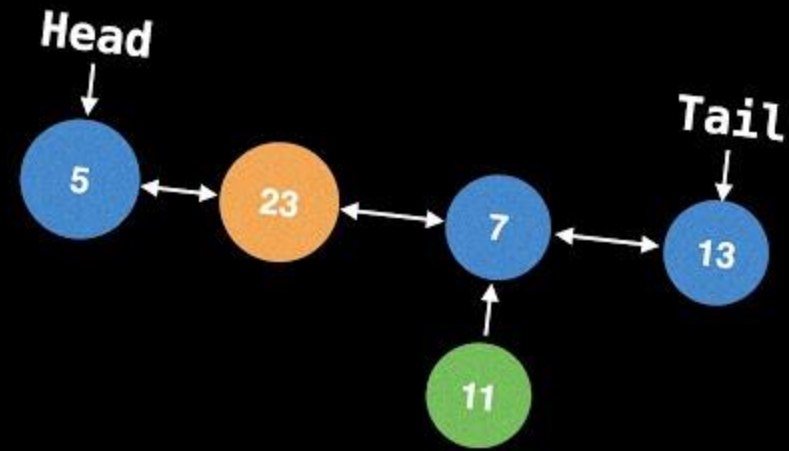
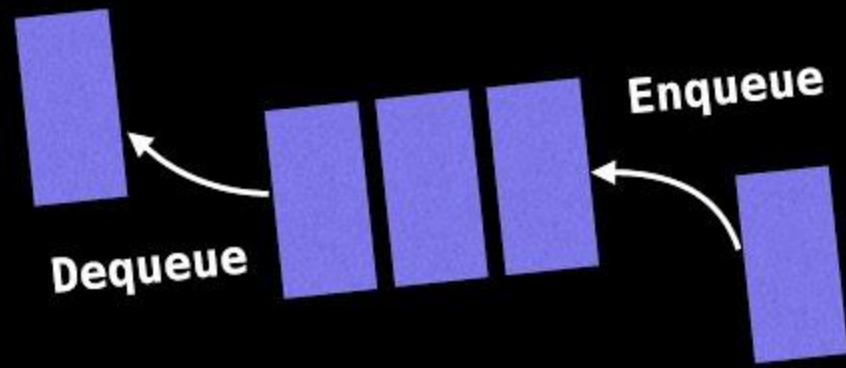
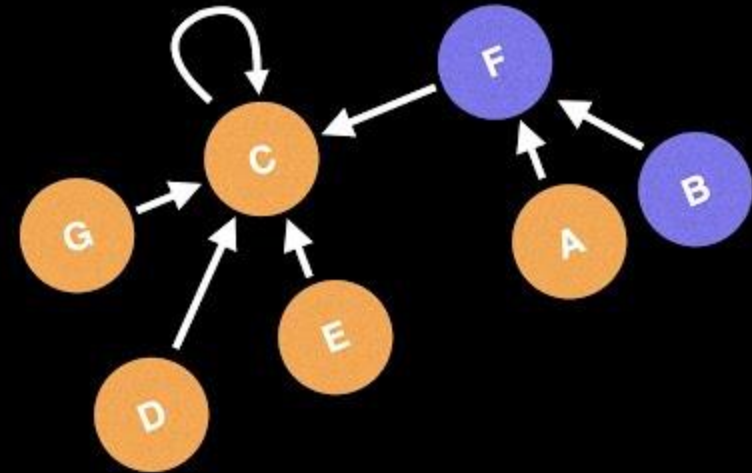
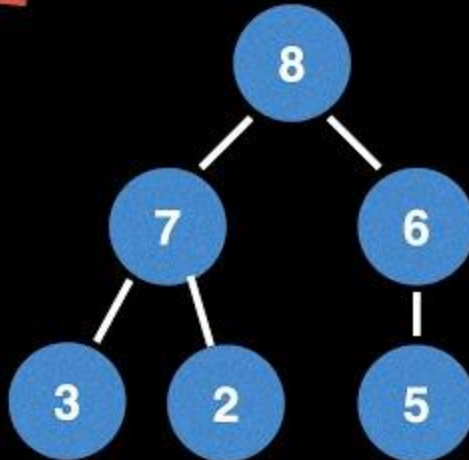
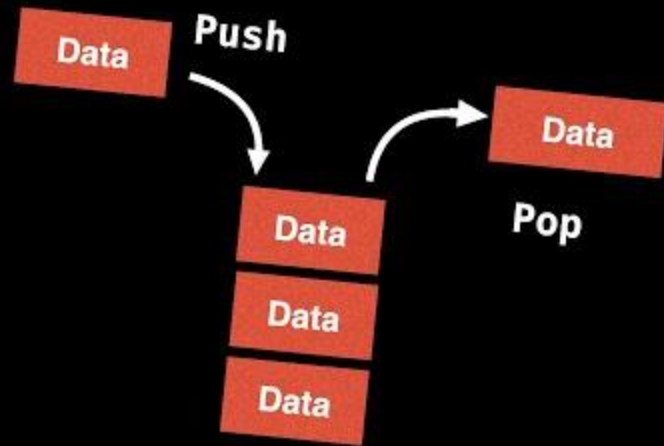


STRUKTUR DATA

Fredian Simanjuntak, S.Kom., M. Kom



Data Structures



Struktur Data

Cara menyimpan, mengambil, dan menyusun data. Dengan ini, data yang ada di komputer lebih mudah diakses dan diperbarui.





