

## BAB 4 SUSUNAN, OPERATOR DAN ARITMATIKAA

### 4.1 Pengenalan

Bab ini membicarakan notasi khusus untuk susunan yang merupakan satu bentuk struktur yang paling mudah dan berguna. Operasi-operasi dasar susunan turut dibicarakan dalam bab ini. Perhitungan aritmatika mudah dan notasi operator akan digunakan dalam penulisan program untuk menambahkan kemampuan dalam program Prolog.

### 4.2 Tujuan

Di akhir bab ini diharapkan anda dapat:

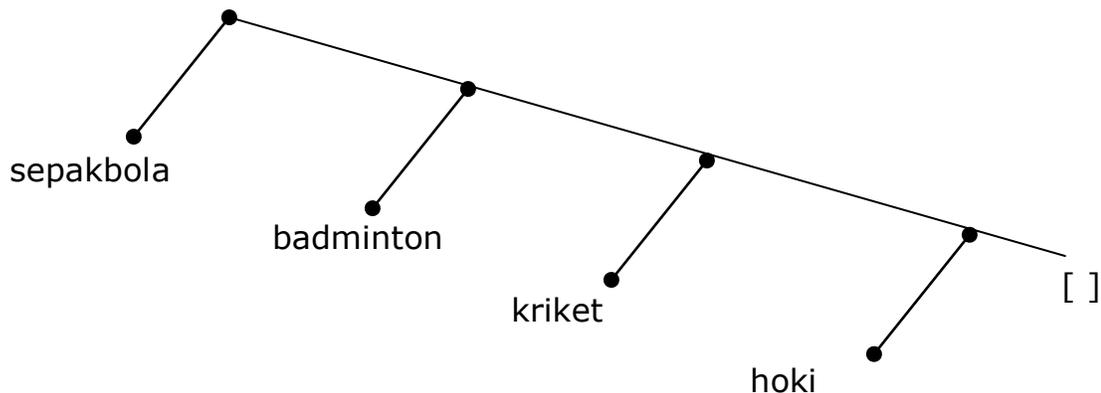
1. Mengetahui bagaimana data diwakilkan dalam bentuk susunan.
2. Mempelajari operasi-operasi dasar susunan.
3. Mempelajari operasi aritmatika dalam bahasa pemrograman Prolog.

### 4.3 Perwakilan Susunan

Susunan merupakan struktur data mudah yang digunakan secara meluas dalam pemrograman bukan numerik. Susunan adalah urutan bilangan item seperti sepakbola, badminton, kriket, hoki atau ahmad, ali, siti, fatima. Susunan dalam Prolog ditulis sebagai:

```
[sepakbola,badminton,kriket,hoki]  
[ahmad,ali,siti,fatima]
```

Namun begitu, kesemua objek berstruktur dalam Prolog diwakilkan dalam bentuk pohon dan susunan juga tidak terkecuali dalam perwakilan ini. Perwakilan pohon untuk susunan di atas ditunjukkan dalam Gambar 4.1.



Susunan dalam sistem Prolog bisa terdiri daripada susunan kosong atau susunan bukan kosong. Bagi kasus yang pertama, susunan bisa ditulis sebagai atom, []. Bagi kasus yang kedua, susunan bisa digambarkan dengan dua perkara yaitu :

- (1) item pertama yang dikenali sebagai kepala susunan
- (2) item selebihnya yang dikenali sebagai ekor susunan

Bagi contoh susunan di bawah:

[sepakbola,badminton,kriket,hoki]

kepala ialah sepakbola dan bagian ekor susunan ialah:

[badminton,kriket,hoki]

Susunan mempunyai cirri-ciri sebagai berikut :

- susunan bisa mengandungi sebarang bilangan elemen apakah konstan, variabel, nomor atau pun susunan yang lain.
- Satu susunan bisa kosong yang diwakilkan sebagai [].
- Elemen pertama dalam susunan dipanggil kepala susunan dan yang selebihnya dipanggil ekor susunan. Ekor bagi susunan juga dipanggil susunan. Contohnya :  
Susunan=[aa,bb,cc] Kepala=aa Ekor=[bb,cc]
- Satu susunan bisa dipisahkan kepada kepala dan ekor dengan menggunakan simbol |. Contoh :  
Susunan=[Kepala|Ekor] ?-[K|E]=[a,b,c] K=a E=[b,c]
- □ Terdapat berbagai cara untuk menulis susunan yang sama. Contohnya :  
[a,b,c] = [a|[b,c]] = [a,b|[c]] = [a,b,c|[]]
- Elemen susunan bisa diulang dalam satu susunan.

