

## SENARAI

adalah runtunan elemen yang bertipe sama.

Setiap elemen di dalam runtunan mempunyai keterurutan tertentu (ada elemen predesesor atau elemen pendahulunya, dan ada elemen suksesor atau elemen berikutnya).

## Konsep Senarai dalam Matematika

- Senarai dinyatakan sebagai runtunan elemen yang dipisahkan dengan tanda koma, yaitu:

$a_1, a_2, \dots, a_n$ ,  $n \geq 0$ , dan setiap  $a_i$  dari tipe tertentu. Jumlah  $n$  elemen disebut panjang senarai. Andaikan  $n \geq 1$ , menyatakan bahwa  $a_1$  adalah elemen pertama senarai dan  $a_n$  adalah elemen terakhir senarai. Jika  $n = 0$ , adalah senarai kosong, yaitu senarai yang tidak mempunyai elemen satu buah pun.

## Elemen senarai tersusun 2 bagian:

- Bagian informasi (info)
- Bagian alamat (next) yang berisi alamat elemen berikutnya.

```
type ElmSenarai : record <info : TypeInfo,  
  Next : Alamat>
```

TypeInfo = tipe terdefinisi berupa tipe dasar atau tipe terstruktur, sedangkan Alamat adalah tipe khusus.

## Notasi Senarai

Misalkan  $S$  adalah senarai dan  $P$  adalah peubah bertipe Alamat, maka notasi yang dapat digunakan sebagai berikut:

1.  $Awal(S)$ , menyatakan alamat elemen pertama senarai  $S$ .
2.  $Info(P)$ , menyatakan field Info dari elemen yang alamatnya  $P$ .
3.  $Next(P)$ , menyatakan field Next dari elemen yang alamatnya  $P$ .

## Mendeklarasikan Tipe Senarai

Kamus

type TipeInfo : integer

type ElmSenarai : record <Info : TipeInfo, Next :  
Alamat>

type Senarai : ElmSenarai

Senarai Kosong

adalah senarai yang tidak mempunyai elemen satu buah pun. Dalam notasi algoritmik jika S adalah senarai kosong, maka Awal(S) = Nil.

Kamis, 25 Mei 2006

Algoritma dan Pemrograman II

5

## Fungsi Pemeriksaan Senarai Kosong

function Kosong(input S : Senarai) →  
boolean

{Mengembalikan true jika S kosong, false  
jika tidak kosong}

Kamus Lokal

P : Alamat

Algoritma

return Awal(S) = Nil

Kamis, 25 Mei 2006

Algoritma dan Pemrograman II

6

## Penelusuran Senarai

Skema Pemrosesan Beruntun

procedure PenelusuranSenarai(input S : Senarai)

{Mengunjungi elemen senarai berkait dan memproses elemen yang dikunjungi. Setiap elemen senarai diproses sama. Senarai kosong ditangani secara khusus.

K. Awal : Senarai S mungkin kosong

K. Akhir: Semua elemen S dikunjungi dan diproses}

Kamus Lokal

P : Alamat

Algoritma

if Awal(S) = Nil then  
output ('Senarai kosong')

else

Inisialisasi

P ← Awal(S)

repeat

Proses(P)

P ← Next(P)

until P = Nil

Terminasi

endif

Kamis, 25 Mei 2006

Algoritma dan Pemrograman II

7

## Skema Pemrosesan Beruntun 2

procedure PenelusuranSenarai2(input S : Senarai)

{Mengunjungi elemen senarai berkait dan memproses elemen yang dikunjungi.

Setiap elemen senarai diproses sama. Tidak ada penanganan khusus untuk senarai kosong.

