



Teknik Informatika Kampus 3 Nganjuk  
Jurusan Teknologi Informasi

Politeknik Negeri  
Jember

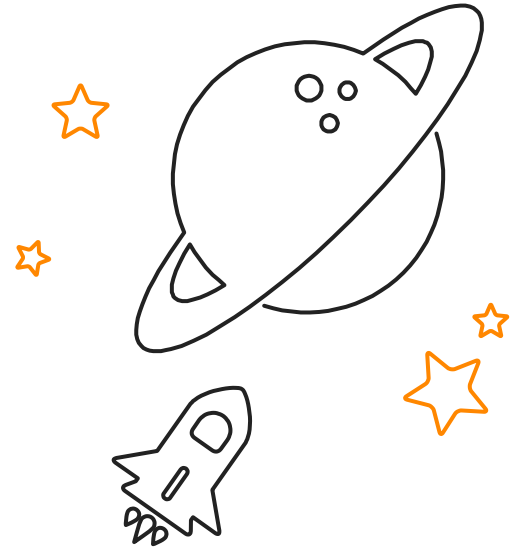
# Single Linked List

Struktur Data

Ulfa Emi Rahmawati, S.Kom., M.Kom.

# Introducing Linked List

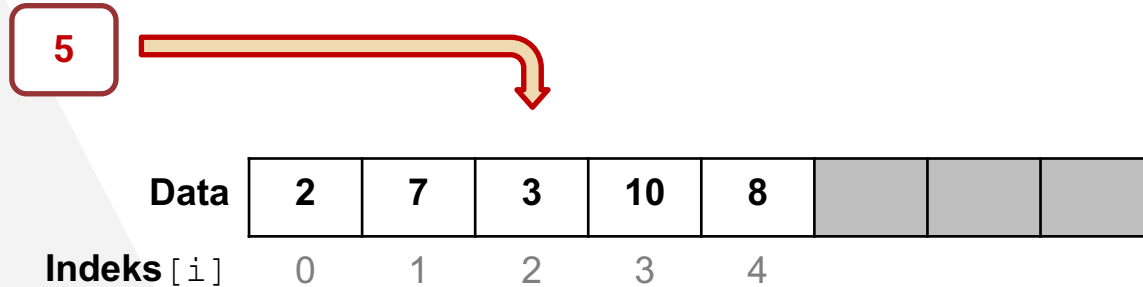
Mengapa diperlukan linked list?



## Pentingnya Linked List

Bagaimanakah cara menambah data ke dalam suatu **Array List** pada indeks tertentu?

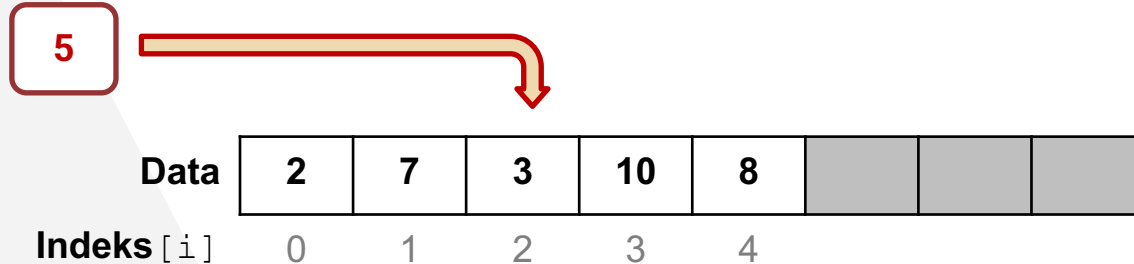
- Tambahkan data **5** ke dalam Array pada indeks ke-2