## 4.2 Operasi Susunan

Operasi susunan adalah sama dengan operasi dalam set. Tetapi, terdapat 2 perbedaan yaitu :

1. Urutan atau susunan elemen dalam satu set tidak penting, tetapi susunan elemen dalam susunan adalah penting.
2. Elemen yang sama dalam susunan bisa diulang tetapi ulangan elemen yang sama dalam set tidak dibenarkan.

Antara operasi susunan yang bisa dilakukan ialah:

- menyimak apakah ada suatu objek itu merupakan anggota kepada susunan yaitu sama dengan menyimak keanggotaan set.
- menggabungkan dua susunan dan menghasilkan susunan ketiga dan ini menyamai kesatuan bagi set.
- menambah objek baru kepada susunan atau menghapus objek dari susunan.

### 4.2.1 Keanggotaan

Andaikan kita mentakrif hubungan keanggotaan sebagai keanggotaan( $X,S$ ) dengan  $X$  ialah objek dan  $S$  ialah susunan. Gol keanggotaan( $X,S$ ) adalah benar jika  $X$  wujud dalam  $S$  yaitu objek  $X$  adalah anggota kepada satu susunan  $S$ . Sebagai contoh:

**keanggotaan ( $b,[a,b,c]$ ) adalah benar**  
**keanggotaan ( $d,[a,b,c]$ ) adalah palsu**  
**keanggotaan ( $[b,c],[a,[b,c]]$ ) adalah benar**  
**keanggotaan ( $b,[a,[b,c]]$ ) adalah palsu**

Program hubungan keanggotaan bisa didasarkan kepada pemerhatian berikut:  $X$  adalah anggota kepada  $S$  jika

- (1)  $X$  adalah kepala bagi  $S$ , atau
- (2)  $X$  adalah anggota kepada ekor bagi  $S$

Hubungan keanggotaan bisa ditulis dalam program Prolog dengan menggunakan 2 klausa. Klausa pertama itu fakta dan klausa kedua merupakan peraturan yaitu:

**keanggotaan( $X,[X|Ekor]$ ).**  
**keanggotaan( $X,[Kepala|Ekor]$ ) :-**  
**keanggotaan( $X,Ekor$ ).**

### 4.2.2 Penggabungan

Bagi penggabungan susunan, kita akan takrifkan hubungan:

**gabung( $S1,S2,S3$ )**

Hubungan gabung bisa menggabungkan 2 susunan  $S1$  dan  $S2$  untuk menghasilkan  $S3$  yang terdiri daripada kandungan  $S1$  dan  $S2$ . Sebagai contoh:

**gabung( $[a,b],[c,d],[a,b,c,d]$ )**

adalah benar, tetapi

**gabung( $[a,b],[c,d],[a,b,a,c,d]$ )**

adalah palsu.

Dalam pentakrifan hubungan gabung, kita perlu mempertimbangkan dua kasus, tergantung kepada argumen pertama yaitu  $S1$ .

