



Struktur Data

# Linked List

---

MUHAMMAD 'ARIFUL FURQON S.PD., M.KOM.

FAKULTAS ILMU KOMPUTER

UNIVERSITAS JEMBER



# NEW RULE!

---

Pengumuman peraturan baru untuk kelas Struktur Data (terutama yang saya ampuh).

1. Presensi dilakukan via attendance e-learning bukan QR Code. Catatan kehadiran yang diutamakan dalam penilaian adalah attendance via e-learning/MMP. Attendance via QR Code hanya berlaku jika e-learning down atau terdapat masalah-masalah yg lain.
2. Harap menselaraskan nama yang digunakan di zoom dengan format NIM\_Nama agar mempermudah proses pencatatan keaktifan mahasiswa.
3. Pada waktu perkuliahan melalui zoom, saya sewaktu-waktu akan melakukan screen shot terhadap kehadiran mahasiswa. Jadi misal kalau di tengah2 zoom terdapat trobel atau ada masalah lain bisa menghubungi saya langsung untuk klarifikasi mengenai trobel atau masalah tersebut.

Demikian peraturan ini saya buat sebagaimana mestinya untuk kelancaran pembelajaran kita ke depannya.

TTD  
Muhammad 'Ariful Furqon, S.Pd., M.Kom.



# Important Links

---

Link PPT : <https://unej.id/SD07materi>

Link Code: <https://unej.id/SD07code>

Link Tugas: <https://unej.id/SD07tugas>

Link Record: <https://unej.id/zoomSD>



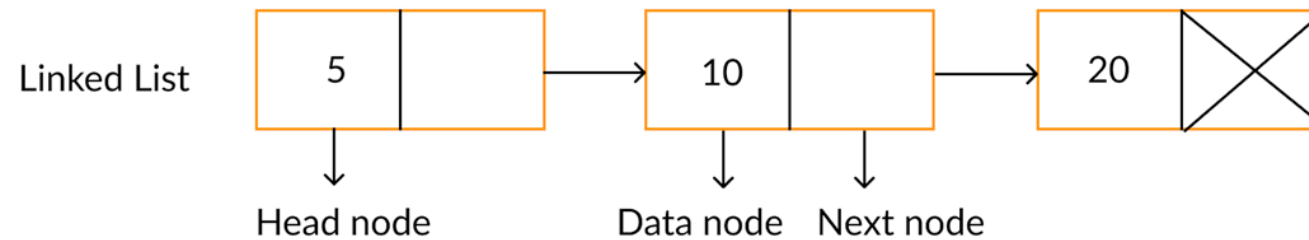
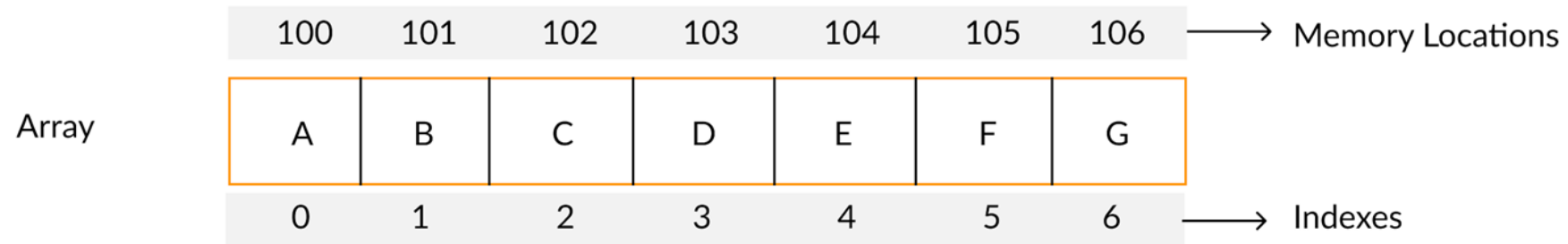
# Pre-Test

---

1. Jelaskan perbedaan linked list dengan array!
2. Jelaskan mengapa pemrosesan data yang disimpan di linked list lebih lambat dibanding array!
3. Jelaskan struktur sebuah node di linked list!
4. Jelaskan struktur linked list!
5. Jelaskan apakah first node atau head. Jelaskan karakteristik first node!