## Pentingnya Linked List

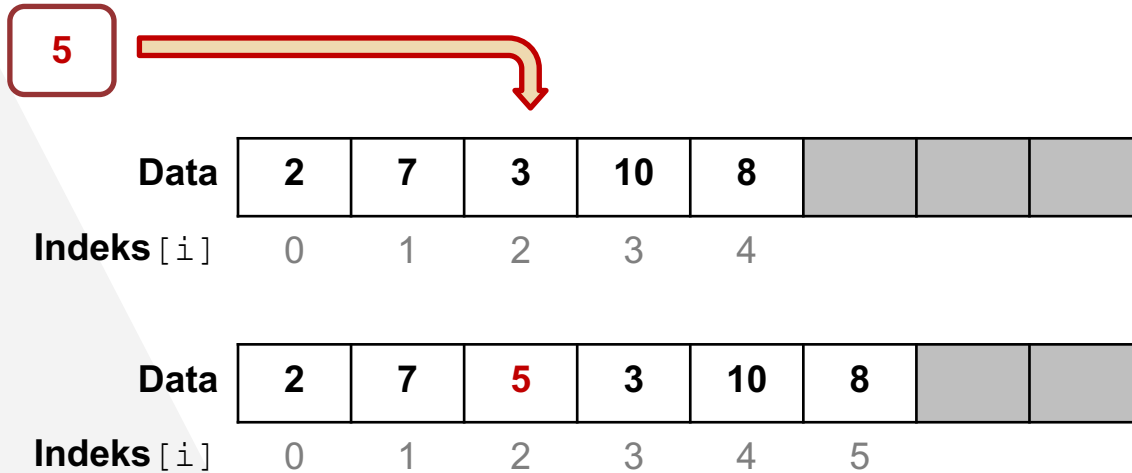
### Langkah-langkah

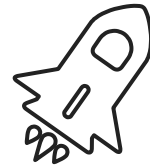
- Tambahkan angka **5**, ke dalam Array {2, 7, 3, 10, 8} pada indeks ke-2
- Diperlukan pemindahan data **3, 10, 8**
- Hasil akhir menjadi Array {2, 7, **5**, 3, 10, 8}



## Pentingnya Linked List

- Diperlukan pemindahan ekor (Tail) ke kanan
- Begitu juga saat **menghapus data**.



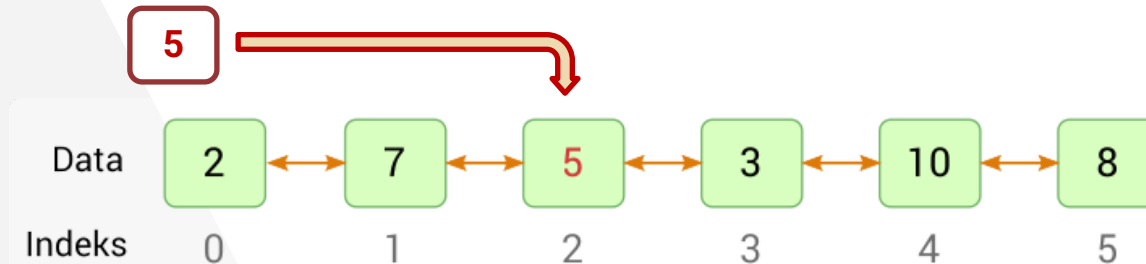


# Butuh Solusi!

Diperlukan alternatif lainnya, agar lebih efisien!

## Solusi

- Perlu alternatif struktur lain untuk menyimpan elemen dalam suatu urutan (**sequence**)
- Efisien saat penambahan dan penghapusan data pada posisi tertentu (**indeks tertentu**)
- **Perbaikan** hanya pada **Link** yang **berdekatan** elemen baru
- Tidak berdampak pada elemen lainnya



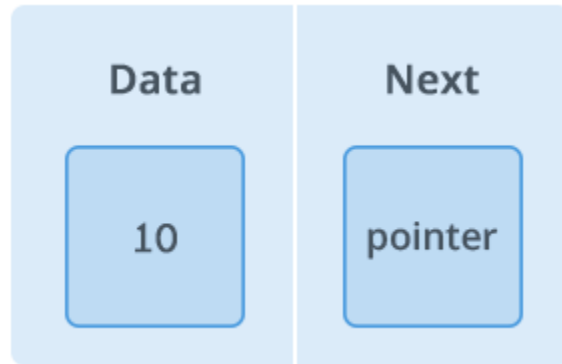
# Pengertian Linked List

Pembahasan Linked List



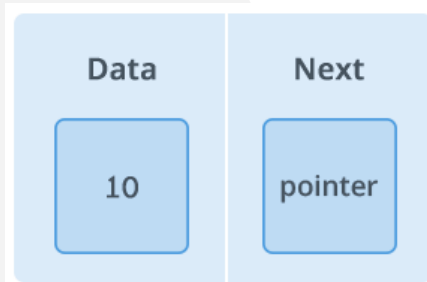
## Pengertian Linked List

- Sebuah cara untuk menyimpan kumpulan elemen.
- Elemen dapat bertipe ***integer*** atau ***character***.
- Setiap elemen tersimpan dalam bentuk **Node**.