STRUKTUR DATA

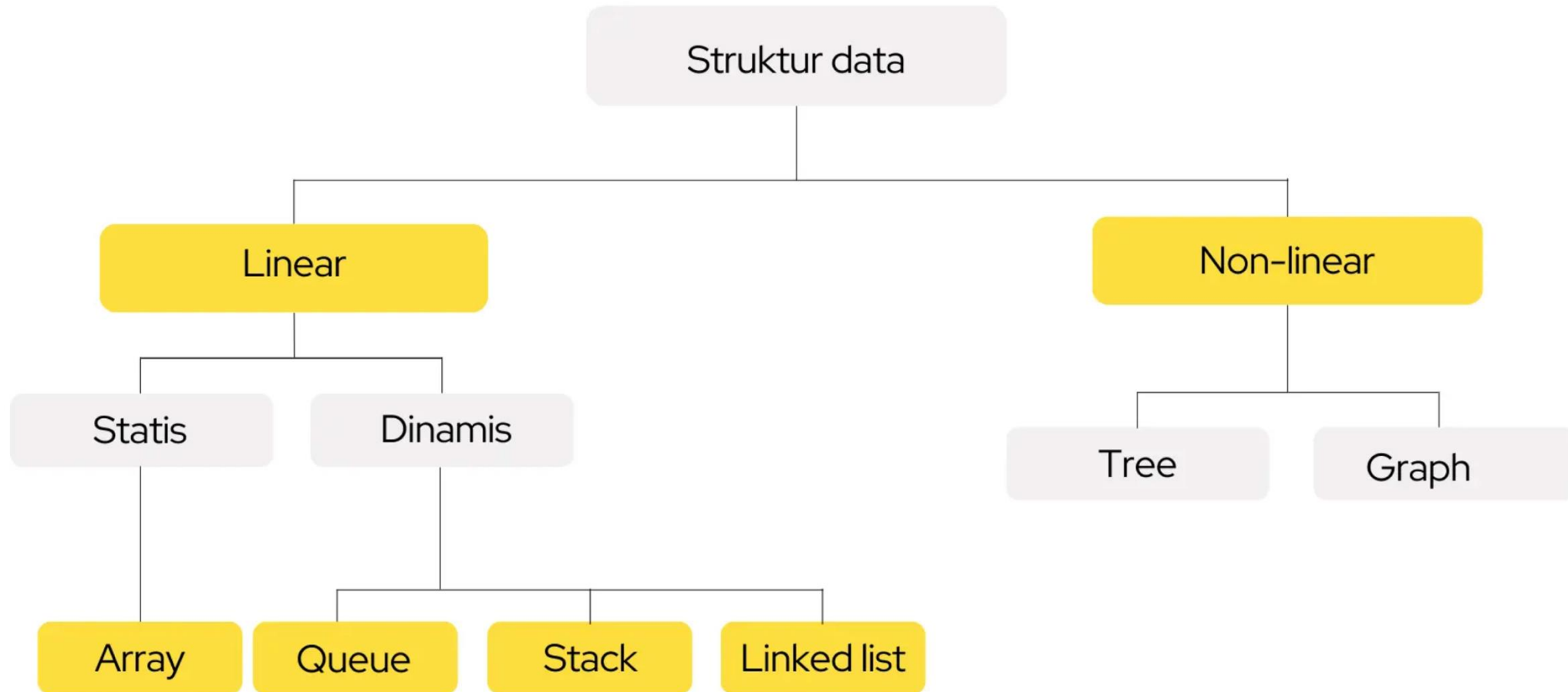
Data bisa berupa angka, huruf dan simbol yang diletakkan pada kolom-kolom tertentu dan disusun dengan sangat baik.

Dalam penyusunan data ini terdapat beberapa istilah yang harus dipahami terlebih dahulu dan istilah tersebut adalah **node** dan **indeks**.

node adalah sebuah elemen yang terdapat pada sebuah struktur yang berisikan pointer ke *node* selanjutnya

indeks adalah sebuah objek yang berada di sistem database yang mampu mempercepat proses pencarian data

KLASIFIKASI STRUKTUR DATA



JENIS STRUKTUR DATA - ARRAY

Array adalah tipe struktur data yang terdiri dari kumpulan elemen dengan tipe yang sama. Setiap elemen dalam array diindeks dengan nomor integer yang mengidentifikasikan posisinya dalam array. Array digunakan untuk menyimpan data berurutan dengan ukuran tetap. Akses ke elemen dalam array dilakukan dengan cepat karena indeksnya langsung menunjuk ke lokasi memori yang sesuai. Array terdiri dari dua jenis, yaitu *one-dimensional array* dan *multi-dimensional array*.

Kelebihan :

- Akses elemen cepat berkat pengindeksan langsung ke lokasi memori.
- Penggunaan memori terpusat karena ukuran array tetap.

Kekurangan :

- Ukuran array tetap, sulit menambah atau menghapus elemen tanpa mengalokasikan ulang memori.
- Penggunaan memori tidak efisien jika array memiliki banyak elemen kosong.

Contoh : Array integer berukuran 5 – [10, 20, 30, 40, 50].

