

Latihan Linked List

1. Jelaskan pengertian Linked List !

Jawaban :

Linked list adalah struktur data linier, di mana elemen tidak disimpan di lokasi memori yang berdekatan. Elemen dalam linked list ditautkan menggunakan pointer. Dengan kata sederhana, daftar tertaut terdiri dari node di mana setiap node berisi bidang data dan referensi (tautan) ke node berikutnya dalam list.

2. Jelaskan fungsi dan kegunaan linked list !

Jawaban :

Fungsi Linked List :

- Linked list dapat digunakan untuk mengimplementasikan struktur data lain seperti stack, queue, graf, dll.
- Digunakan untuk melakukan operasi aritmatika pada bilangan long integer
- Dipakai untuk representasi matriks rongga.
- Digunakan dalam alokasi file yang ditautkan.
- Membantu dalam manajemen memori.

Kegunaan/Penerapan Linked List

- Linked list digunakan dalam penjadwalan Round-Robin untuk melacak giliran dalam permainan multi-pemain.
- Digunakan dalam aplikasi penampil gambar. Gambar sebelumnya dan berikutnya ditautkan, sehingga dapat diakses oleh tombol prev dan next.
- Dalam playlist musik, lagu yang sedang diputar ditautkan ke lagu sebelumnya dan berikutnya.

3. Jelaskan kelebihan dan kelemahan linked list !

Jawaban :

Kekurangan	Kelebihan
Diperlukan ruang tambahan untuk menyatakan/tempat field pointer	Jenis data yang berbeda dapat di-link
Diperlukan waktu yang lebih banyak untuk mencari suatu node dalam linked list	Operasi remove atau insert hanya dilakukan dengan mengubah pointer-nya saja

--	--

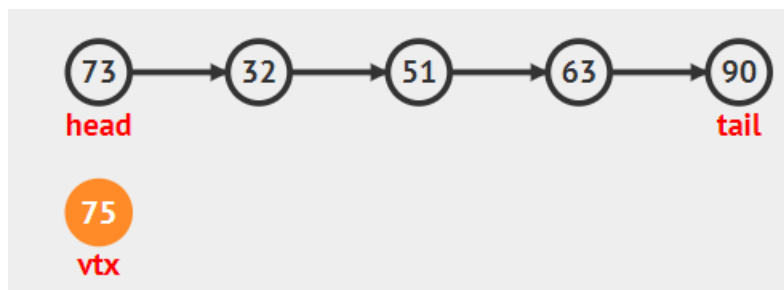
4. Sebut dan jelaskan apa saja operasi di dalam linked list !

Method	Description
<u>add(int index, E element)</u>	Metode ini Menyisipkan elemen yang ditentukan pada posisi yang ditentukan dalam daftar ini.
<u>add(E e)</u>	Metode ini Menambahkan elemen yang ditentukan ke akhir daftar ini.
<u>addFirst(E e)</u>	Metode ini Menyisipkan elemen yang ditentukan di awal daftar ini.
<u>addLast(E e)</u>	Metode ini Menambahkan elemen yang ditentukan ke akhir daftar ini.
<u>element()</u>	Metode ini mengambil tetapi tidak menghapus, kepala (elemen pertama) dari daftar ini.
<u>get(int index)</u>	Metode ini mengembalikan elemen pada posisi yang ditentukan dalam daftar ini.
<u>getFirst()</u>	Metode ini mengembalikan elemen pertama dalam daftar ini.
<u>getLast()</u>	Metode ini mengembalikan elemen terakhir dalam daftar ini.
<u>remove()</u>	Metode ini mengambil dan menghapus kepala (elemen pertama) dari daftar ini.
<u>remove(int index)</u>	Metode ini menghapus elemen pada posisi yang ditentukan dalam daftar ini.
<u>remove(Object o)</u>	Metode ini menghapus kemunculan pertama elemen yang ditentukan dari daftar ini jika ada.
<u>removeFirst()</u>	Metode ini menghapus dan mengembalikan elemen pertama dari daftar ini.
<u>removeLast()</u>	Metode ini menghapus dan mengembalikan elemen terakhir dari daftar ini.
<u>size()</u>	Metode ini mengembalikan jumlah elemen dalam daftar ini.
<u>toString()</u>	Metode ini mengembalikan string yang berisi semua elemen dalam daftar ini dalam urutan yang tepat (dari elemen pertama hingga terakhir), setiap elemen dipisahkan oleh koma dan String diapit dalam tanda kurung siku.

