PRAKTIKUM PEMROGRAMAN I

Operasi Remove Head & Tail

Ade Sukendar

Teknik Informatika Universitas Pasundan

2024

Ready !!! Go !!!

Hapus (*Remove)* Elemen List

- Operasi hapus (*remove*) list adalah proses untuk menghilangkan elemen di dalam list
- Operasi hapus elemen list dapat dilakukan di
  - Awal (*head)*
  - Tengah (*middle)*
  - Akhir (*Tail)*
- Dampak dari operasi hapus list adalah elemen list akan berkurang sehingga alokasi memori pun akan mengecil sesuai dengan jumlah elemen yang ada di dalam list

Operasi List
Remove Elemen di Head

Ilustrasi Operasi Hapus/*Remove*

- Misalkan sebuah list melakukan operasi menambah elemen ke list seperti ini
  - addTail(80)
  - addTail(60)
  - addTail(72)

- Dari operasi tersebut bagaimana penggambaran elemen di dalam List!!

Ilustrasi Operasi Hapus/*Remove* ...

- Operasi hapus elemen list di awal / *head* direpresentasikan dengan fungsi ***removeHead***
- Misalkan list sebelumnya dilakukan operasi removeHead 2x, apa yang terjadi dengan elemen List?
- Jika dilakukan kembali operasi removeHead sekali, apa yang terjadi dengan List?
- Jika dilakukan kembali operasi removeHead sekali lagi, apa yang terjadi dengan List?

Kondisi Operasi *Remove* Head

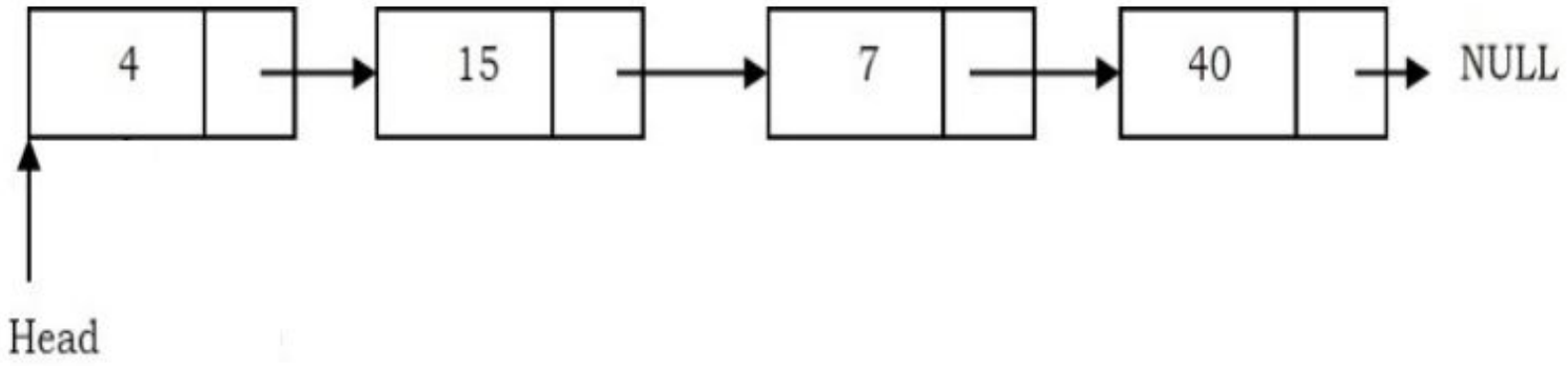
- Menghapus elemen di Head mempunyai kondisi yaitu elemen List tidak kosong (*not empty*)

Proses: Remove Head Jika List Not Empty

1. Pastikan bahwa HEAD tidak NULL
2. Buat node temp (*temporary*) dan isi dengan node HEAD
3. HEAD diisi dengan next node HEAD
4. Hapus node temp

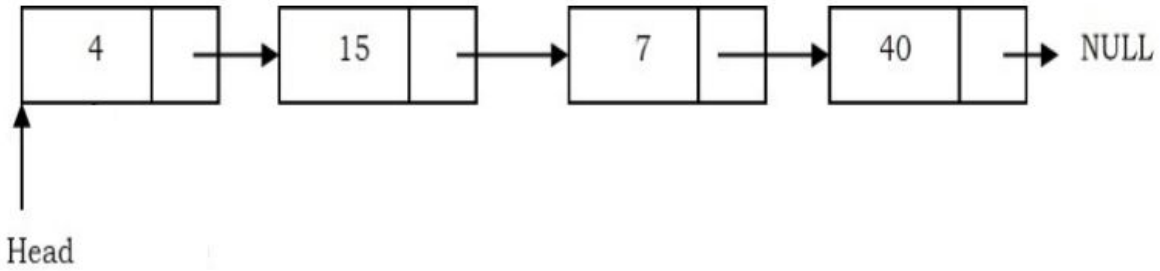
Proses: Remove Head Jika List Not Empty …

