Implementasi Struktur Data

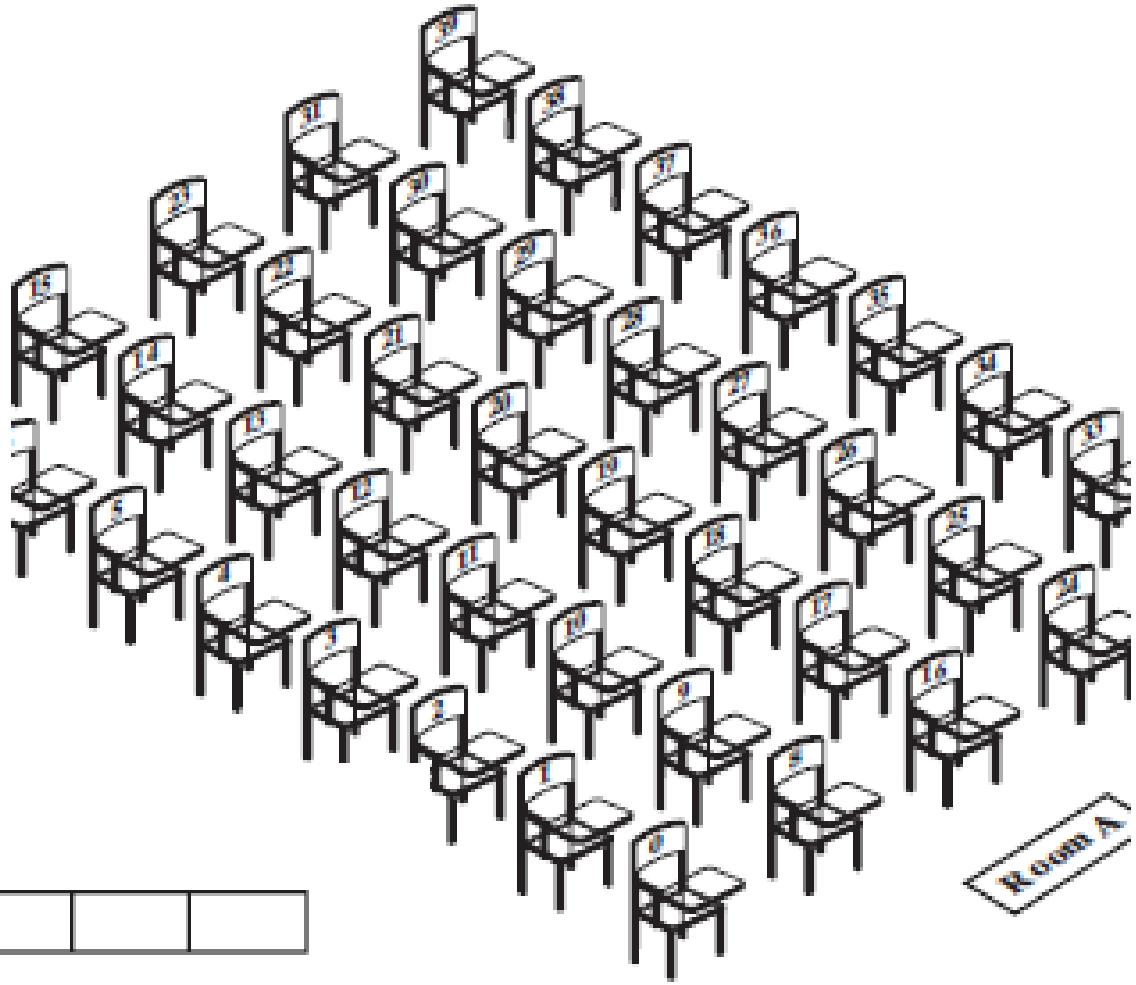
Linked List

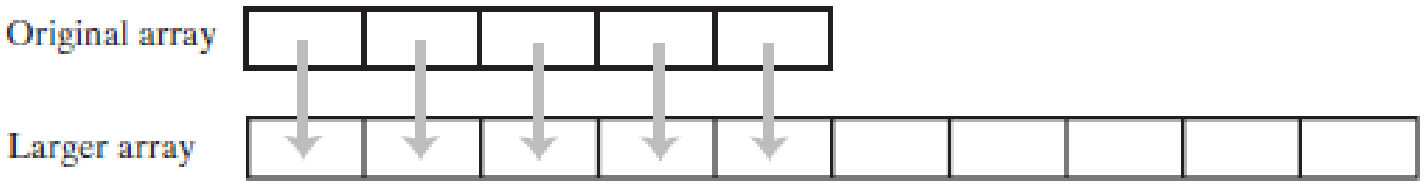
Program Studi Diploma III Teknik Informatika
Fakultas Ilmu Terapan
Universitas Telkom


LIST

Array has a fixed size, and so it can either become full or have several unused elements.