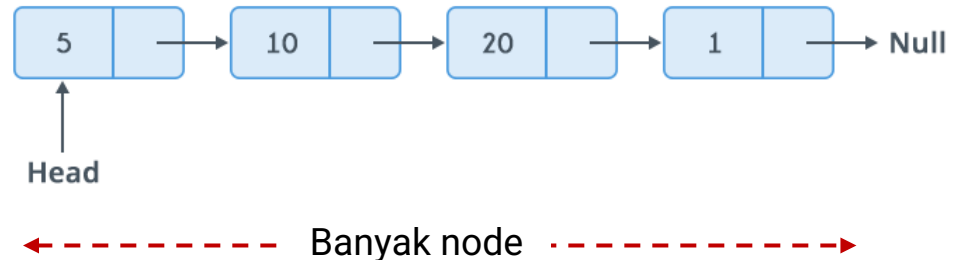


# Struktur Linked List

- Node terdiri dari 2 bagian, **data** dan **next**
- **Data** berfungsi menyimpan elemen (nilai)
- **Next** berfungsi menyimpan **link** pada **node** selanjutnya
- Linked list terbentuk ketika banyak node saling terhubung sehingga membentuk rantai

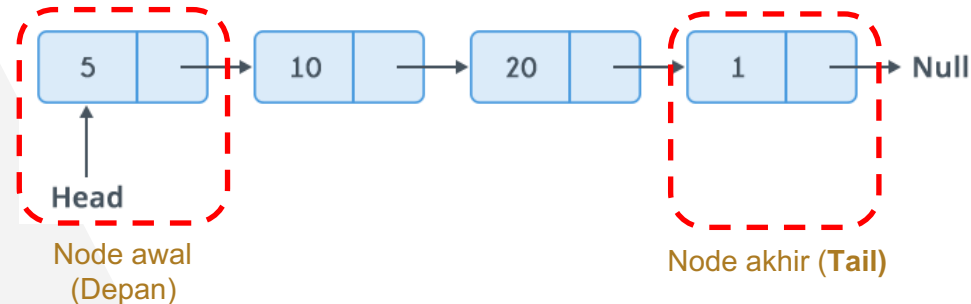


← - - Sebuah node - - →



## Struktur Linked List

- Node baru ditambahkan diawal menjadi Node Awal (**Head**)
- **Next** pada node terakhir menunjuk ke **null** sebagai penanda node terakhir (**Tail**)
- Single Linked List hanya memungkinkan pergerakan maju (**Forward**)



# Operasi pada Linked List

Jenis-jenis operasi pada Linked List

## Operasi Linked List

Operasi pada linked list terbagi menjadi:

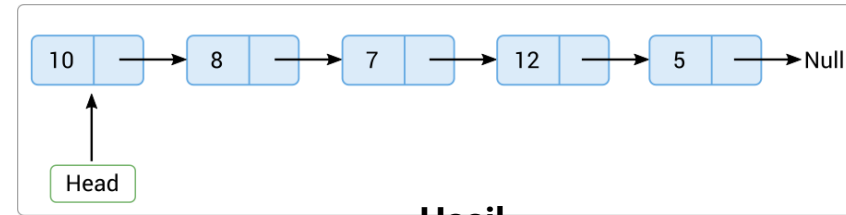
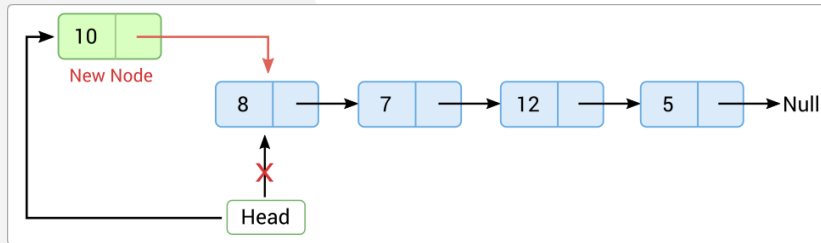
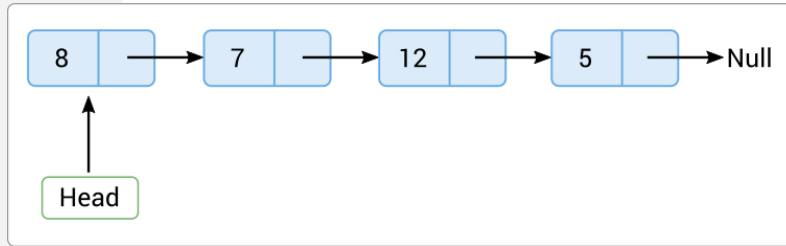
- Insertion, penyisipan node baru pada suatu linked list
- Deletion, penghapusan node pada suatu linked list
- Traversal, menampilkan node pada suatu linked list
- Search, mencari node tertentu pada suatu linked list
- Sort, mengurutkan node pada suatu linked list

