



PRAKTIKUM PEMROGRAMAN I

List (Operasi Add Middle)

Ade Sukendar

Teknik Informatika Universitas Pasundan
2025

Ready !!! Go !!!



Operasi List

Add Middle / Sisip di Tengah

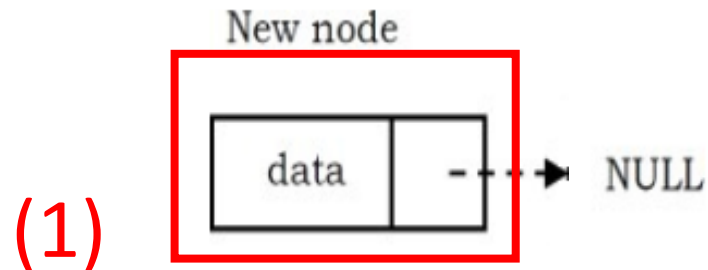
Kondisi Operasi Add Middle

- Add elemen di tengah mempunyai dua kondisi yaitu
 - Jika elemen List kosong (*empty*)
 - Jika elemen List tidak kosong (*not empty*)

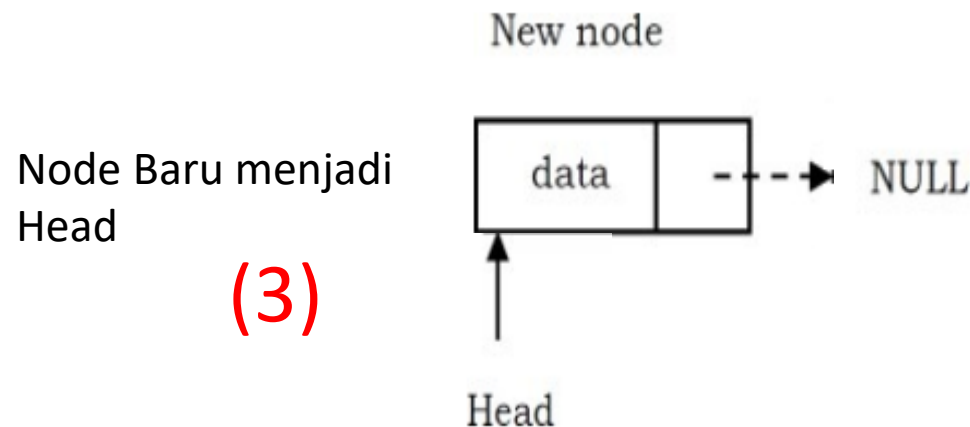
Proses 5: Add Middle Jika List Empty

1. Buat node baru
 - a. Isi value elemen dengan nilai tertentu
 - b. Isi next dengan NULL
2. Pastikan bahwa Head berisi NULL
3. Jadikan node baru sebagai HEAD

Proses 5: Add Middle Jika List Empty (cont..)



Head = NULL (2)

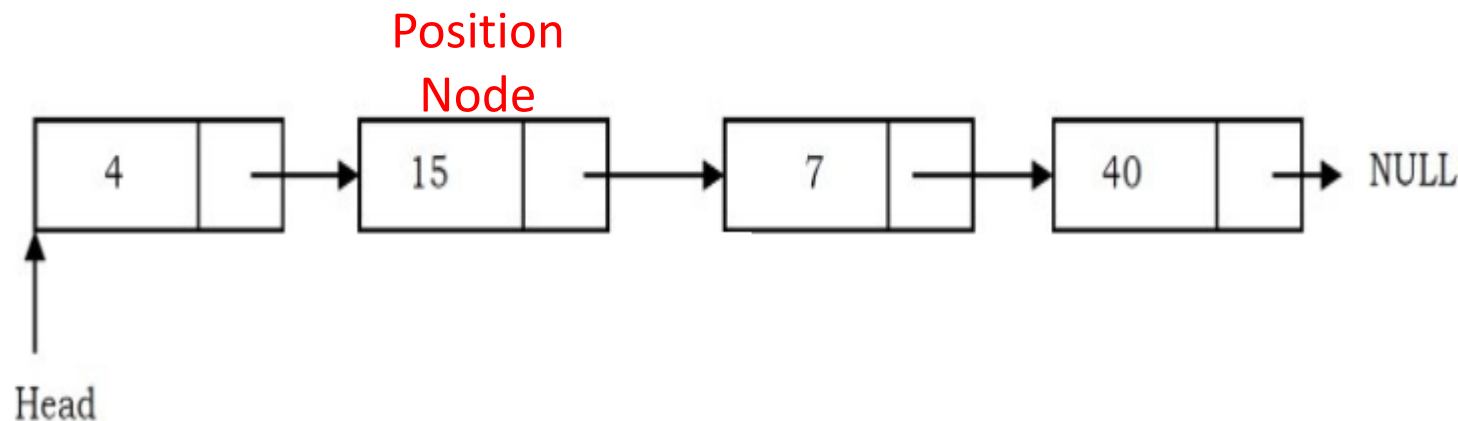


Proses 6: Add Middle Jika List Not Empty

1. Buat node baru
 - a. Isi value elemen dengan nilai tertentu
 - b. Isi next dengan NULL
2. Pastikan bahwa HEAD tidak berisi NULL
3. Pengecekan setiap elemen node sampai menemukan posisi node (*position node*) yang dicari
4. Next node baru diisi oleh next node *position node*
5. Next node *position node* diisi relasi ke node baru