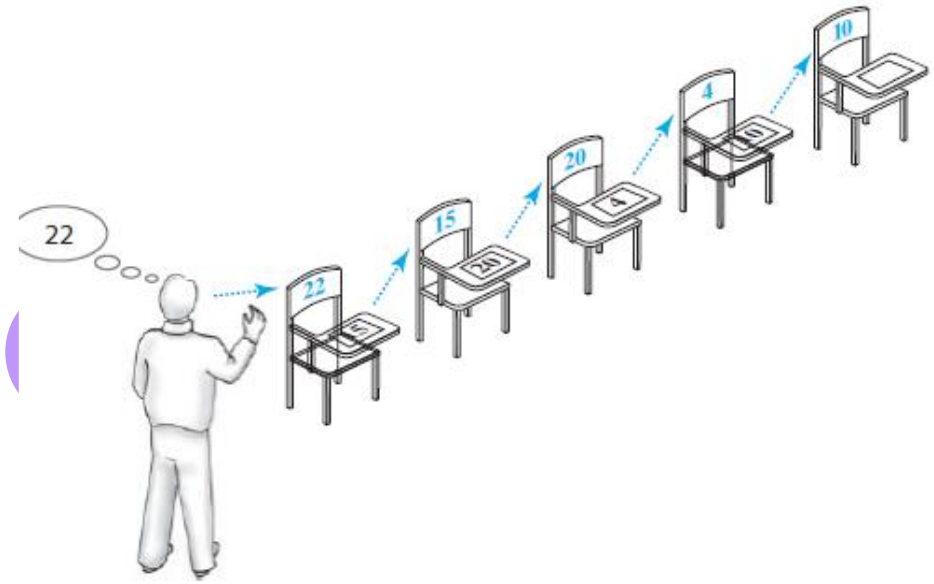
Resizing an array -> move data each time





LIST

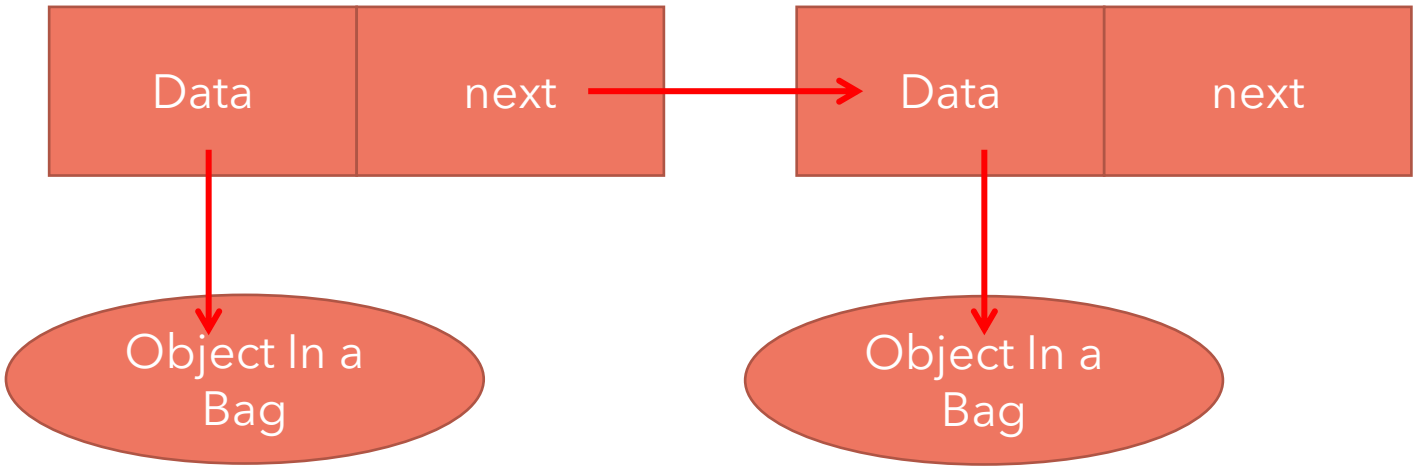
- Data organization that uses memory only as needed for a new entry and returns the unneeded memory to the system after an entry is removed.
- By linking data, it avoids moving data when adding or removing entries.



Linked list and array (konvensional)

Array	List
Alokasi memory bersifat statis	Alokasi memory bersifat dinamis
Lokasi memory continue (fisik dan logik terurut)	Lokasi pada memori random (fisik dapat terpisah, logik berkaitan)
Operasi pengubahan susunan data relatif memakan waktu	Operasi pengubahan susunan data lebih mudah dan ringkas
Akses data lebih mudah (menggunakan indeks)	Akses data lebih sulit (menggunakan bantuan)

LINKED LIST



- A **linked list** is a data structure used for collecting a sequence of objects that allows efficient addition and removal of elements in the middle of the sequence.
- A linked list consists of a number of nodes, each of which has a reference to the next node. We call this the next link. The last cell's next link references null.

Linked List

- Linked List bersifat linear, dan biasanya diakses melalui referensi dari node pertama pada linked list tersebut.
- Node-node berikutnya diakses secara berturut-turut melalui referensi link dari node sebelumnya
- Berdasarkan kesepakatan, referensi link pada node terakhir di-set ke null yang menandakan “akhir dari list”
- Data disimpan dan dihapus dari linked list secara dinamis, node akan dibuat dan dihapus sesuai kebutuhan

SINGLY LINKED LIST

Singly Linked List

- Bentuk linked list yang paling sederhana
- Objek linked list (head/ first) memiliki reference ke node pertama dari linked list
- Merupakan list satu arah: setiap node memiliki reference ke node selanjutnya pada linked list

Operasi pada Singly Linked List

Insert (Sisip) :

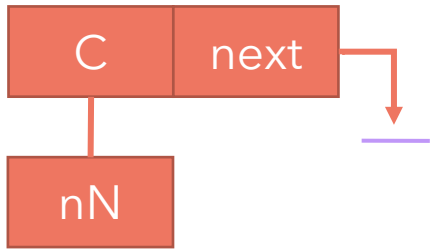
- Sisip Depan
- Sisip Belakang
- Sisip Tengah

