



Pertemuan – 6

List Berkait

Dipersiapkan oleh : Boldson Herdianto S., S.Kom., MMSI.

Kita lanjutkan
untuk yang satu ini



SENARAI / LINK LIST

Struktur Data

★ Definisi :

- struktur data yang terdiri dari rangkaian elemen sejenis yang saling berhubungan. Setiap elemen memiliki pendahulu dan penerusnya (kecuali elemen terakhir)

★ Contoh :

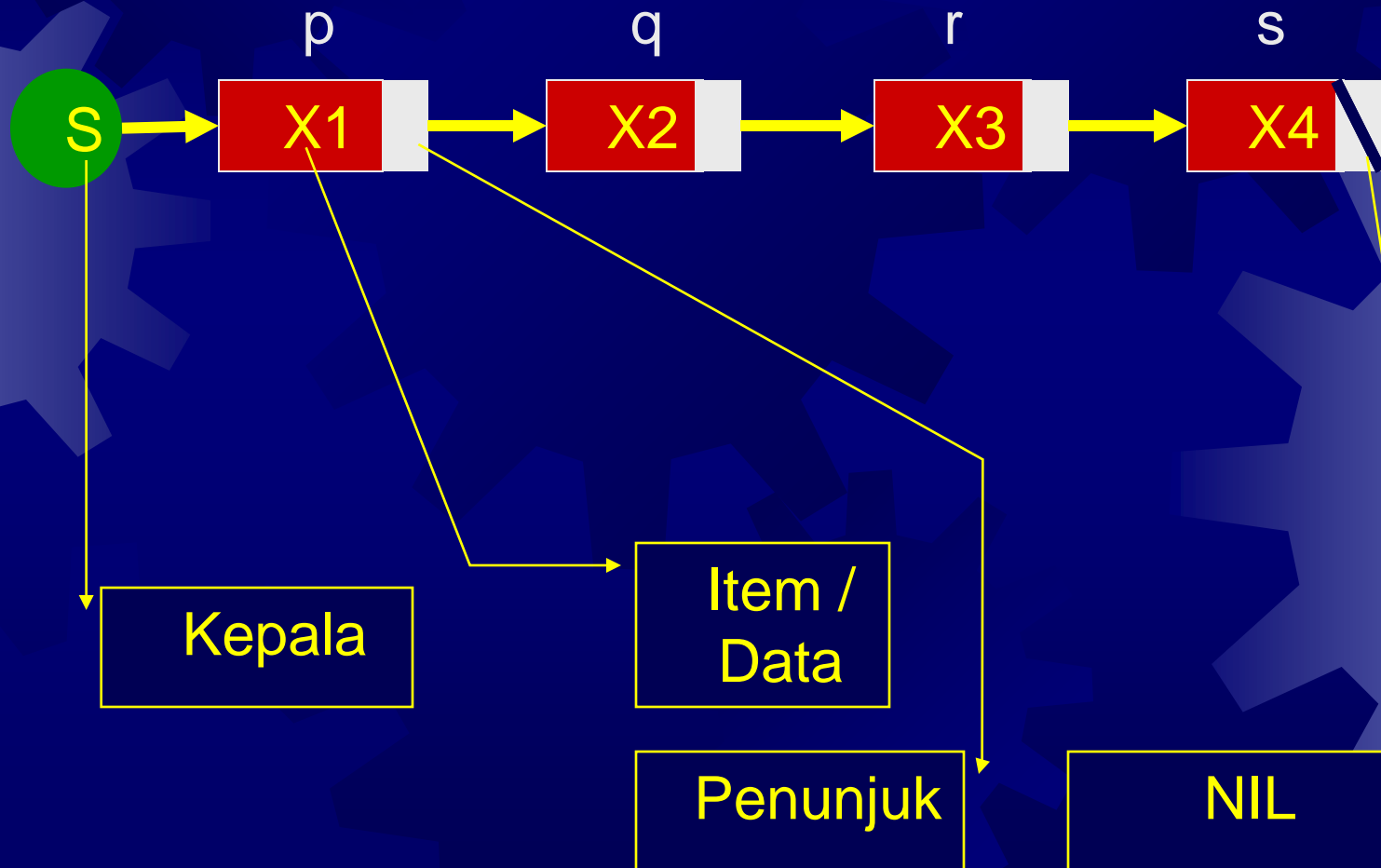
- Struktur ini mirip kereta api, dimana kepalanya seperti lokomotif, elemennya seperti gerbong kereta dan datanya seperti penumpang/barang