## Insertion

- Insertion, penyisipan node baru pada suatu linked list
  - Penyisipan di awal, penyisipan elemen di awal list.
  - Penyisipan di akhir, penyisipan elemen di akhir list.
  - Penyisipan setelah elemen/node tertentu.

# Insertion

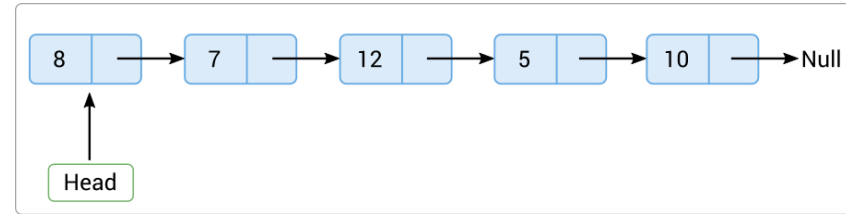
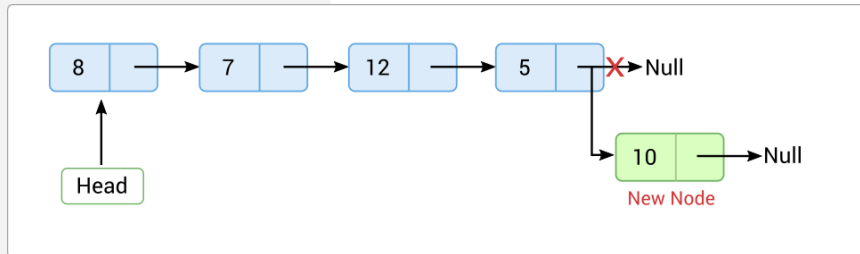
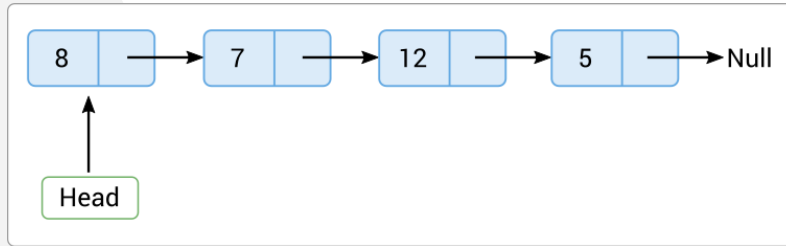
- Penyisipan elemen di awal list



**Hasil**

# Insertion

- Penyisipan elemen di akhir list

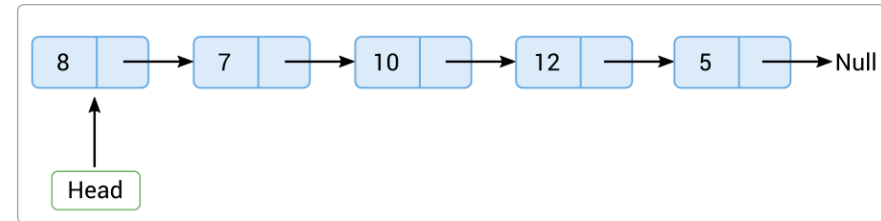
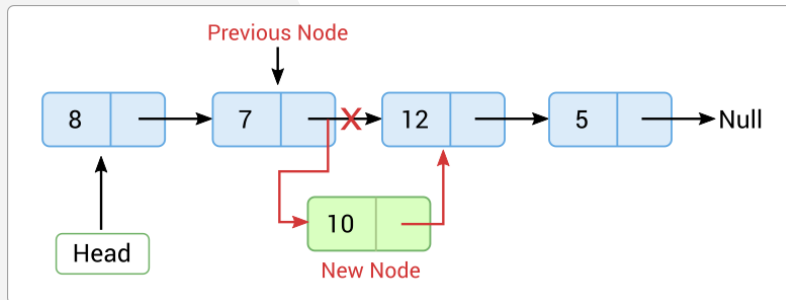
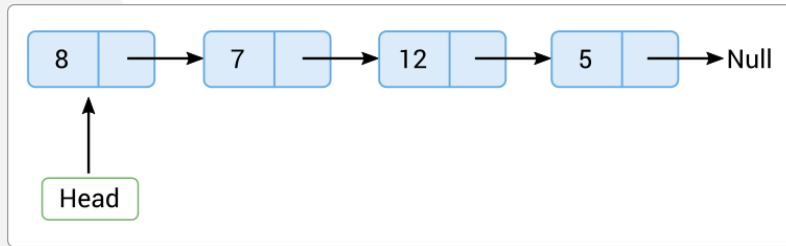