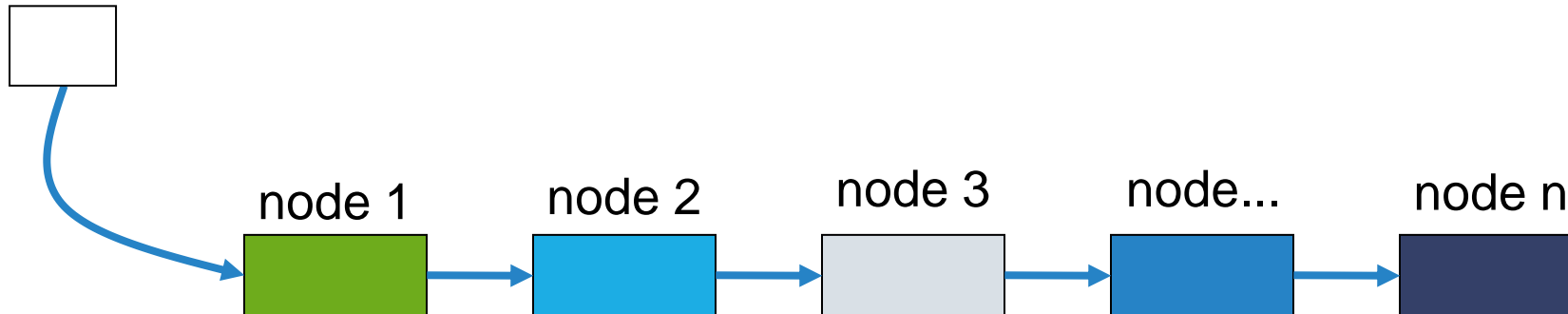
# Preview – Array vs Linked List





# Preview – Struktur Linked List

head / start  
node

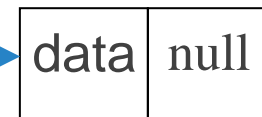




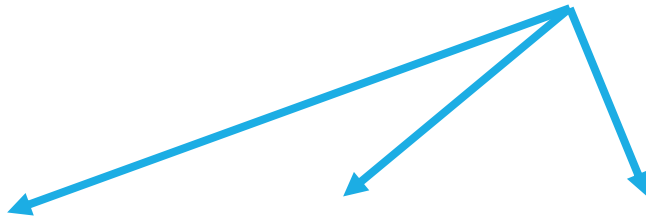
# Preview – Node

Head / start node adalah pointer yang menunjuk node pertama

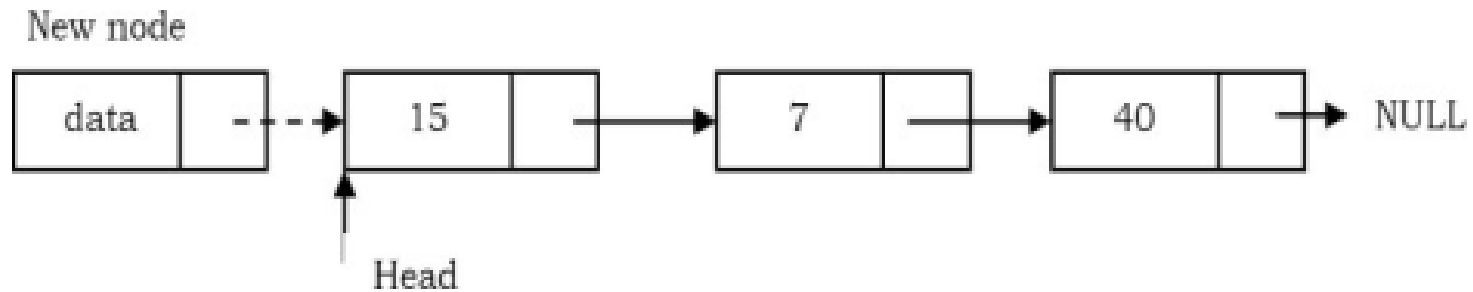
Head/Start Node



