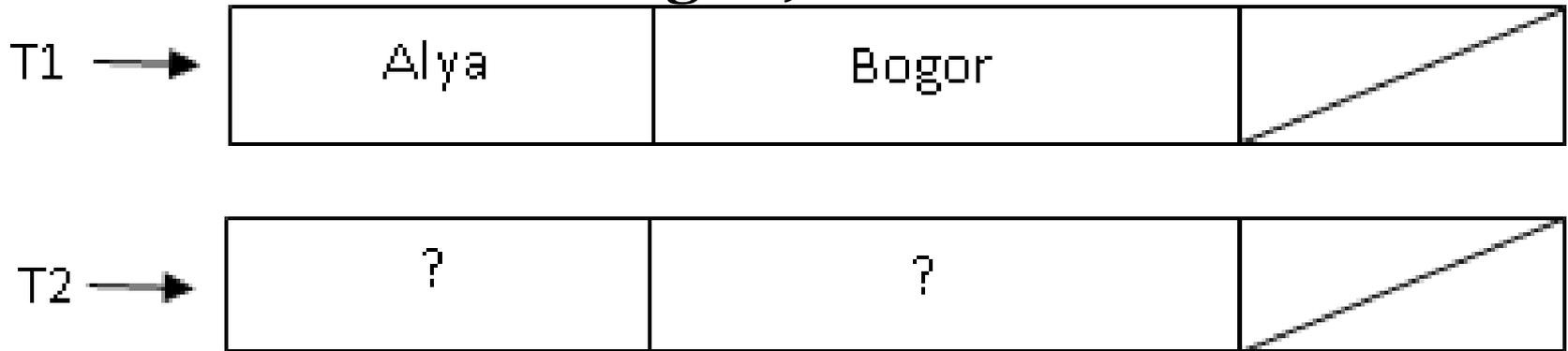
```
Type simpul = ^data
  Data = record
    Nama : string;
    Alamat : string;
    Berikut: simpul;
End;
Var T1, T2 : simpul;

New (T1);
New (T2);
```

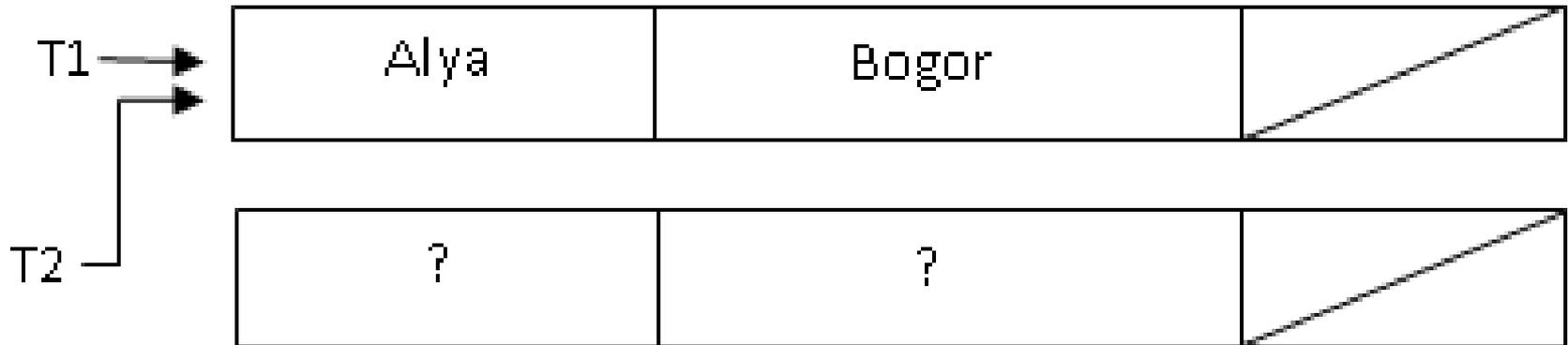
- **Contoh :**
- Kita mempunyai dua simpul :



- Dengan menggunakan statement :
- $T1^{.nama} := 'Alya';$
- $T1^{.alamat} := 'Bogor';$