**Hasil**



# Insertion

- Penyisipan elemen setelah node tertentu



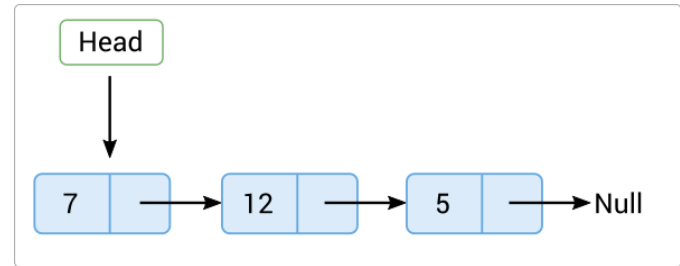
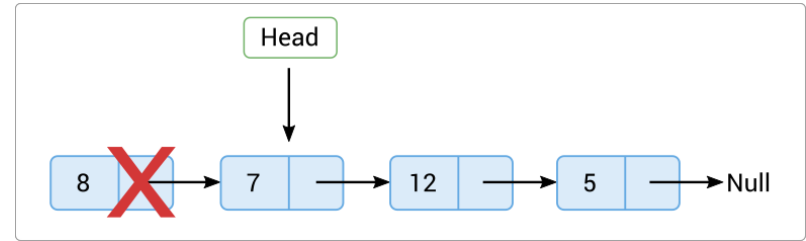
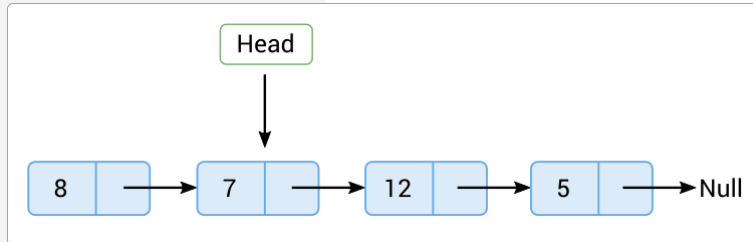
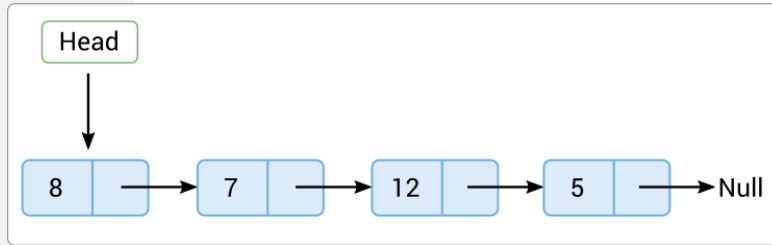
**Hasil**

## Deletion

- Deletion, penghapusan node pada suatu linked list
  - Penghapusan di awal, penghapusan elemen di awal list.
  - Penghapusan di akhir, penghapusan elemen di akhir list.
  - Penghapusan setelah elemen/node tertentu.

