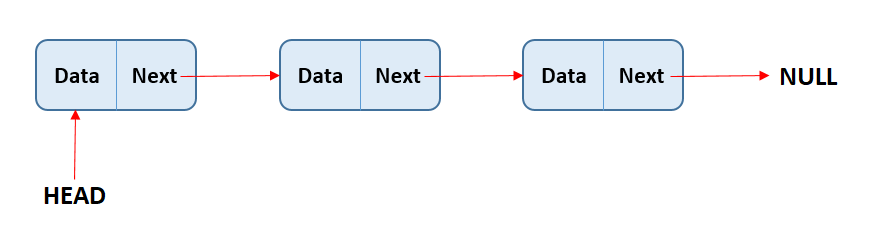
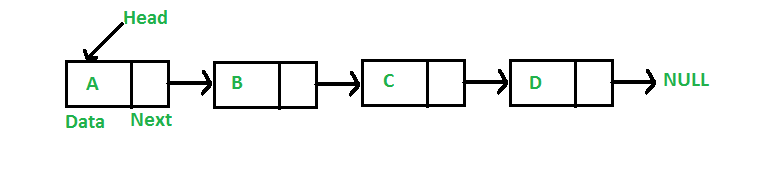
**Pengertian Linked List**

Linked list adalah suatu kumpulan dari beberapa elemen sejenis dari suatu struktur data linier berbentuk simpul rantai dengan 2 item di tiap simpul. Dalam tiap linked list terdapat 1 atau lenih node, tang tiap node nya berisi info/data dan Di dalam Info/data, berisi informasi mulai dari integer, float, char, dan lain-lain. Juga untuk Link/next berguna untuk menyimpan alamat nde berikutnya.

Berbeda dengan array , elemen linked list tak dapat di posisikan oleh kita dengan alamat memori yang berdekata, melainkan ditautkan atau di cantumkan menggunakan pointer.

Linked list adalah suatu cara untuk menyimppann data dengan struktur sehingga programmer dapat secara otomatis menciptakan suatu tempat baru untuk menyimpan data kapan saja diperlukan.Pula, linked list di kenal juga denagn istilah senarai berantai, karena struktur sata linked list terdiri dari urutan atau record data yang memiliki field untuk memungkinkan berfungsi untuk menyimpan alamat atau referensi field selanjuutunya(yang berada dalam urutan).

Elemen linked list yang dihubungkan dnegan link pada linked list biasa disebut Node. Node dalam linked list pun di bagi menjadi dua, yaitu Head dan juga Tail.

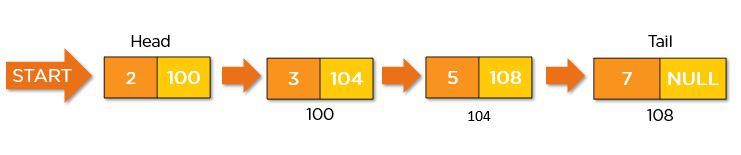
1. Head linked list adalah elemen yang berada di posisi pertama dalam suatu linked list
2. Tail addlah elemen yang berada di posisi terakhir dalam suatu linked list.

**Jenis Jenis Linked List**

Pada umumnya, linked list di bagi mejadi 4 jeins, yaitu **Singly linked, Doubly linked list, Circular linked list,** dan **Circular doubly linked list.**

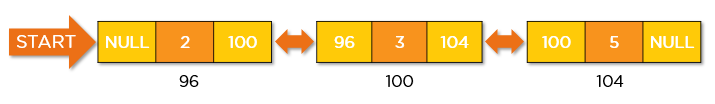
1. Singly linked list

adalah linked list unidirectional, Jadi, kita hanya dapat melintasnya dalam suatu arah, yaituh dari simpul kepala ke simpul ekor.



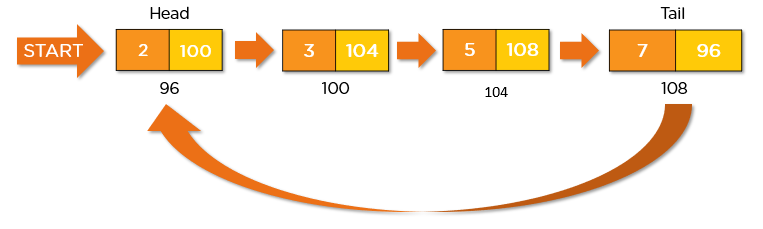
1. Doubly llinked list

adalah linked list unidirectional. Jadi kita bisaa melintasinya dari dua arah. Tak sama dengan singly, simpull doubly linked list berisi 1 pointer yangbiasa disebut previous pointer. Pointer ini menunjuk ke simpul sebelumnya.