JENIS STRUKTUR DATA - QUEUE

Queue adalah struktur data antrian yang bekerja dengan prinsip **FIFO (First In First Out)**. Elemen pertama yang masuk akan menjadi elemen pertama yang keluar dari antrian. Queue digunakan dalam penjadwalan tugas, manajemen antrian, dan simulasi proses antrian.

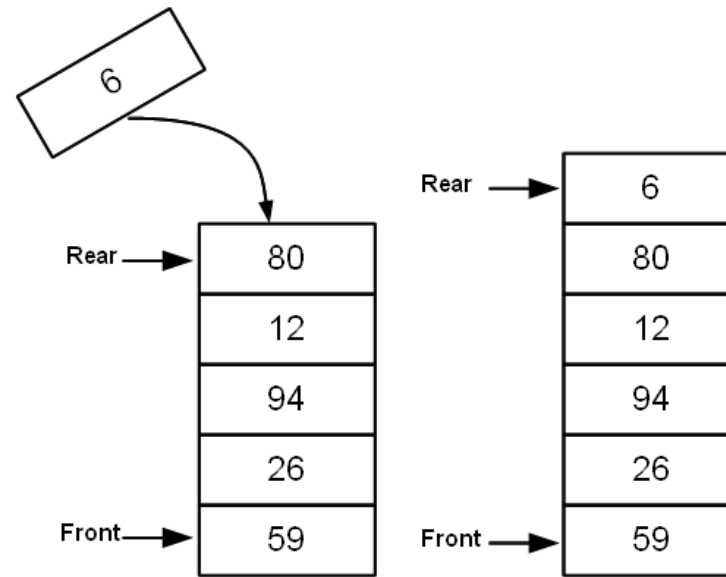
Kelebihan :

- Digunakan dalam penjadwalan tugas dan manajemen antrian.
- Mengatur proses berbasis antrian secara efisien.

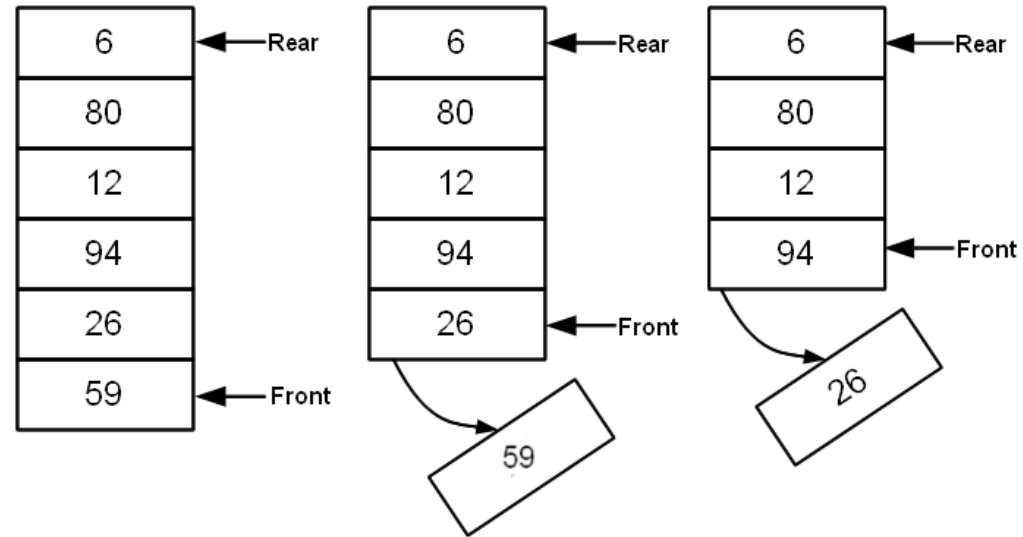
Kekurangan :

- Akses elemen di tengah antrian tidak dapat dilakukan secara langsung.
- Ukuran antrian terbatas oleh kapasitas memori.

Contoh : Queue yang berisi elemen [Red, Green, Blue].



Data baru dimasukkan (*insert*) dalam queue



Dua data dikeluarkan (*remove*) dari queue

JENIS STRUKTUR DATA - STACK

Stack adalah struktur data tumpukan yang mengikuti prinsip **LIFO (Last In First Out)**. Elemen terakhir yang ditambahkan akan menjadi elemen pertama yang dihapus. Stack digunakan dalam berbagai konteks, seperti evaluasi ekspresi matematika, penanganan rekursi, dan manajemen panggilan fungsi.