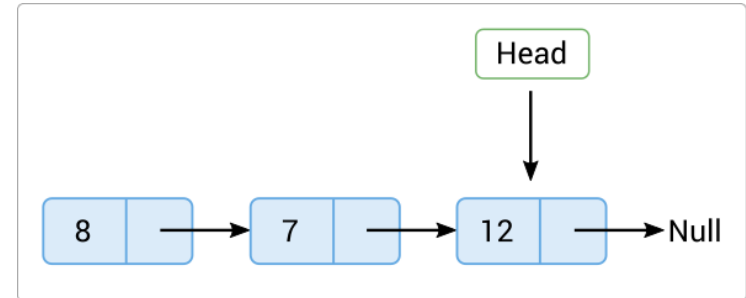
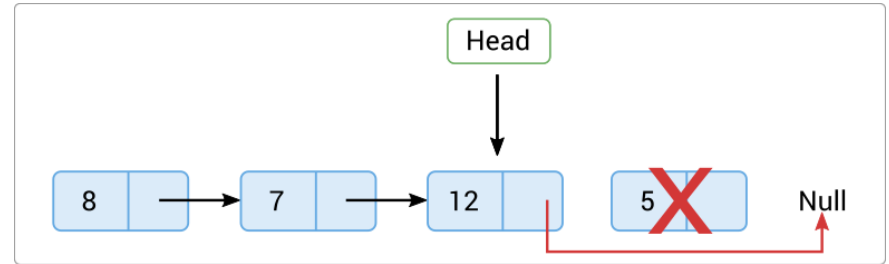
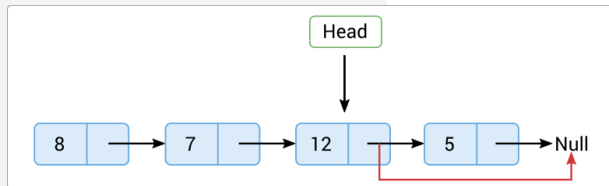
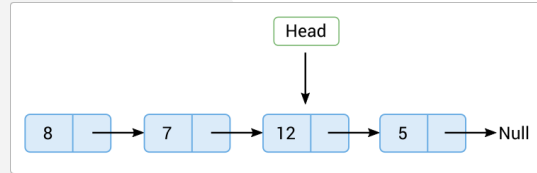
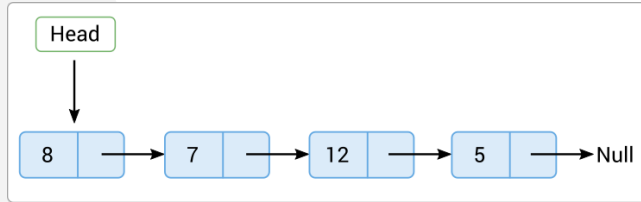
# Deletion

- Deletion di Awal



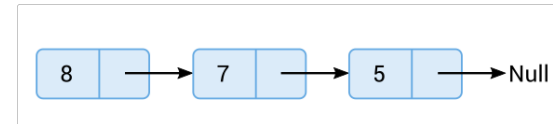
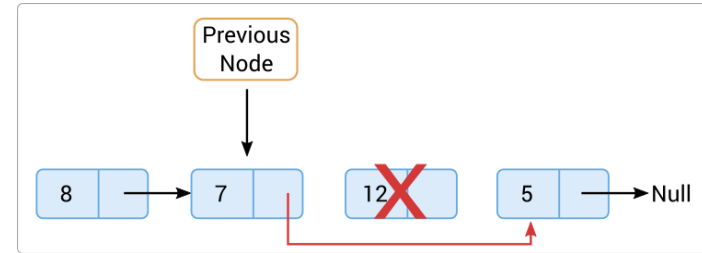
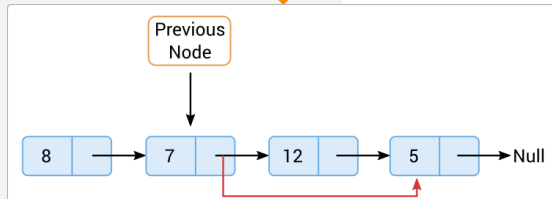
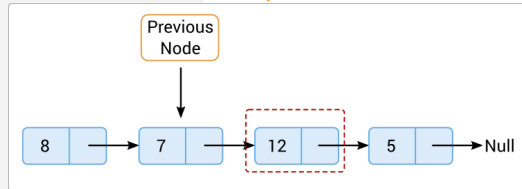
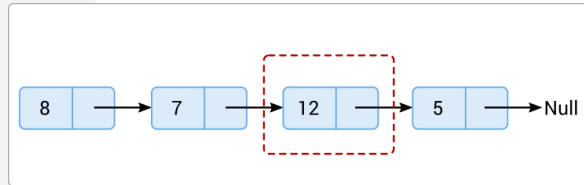
# Deletion

- Deletion di Akhir



# Deletion

- Deletion pada node tertentu



## Pemanfaatan Linked list

Linked list dapat dimanfaatkan pada

- Antrian (**Queues**)
- Tumpukan (**Stacks**)
- **Graph**

# THANKS!

**Any questions?**