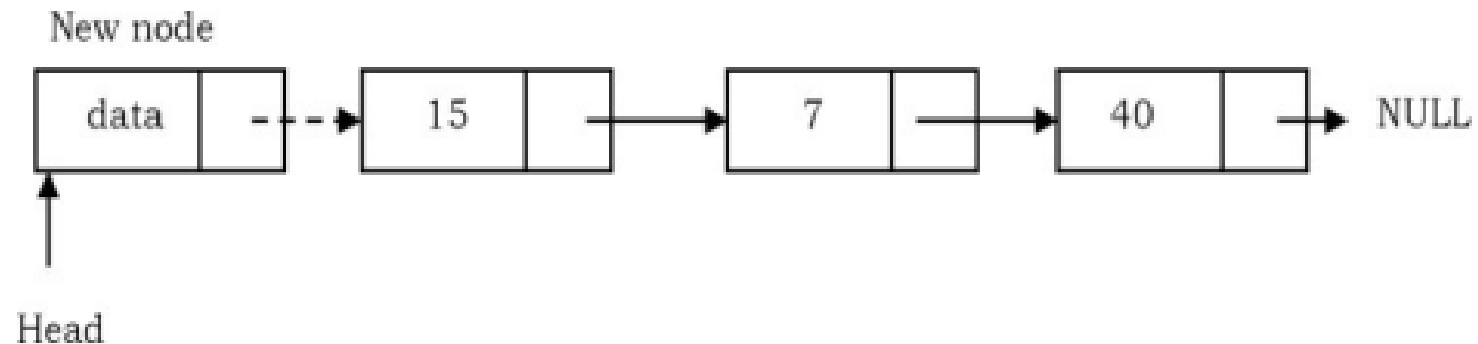
*Node*



# Preview – Menambah Node Baru di Awal Linked List

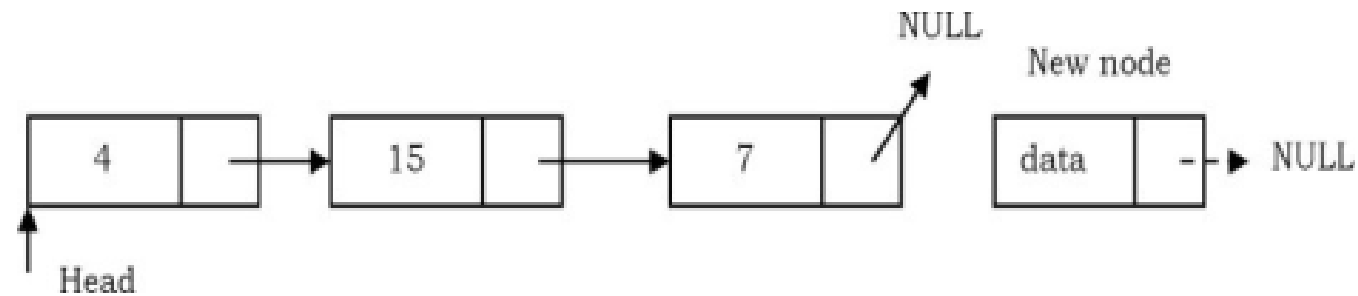


- Update head pointer to point to the new node.

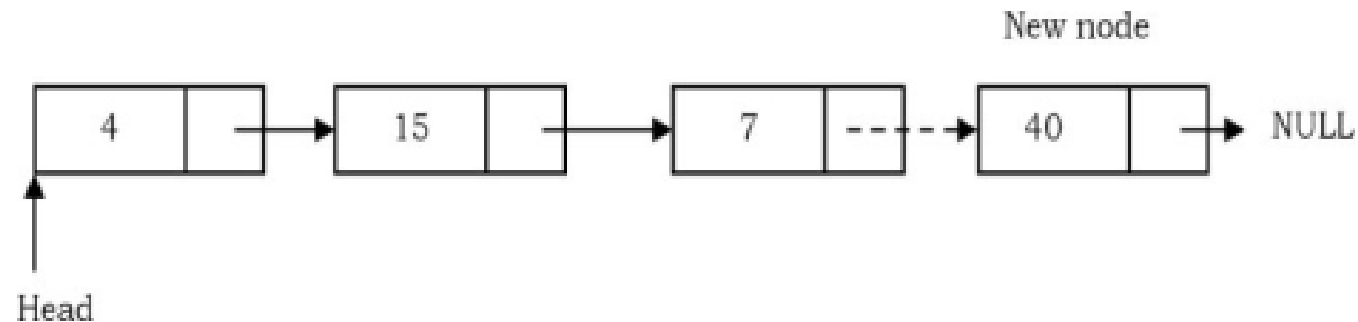




# Preview – Menambah Node Baru di Akhir Linked List



- Last nodes next pointer points to the new node.