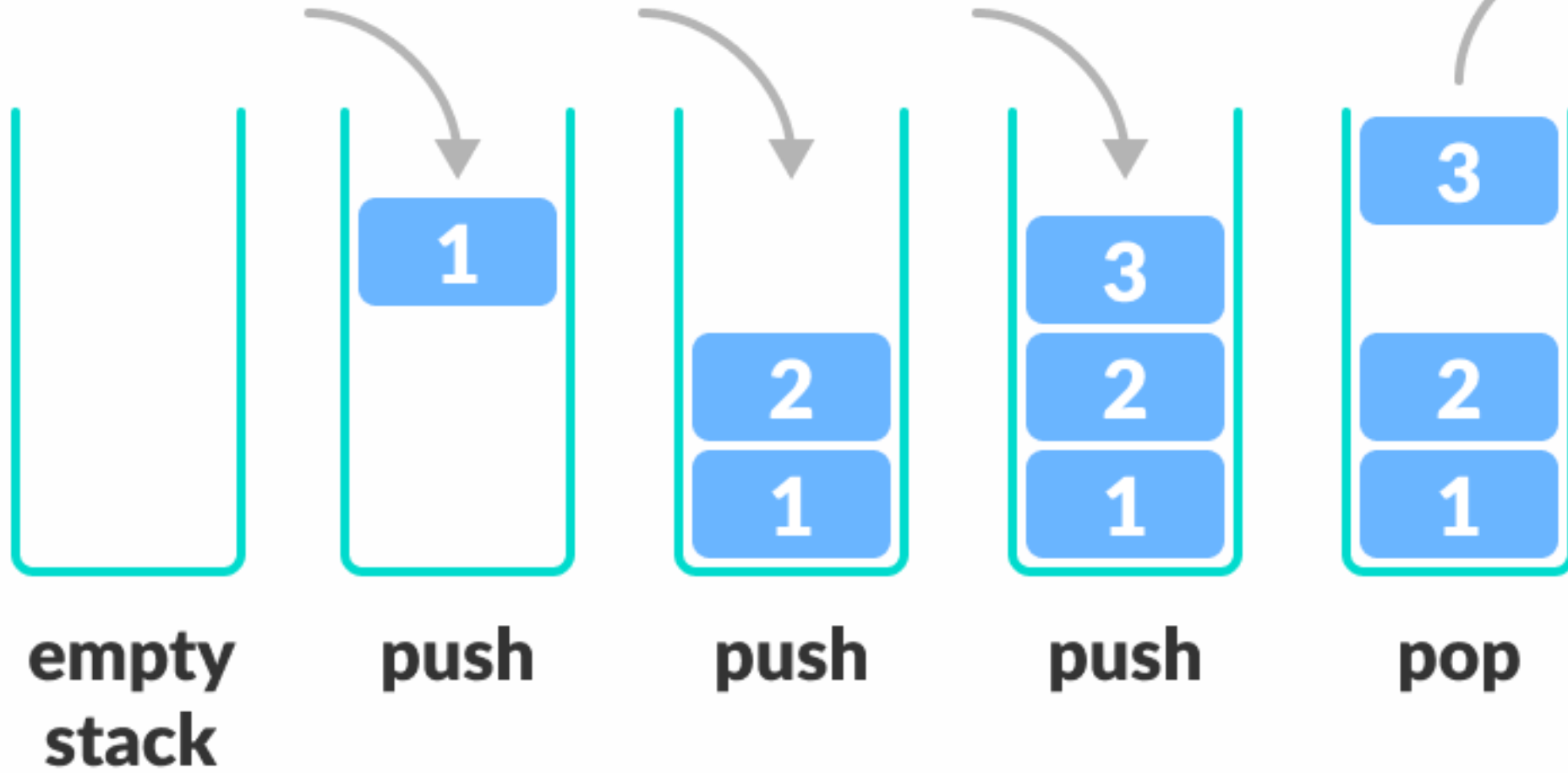
Kelebihan :

- Implementasi sederhana dan efisien untuk menyimpan data sementara.
- Penanganan rekursi dan undo operasi dapat diimplementasikan dengan stack.

Kekurangan :

- Ukuran stack terbatas oleh kapasitas memori.
- Akses elemen di tengah stack tidak dapat dilakukan secara langsung.

Contoh : Stack yang berisi elemen [5, 10, 15].



JENIS STRUKTUR DATA – LINKED LIST

Linked list adalah struktur data dinamis yang terdiri dari simpul-simpul yang saling terhubung melalui pointer. Setiap simpul memiliki data dan pointer ke simpul berikutnya dalam list.

Linked list cocok untuk menyimpan data dengan ukuran dinamis karena dapat dengan mudah menambah atau menghapus elemen tanpa perlu mengalokasikan ulang memori.

Kelebihan :

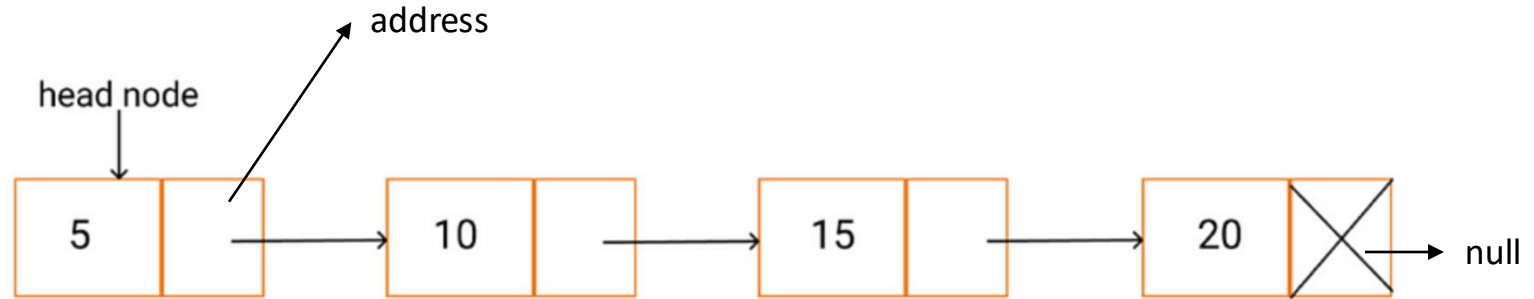
- Mudah menambah dan menghapus elemen tanpa perlu mengalokasikan ulang memori.
- Ukuran list fleksibel karena elemen dapat ditambahkan atau dihapus sesuai kebutuhan.