Delete (Hapus):

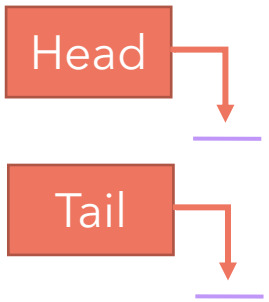
- Hapus Depan
- Hapus Belakang
- Hapus Tengah

Sisip Depan (dengan head dan tail)

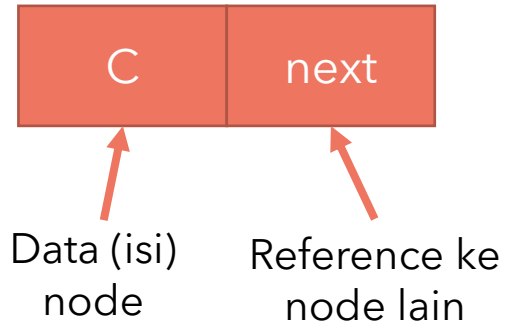
Keadaan 1:
Linked List masih kosong, dilakukan sisip Depan

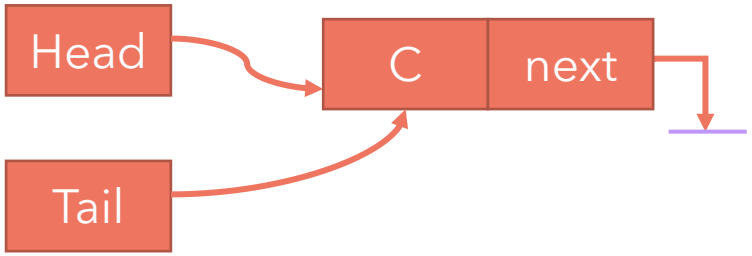


Head = Tail = Null



Node dengan isi "C" akan dimasukkan pada linked list



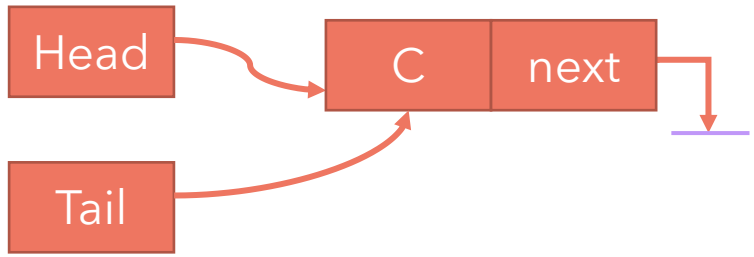


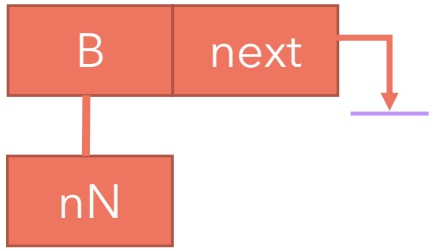
Head = Tail = nN

Pastikan agar Head selalu berada di awal linked list, dan Tail di akhirnya

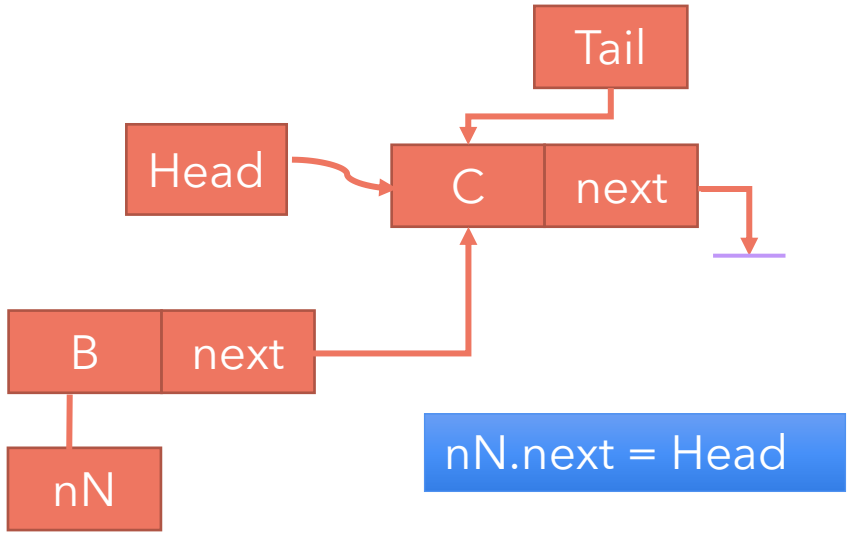
Pastikan agar Head selalu berada di awal linked list, dan Tail di akhirnya