# Menyisipkan Node Setelah Node Tertentu



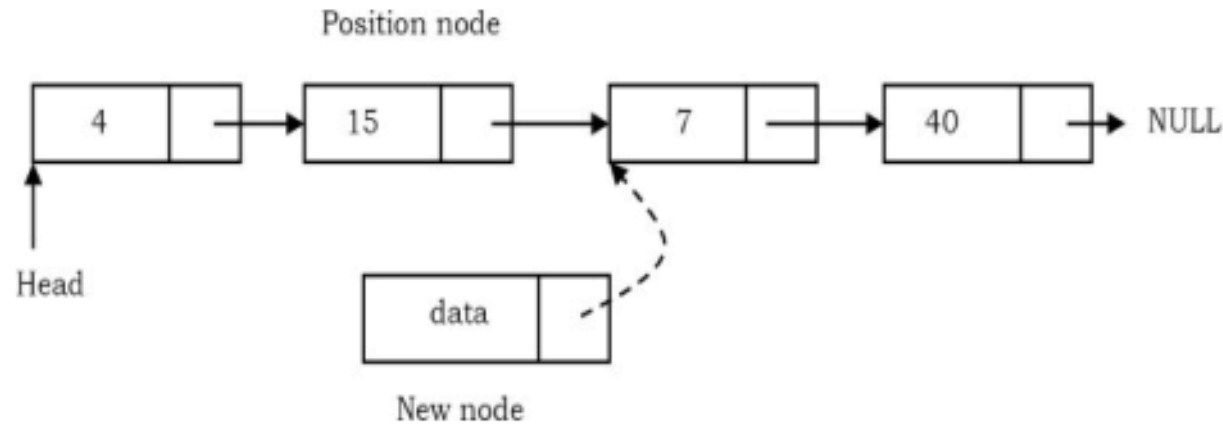
```
1 def insert_after_item(self, x, data):
2     n = self.start_node
3     print(n.ref)
4     while n is not None:
5         if n.item == x:
6             break
7         n = n.ref
8     if n is None:
9         print("item not in the list")
10    else:
11        new_node = Node(data)
12        new_node.ref = n.ref
13        n.ref = new_node
```

Pencarian node yang bernilai X tidak boleh dilakukan oleh first node. Di sini dilakukan oleh n

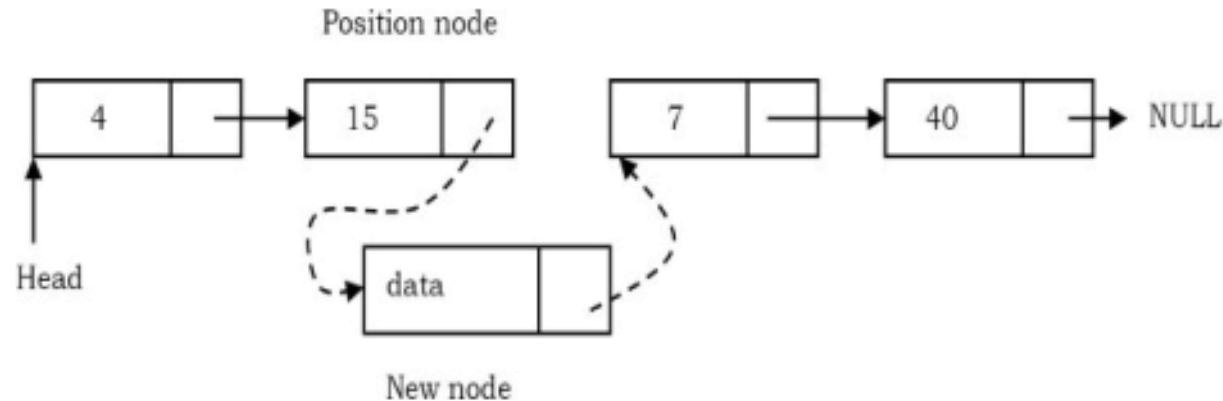
Pencarian node yang mengandung x

Menyisipkan node data setelah node x

# Menyisipkan Node Setelah Node Tertentu



- Position node's next pointer now points to the new node.



# Menambah Data Sebelum Node Tertentu



```
1 def insert_before_item(self, x, data):
2     if self.start_node is None:
3         print("List has no element")
4         return
5
6     if x == self.start_node.item:
7         new_node = Node(data)
8         new_node.ref = self.start_node
9         self.start_node = new_node
10    return
```

Pengecekan apakah  
linked list kosong

Jika data  
disisiplan  
sebelum start  
node

# Menambah Data Sebelum Node Tertentu



```
1  n = self.start_node
2  print(n.ref)
3  while n.ref is not None:
4      if n.ref.item == x:
5          break
6      n = n.ref
7  if n.ref is None:
8      print("item not in the list")
9  else:
10     new_node = Node(data)
11     new_node.ref = n.ref
12     n.ref = new_node
```