Review

MODEL SENARAI

Struktur Data

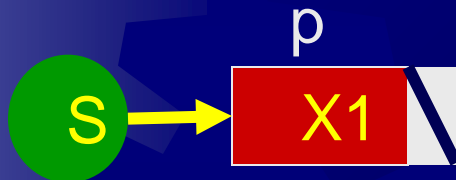


KONDISI SENARAI

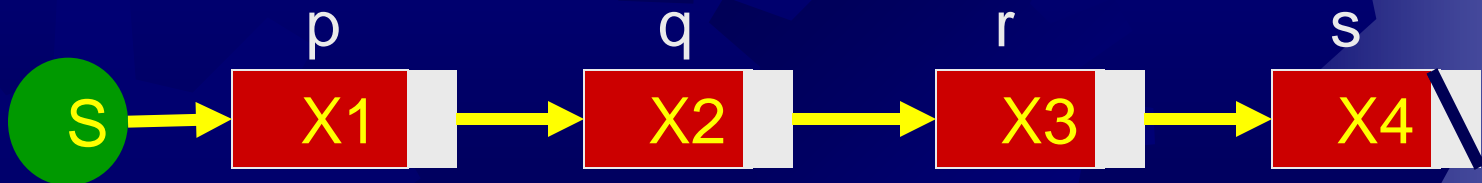
Struktur Data

S

Senarai / List KOSONG



Senarai / List dengan 1 elemen



Senarai / List dengan 4 elemen

Review

KAMUS DATA SENARAI

Struktur Data

Elemen Senarai



Kamus Data :

Info : array [1..4] of String

Next : array [1..4] of Integer

Awal : Integer

| | Info | Next |
|---|------|------|
| 4 | D | Nil |
| 3 | B | 2 |
| 2 | C | 4 |
| 1 | A | 3 |

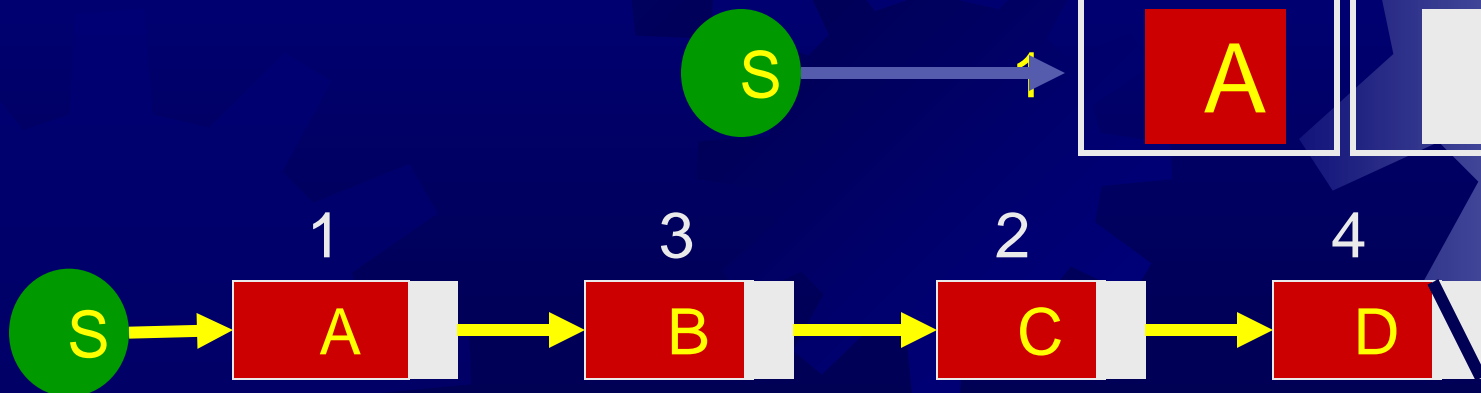


Review

REPRESENTASI SENARAI DENGAN ARRAY

Link List / List Berkait

| | Info | Next |
|---|------|------|
| 4 | D | Nil |
| 3 | B | 2 |
| 2 | C | 4 |
| 1 | A | 3 |