Keadaan 2:
Sisip depan pada linked list yang memiliki isi

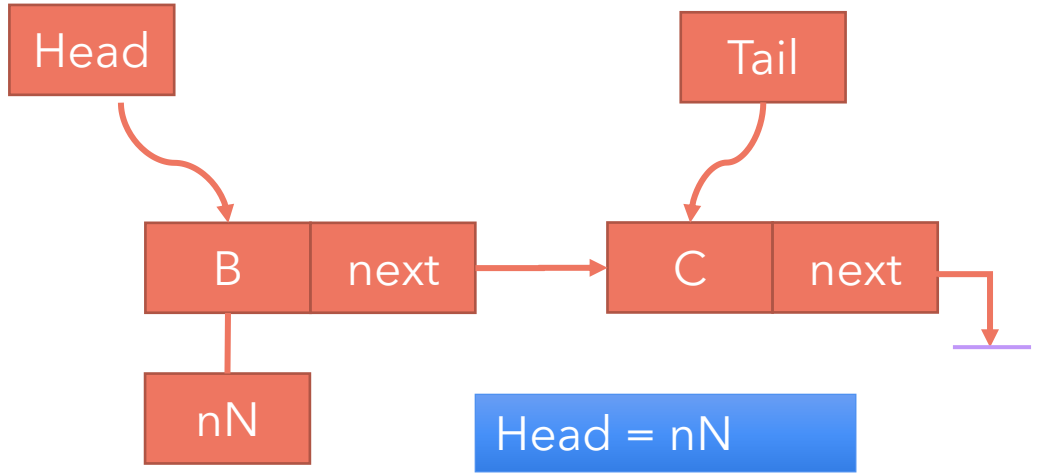




Sisipkan node B di depan List

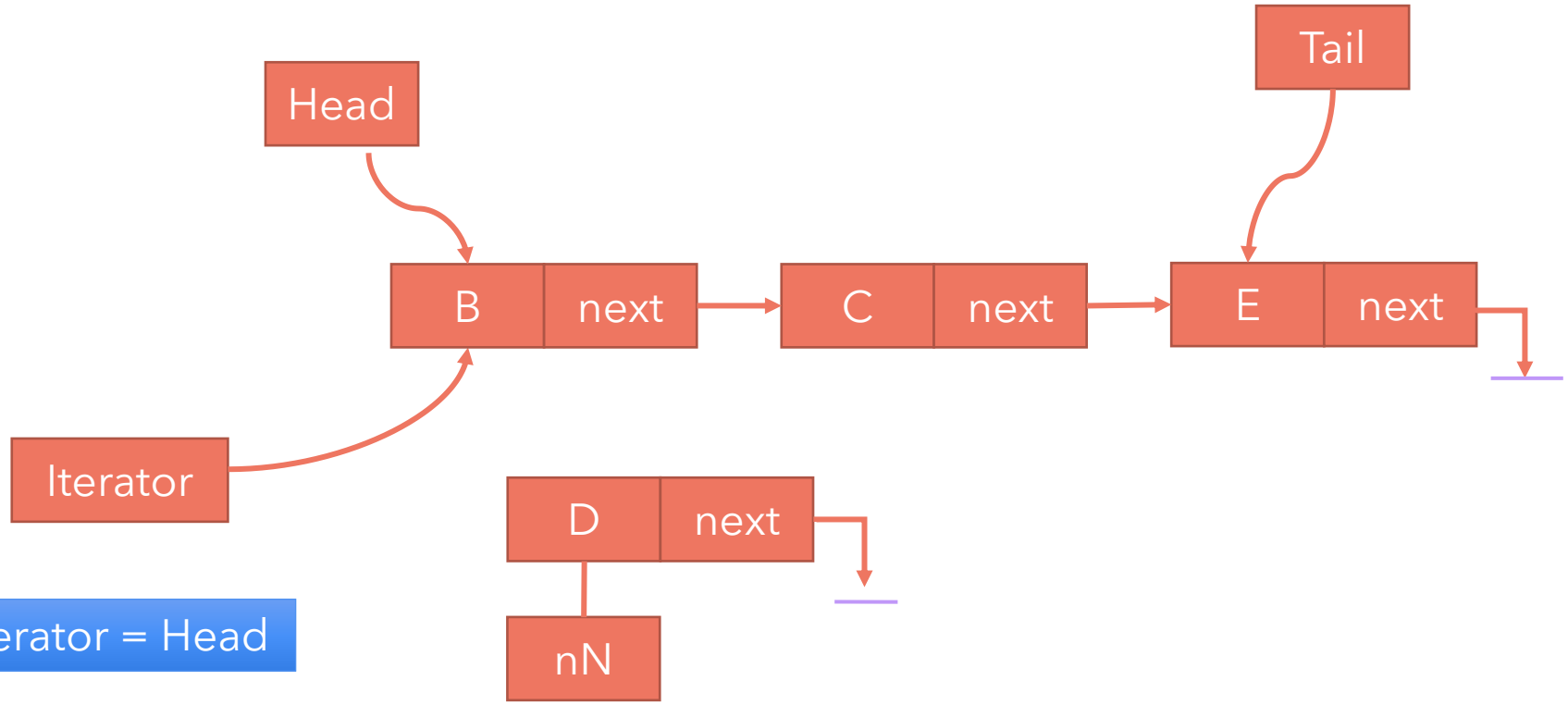


nN.next = Head



Head = nN

Sisip Tengah (dengan head dan tail)