- Misalkan ada sebuah list yang sudah berisi elemen yaitu (4, 15, 7, 40)
- Awal list di sebut *head, Head* menunjuk ke elemen bernilai 4
- Penanda akhir list yaitu NULL

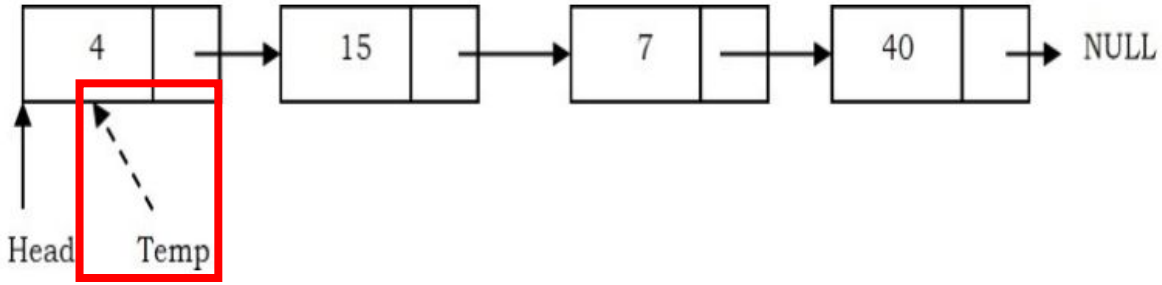


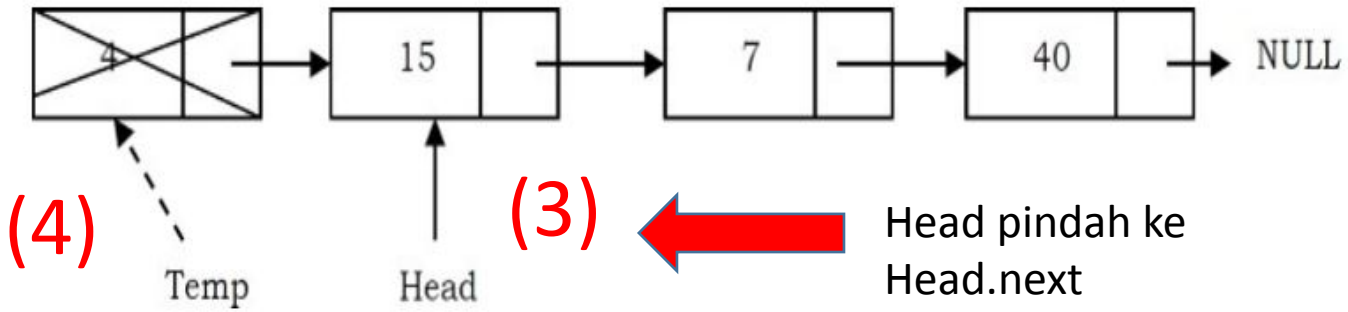
Proses: Remove Head Jika List Not Empty …

Head != NULL (1)



(2)





Head pindah ke Head.next

Algoritma Remove Head List

```
procedure removeHead()
deklarasi
  temp: Node
deskripsi
  IF (HEAD <> null) THEN
    temp ← HEAD
    HEAD ← HEAD.next
    dispose(Temp)
   ENDIF
```