Review

NOTASI Info dan Next

Link List / List Berkait

Dua Notasi :

- ✓ INFO (x) : Data yang ada di alamat X
- ✓ NEXT (x) : Alamat elemen berikut setelah X

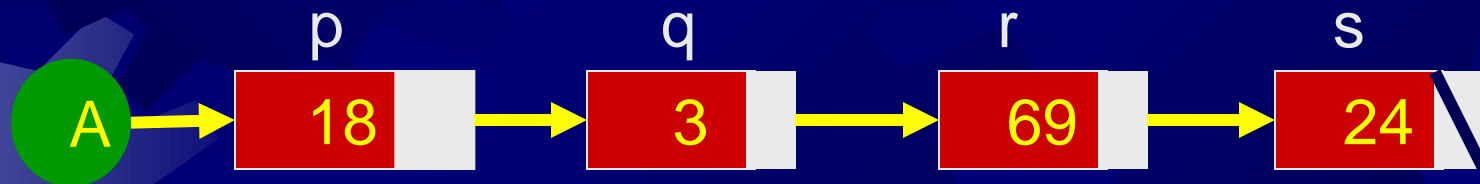


Contoh : Next (1) = 3

Info (1) = A

Latihan Info dan Next

Link List / List Berkait



Contoh : Next (p) = q Info (r) = 69

✓ A =

✓ Next(A)

✓ Next(Next(p))

✓ Next(Next(Next(p)))

✓ Next (Next (r))

✓ Info(A) =

✓ Info(Next(A))

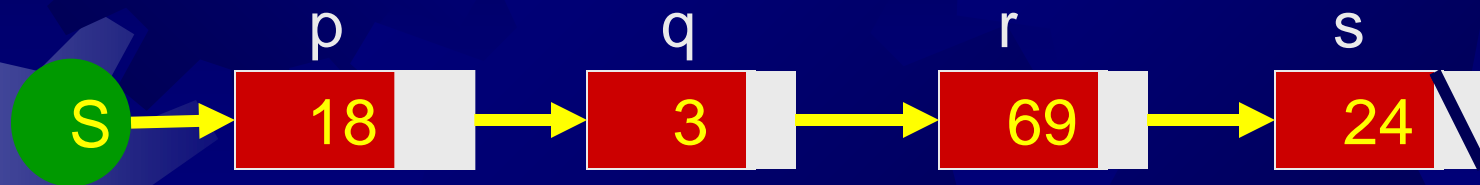
✓ Info(Next(p))

✓ Info(Next(Next(p)))

✓ Info (r)

Operasi Dasar Senarai

Link List / List Berkait



1. **Cipta_Senarai** : membuat senarai kosong
2. **Sisip_Awal** : menyisipkan elemen di awal
3. **Sisip_Akhir** : menyisipkan elemen di akhir
4. **Hapus_Awal** : menghapus elemen di awal
5. **Hapus_Akhir** : menghapus elemen di akhir