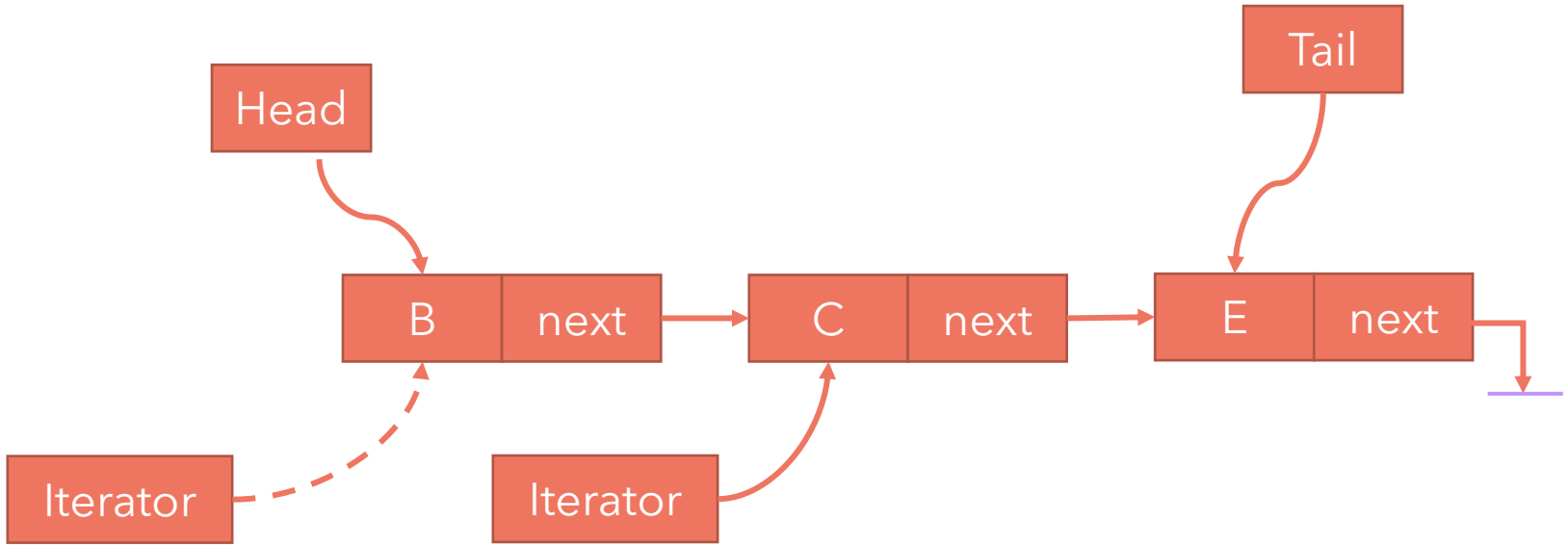
Iterator = Head

Sisipkan D
antara C dan E

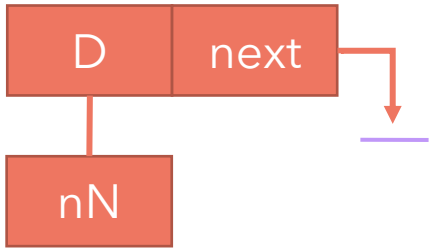
Posisi yang tepat??