Operasi List
Remove Elemen di Tail

Kondisi Operasi Remove Tail

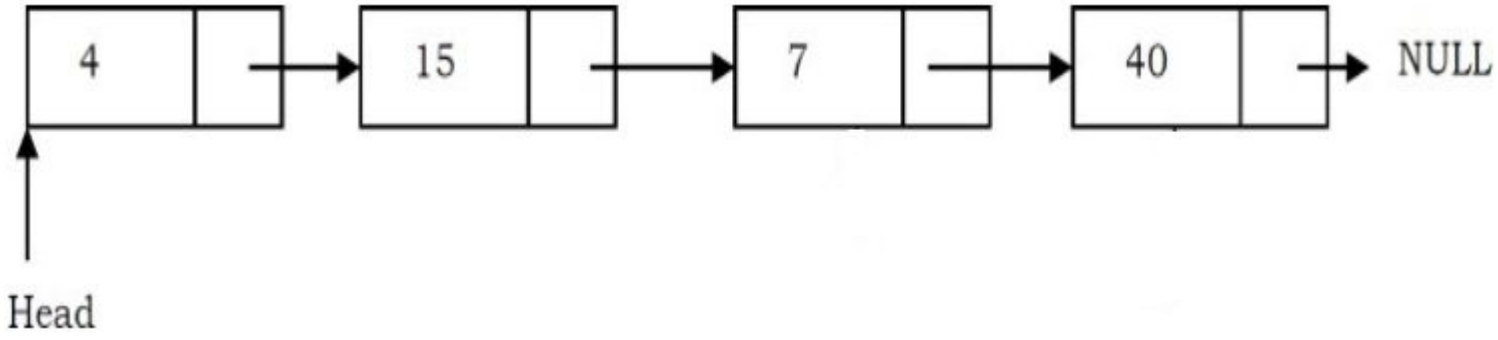
- Menghapus elemen di tail mempunyai kondisi yaitu
  Jika elemen List tidak kosong (*not empty*)

Proses: Remove Tail Jika List Not Empty

1. Pastikan bahwa HEAD tidak berisi NULL
2. Pengecekan setiap elemen node, untuk menemukan node sebelum node terakhir (**preNode**) dan node terakhir
3. Next preNode diisi dengan nilai NULL
4. Node terakhir di hapus/dispose

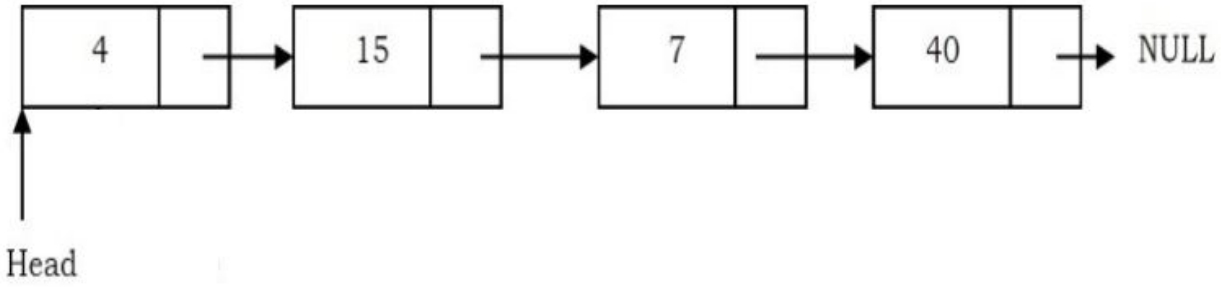
Proses: Remove Tail Jika List Not Empty …

- Misalkan ada sebuah list yang sudah berisi elemen yaitu (4, 15, 7, 40)
- Awal list di sebut *head, Head* menunjuk ke elemen bernilai 4
- Penanda akhir list yaitu NULL