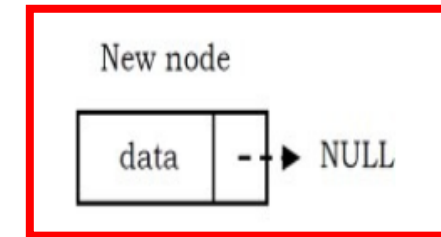
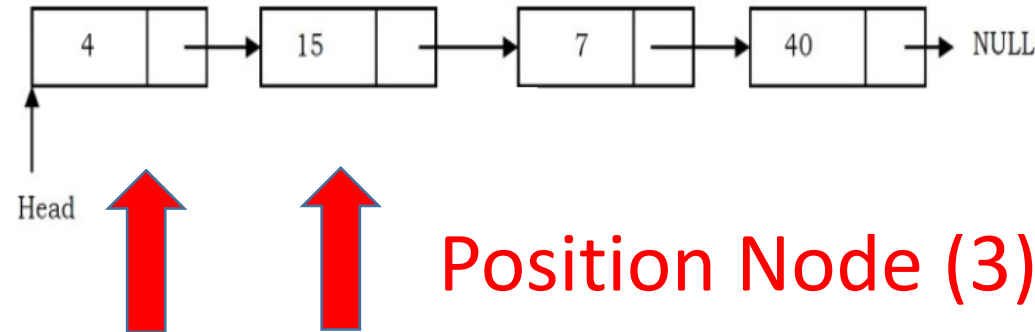
Proses 6: Add Middle Jika List Not Empty (cont..)

- 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 ditambahkan node baru di posisi ke-3, sehingga harus berhenti di posisi ke-2
- Posisi ke-2 disebut sebagai *position node*

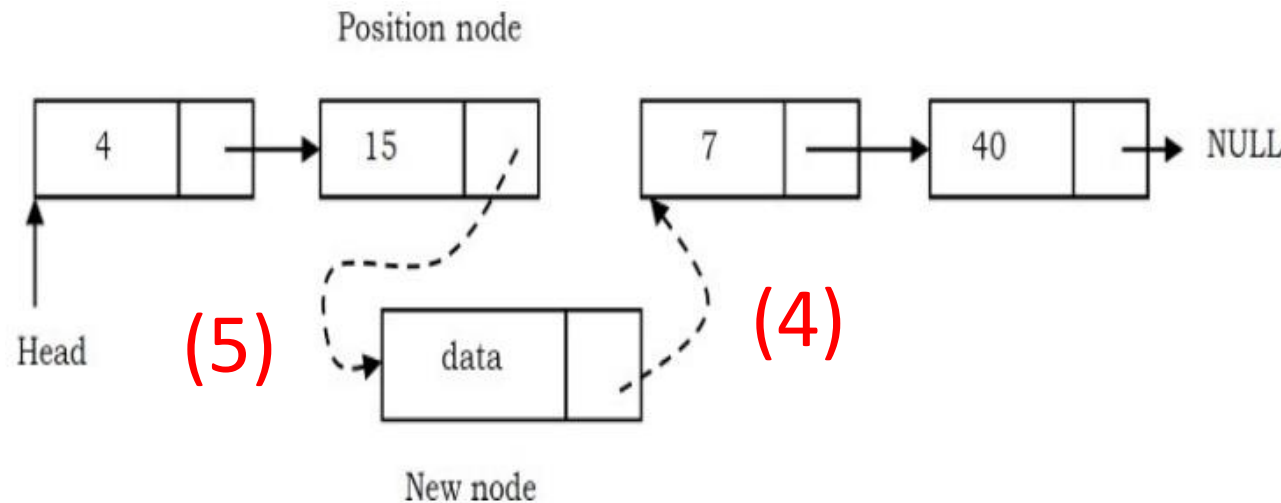


Proses 6: Add Middle Jika List Not Empty (cont..)

head !=
NULL (2)



(1)



(5)

(4)

Algoritma Operasi Add Middle

```
procedure addMid(data: integer, position: integer)
```

```
  deklarasi
```

```
    posNode, curNode: Node {current node}
```

```
    i: integer
```

```
  deskripsi
```

```
    newNode ← new Node(data)
```

```
    IF (HEAD = null) THEN
```

```
      HEAD ← newNode
```

```
    ELSE
```

```
      curNode ← HEAD;
```

```
      IF (position = 1) THEN           {tambah di awal}
```

```
        newNode.next ← curNode
```

```
        HEAD ← newNode
```

```
        {slide berikutnya}
```

Algoritma Operasi Add Middle (cont..)

```
ELSE
    i ← 1
    WHILE (curNode <> null AND i < position) DO
        posNode ← curNode
        curNode ← curNode.next
        i++
    ENDWHILE
    posNode.next ← newNode
    newNode.next ← curNode
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