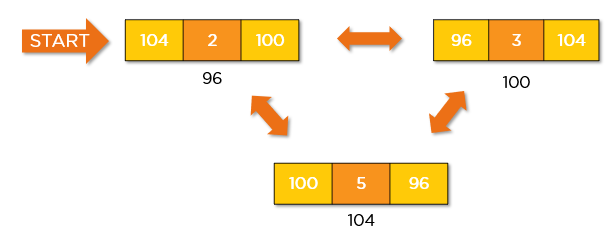
1. Circular linked list

adalah linked list unidirectional. Kita hanay dapat mmelintaskan dalam satu arah. Tetapi jenis linked list ini memiliki simpul terakhir yang menunjukan ke simpul kepala. Jadi saat melintas, kita harus berhati-hati dan berhenti saat mengunjungi kembali simpul kepala.



1. Circular doubly linked list

adalah gabungan dari Doubly linked list dan Circular linked list. Sepeeti Doubly linke list, linked list ini memiliki pointer tambahan yang di sebut previous pointer, dan mirip dengan Circular llinked list, simpul terakhirnya menunjuk pada simpul kepala. Jenis liinke di list ini adalah bidirectional. Jadi, kita bias melintasinya dua arah.



**Karakteristik Linked List**

Sebuah linked list memilki beberapa karakteristik sebagai berikut:

* Linked list menggunkana memimori tambahan untuk save link atau tautan
* Linked bias menyusut atau bertambah kapan saja dengan mudah
* Untuk inisialisasi awal linked list, kita tidak peril tahu ukuran dan elemen
* Linked list umumnya dapat digunakan untuk mengimplementasikan struktur data lain seperti stack, queue, ataupun graf
* Simpul pertama dari linked list disebut sebagai Geaad
* Pointer setelah simpul terakhir selalu bernilai NULL

**Operasi-operasi pada Linked Linked List**

Banyak operasi yang dapt di lakukan dalam linked list. Contoh, seperti operasi insertion, yaitu tindakkan menambahkkan elemen baru ke linked list.

Berikut adalah beberapa daftar operasi dasaar pada linked list:

* **Transversal -** mengaksses setiap elemen dari linked list
* **Insertion -** menambahkan elemen baru ke linked list
* **Delecyion -** menghapus elemen yang ada
* **Searching -** menemukan simpul pada linked list list
* **Sorting –** mengurutkan simpulkan dari struktur linked list

**Fungsi dan Kegunaan Linked List**

Ada juga fungsi dari kegunaann linked list, seperti poin-poin berikut:

* Linked list dapat digunkan untuk mengimplementasikan struktur data lain seperti stack, queue, graf, dan yang lain.
* Digunakan untuk melakukan operasi aritmatikan pada bilangan long integer
* Digunakan untuk representasi matriks rongga
* Digunakan dalam alokasi file
* Membantu dalam manajemen memori.

Penerapan linked list banyak ditemui dalam beberapa kasus berikut:

* 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

* Berikut ini dalah keunggulan menggunakan linked list:
* Struktur data dinamis: Linked list adalah himpunan dinamis sehingga dapat bertambah dan menyusut saat runtime dengan mengalokasikan dan membatalkan alokasi memori. Jadi kita tidak perlu memberikan ukuran awal dari linked list.
* Tidak boros memori: Dalam linked list, pemanfaatan memori yang efisien dapat dicapai karena ukuran linked list bertambah atau berkurang pada runtime sehingga tidak ada pemborosan memori dan tidak perlu mengalokasikan memori sebelumnya.
* Implementasi: Struktur data linier seperti stack dan queue seringkali mudah diimplementasikan menggunakan linked list.
* Operasi penyisipan dan penghapusan: Operasi penyisipan dan penghapusan cukup mudah dalam linked list. Kita tidak perlu menggeser elemen setelah operasi penyisipan atau penghapusan elemen, hanya alamat yang ada di pointer berikutnya saja yang perlu diperbarui.

Kelemahan Linked List

Adapun kelemahan menggunakan linked list adalah sebagai berikut:

* Penggunaan memori: Linked list memerlukan lebih banyak memori dibandingkan dengan array. Karena dalam linked list, pointer juga perlu menyimpan alamat elemen berikutnya dan membutuhkan memori tambahan untuk dirinya sendiri.
* Traversal: Dalam traversal, linked list lebih banyak memakan waktu dibandingkan dengan array. Akses langsung ke elemen tidak bisa dilakukan pada linked list seperti array yang dapat akses elemen berdasarkan indeks. Untuk mengakses sebuah simpul pada posisi n dari linked list, kita harus melintasi semua simpul sebelumnya.
* Reverse Traversing: 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.