Cipta_Senarai

Membuat senarai kosong

S

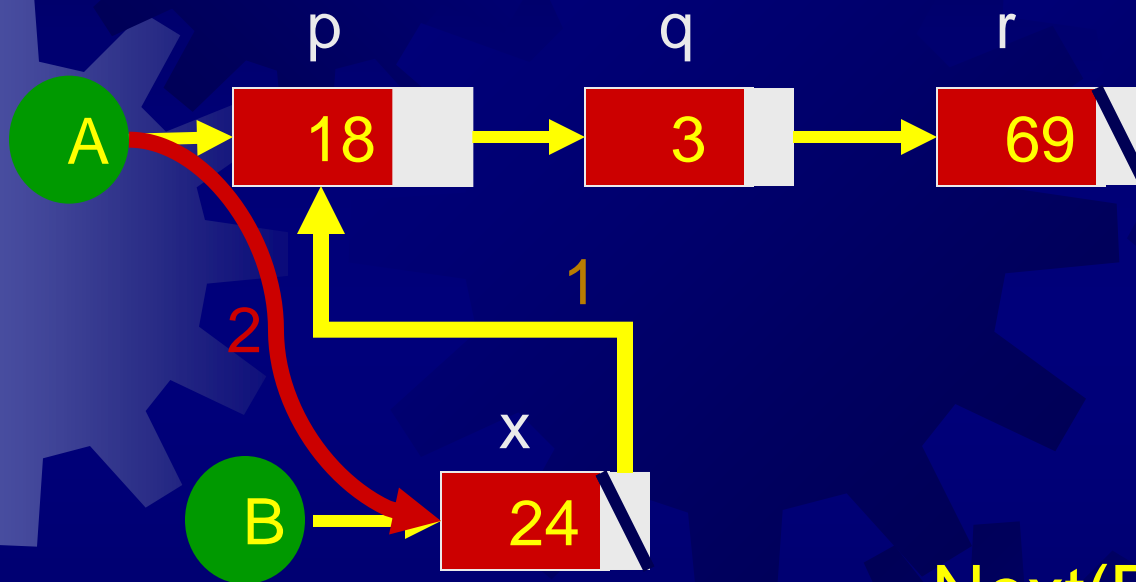
S = Nil

S

| | Info | Next |
|---|------|------|
| 4 | | |
| 3 | | |
| 2 | | |
| 1 | | |

Sisip_Awal

Menyisipkan elemen di awal



$\text{Next}(B) = \text{Awal}(A)$

$\text{Awal}(A) = \text{Awal}(B)$

Sisip_Akhir

Menyisipkan elemen di akhir

Akhir

Awa

1

18

2

3

3

69

B

P

24

Akhir = Awal

While (next(Akhir) <> Nil do

Akhir = Next(Akhir)

Enddo

Next(Akhir) = P

Sisip_Akhir

Menyisipkan elemen di akhir

Akhir

1

2

3

Awa

18

3

69

B

P

24

Akhir = Awal

While (next(Akhir) <> Nil do

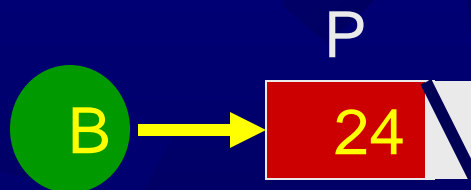
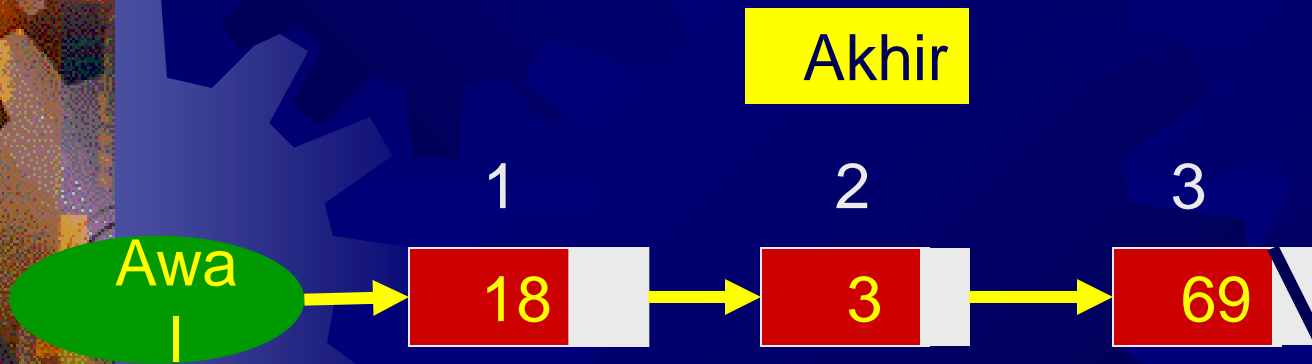
➔ Akhir = Next(Akhir)