Kekurangan :

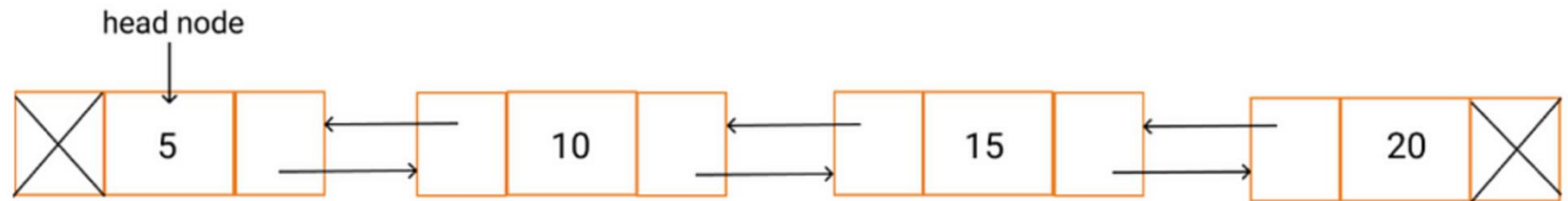
- Akses elemen lebih lambat karena harus melalui traversal dari awal list.
- Membutuhkan memori ekstra untuk menyimpan pointer yang menghubungkan simpul-simpul.

Tipe ini dibagi menjadi tiga jenis, yaitu *doubly linked list*, *singly linked list* dan *circular linked list*.

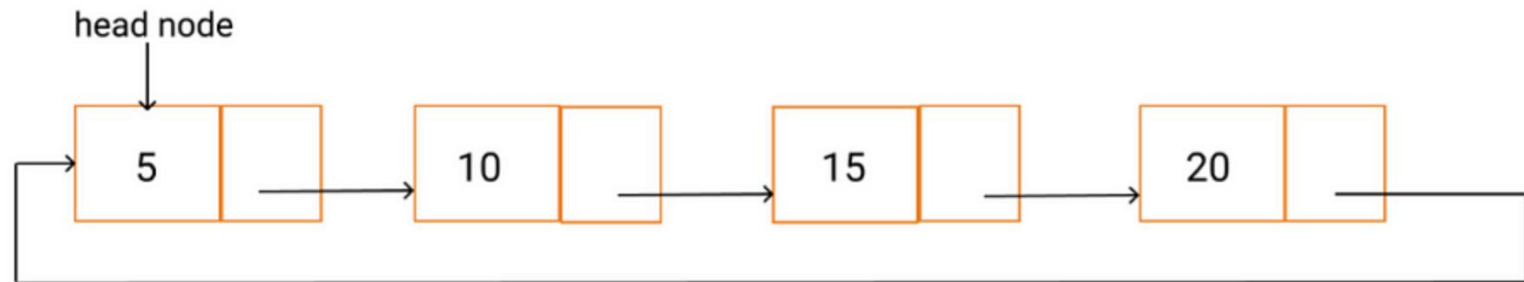
Singly Linked List



Doubly Linked List



Circular Linked List



JENIS STRUKTUR DATA – TREE

Tree adalah struktur data yang terdiri dari simpul (node) yang terhubung dengan tepi (edge) dan membentuk hierarki.

Setiap simpul dalam tree memiliki satu simpul induk, kecuali simpul akar yang tidak memiliki simpul induk. Tree digunakan dalam representasi struktur hirarkis, seperti direktori file dalam sistem operasi dan pengindeksan data.

Node paling bawah disebut sebagai **leaf node** dan *node* paling atas disebut **root node**.

Setiap *node* berisi *pointer* ke titik *node* yang berdekatan.