- (1) Jika argumen pertama adalah susunan kosong, maka argumen kedua dan ketiga adalah sama (dikenali sebagai  $S$ ). Ini bisa ditunjukkan oleh fakta Prolog berikut:

**gabung( $[],S,S$ ).**

- (2) Jika argumen pertama bukan susunan kosong, maka ini bermakna ia mempunyai kepala dan ekor yang bisa digambarkan sebagai:

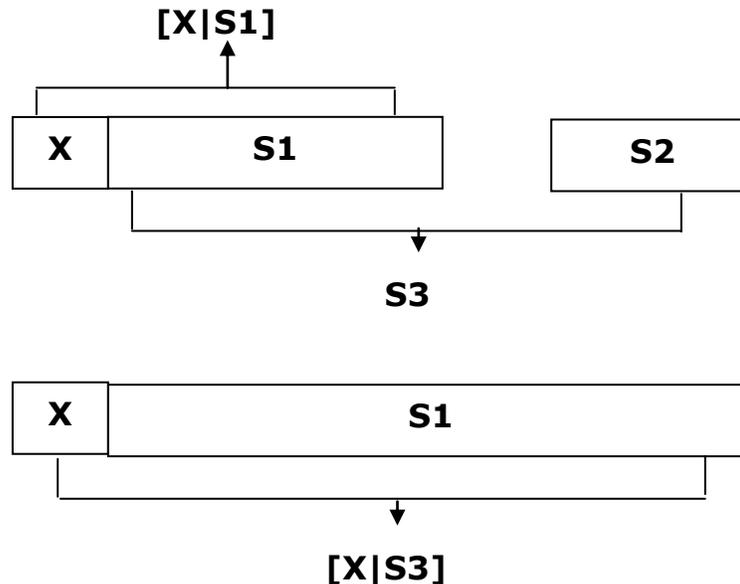
**$[X|S1]$**

Gambar 4.2 menunjukkan penggabungan bagi [X|S1] dengan satu susunan S2. Hasil dari penggabungan ini ialah satu susunan [X|S3] dengan S3 adalah penggabungan di antara S1 dengan S2. Dalam Prolog, penggabungan susunan bagi kasus kedua ini bisa ditulis sebagai:

**gabung([X|S1],S2,[X|S3]) :-  
gabung(S1,S2,S3).**

Bagi kedua-dua kasus di atas, operasi penggabungan susunan bisa ditulis dalam program Prolog melalui dua klausa yaitu:

**gabung([],S,S).**  
**gabung ([K|S1],S2,[K|S3]):-  
gabung (S1,S2,S3).**



Gambar 4.2: Penggabungan susunan

Program ini seterusnya bisa digunakan untuk menggabungkan sebarang susunan yang diberi. Sebagai contoh:

**?-gabung([a,b],[c,d],S).**  
**S = [a,b,c,d]**  
**?-gabung([a,[b,c]],[p,[],q,r],S).**  
**S = [a,[b,c],p,[],q,r]**

Hubungan gabung adalah satu hubungan fleksibel dan bisa digunakan dalam berbagai cara seperti:

(1) Untuk memecahkan satu susunan

**?- gabung(S1,S2,[a,b,c]).**  
**S1=[],S2=[a,b,c];**  
**S1=[a],S2=[b,c];**  
**S1=[a,b],S2=[c];**  
**S1=[a,b,c],S2=[];**

(2) Untuk melihat corak

**?-gabung(Sebelum,[d|Setelah],[a,b,c,d,e,f,g,h,i]).**  
**Sebelum=[a,b,c,d]**  
**Setelah=[f,g,h,i]**  
**S1 X S2**  
**S3 X**  
**S3**  
**[X|S1]**  
**[X|S3].**

#### 4.2.3 Penambahan Elemen

Untuk menambah elemen kepada susunan adalah mudah dengan memasukkan elemen baru itu di depan susunan supaya ia menjadi kepala baru susunan. Jika X ialah item baru dan S ialah susunan untuk dimasukkan item X, maka hasil yang diperolehi dari operasi penambahan ini ialah:

**[X|S]**

Secara ringkas satu elemen baru bisa ditambah di depan susunan dan bisa dicapai melalui klausa:

**tambah(X,S[X|S]).**

Contohnya:

**?- tambah(d,[a,b,c],S).**  
**S=[d,a,b,c]**

Untuk menambah elemen pada susunan belakang, kita bisa guna hubungan *append* yang telah ditakrifkan dalam sebagian implementasi sistem Prolog. Contohnya:

