

## BAB III

### KOMBINASI TIPE DATA LARIK DAN SENARAI

#### III.1 Struktur data dalam bahasa C.

Pada bab sebelumnya telah dijelaskan beberapa tipe data dari bahasa C seperti int (bulat) ,float (pecahan), char (karakter) dsb. Pada bab ini akan dijelaskan bermacam-macam tipe data lebih lanjut yang sering digunakan dalam pembuatan program.

##### III.1.1 Larik

Larik adalah kumpulan data dengan setiap elemen data menggunakan nama dan tipe data yang sama. Sedangkan setiap elemen larik dapat diakses dan dibedakan melalui indeks larik.

##### A. Larik berdimensi satu.

Suatu larik dimensi satu dapat dideklarasikan dalam bentuk umum :

```
tipe nama_variabel[ukuran];
```

- tipe menyatakan tipe data dari elemen larik.
  - nama\_variabel merupakan nama yang dibuat sendiri oleh penulis program yang namanya akan menjadi variabel larik.
  - ukuran untuk menyatakan jumlah maksimal elemen larik.
- dalam bahasa C elemen pertama dari suatu larik dimulai dengan indeks 0 (nol).

contoh : `int nilai[5];`

menunjukkan suatu larik dimensi satu dengan nama nilai dan mengandung 5 elemen bertipe int.

Perhatikan contoh program berikut :

```
#include <stdio.h>
main()
{
    register int i;
    float total,rata,nilai[5];

    for(i=0 ; i<3 ; i++)
    < printf("Nilai tes ke %d = ",i+1);
      scanf("%f",&nilai[i]);
      total=total+nilai[i];
    >
    rata=total/3;
    printf("\nRata-rata nilai = %0.2f ",rata);
}

```

Hasil program :

Nilai tes ke 1 = 60  
 Nilai tes ke 2 = 70.5  
 Nilai tes ke 3 = 50.5

Rata-rata nilai = 60.33

#### B. Larik berdimensi dua.

Larik dimensi dua dapat digunakan untuk menyimpan data yang berupa tabel atau matrik.

Bentuk umum pendeklarasian :

```
type nama_variabel[baris][kolom];
```

contoh : `int matrik[2][4];`

menunjukkan suatu larik dimensi dua dengan nama matrik dan mengandung 2 elemen baris dan 4 elemen kolom yang bertipe data int.

contoh :

$$X = \begin{bmatrix} 8 & 5 & 9 & 8 \\ 8 & 2 & 1 & 0 \end{bmatrix}$$

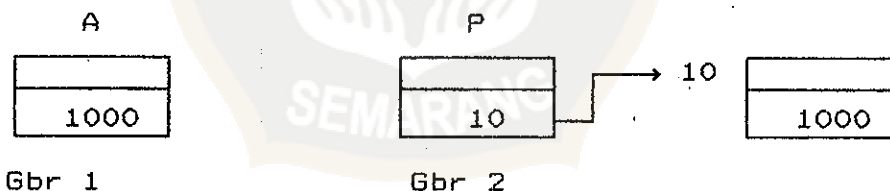
data matrik X di atas dapat dimasukan kedalam variabel matrik yang ditunjukkan dalam program berikut :

```
#include <stdio.h>
main()
< int matrik[2][4],i,j;

for(i=0 ; i<2 ; i++)
for(j=0 ; j<4 ; j++)
< printf(" X[%d %d] = ",i,j);
scanf("%d",&matrik[i][j]);
>
>
```

### III.1.2 Pointer.

Untuk mengetahui pengertian dari pointer, perhatikan kedua gambar berikut :



Pada Gbr 1 variabel A adalah bukan variabel pointer, sebab nilai 1000 adalah nilai yang sesungguhnya. Sedangkan pada Gbr 2 variabel P merupakan variabel pointer, sebab nilai 10 adalah bukan nilai yang sebenarnya tetapi merupakan lokasi dimana data yang sesungguhnya berada (nilai data yang sesungguhnya tersimpan di lokasi 10 ).

Dari ilustrasi di atas dapat dilihat bahwa variabel pointer akan digunakan untuk menunjuk ke lokasi lain yang berisi data sesungguhnya yang akan diproses.

#### A. Mendeklarasikan variabel pointer.

Bentuk umum pendeklarasian variabel pointer :

```
tipe *nama_variabel;
```

tipe menunjukkan tipe dari data yang ditunjuk oleh pointer tersebut.

contoh :

```
int *px; /* contoh 1*/
char *pc; /* contoh 2*/
```

pada contoh 1 diatas, menyatakan bahwa px adalah variabel pointer yang menunjuk suatu data bertipe int. Sedangkan contoh 2 menyatakan bahwa pc adalah variabel pointer yang menunjuk suatu data bertipe char.