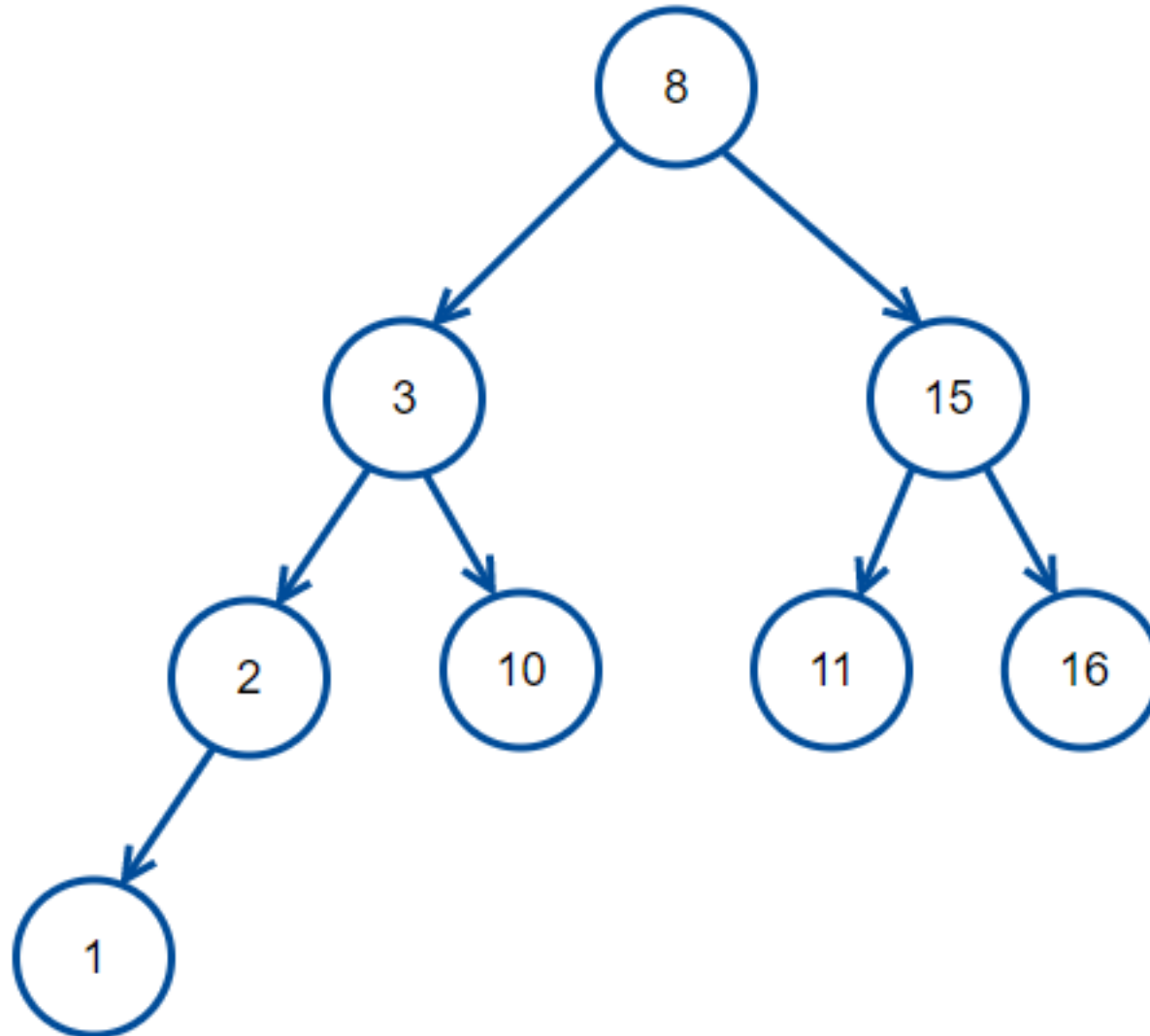
Kelebihan :

- Representasi struktur hirarkis secara efisien.
- Cepat dalam pencarian data terurut dan mengurutkan data.

Kekurangan :

- Operasi dalam tree seperti pencarian dan penyisipan memerlukan traversal yang kompleks.
- Jika tidak seimbang, dapat menyebabkan kinerja buruk.

Trees memiliki beberapa jenis struktur data, seperti *binary tree*, *binary search tree*, *B-tree*, dan *AVL tree*. Di antara jenis-jenis tersebut, ***binary tree* adalah salah satu yang paling sering dipakai.**



JENIS STRUKTUR DATA – GRAPH

Dalam *Graph*, setiap *node* disebut sebagai *vertices*. *Graph* terdiri dari kumpulan *vertices* dan koneksi (*edges*).

Graph digunakan dalam analisis jaringan sosial, algoritma pencarian jalur terpendek, dan permasalahan yang melibatkan relasi antara entitas.