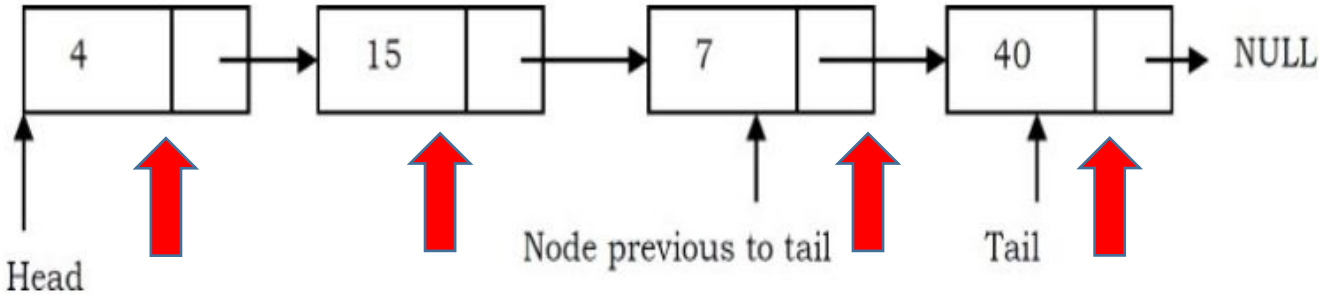


Proses: Remove Tail Jika List Not Empty …

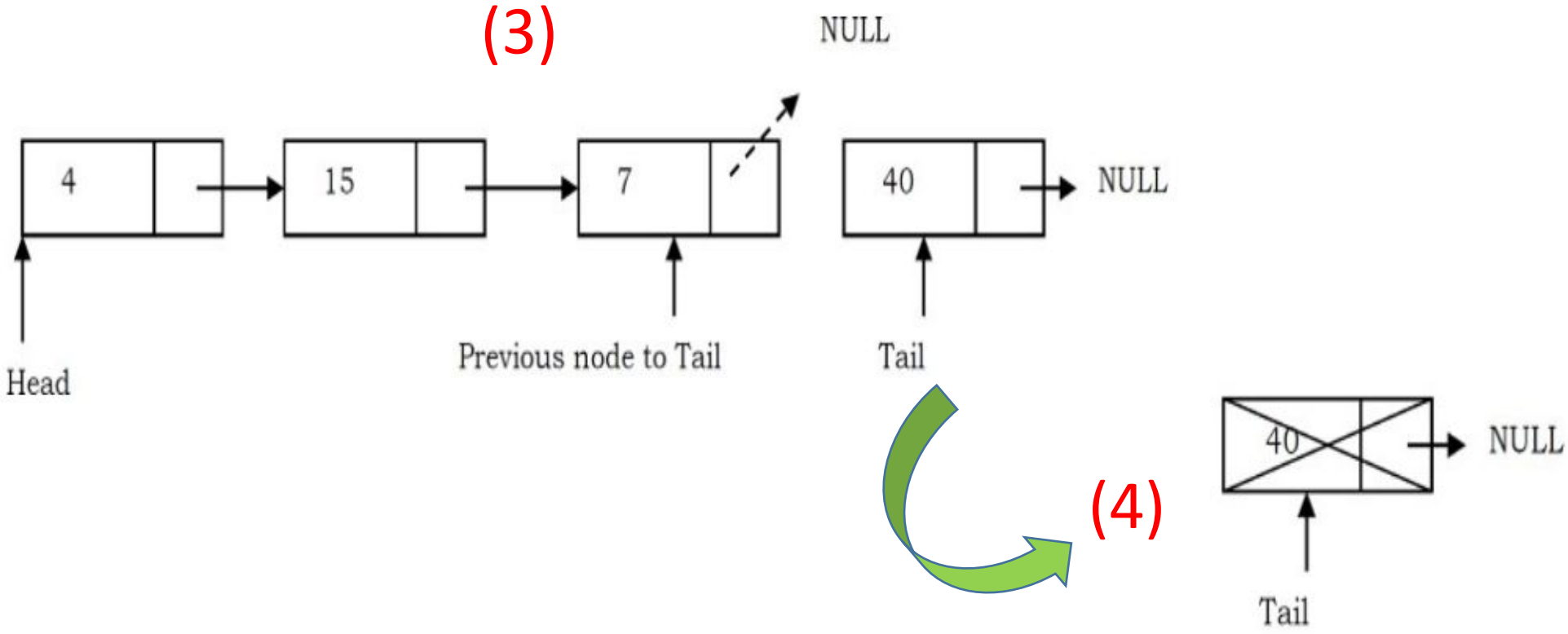
Head != NULL (1)



(2)



Proses: Remove Tail Jika List Not Empty ...



Algoritma Remove Tail List

```
procedure removeTail()
deklarasi
  preNode, lastNode: Node
deskripsi
  IF (HEAD <> null) THEN
    IF (HEAD.next = null) THEN {Jika satu elemen list}
      HEAD ← null
    ELSE
      lastNode ← HEAD
      WHILE (lastNode.next <> null)
        preNode ← lastNode
        lastNode ← lastNode.next
      ENDWHILE
      preNode.next ← NULL
      dispose(lastNode)
    ENDIF
  ENDIF
```

Terima Kasih

Referensi

- Foundation of Computer Science – C Edition, Alfred V. Aho dan Jeffrey D. Ullman, 1994.
- Data Structures and Algorithms in Java, 2nd Edition by Robert Lafore
- Data Structures and Algorithms Made Easy: Data Structures and Algorithmic Puzzles, Fifth Edition - Narasimha Karumanchi
- The Algorithm Design Manual - Steven S Skiena
- Algorithms (4th Edition) - Robert sedgewick