Nama: Joshua Taftfarel Petra Chrisensa

NPM : 232310002

Kelas : TI-23-PA2

Mata Kuliah : Lab. Algoritma &  Struktur data

Linked List

Linked list adalah [strukur data](https://www.trivusi.web.id/2022/06/mengenal-struktur-data.html" \t "_blank) linier berbentuk rantai simpul di mana setiap simpul menyimpan 2 item, yaitu nilai data dan pointer ke simpul elemen berikutnya. Berbeda dengan [array](https://www.trivusi.web.id/2022/07/struktur-data-array.html), elemen linked list tidak ditempatkan dalam alamat memori yang berdekatan melainkan elemen ditautkan menggunakan pointer. Simpul pertama pada linked list disebut dengan head. Jika linked list berisi elemen kosong maka akan disebut dengan null. Linked list biasa digunakan untuk membuat file system, adjacency list dan hash table.

**Jenis-Jenis Linked List**

1. **Singly Linked List**

Linked list unidirectional, linked list yang melintas dalam satu arah (dari simpul kepala sampai simpul ekor).

1. **Doubly Linked List**

Linked list bidirectional, linked list yang melintas secara dua arah dan berisi satu pointer tambahan disebut previous pointer.

1. **Circular Linked List**

Linked list unidirectional, sama seperti singly linked list namun pada linked list ini simpul akhir akan menunjuk pada simpul kepala.

1. **Circular Doubly Linked List**

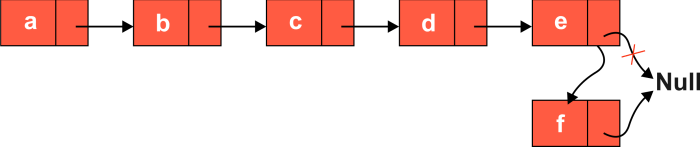
Gabungan dari dua linked list yaitu doubly linked list dan circular linked list, linked list ini memiliki pointer tambahan seperti doubly linked list, dan simpul terakhir mengarah pada simpul kepala.

**Karakteristik Linked List**

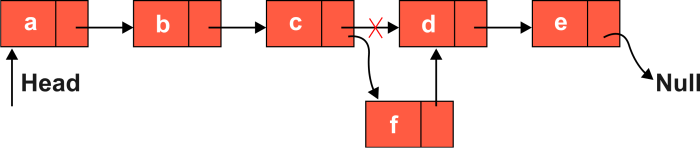
* Sebuah linked list memiliki beberapa karakteristik sebagai berikut:
* Linked list menggunakan memori tambahan untuk menyimpan link (tautan)
* Untuk inisialiasi awal linked list, kita tidak perlu tahu ukuran dari elemen.
* Linked list umumnya dapat digunakan untuk mengimplementasikan struktur data lain seperti stack, & queue
* Simpul pertama dari linked list disebut sebagai Head.
* Pointer setelah simpul terakhir selalu bernilai NULL
* Dalam struktur data linked list, operasi penyisipan dan penghapusan dapat dilakukan dengan mudah
* Tiap-tiap simpul dari linked list berisi pointer atau tautan yang menjadi alamat dari simpul berikutnya
* Linked list bisa menyusut atau bertambah kapan saja dengan mudah.

**Operasi pada Linked List**

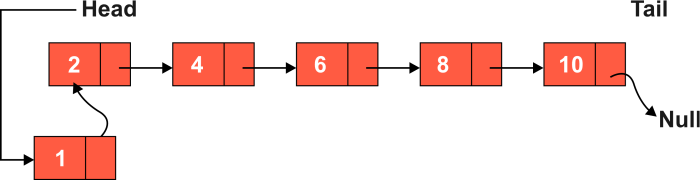
* **Insertion** – Menambahkan elemen baru ke linked list. Operasi ini dapat digunakan untuk menambahkan elemen didepan dari elemen yg sudah ada.
  + Setelah node/elemen terakhir