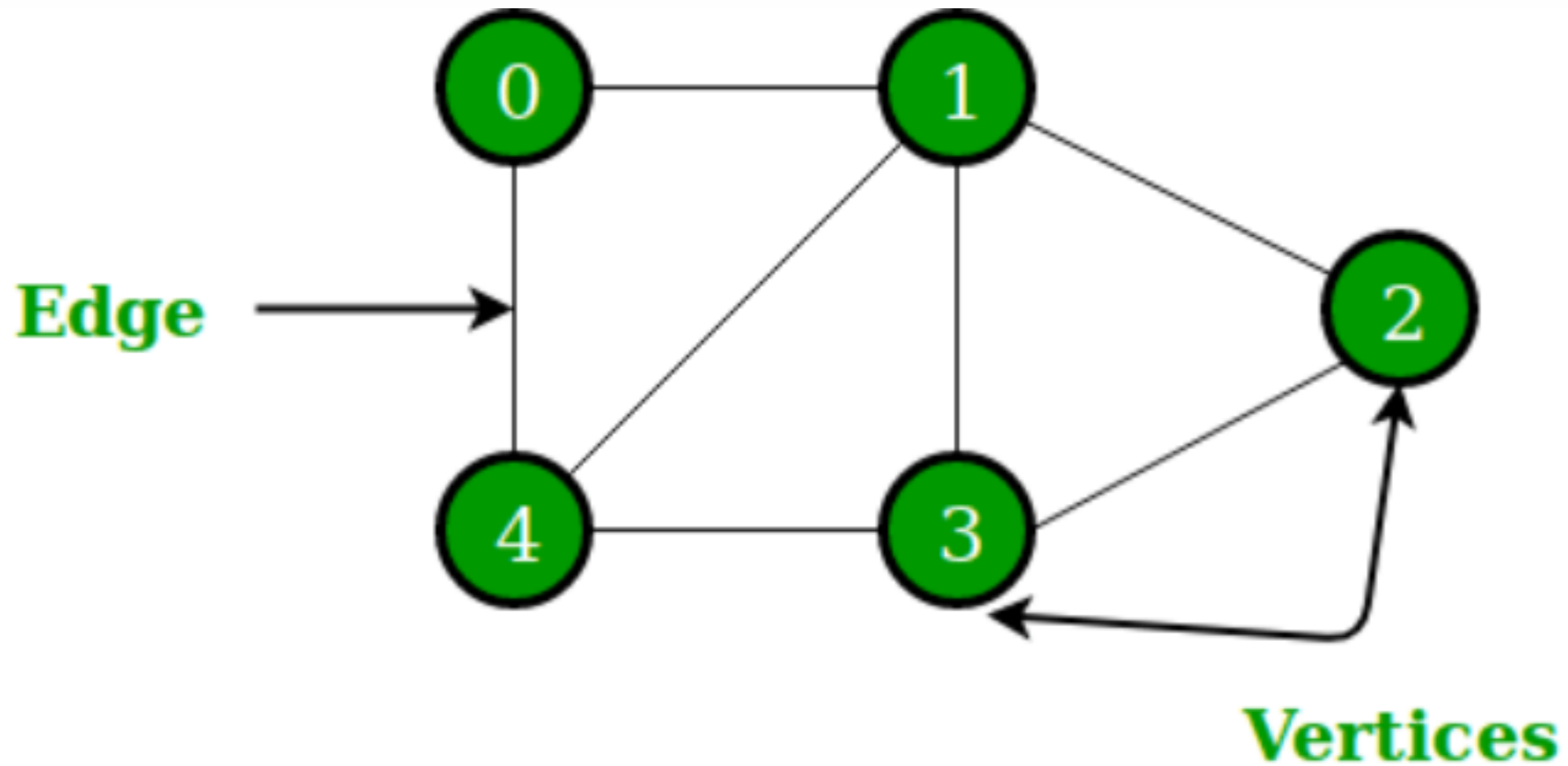
Kelebihan :

- Merepresentasikan hubungan dan koneksi antara entitas dengan baik.
- Digunakan dalam analisis jaringan sosial dan algoritma pencarian jalur terpendek.

Kekurangan :

- Pencarian jalur optimal dalam graph memerlukan algoritma khusus.
- Penggunaan memori besar jika graph besar.

Analogi sederhana dari *graph* adalah *social network*. Manusia diibaratkan sebagai *nodes* dan persahabatan atau hubungan adalah *edges*. Setiap pengenalan manusia dengan manusia lainnya menghasilkan jaringan atau persahabatan baru.



FUNGSI STRUKTUR DATA

1. **Organisasi Data** : Struktur data membantu mengorganisasi data dengan cara yang efisien dan terstruktur. Ini memungkinkan data diakses, dicari, dan dimanipulasi dengan lebih mudah dan cepat.
2. **Efisiensi** : Struktur data dirancang untuk memberikan performa yang optimal dalam berbagai operasi, seperti pencarian, pengurutan, dan penyisipan data.
3. **Penghematan Memori** : Struktur data membantu mengelola alokasi memori dengan lebih efisien. Penggunaan memori yang efisien adalah aspek penting dalam aplikasi yang memerlukan pengolahan data yang besar dan kompleks.
4. **Pengurangan Kompleksitas** : Struktur data membantu mengurangi kompleksitas dalam pemrograman dengan menyediakan metode yang lebih terorganisir dan mudah diimplementasikan untuk menyimpan dan memanipulasi data.
5. **Pencarian dan Pemrosesan Data** : Struktur data menyediakan algoritma dan metode yang efisien untuk mencari dan memproses data. Misalnya, hash table memungkinkan pencarian data dengan waktu konstan $O(1)$, sedangkan pohon biner dapat digunakan untuk pencarian data terurut dengan waktu logaritmik $O(\log n)$.

FUNGSI STRUKTUR DATA

6. **Kemudahan Pengembangan Aplikasi** : Dengan menggunakan struktur data yang tepat, pengembang dapat mengoptimalkan kode dan menyederhanakan algoritma dalam aplikasi. Hal ini membantu dalam menghadapi tantangan dalam pengelolaan data yang kompleks.
7. **Pengelompokan Data** : Struktur data memungkinkan pengelompokan data berdasarkan kriteria tertentu, yang dapat membantu dalam analisis dan visualisasi data.
8. **Representasi Informasi** : Struktur data membantu merepresentasikan informasi dalam bentuk yang lebih mudah dipahami dan dimanipulasi. Contohnya, graf digunakan untuk merepresentasikan hubungan antara entitas dalam bentuk yang visual dan terstruktur.
9. **Reusabilitas** : Penggunaan struktur data yang tepat memungkinkan reusabilitas kode dalam berbagai aplikasi. Hal ini membantu dalam pengembangan aplikasi yang efisien dan menghemat waktu dan usaha dalam pengembangan perangkat lunak.
10. **Manajemen Data** : Struktur data membantu dalam manajemen data dengan cara yang lebih terstruktur dan terorganisir. Dengan pengaturan data yang baik, aplikasi dapat diatur dengan lebih baik dan mudah dikelola.