Bagian next (iterator)
adalah node E

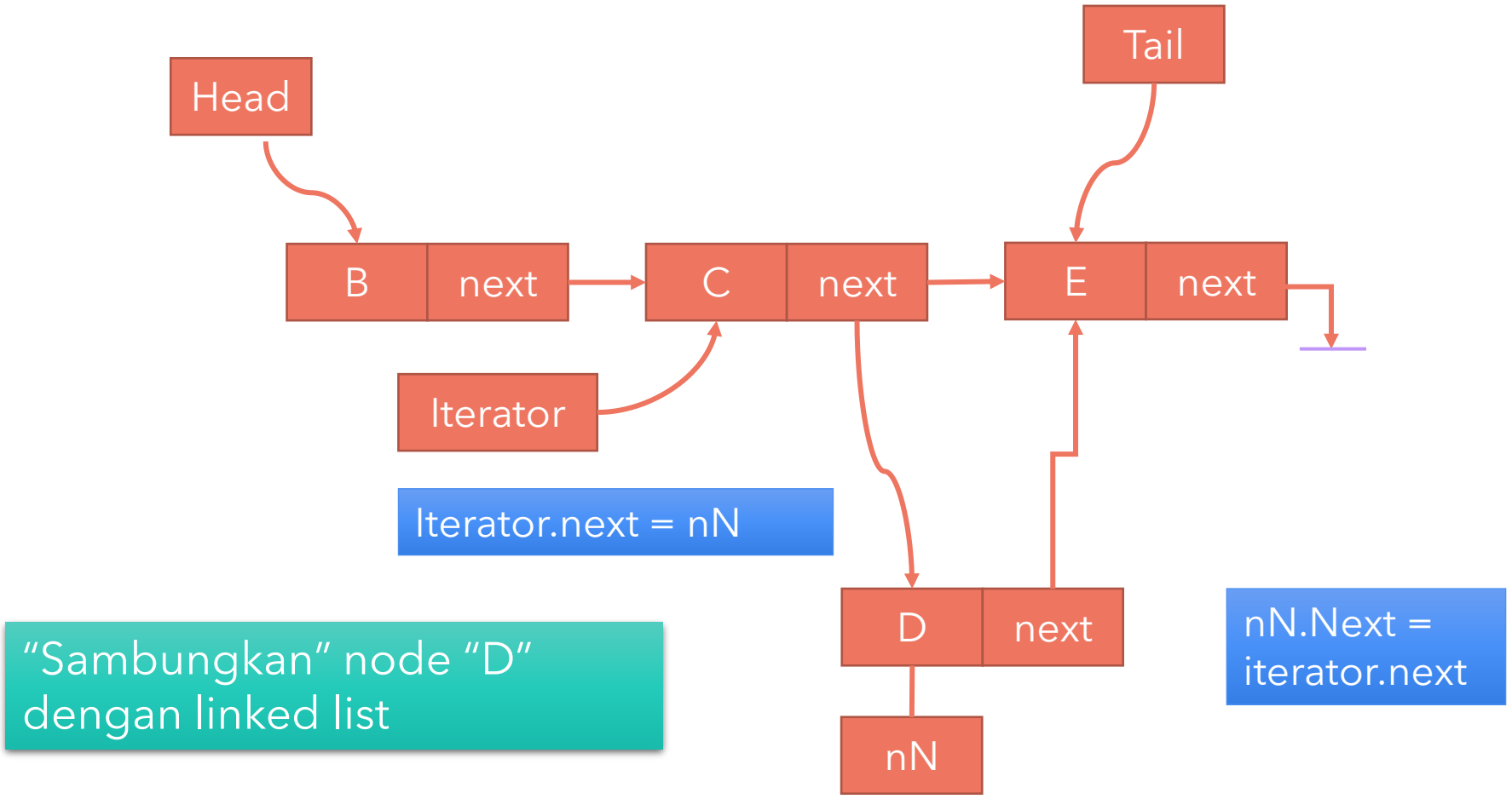
Sisip Tengah



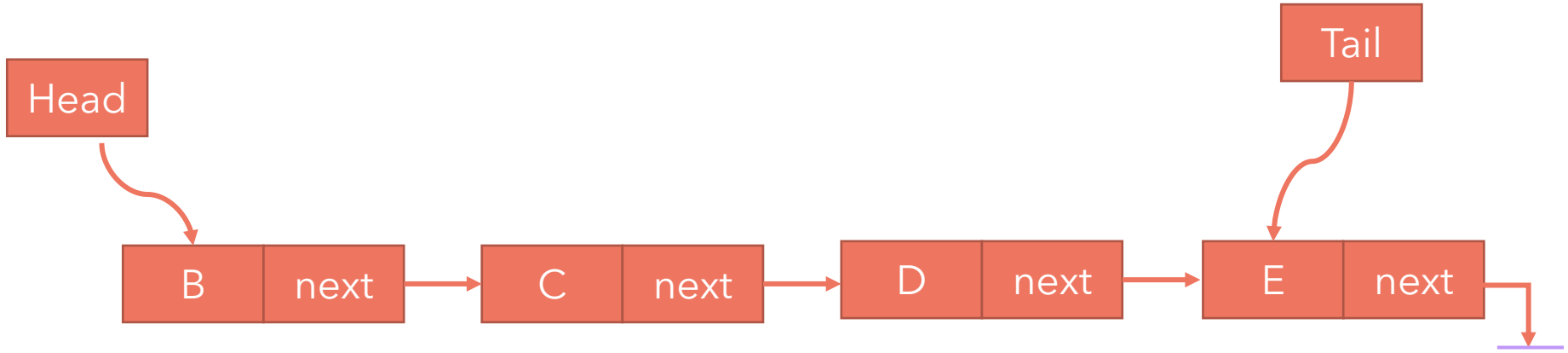

```
while (iterator.next != "E")
    iterator = iterator.next
```



Sisip Tengah



Sisip Tengah

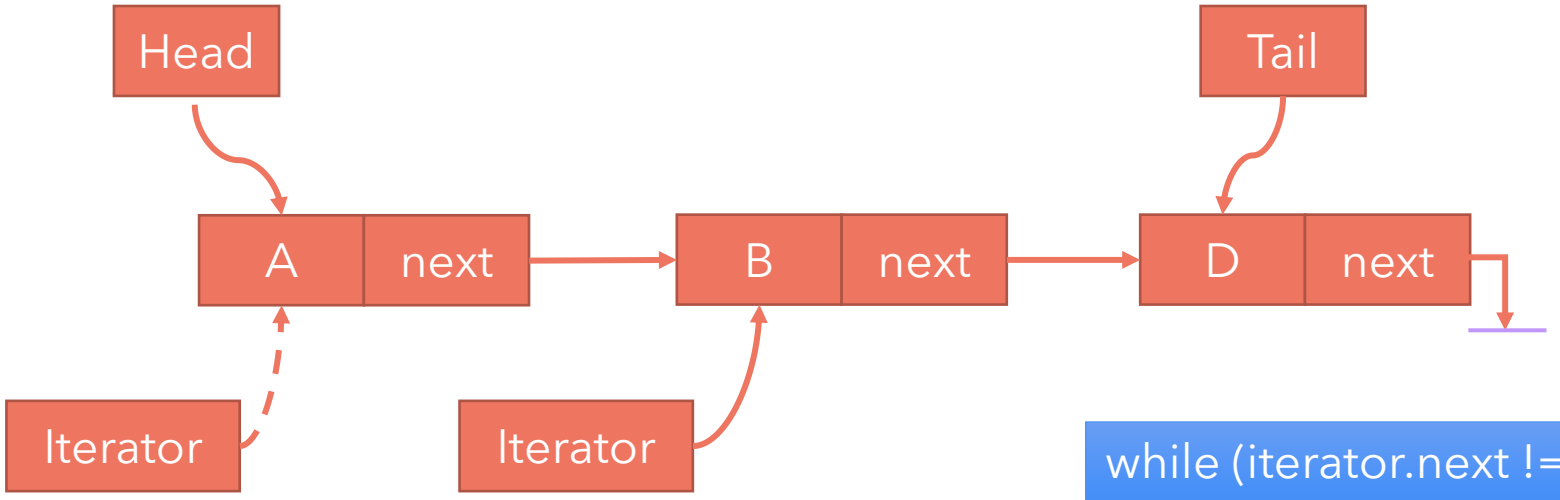


Hapus Belakang (dengan head dan tail)