#### B. Mengatur pointer menunjuk variabel lain.

Seperti telah dijelaskan di atas, bahwa variabel pointer haruslah berisi alamat, sehingga agar suatu pointer menunjuk ke variabel lain, mula-mula pointer harus diisi dengan alamat variabel yang akan ditunjuk.

contoh :

```
#include <stdio.h>
main()
< int x=87;
  int *px;

  px=&x;
  printf("Variabel px berisi =%p\n",px);
  printf("Variabel x berisi =%d\n",x);
>
```

Hasil program :

```
Variabel px berisi= FFEO
Variabel x berisi = 87
```

### C. Mengakses suatu variabel melalui pointer.

Jika suatu variabel sudah ditunjuk oleh pointer tertentu maka variabel tersebut dapat diakses melalui pointer tersebut (disebut pengaksesan cara tak langsung).

Pengaksesan cara ini dilakukan dengan menggunakan operator \* yang ditulis didepan pointer-nya.

contoh :

```
#include <stdio.h>
main()
{ int x=20,y=30;
  int *px,*py;

  px=&x;py=&y;
  *px=*px+*py;
  *py=*py+10;
  printf(" x = %d, y=%d ",x,y);
}
```

Hasil program :

```
x = 50    y = 40
```

### III.1.3 Struktur.

Berbeda dengan larik, struktur adalah pengelompokan dari variabel-variabel yang bernaung dalam suatu nama, dengan sifat setiap variabel yang bernaung tersebut dapat memiliki tipe yang berlainan.

#### A. Mendeklarasikan struktur.

Suatu struktur dapat dideklarasikan dengan kata kunci **struct**.

contoh :

```
struct Data_Brg
{
  char Brg[30];
  int Unit;
  float Harga;
}
```

Perlu diketahui bahwa Data\_Brg merupakan nama dari tipe data struktur (bukan nama variabel). Variabel struktur itu sendiri dapat dideklarasikan dengan cara sbb:

```

struct Data_Brg  Beli, Jual;
    |           |
    |           |
    v           v
tipe data  Variabel-variabel

```

cara lain adalah dengan menggabungkan kedua deklarasi tersebut.

```
struct Data_Brg
{
  char Brg[30];
  int Unit;
  float Harga;
}>Beli, Jual;
```

B. Mengakses elemen struktur.

Elemen-elemen struktur dapat diakses secara individual dengan menggunakan bentuk :

Variabel\_struktur.nama\_field

antara variabel struktur dan nama field dipisah dengan tanda titik.

contoh :

```
#include <stdio.h>
main()
{
  struct data_rekan
  {
    char nama[40];
    int tgl;
  };
```

```

int bln;
int thn;
>temanku={"Sumarno",04,09,72};

printf("Nama      : %s\n",temanku.nama);
printf("Tgl lahir: %d - %d - %d\n",
      temanku.tgl,temanku.bln,temanku.thn);
>

```

Hasil program :

```

Nama      : Sumarno
Tgl lahir : 04 - 09 -72

```

### C. Pointer untuk data struktur.

Seperti halnya pointer untuk variabel biasa, pointer untuk struktur juga menunjuk alamat letak dari variabel strukturnya.

Perhatikan tabel di bawah ini :

Pointer untuk Variabel biasa	Pointer untuk Struktur
<pre> int x,*ptr_x ptr_x=&amp;x; </pre>	<pre> struct data { char nama[40];   float IP; }mhs,*ptr_mmhs; ptr_mhs=&amp;mhs; </pre>

Pengaksesan elemen struktur melalui pointer dapat dilakukan dengan penulisan :

```
ptr_mhs->IP ;
```

Pernyataan diatas menyatakan pengaksesan elemen IP melalui variabel pointer ptr\_mhs.

contoh :

```
#include <stdio.h>

main()
{ struct data_rekan
  { char nama[40];
    int tgl;
    int blnn;
    int thn;
  }temanku={"Sumarno",04,09,72},*ptr;

  printf("Nama      :%s\n",ptr->nama);
  printf("Tgl lahir:%d - %d - %d\n",
    ptr->tgl,ptr->bln,ptr->thn);
```

Hasil program:

```
Nama      : Sumarno
Tgl lahir : 04 - 09-72
```

### III.1.4 Senarai.

Di atas telah dijelaskan bahwa, salah satu cara untuk menyimpan sekumpulan data yang dimiliki adalah dengan menggunakan tipe data larik. Namun pemakaian tipe data larik tidak selalu tepat untuk program-program terapan yang kebutuhan penyimpanannya selalu bertambah selama eksekusi program berlangsung. Untuk itu diperlukan suatu tipe data lain yang bisa digunakan untuk mengalokasikan dan mendealokasikan penyimpanan secara dinamis sesuai kebutuhan pada saat suatu program dieksekusi. Tipe data inilah yang disebut tipe data senarai.