Enddo

Next(Akhir) = P

Sisip_Akhir

Menyisipkan elemen di akhir



Akhir = Awal

While (next(Akhir) <> Nil do

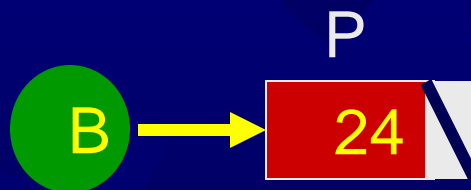
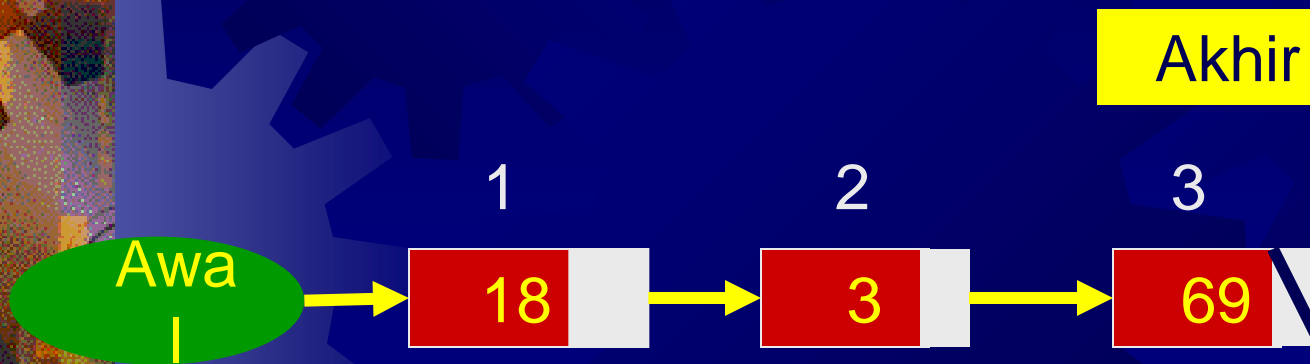
→ Akhir = Next(Akhir)

Enddo

Next(Akhir) = P

Sisip_Akhir

Menyisipkan elemen di akhir



Akhir = Awal

While (next(Akhir) <> Nil do

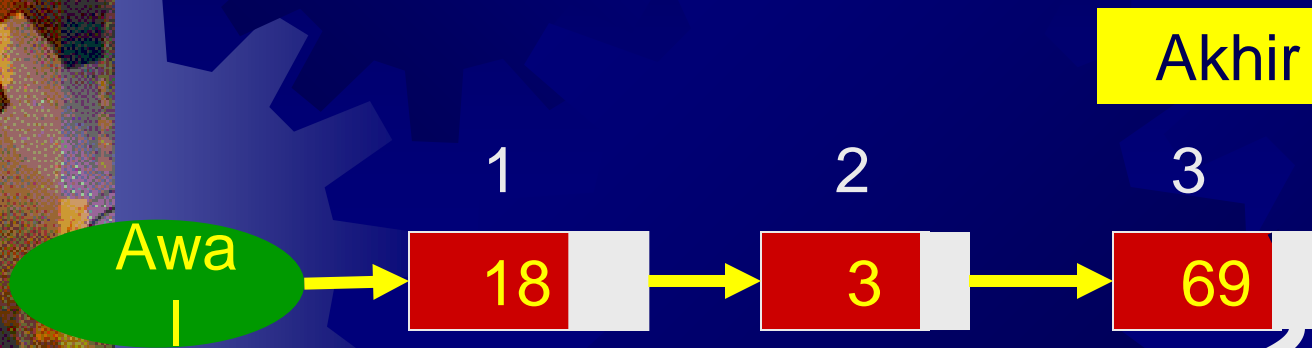
→ Akhir = Next(Akhir)