5. Gambar&jelaskan cara addFirst & AddLast !

addFirst

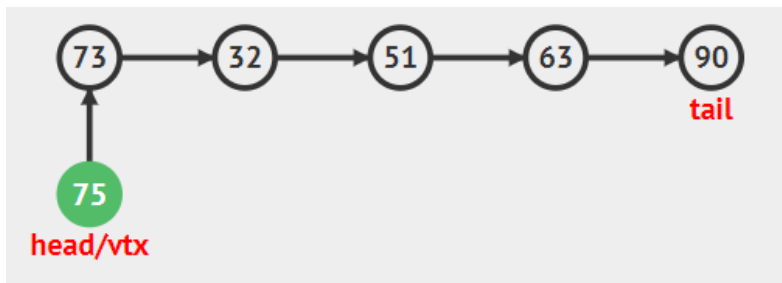
Buat vertex baru untuk menyimpan nilai 75,



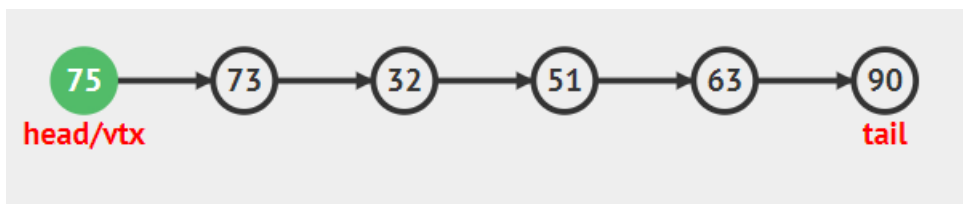
Kemudian operasi next akan menunjuk ke head saat ini,



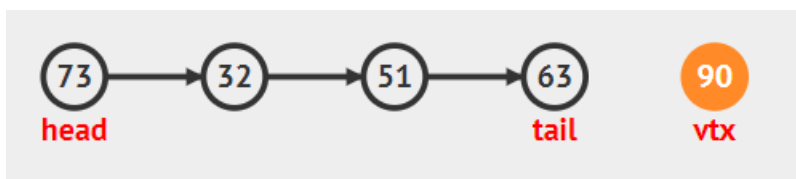
Head menunjuk ke vtx



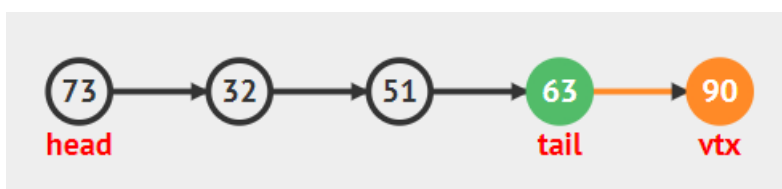
Tata letak ulang linked list untuk visualisasi (bukan dalam linked list yang sebenarnya). Seluruh proses adalah $O(1)$



Buat vertex baru untuk menyimpan nilai 75

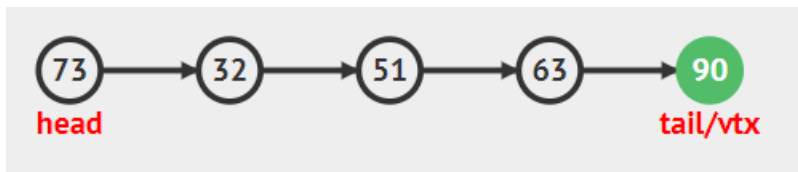


next saat ini menunjuk ke vtx, Tail menunjuk ke vtx.



Seluruh operasi adalah $O(1)$ jika kita mempertahankan penunjuk ekor.

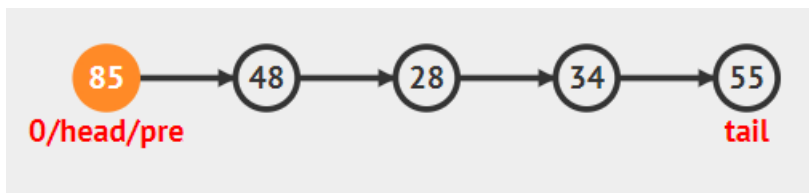
addLast



6. Gambar & jelaskan cara addMid & removeMid!

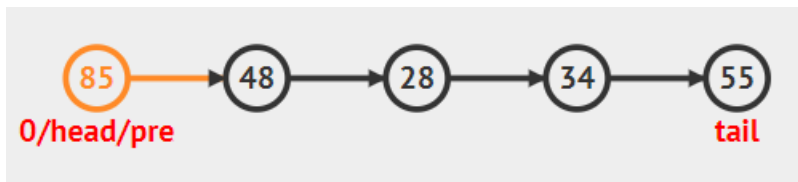
addMid

Atur temp1 ke head.