Pastikan agar Head selalu berada di awal linked list, dan Tail di akhirnya

Pindahkan Tail dari node paling belakang

Gunakan iterator untuk "memegang" node sebelum Tail



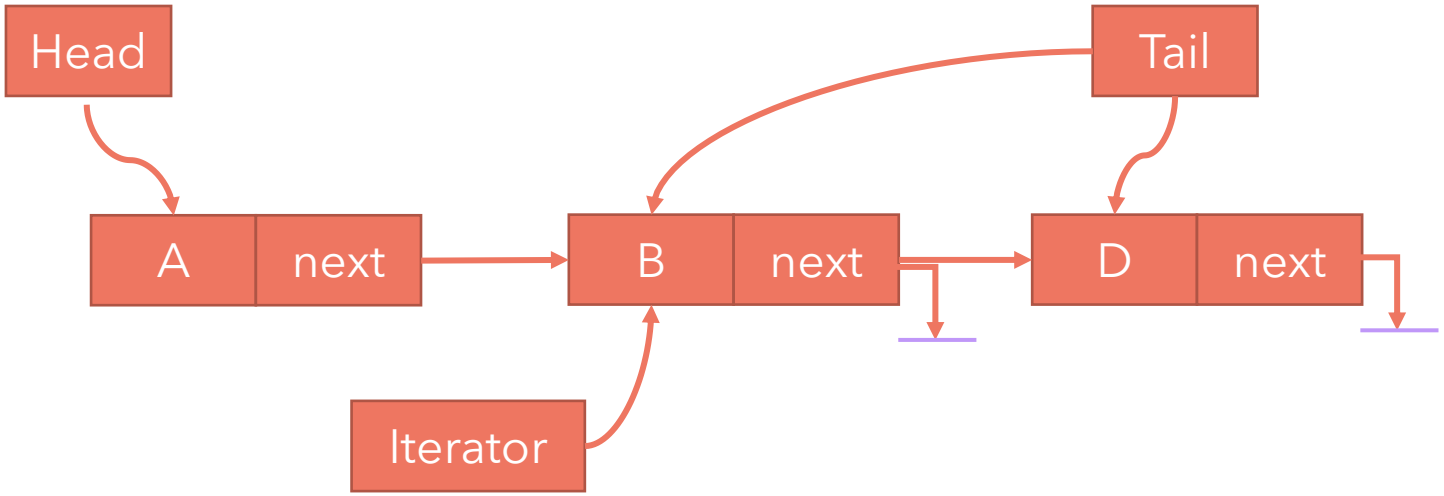

```
while (iterator.next != Tail)
    iterator = iterator.next
```