Pencarian node  
yang bernilai x

Penyisipan data  
sebelum node x

# Menghapus Node di Awal dan Akhir



```
1 def delete_at_start(self):
2     if self.start_node is None:
3         print("The list has no element to delete")
4         return
5     self.start_node = self.start_node.ref
6
7 def delete_at_end(self):
8     if self.start_node is None:
9         print("The list has no element to delete")
10        return
11
12    n = self.start_node
13    while n.ref.ref is not None:
14        n = n.ref
15    n.ref = None
```

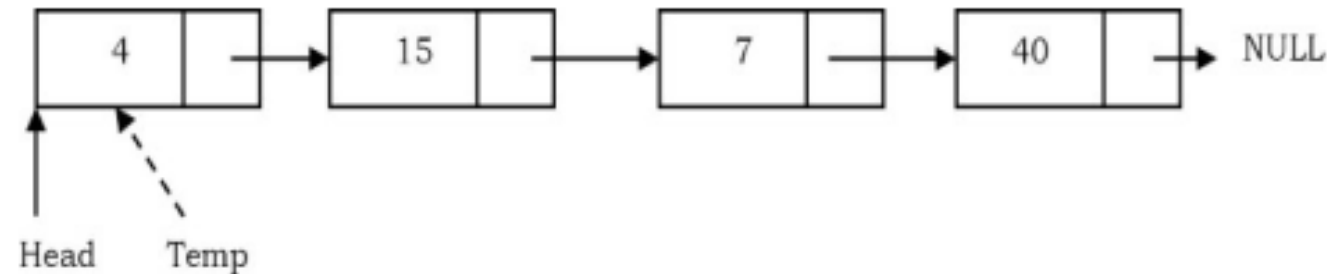
Penghapusan node di awal

Pengecekan apakah list kosong

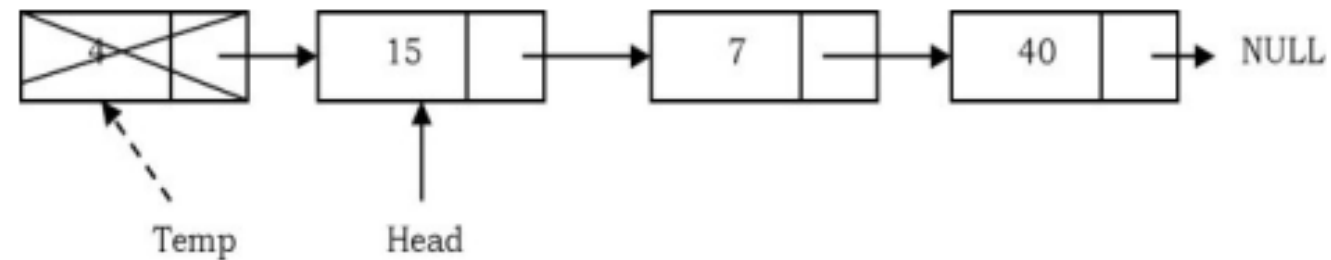
Penghapusan node di akhir

# Menghapus Node di Awal

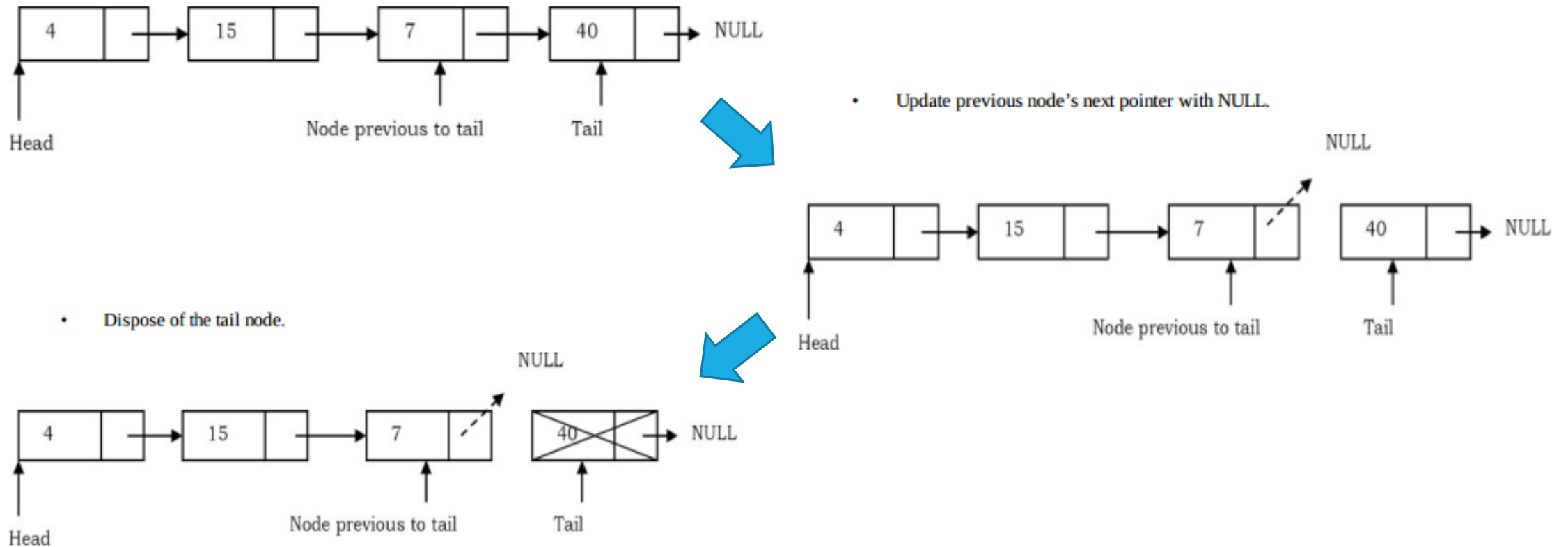
- Create a temporary node which will point to the same node as that of head.