K. Awal : Senarai S mungkin kosong

K. Akhir: Semua elemen S dikunjungi dan diproses}

Kamus Lokal

P : Alamat

Algoritma

Inisialisasi

P ← Awal(S)

while P ≠ Nil do

Proses(P)

P ← Next(P)

endwhile

Terminasi

Kamis, 25 Mei 2006

Algoritma dan Pemrograman II

8

## Pencetakan dengan Skema Pemrosesan Beruntun

```
procedure CetakSenarai(input S : Senarai)
{Mencetak elemen senarai ke piranti keluaran. Senarai kosong ditangani secara khusus.
K.Awal : Senarai S mungkin kosong
K. Akhir: jika S kosong pesan "senarai kosong" tercetak, kalau tidak semua elemen ditulis ke piranti keluaran.}
Kamus Lokal
P : Alamat
```

```
Algoritma
if Awal(S) = Nil then
  output ('Senarai kosong')
else
  P ← Awal(S)
  repeat
    output(Info(P))
    P ← Next(P)
  until P = Nil
endif
```

Kamis, 25 Mei 2006

Algoritma dan Pemrograman II

9

## Pencetakan dengan skema Pemrosesan Beruntun 2

```
procedure CetakSenarai2(input S : Senarai)
{Mencetak elemen senarai ke piranti keluaran. Senarai kosong tidak ditangani secara khusus.
K.Awal : Senarai S mungkin kosong
K. Akhir: Semua elemen ditulis ke piranti keluaran.}
```

```
Kamus Lokal
P : Alamat
```

```
Algoritma
P ← Awal(S)
while P ≠ Nil do
  Output(Info P)
  P ← Next(P)
endwhile
```

Kamis, 25 Mei 2006

Algoritma dan Pemrograman II

10

## Pencarian di dalam Senarai

Pencarian pada senarai didasarkan pada dua parameter:

1. Pencarian berdasarkan nilai tertentu;
2. Pencarian berdasarkan alamat.

Kamis, 25 Mei 2006

Algoritma dan Pemrograman II

11

## Pencarian berdasarkan nilai

```
Procedure CariElemenSenarai(input S : Senarai, input X : TipeInfo; output ketemu : boolean,output P: Alamat)
```

```
Algoritma
P ← Awal(S)
ketemu ← false
While (P ≠ Nil) and (not ketemu) do
  if Info(P) = X then
    ketemu ← true
  else
    P ← Next(P)
  endif
endwhile
```

Kamis, 25 Mei 2006

Algoritma dan Pemrograman II

12

## Pencarian Berdasarkan Alamat

Procedure CariElemenSenarai2(input S : Senarai, input P : Alamat;  
output ketemu : boolean)

Kamus lokal

PS : Alamat

Algoritma

PS  $\leftarrow$  Awal(S)

ketemu  $\leftarrow$  false

While (PS  $\neq$  Nil) and (not ketemu) do

  if PS = X then

    ketemu  $\leftarrow$  true

  else

    PS  $\leftarrow$  Next(PS)

  endif

endwhile

## Penyisipan Elemen pada Senarai

1. Penyisipan elemen di awal senarai
2. Penyisipan elemen di tengah senarai
3. Penyisipan elemen di akhir senarai

## Penyisipan Elemen di Awal Senarai

Procedure SisipDiawal(input/output S :  
Senarai, input P : Alamat)

Algoritma

Next(P)  $\leftarrow$  Awal(S)

Awal(S)  $\leftarrow$  P

## Penyisipan Elemen di Tengah Senarai

Procedure SisipDiTengah(input P : Alamat,  
input PredP : Alamat)

Algoritma

Next(P)  $\leftarrow$  Next(PredP)

Next(PredP)  $\leftarrow$  P

## Penyisipan Elemen di Akhir Senarai

```
Procedure SisipDiAkhir(input/output S : Senarai, input P : Alamat)
Kamus lokal
  Akhir : Alamat
Algoritma
  if Awal(S) = Nil then
    Awal(S) ← P
  else
    Akhir ← Awal(S)
    while Next(Akhir) ≠ Nil do
      Akhir ← Next(Akhir)
    endwhile

    Next(Akhir) ← P
  endif
```