Dengan demikian tipe data senarai adalah : kumpulan komponen-komponen yang disusun secara berurutan dengan bantuan pointer. Masing-masing komponen dari tipe data senarai disebut dengan simpul. Setiap tubuh simpul dalam senarai terbagi menjadi dua bagian, yaitu :



1. Medan Informasi : Berisi informasi atau data yang disimpan dan diolah.
2. Medan Penyambung : Berisi alamat simpul-simpul lain yang mengapitnya.

perlu diingat bahwa medan penyambung adalah suatu pointer, sehingga nilai dari medan ini adalah alamat dari suatu lokasi tertentu.

#### A. Fungsi malloc() dan fungsi free().

Dalam kaitannya bekerja dengan tipe data senarai, ada dua buah fungsi yang sering digunakan dalam mengolah tipe data senarai.

##### 1. Fungsi malloc().

Fungsi ini digunakan untuk membuat simpul dengan cara memesan atau mengalokasikan memori.

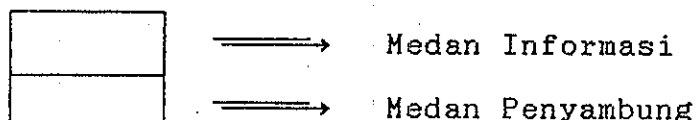
##### 2. Fungsi free().

Fungsi ini merupakan kebalikan dari fungsi malloc(), yaitu digunakan untuk menghapus simpul yang sudah pernah dibuat.

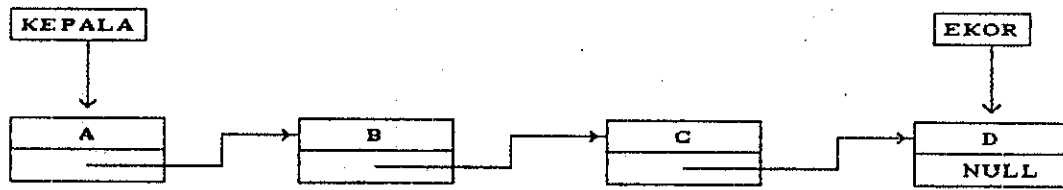
#### B. Senarai Model Single

Yang dimaksud dengan Senarai Model Single yaitu : Suatu senarai dimana medan penyambung dari tiap-tiap simpulnya hanya memiliki satu buah pointer.

Gbr Simpul pada Senarai Model Single



Gambar berikut menunjukkan diagram skematis dari senarai model single yang memiliki 4 buah simpul.

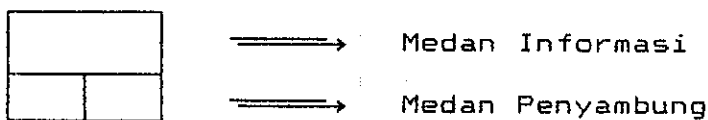


KEPALA dan EKOR bukan merupakan bagian dari senarai akan tetapi digunakan untuk menunjuk ke simpul pertama dan terakhir dari senarai tersebut. Medan penyambung dari suatu simpul yang tidak menunjuk simpul lain akan diisi dengan nilai NULL (seperti pada simpul D).

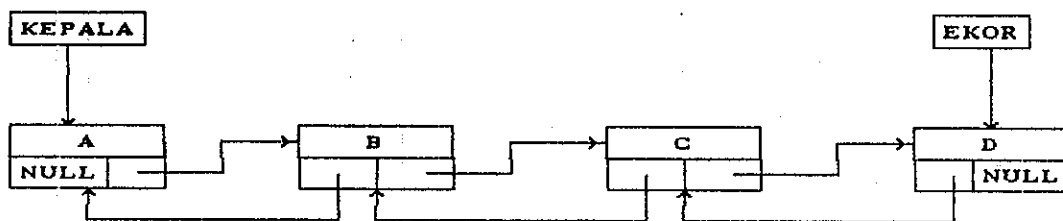
C. Senarai Model Double.

Berbeda dengan model single, senarai model double memiliki dua buah pointer pada medan penyambungannya, dengan pointer pertama menunjuk simpul disebelah kirinya, sedangkan pointer kedua menunjuk simpul disebelah kanannya.