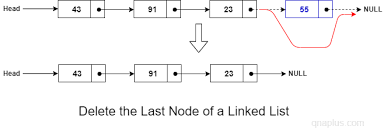
* + Sesudah node/elemen



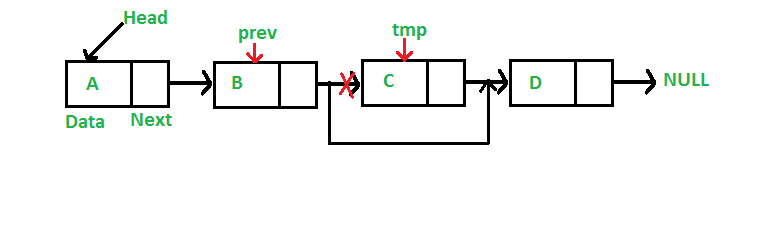
* + Diawal linked list



* **Delection** – Menghapus Elemen yang ada pada linked list. Operasi ini dapat digunakan untuk menghapus elemen baik itu di depan, belakang atau pun di node lain. 2 methode untuk menghapus ialah dengan inisial node dan dengan node terakhir.
  + Metode elemen terakhir



* + Metode Inisial



* **Travesal** – Mengakses setiap elemen dari linked list.
* **Searching** – Mencari/Menemukan simpul pada linked list.
* **Sorting** – Mengurutkan simpul dari struktur linked list

**Fungsi dari Linked List**

* Digunakan untuk mengimplementasikan struktur data lain seperti antrian, dan tumpukan.
* Digunakan untuk meakukan operasi aritmatika pada bilangan long int.
* Dipakai untuk representasi matriks rongga.
* Membantu manajemen memori.

**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.

**Kelebihan Linked List**

* **Struktur data dinamis :** dapat bertambah atau berkurang Ketika runtime dengan mengalokasikan dan membatalkan alokasi.
* **Tidak boros memori :** pemanfaatan memori yang efesien.
* **Implementasi :** Struktur data linear seperti stack dan queue seringkali mudah diimplementasikan.
* **Operasi penyisipan dan penghapusan :** tidak perlu menggeser elemen setelah operasi penyisipan atau penghapusan elemen, hanya alamat yang ada di pointer berikutnya saja yang perlu diperbarui.

**Kelemahan Linked List**

* **Penggunaan memori :** memerlukan lebih banyak memori, karena , pointer juga perlu menyimpan alamat elemen berikutnya dan membutuhkan memori tambahan untuk dirinya sendiri.
* **Traversal :** lebih banyak makan waktu dibandingkan array, karena tidak bisa akses langsung.
* **Reverse traversal :** dalam single linked list, reverse traversing tidak dimungkinkan, tetapi dalam kasus double-linked list, ini dapat dimungkinkan karena berisi pointer ke node yang terhubung sebelumnya dengan setiap node. Untuk melakukannya, diperlukan memori tambahan untuk pointer sebelumnya sehingga ada pemborosan memori.
* **Akses acak :** akses acak tidak bisa dilakukan dalam linked list karena alokasi memorinya yang dinamis.

**Perbedaan Array dengan Linked List**

|  |  |
| --- | --- |
| Array | Linked List |
| Ukuran pada array bersifat tetap | Ukuran pada linked list dapat berubah-ubah atau dinamis |
| Dapat mengakses random elemen | Tidak dapat mengakses random elemen |
| Sulit menambahkan elemen baru | Mudah untuk menambahkan elemen baru |
| Tidak ada ruang tambahan untuk pointer berikutnya | Terdapat ruang/area tambahan untuk pointer berikutnya |
| Elemen relative tertutup dan berdekatan | Elemen-elemennya tidak berdekatan |

<https://www.trivusi.web.id/2022/07/struktur-data-linked-list.html>

<https://www.javatpoint.com/linked-list-data-structure-in-cpp-with-illustration>

<https://www.softwaretestinghelp.com/linked-list/>

<https://youtu.be/VVemCxif9vg?si=GLLcXEuracbRyUHK>