Enddo

Next(Akhir) = P

Sisip_Akhir

Menyisipkan elemen di akhir



Akhir = Awal

While (next(Akhir) \neq Nil do

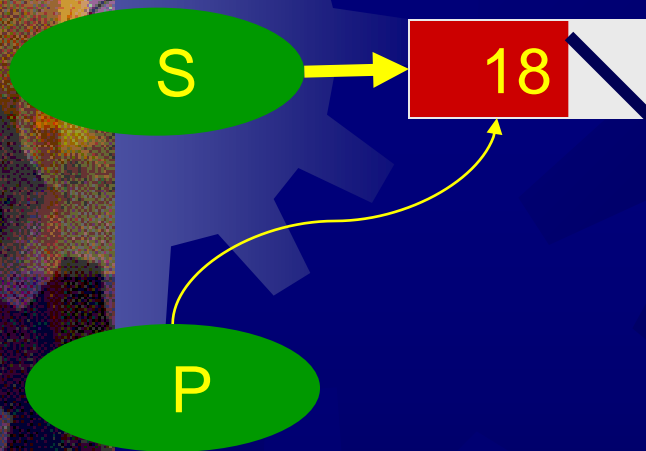
Akhir = Next(Akhir)

Enddo

➔ Next(Akhir) = P

Hapus Awal (kondisi 1 elemen)

Menghapus elemen di awal



P : untuk menyimpan
alamat elemen yang akan
dihapus

IF Next (S)) = Nil then

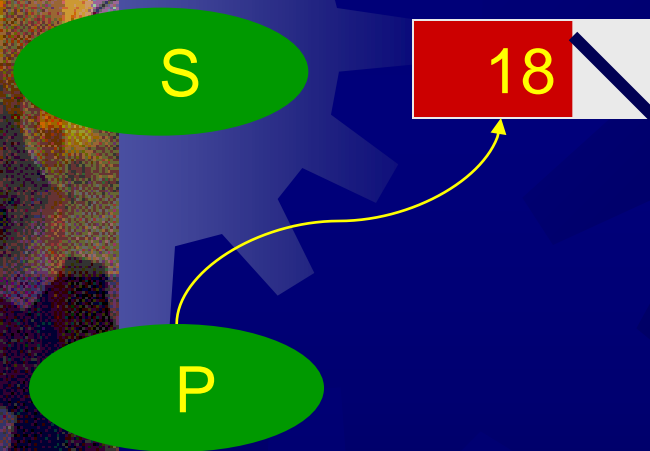
→ P = S

S = Nil

Enddo

Hapus Awal (kondisi 1 elemen)