Kamis, 25 Mei 2006

Algoritma dan Pemrograman II

17

## Menghapus Elemen pada Senarai

1. Penghapusan Elemen di Awal Senarai;
2. Penghapusan Elemen di Tengah Senarai;
3. Penghapusan Elemen di Akhir Senarai.

Kamis, 25 Mei 2006

Algoritma dan Pemrograman II

18

## Penghapusan Elemen di Awal Senarai

```
procedure HapusDiawal(input/output S : Senarai, output P :
  Alamat)
{ K.Awal : S tidak kosong, minimal satu buah elemen.
  K.Akhir : P berisi alamat elemen pertama S.}
```

Algoritma

```
P ← Awal (S)
Awal(S) ← Next(Awal (S))
```

Kamis, 25 Mei 2006

Algoritma dan Pemrograman II

19

## Penghapusan Elemen di Tengah Senarai

```
procedure HapusdiTengah(input PredP : Alamat,
  output P : Alamat)
{ K.Awal : PredP adalah alamat elemen senarai S,
  Next(PredP) ≠ nil.
  K.Akhir : P berisi alamat elemen yang dihapus
  dari S.}
```

Algoritma

```
P ← Next (PredP)
Next(PredP) ← Next(Next (PredP))
```

Kamis, 25 Mei 2006

Algoritma dan Pemrograman II

20

## Penghapusan Elemen di Akhir Senarai

```
procedure HapusdiAkhir(input/output S : Senarai, output P : Alamat)
{ K.Awal : S tidak kosong, minimal satu buah elemen.
  K.Akhir : P berisi elemen terakhir senarai S sebelum penghapusan.}
Kamus Lokal
PredAkhir, Akhir : Alamat
```

```
Algoritma
{ cari alamat elemen terakhir, akhir}
Akhir ← Awal(S)
PredAkhir ← Nil
while Next (Akhir) ≠ Nil do
  Akhir ← Next(Akhir)
endwhile
P ← Akhir (ambil alamat elemen yang dihapus)
If PredAkhir = Nil then
  Awal(S) ← Nil
else
  Next(PredAkhir) ← Nil
endif
```

Kamis, 25 Mei 2006

Algoritma dan Pemrograman II

21

## Operasi Terhadap Dua Buah Senarai

1. Menyambung Dua Buah Senarai,
2. Membandingkan Dua Buah Senarai.

Kamis, 25 Mei 2006

Algoritma dan Pemrograman II

22

## Menyambung Dua Buah Senarai

```
procedure Sambung(input S1 : Senarai, input S2 : Senarai, output S3 : Senarai)
{K. Awal : S1 atau S2 mungkin kosong.
  K.Akhir : S3 adalah hasil senarai S1 dengan senarai S2}
```

```
Kamus Lokal
Akhir1 : Alamat
```

```
Algoritma
Awal(S3) ← Nil
if Awal(S1) = Nil then
  Awal(S3) ← Awal(S2)
else
  Awal(S3) ← Awal(S1)
  Akhir1 ← Awal(S)
  while Next(Akhir1) ≠ Nil do
    Akhir1 ← Next(Akhir1)
  endwhile
  Next(Akhir1) ← Awal(S2)
endif
```

Kamis, 25 Mei 2006

Algoritma dan Pemrograman II

23

## Membandingkan Dua Buah Senarai

```
function sama(input S1 : Senarai, input S2 : Senarai) → Boolean
```

```
Kamus Lokal
P1, P2 : Alamat
Same : Boolean
```

```
Algoritma
Same ← true
P1 ← Awal(S1)
P2 ← Awal(S2)
while (P1 ≠ Nil) and (P2 ≠ Nil) and (Same) do
  if Info(P1) = Info(P2) then
    P1 ← Next(P1)
    P2 ← Next(P2)
  else
    Same ← false
  endif
endwhile
return Same
```

Kamis, 25 Mei 2006

Algoritma dan Pemrograman II

24