- Now, move the head nodes pointer to the next node and dispose of the temporary node.



# Menghapus Node di Akhir







# Fungsi Count Jumlah Node

---

```
1 def get_count(self):
2     if self.start_node is None:
3         return 0;
4     n = self.start_node
5     count = 0;
6     while n is not None:
7         count = count + 1
8         n = n.ref
9     return count
```

# Memeriksa Keberadaan Node yang Memiliki Nilai Tertentu



```
1 def search_item(self, x):
2     if self.start_node is None:
3         print("List has no elements")
4         return
5     n = self.start_node
6     while n is not None:
7         if n.item == x:
8             print("Item found")
9             return True
10        n = n.ref
11    print("item not found")
12    return False
```

# Menghapus Node yang Memiliki Nilai Tertentu



```
1 def delete_element_by_value(self, x):
2     if self.start_node is None:
3         print("The list has no element to delete")
4         return
5
6     # Deleting first node
7     if self.start_node.item == x:
8         self.start_node = self.start_node.ref
9     return
```

Pengecekan list  
kosong

Penghapusan  
pada elemen  
pertama

# Menghapus Node yang Memiliki Nilai Tertentu



```
1  n = self.start_node
2  while n.ref is not None:
3      if n.ref.item == x:
4          break
5      n = n.ref
6
7      if n.ref is None:
8          print("item not found in the list")
9      else:
10         n.ref = n.ref.ref
```

Pencarian node yang posisinya sebelum node yang akan dihapus

Menghapus node



# Latihan

---

- Buatlah sebuah linked list untuk menyimpan data bertipe integer
- Simpanlah 3,1, 65, 3,10, 9, 12, 5 ke linked list tersebut. Data 1 – 4 dimasukkan menggunakan tambah di depan. Data 5 – 8 dimasukkan tambah di belakang
- Tambahkan data 40 di urutan 4 dan data 13 di urutan 7
- Hapus 1 data di depan dan satu data di belakang
- Hapus data yang bernilai 65



# Tugas

---

- Buatlah method untuk mencari banyaknya anggota sebuah linked list
- Buatlah method untuk menghitung jumlah (sum) semua bilangan di sebuah linked list
- Buatlah method untuk menyambung dua buah linked list menjadi sebuah linked list. Pengguna bisa menentukan susunannya apakah linkedlist1linkedlist2 atau linkedlist2linkedlist1
- Buatlah method untuk menambahkan data ke linkedlist sedemikian hingga data yang ada di linkedlist dalam kondisi terurut
- Buatlah method untuk membalik arah semua pointer di linkedlist

Petunjuk :

1. Kerjakan tugas berikut dg menggunakan python!
2. Kirim dalam format zip!  
dengan nama file : **NIM\_Nama.zip**  
ke : <https://unej.id/SD07tugas>
3. Deadline : **Minggu, 11 April 2021 pukul 23.59**

# Terima Kasih!



---

MUHAMMAD 'ARIFUL FURQON S.PD., M.KOM.

FAKULTAS ILMU KOMPUTER

UNIVERSITAS JEMBER