PENERAPAN STRUKTUR DATA

Penerapan Struktur Data dalam Database :

Dalam basis data, struktur data digunakan untuk menyimpan dan mengelola data dengan efisien. Contoh penerapan adalah penggunaan tabel (array atau linked list) untuk menyimpan data entitas, seperti pengguna, produk, atau pesanan. Indeks (hash table) digunakan untuk mempercepat pencarian dan pengambilan data berdasarkan kunci unik, seperti ID pengguna atau nomor pesanan.

Penerapan Struktur Data dalam Jaringan

Dalam jaringan komputer, struktur data digunakan untuk menyimpan dan mengorganisir informasi tentang perangkat dan koneksi dalam jaringan. Contoh penerapan adalah penggunaan graf berbobot untuk memetakan topologi jaringan dan tabel routing (hash table) untuk mengoptimalkan proses pengiriman data.

Penerapan Struktur Data dalam Grafik Komputer

Dalam grafik komputer, struktur data digunakan untuk menyimpan informasi tentang objek dalam dunia 3D, seperti titik, garis, dan poligon. Contoh penerapan adalah penggunaan pohon (tree) untuk menyimpan hierarki objek dalam adegan dan buffer (array) untuk menyimpan data gambar dalam format komputer.

PENERAPAN STRUKTUR DATA

Penerapan Struktur Data dalam Kecerdasan Buatan

Dalam kecerdasan buatan, struktur data digunakan untuk merepresentasikan pengetahuan dan model pembelajaran mesin. Contoh penerapan adalah penggunaan pohon keputusan (tree) untuk klasifikasi data dan graf (graph) untuk mewakili aliran informasi dalam jaringan syaraf tiruan.

Penerapan Struktur Data dalam Pemrosesan Bahasa Alami

Dalam pemrosesan bahasa alami, struktur data digunakan untuk menganalisis dan memahami teks. Contoh penerapan adalah penggunaan trie untuk mencari kata-kata dalam kamus dan pohon (tree) untuk memahami struktur kalimat.

Penerapan Struktur Data dalam Sistem Operasi

Dalam sistem operasi, struktur data digunakan untuk mengelola dan mengorganisir berbagai informasi tentang proses, file, dan memori. Contoh penerapan adalah penggunaan tabel hash untuk memetakan informasi proses dan linked list untuk mengelola daftar file.

PENERAPAN STRUKTUR DATA

Penerapan Struktur Data dalam Permainan Komputer

Dalam permainan komputer, struktur data digunakan untuk mengatur dan menyimpan data permainan, seperti karakter, level, dan objek. Contoh penerapan adalah penggunaan pohon (tree) untuk membangun peta permainan dan stack untuk menyimpan status permainan saat melakukan undo atau checkpoint.

Penerapan Struktur Data dalam Sistem Informasi Geografis

Dalam sistem informasi geografis (GIS), struktur data digunakan untuk menyimpan dan mengorganisir data spasial, seperti titik, garis, dan poligon. Contoh penerapan adalah penggunaan graf (graph) untuk memetakan relasi spasial antara entitas dan quadtree untuk mencari data dalam peta dengan cepat.

Penerapan Struktur Data dalam Analisis Keuangan

Dalam analisis keuangan, struktur data digunakan untuk menyimpan data historis dan menghitung indikator keuangan. Contoh penerapan adalah penggunaan array untuk menyimpan data harga saham dan stack untuk menghitung rata-rata pergerakan harga.

PENUTUP

APA KEGUNAAN DARI STRUKTUR DATA?

Bayangkan struktur data sebagai kumpulan buku yang ada di perpustakaan. Di sana, buku-buku diatur sesuai bidang untuk memudahkan pengunjung mencari buku yang mereka butuhkan. Misalnya, ada seseorang yang membutuhkan buku tentang sistem politik Indonesia, ia bisa langsung menuju rak buku berlabel “politik dan hukum”.

Bayangkan jika perpustakaan tidak menyusun buku sesuai bidang dan tidak memberi label di raknya. Setiap orang yang berkunjung pasti kesulitan dan membutuhkan waktu lama menemukan buku yang dicari di antara puluhan rak buku lain.

Hal yang sama juga bisa dilihat dalam sebuah kamus. Ketika seseorang ingin mencari arti kata “sertifikasi”, ia bisa langsung mencari kata tersebut di abjad S.

PENUTUP

APA KEGUNAAN DARI STRUKTUR DATA?

Dalam bidang IT yang aplikasi dan jumlah datanya terus bertambah, struktur data memberikan banyak keuntungan, antara lain:

- Memproses data besar lebih cepat dengan cara mengatur data ke bentuk yang lebih mudah dikerjakan.
- Memudahkan pencarian data dengan cara mengatur informasi dalam bentuk yang lebih mudah ditemukan.
- Setelah selesai membuat struktur data, struktur tersebut bisa digunakan berulang kapan pun dibutuhkan. *Analyst* tidak perlu membuat struktur baru sehingga menghemat waktu dan tenaga.
- Mengatur informasi agar pengguna bisa langsung mencari item yang diperlukan, tidak perlu mencari setiap item.

Terima
Kasih