Hapus Belakang

Pastikan agar Head selalu berada di awal linked list, dan Tail di akhirnya

Pindahkan Tail dari node paling belakang

Gunakan iterator untuk "memegang" node sebelum Tail



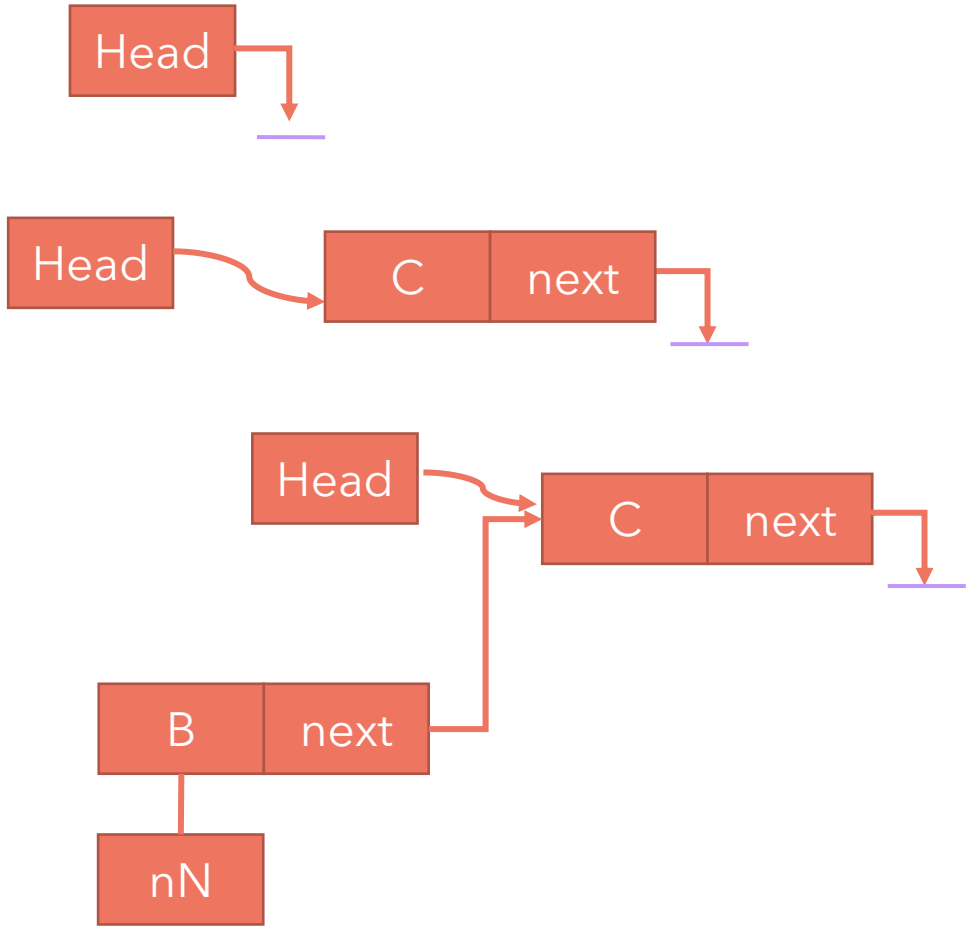
Tail = iterator

Tail.next = null

PENGAYAAN

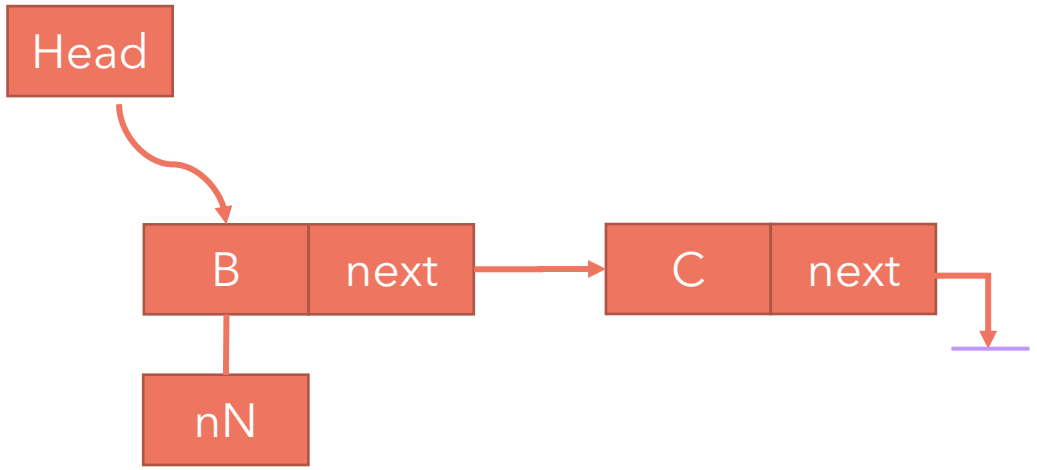
Sisip Depan (dengan head saja)

- Linked List masih kosong, head menunjuk ke NULL
- Ketika disisipkan node baru, maka head akan menunjuk node tersebut
- Ketika kemudian akan disisipkan node baru di depan linked list yang ada, maka newNode akan menunjuk ditunjuk oleh head



Sisip Depan

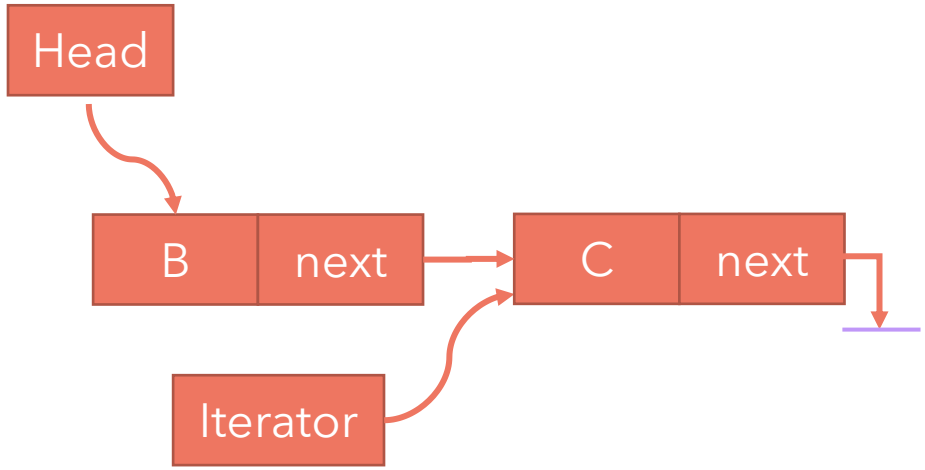
Terakhir, head menunjuk node baru tersebut

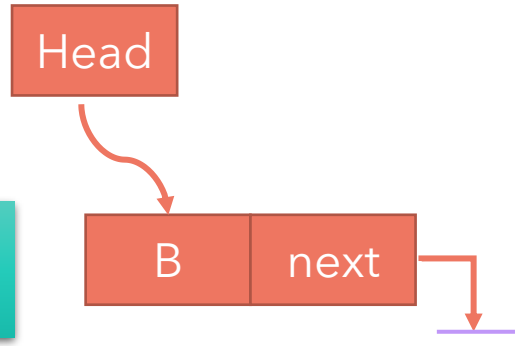


Latihan :Bagaimana dengan Sisip Belakang dan Sisip Tengah?

Hapus Belakang (dengan head saja)

- Buat iterator mencapai akhir linked list.
- Hapus node yang ditunjuk iterator. Arahkan node sebelum terakhir untuk menunjuk null





Latihan : Bagaimana dengan Hapus Depan dan Hapus Tengah?

Reference

- https://www.cpp.edu/~ftang/courses/CS240/lectures/adt.htm
- Carrano, F., M., *Data Structures and Abstraction with Java,* 3rd Ed, Prentice Hall. (2012)
- Hortsmann, C., *Big Java,* 4th Ed., John Waley & Sons, Inc. (2010)
- Deitel P., Deitel H., *Java How to Program, Early Objects (Deitel: How to Program),* 11th Edition, Pearson (2017)