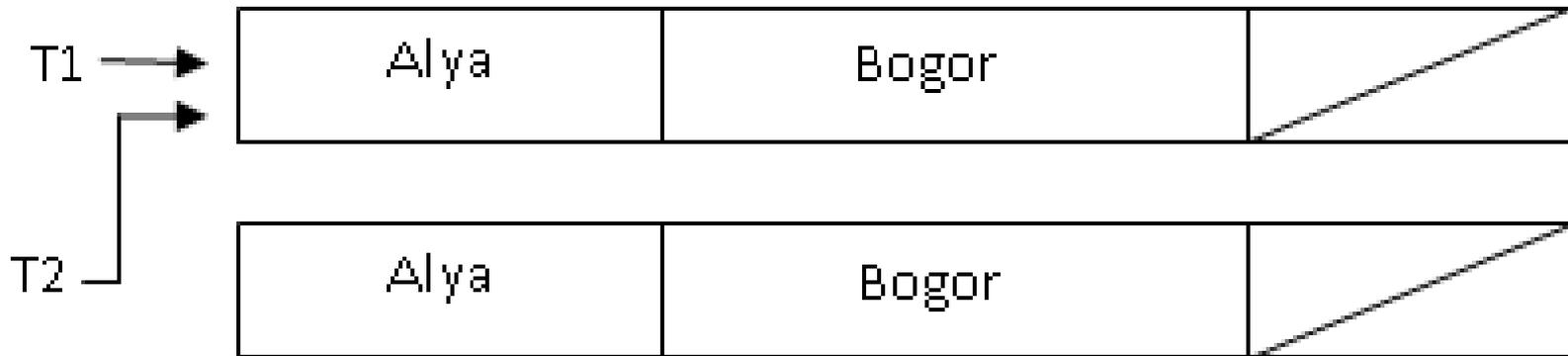


- Jika kita memberikan statement :
- $T2 := T1;$
- Maka gambar diatas berubah menjadi



Note : pointer T2 juga menunjuk ke simpul yang ditunjuk oleh pinter T1; simpul yang semula ditunjuk T2, menjadi terlepas. Dalam keadaan seperti ini, karena tidak ditunjuk oleh suatu pointer lain, maka simpul tersebut tidak bisa dimasup lagi karena lokasi simpul tersebut dalam pengingat tidak diketahui lagi (kecuali apabila simpul ini ditunjuk oleh pointer yang lain). Operasi inilah yang disbut dengan operasi **mengkopi pointer**.

- Jika kita memberikan statement :
- $T2^{\wedge} := T1^{\wedge};$
- Maka hasil yang kita peroleh adalah :



- Operasi inilah yang disebut operasi mengkopi isi simpul.

- **Note :**
- Jika dalam statement pemberian tanda ^ tidak ditulis, operasinya disebut operasi mengkopi pointer dengan konsekuensi, simpul yang semula ditunjuk oleh suatu pointer akan bisa terlepas dan tidak bias dimasup lagi
- Jika dalam statement pemberian tanda ^ ditulis, operasinya disebut operasi mengkopi isi simpul pointer dengan konsekuensi bahwa isi dua simpul atau lebih akan menjadi sama.

OPERATOR @

- Bentuk Umum :

```
⟨nama_var⟩ := @⟨variabel_yang_alamatnya_diambil⟩;
```

- Operator ini digunakan untuk mengambil alamat variabel yang akan ditunjuk

MENGHAPUS POINTER

- Bentuk umum :

```
dispose (perubah);
```

- Diatas telah dijelaskan bawa pointer yang telah dialokasikan (dibentuk) bisa didealokasikan(dihapus) kembali pada saat prigram diekekusi. Setelah suatu pointer dihapus, makalokasi yang semula ditempati oleh simpul yang ditunjuk oleh pointer tersebut akan bebas, sehingga bias digunakan oleh perubah lain

CONTOH PROGRAM

- **program pointer;**
- uses wincrt;
- var p : ^integer;
- m,n : integer;
- begin
- new(p);
- m := 10;
- n := 15;
- p := @m; {p sekarang menunjuk ke m}
- p^ := 12; {hal ini sama dengan m := 12}
- p := @n; {p sekarang menunjuk ke n}
- p^ := m; {hal ini sama dengan n:=m}
- writeln('m=',m,',n=',n);
- readln;
- dispose(p);
- end.

- Latihan :
 1. Apa output dari program diatas?
 2. Jelaskan penggunaan operator @, New, dispose,tanda ^ pada program diatas !