**?-append([1,2,3],[4,5,6],X)**  
**X=[1,2,3,4,5,6]**

#### 4.2.4 Penghapusan Elemen

Penghapusan elemen X dari susunan S bisa diprogramkan sebagai hubungan:

### **hapus(X,S,S1)**

dengan S1 adalah sama dengan susunan S yang telah dihapuskan elemen X yaitu S1 tidak mengandung elemen X. Hubungan hapus bisa ditakrifkan seperti hubungan keanggotaan. Terdapat dua kasus yang perlu dipertimbangkan, yaitu:

- (1) Jika X ialah kepala susunan, maka hasil penghapusan elemen ialah ekor susunan.
- (2) Jika X berada di bagian ekor susunan, maka ia akan dihapuskan di bagian tersebut.

Satu elemen bisa dihapuskan dari satu susunan dengan menggunakan 2 klausa yaitu :

**hapus(X,[X|Ekor], Ekor).**  
**hapus(X,[Y|Ekor],[Y|Ekor1]):-**  
**delete(X,Ekor,Ekor1).**

Contoh operasi hapus elemen ialah:

?- **hapus(a,[a,b],S).**

**S=[b,c]**

?- **hapus(a[b,a,c,a],S).**

**S=[b,a,c,a];**

**S=[a,b,c,a];**

**S=[a,b,a,c];**

**No**

Hubungan hapus adalah tidak deterministik. Jika X ada beberapa kali dalam susunan, maka hubungan hapus akan menghapuskan sebarang elemen (yang sama) di antaranya secara penjelajahan belakang. Proses penjelajahan belakang tidak dibicarakan dalam penyediaan modul ini.

### **4.3 Operasi Aritmatika**

Prolog menyediakan beberapa operator tertakrif untuk operasi dasar aritmatika seperti:

<b>+</b>	<b>tambah</b>
<b>-</b>	<b>kurang</b>
<b>*</b>	<b>kali</b>
<b>/</b>	<b>bagi</b>
<b>mod</b>	<b>modulas</b>

Contoh penggunaan operasi aritmatika dalam Prolog ialah:

?-**X=1+2.** (Satu percobaan yang salah dalam operasi aritmatika)

**X=1+2** (X disamakan kepada 1+2)

**?-100 \* 2 > 100**

**yes**

**?-X is 3/2, Y is 3 div 2**

Jawaban yang diberi oleh sistem Prolog ialah:

**X = 1.5**

**Y = 1**

Operator '/' menandakan pembagian interer atau nomor nyata bergantung kepada implementasi. Dalam modul ini kita mengandaikan operator '/'bererti pembahaian nomor nyata dan operator div bererti pembagian nomor. integer. Penggunaan is dan operator '>' akan mendesak sistem Prolog untuk membuat penilaian.

Operasi-operasi +, -, \*, / dan div akan dilaksanakan dari kiri ke kanan. Sebagai contoh:

**X is 5 - 2 - 1**

akan diterjemahkan sebagai:

**X is (5-2) - 1**

#### **4.4 Kesimpulan**

Bab ini memperkenalkan beberapa konsep seperti susunan, kepala susunan, ekor susunan, notasi susunan dan prosedur dalaman aritmatika. Bab ini juga telah membicarakan beberapa operasi dasar yang bisa dilaksanakan pada susunan. Kefahaman kita dalam bab ini sangat penting karena susunan banyak diguna dalam aplikasi sistem cerdas.

#### **4.5 Latihan**

1. Takrifkan predikat atau hubungan  
jumsusunan(Susunan,Jumlah)  
supaya Jumlah ialah jumlah bagi keseluruhan susunan nomor dalam Susunan.
2. Takrifkan hubungan  
maksimum(X,Y,Mak)  
supaya Mak merupakan nilai yang lebih besar di antara dua nomor X dan Y.
3. Takrifkan hubungan  
**songsangan(Susunan,SusunanSongsang)**  
yang akan membuat songsangan terhadap susunan yang diinput.  
Contohnya **songsangan([a,b,c,d],[d,c,b,a]).**