Menghapus elemen di awal



P : untuk menyimpan
alamat elemen yang akan
dihapus

IF Next (S) = Nil then

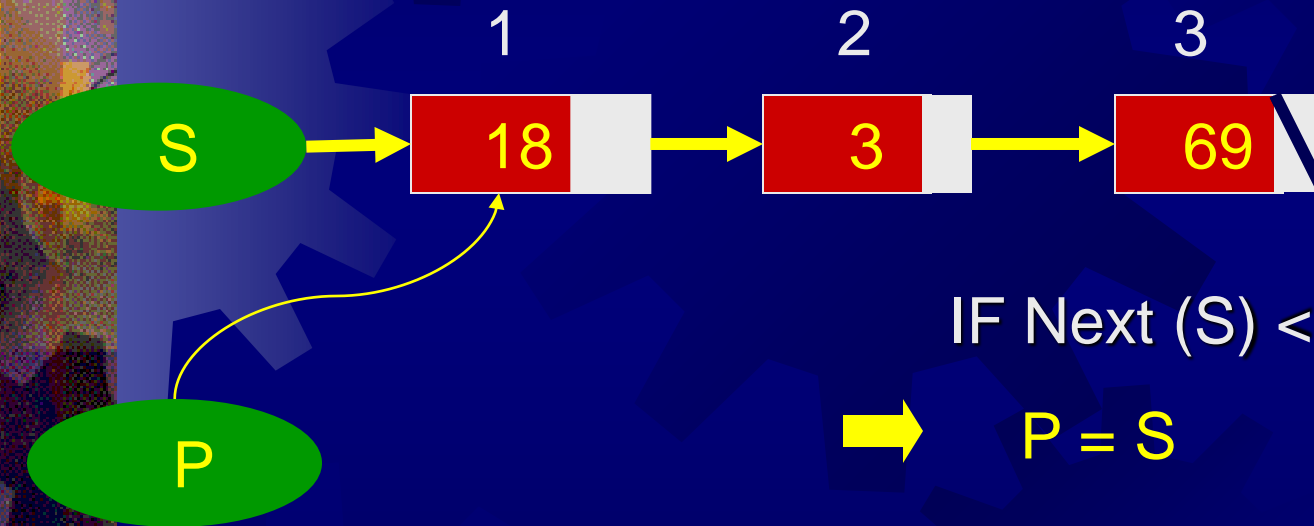
P = S

→ S = Nil

Enddo

Hapus Awal (kondisi N elemen)

Menghapus elemen di awal



P : untuk menyimpan alamat elemen yang akan dihapus

IF Next (S) \neq Nil then

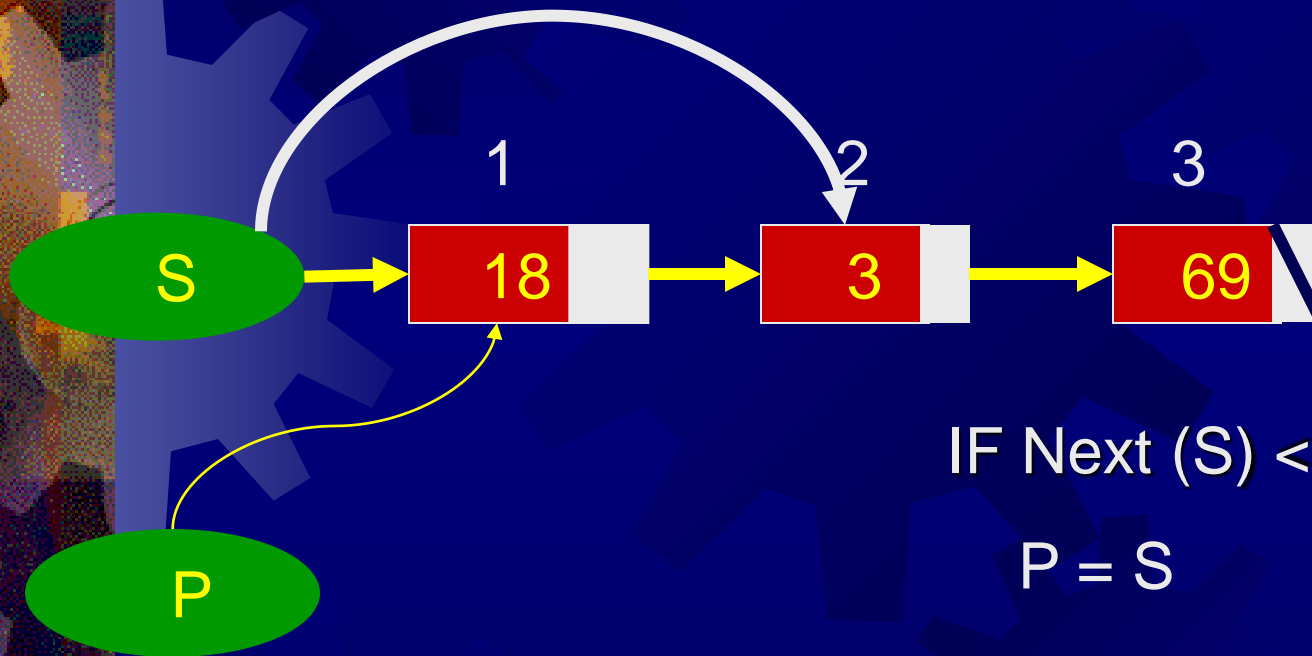
→ P = S

S = Next(S)

Enddo

Hapus Awal (kondisi N elemen)

Menghapus elemen di awal



P : untuk menyimpan
alamat elemen yang akan
dihapus

IF Next (S) \neq Nil then

P = S

➔ S = Next(S)

Enddo