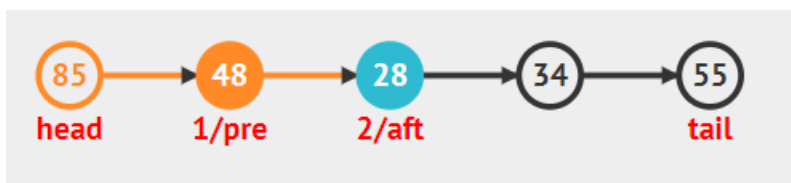
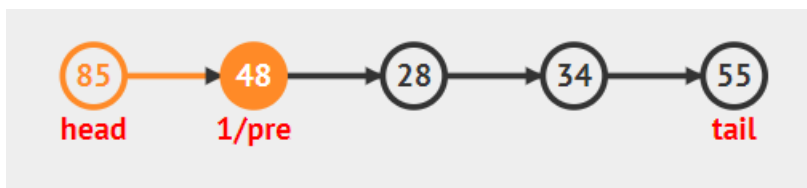
Kami telah menemukan titik penyisipan.

Kami melanjutkan langkah penyisipan berikutnya.

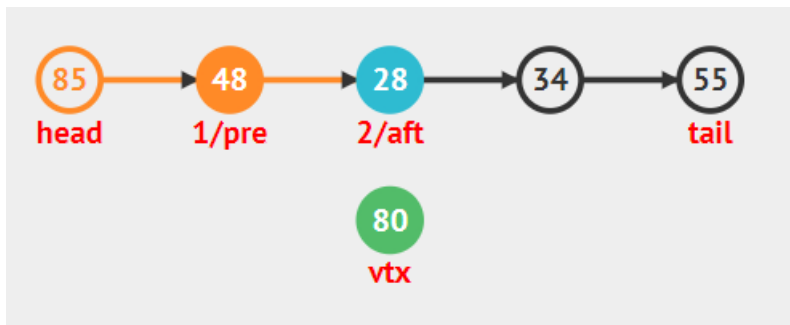


Indeks sebelum titik penyisipan ditemukan.

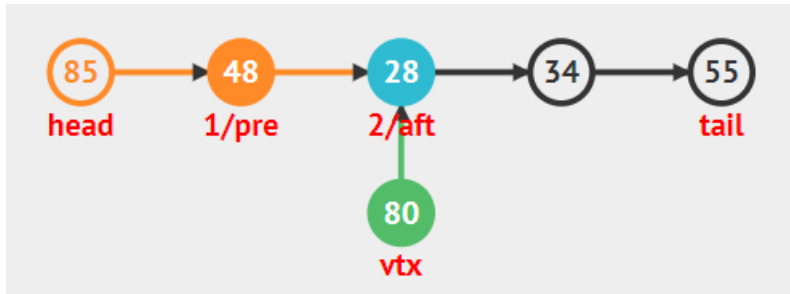
pre adalah sebelum titik penyisipan dan buritan adalah titik penyisipan.



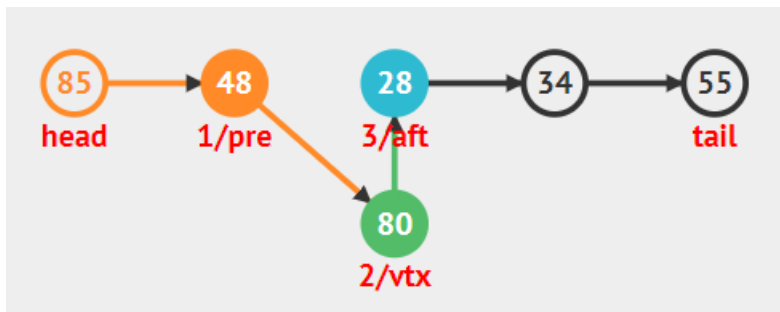
Buat vertex baru untuk menyimpan nilai 80



next saat ini menunjuk ke vtx,

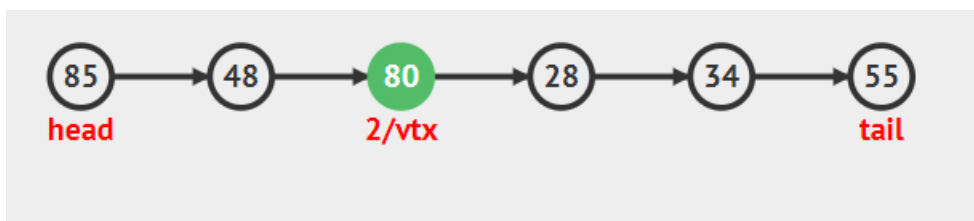


Tail menunjuk ke vtx.



