



# PRAKTIKUM PEMROGRAMAN I

## Operasi Remove List (Mid)

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 Tengah

# Kondisi Operasi Remove di Tengah

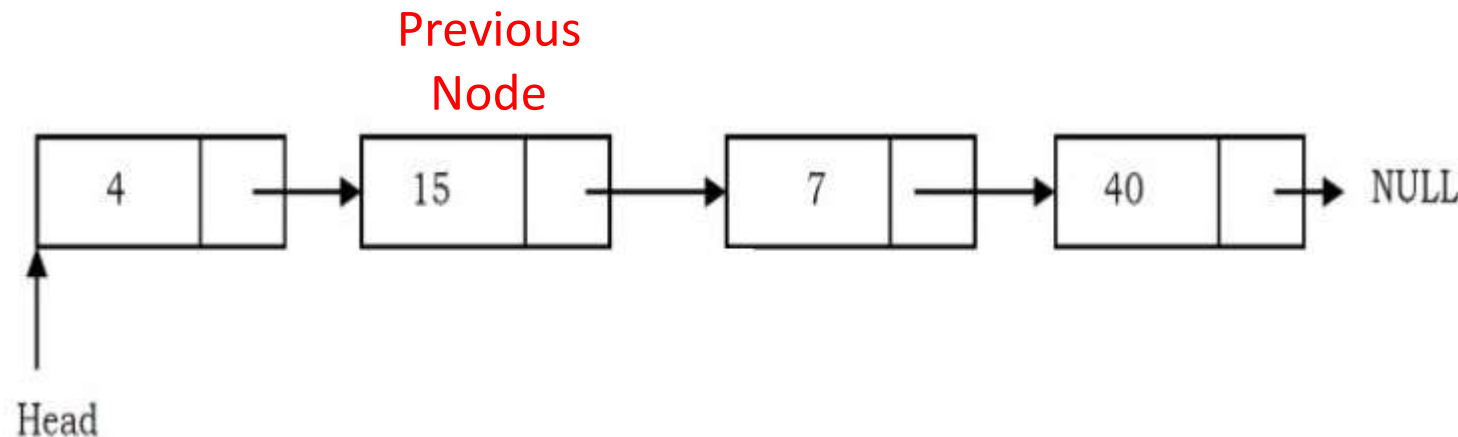
- Menghapus elemen di tengah mempunyai kondisi yaitu elemen List tidak kosong (*not empty*)

# Proses: Remove di Tengah Jika List Not Empty

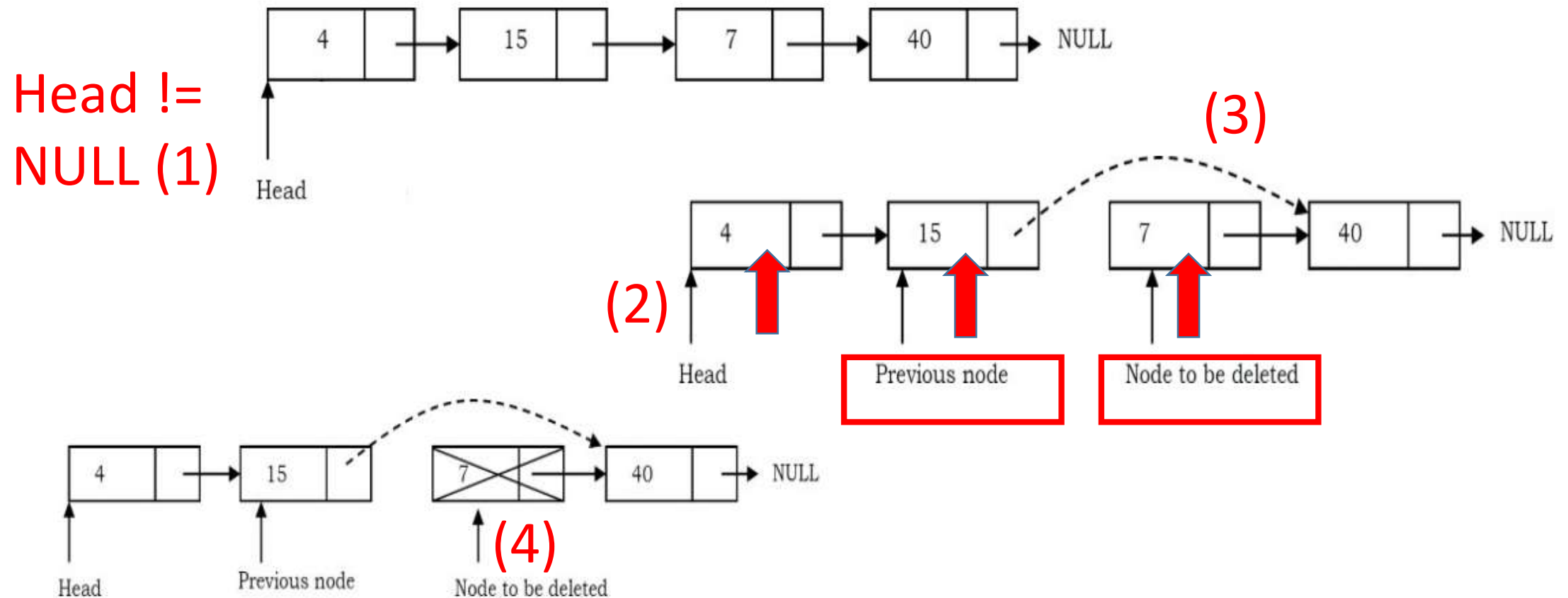
1. Pastikan bahwa Head tidak berisi NULL
2. Pengecekan setiap elemen node berdasarkan nilai node, untuk menemukan posisi node yang akan dihapus (**tempNode**) dan node sebelum node yang akan di hapus (**preNode**)
3. Next preNode diisi dengan nilai next tempNode
4. Node tempNode di hapus/dispose

# Proses: Remove di Tengah 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
- Asumsi akan dihapus node yang bernilai 7
- Posisi ke-2 disebut sebagai *previous node*



# Proses: Remove di Tengah Jika List Not Empty





# Algoritma Remove Middle List

```
procedure removeMid(x: integer)
```

```
deklarasi
```

```
    preNode, delNode: Node
```

```
    i: integer, ketemu: boolean
```

```
Deskripsi
```

```
IF (HEAD <> null) THEN
```

```
    ketemu ← false, i ← 1
```

```
    delNode ← HEAD
```

```
    WHILE (delNode.next <> null AND not ketemu)
```

```
        IF (delNode.data = x) THEN
```

```
            ketemu ← true
```

```
        ELSE
```

```
            preNode ← delNode
```

```
            delNode ← delNode.next
```

```
            i++
```

```
        ENDIF
```

```
    ENDWHILE
```

```
    {slide berikutnya}
```

```
IF (ketemu = true) THEN
```

```
    IF (i = 1) THEN {Jika satu elemen list}
```

```
        HEAD ← null
```

```
    ELSE
```

```
        preNode.next ← delNode.next
```

```
        dispose(delNode)
```

```
    ENDIF
```

```
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