Gbr Simpul pada Senarai Berantai Model Double



Perhatikan senarai model double berikut :



Dari Implementasi gambar di atas dapat di lihat bahwa masing-masing simpul menyimpan alamat dari simpul yang berada di sebelah kanan dan kirinya , kecuali untuk simpul yang berada di ujung ,dimana salah satu dari medan penyambungannya berisi nilai NULL.

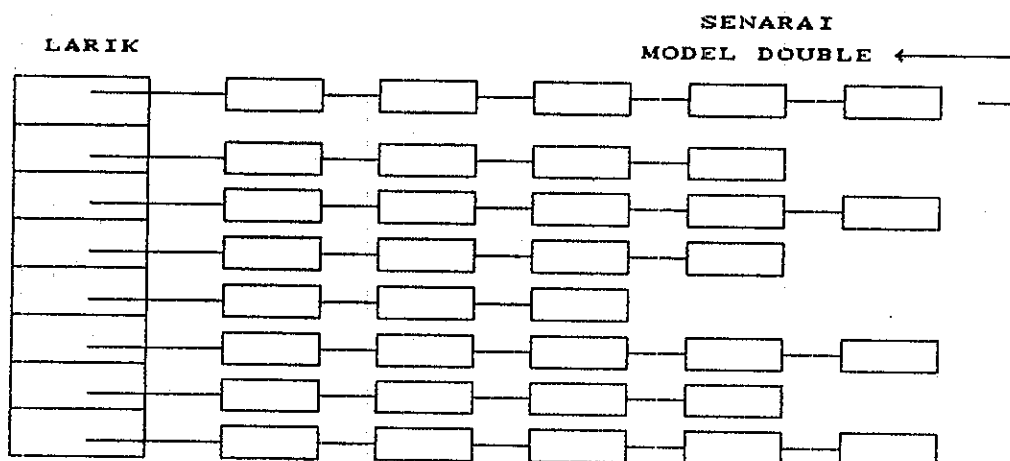
### III.2 Kombinasi tipe data larik dengan senarai.

Untuk menggabungkan tipe data larik dan senarai ada beberapa syarat atau aturan yang harus dipenuhi, yaitu :

1. Larik tidak digunakan untuk menyimpan data akan tetapi digunakan untuk menyimpan alamat kepala dari senarai.
2. Elemen-elemen dari larik harus bertipe pointer.
3. Satu elemen larik hanya digunakan untuk menyimpan satu buah alamat kepala dari senarai.

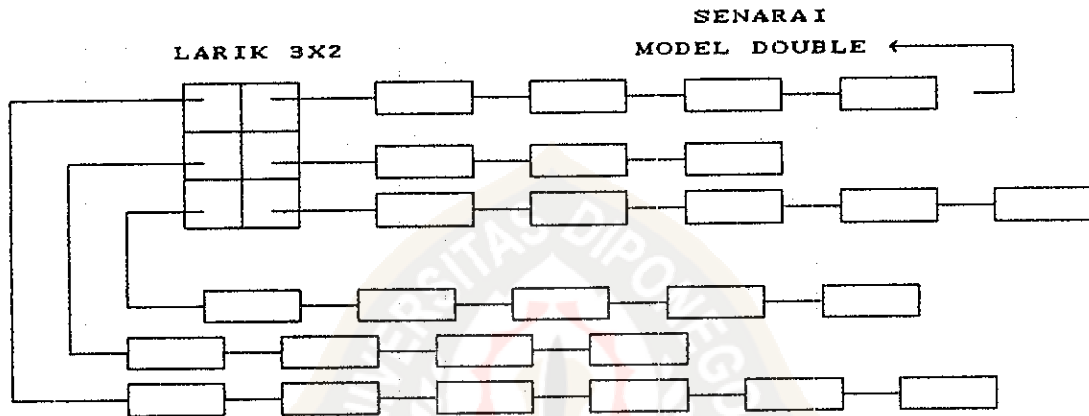
#### A. Kombinasi larik dimensi satu dengan senarai model double.

Secara implementasi gambar, kombinasi tipe data larik dimensi satu dengan senarai model double dapat dilihat sebagai berikut :



B. Kombinasi larik dimensi dua dengan senarai model double.

Secara implementasi gambar, kombinasi tipe data larik dimensi dua dengan senarai model double dapat dilihat sebagai berikut :



Dalam pembuatan rancangan penambahan fasilitas sistem operasi, penulis akan menggunakan kombinasi/gabungan tipe data larik dimensi satu dengan senarai model double.