Tata letak ulang Daftar Tertaut untuk visualisasi (bukan dalam Daftar Tertaut yang sebenarnya).

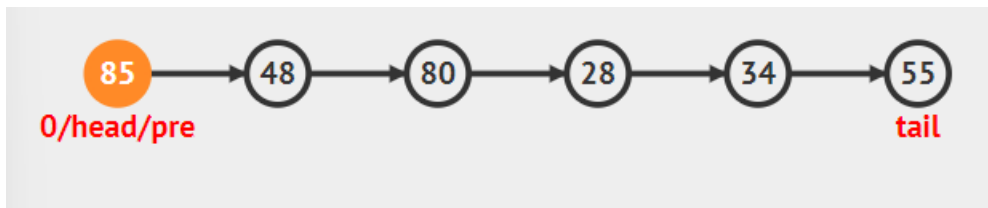
Seluruh proses adalah $O(N)$.



removeMid

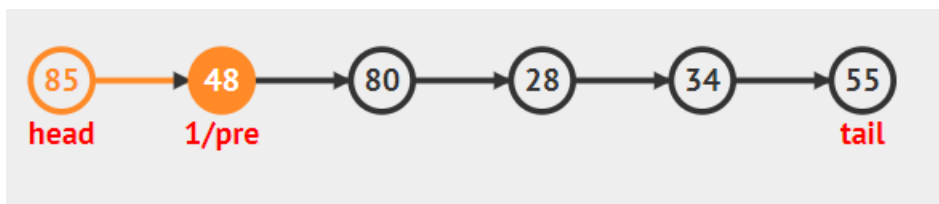
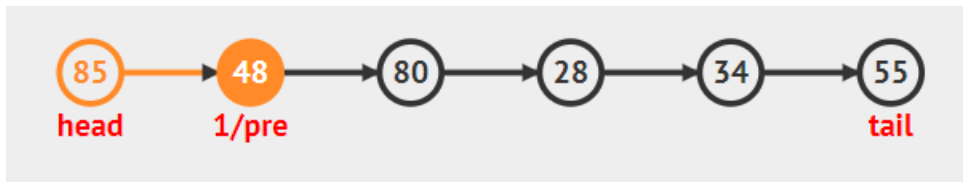
Atur prev ke head.

Pointer prev akan berhenti di satu vertex sebelum vertex yang dihapus.



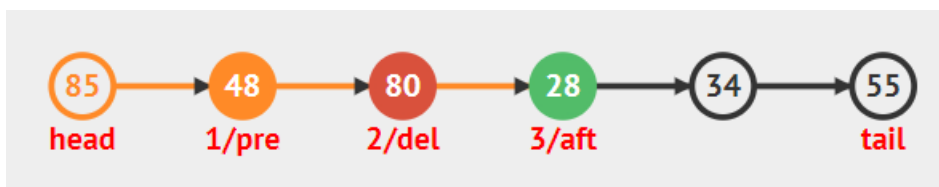
k adalah 0, sebelumnya sekarang menunjuk ke satu titik di belakang verteks yang akan dihapus.

Kami berhenti mencari dan melanjutkan penghapusan.

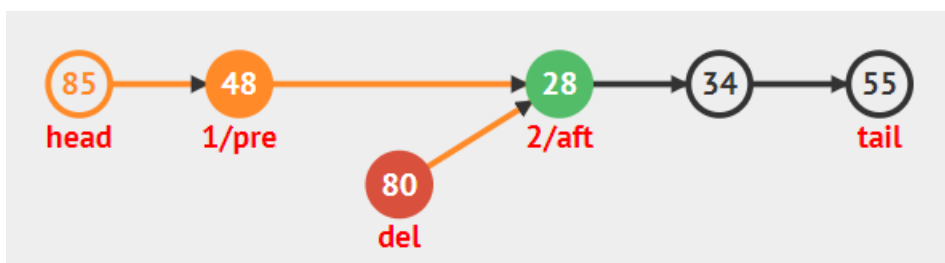


Kami menyimpan referensi ke verteks yang akan dihapus.

Kami juga menyimpan referensi ke verteks setelah verteks yang akan dihapus



Kami menghubungkan vertex di belakang vertex yang akan dihapus (pointer prev) dengan vertex berikutnya setelah vertex yang akan dihapus (pointer after).



Sekarang kita menghapus vertex ini



