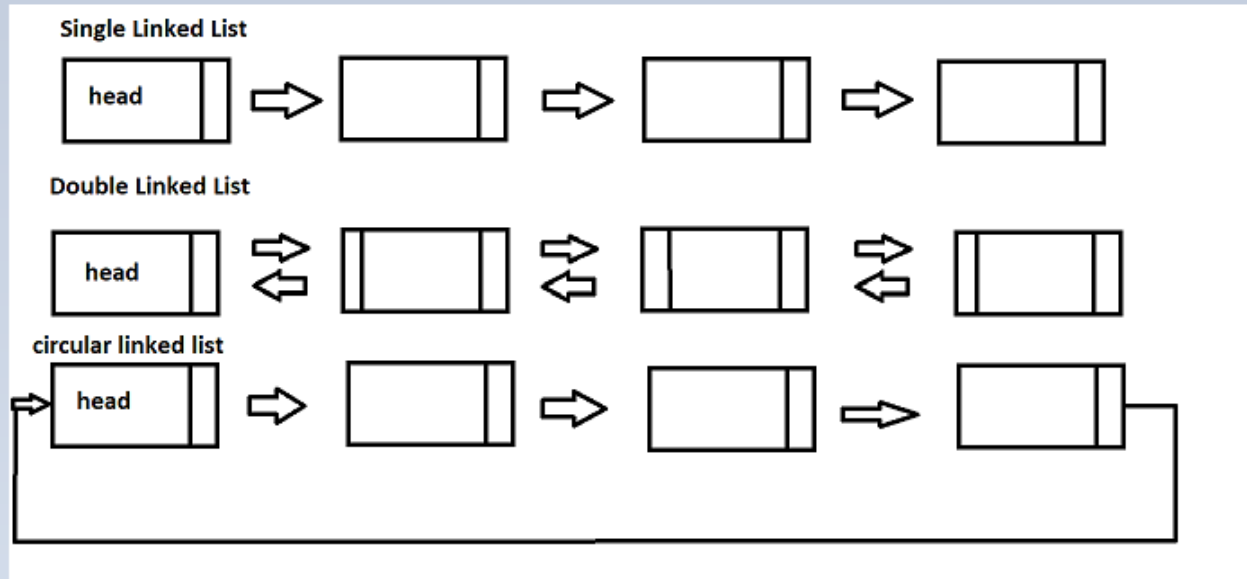
Linked List

1.



2. perbedaan utama antara linked list dan array adalah sifat linked list lebih dinamis sedangkan array lebih statis. Kita hanya dapat menentukan besar array di awal saja, sedangkan linked list kita bisa tambah kapan pun. Contohnya Ketika kita deklarasi variable int angka[10], artinya variable angka ini hanya dapat menampung 10 elemen saja, sedangkan linked list itu tidak terbatas sesuai kebutuhan.

Stack and Queue

1. perbedaan utama antara stack and queue. Konsep dari stack adalah penumpukan, sedangkan queue adalah antrian. Jadi pada stack kita menggunakan istilah FILO yaitu first in last out. Yang masuk pertama akan keluar paling terakhir. Contohnya jika kita memasukkan angka secara berurutan yaitu 1,2,3,4,5. Bayangkan jika angka 1 dimasukkan ke toples terlebih dahulu lalu di lanjutkan oleh 2 , 3 ,4 ,5, jika kita ingin mengeluarkan angka 1, pasti kita harus mengeluarkan angka 5,4,3, dan 2. Sedangkan queue menggunakan istilah FIFO yatu first in first out. Yang masuk pertama akan keluar pertama. Contoh ada antrian 1,2,3,4,5 secara berurutan. Angka 1 antri terlebih dahulu, lalu dibelakangnya ada 2,3,4,5 yang akan dilayani pasti angka 1 lalu 2 lalu 3 lalu 4 lalu 5. Maka dari itu queue menggunakan istilah FIFO

2. Infix adalah perhitungan dimana operator terdapat diantara operand

Contoh 1 + 2 +3 = 6

Implementasi stack

Dari 1 + 2 + 3

- 2 masuk

- + masuk
- 1 + masuk

Lalu

- 1 keluar
- + keluar
- 2 keluar

Jadinya

1 + 2 = 3

3 + 3

- 3 masuk
- + masuk
- 3 masuk

Lalu

- 3 keluar
- +keluar
- 3 keluar

Jadinya 3 + 3 = 6

Prefix adalah perhitungan dimana operator ditulis sebelum operand

Contoh +3 + 2 1 -> ini artinya 1 + 2 lalu hasilnya ditambah 3

Implementasi stack

- 1 lalu ketemu 2, lalu ketemu +
- Dijumlah jadi 3
- 3 ketemu 3 ketemu +
- Dijumlah jadi 6

Postfix adalah perhitungan dimana operator ditulis setelah operand

Contoh 1 2 + 3 + -> artinya setelah melewati 2 operand, dia akan bertemu operator. Contohnya disini bertemu angka 1 dan 2 lalu +, artinya 1 + 2 = 3, lalu 3 ketemu 3 baru ketemu + jadinya 3 + 3 =6

Implementasi stack

Ketika dia bertemu dengan operand ,maka operand tersebut masuk ke stack

1 dan 2 masuk ke stack

- 2
- 1
  Ketemu tanda +
  1 + 2 = 3

  Lalu ketemu 3 masukkan ke stack lagi.

- 3
- 3
  Ketemu tanda +
  3+ 3 = 6

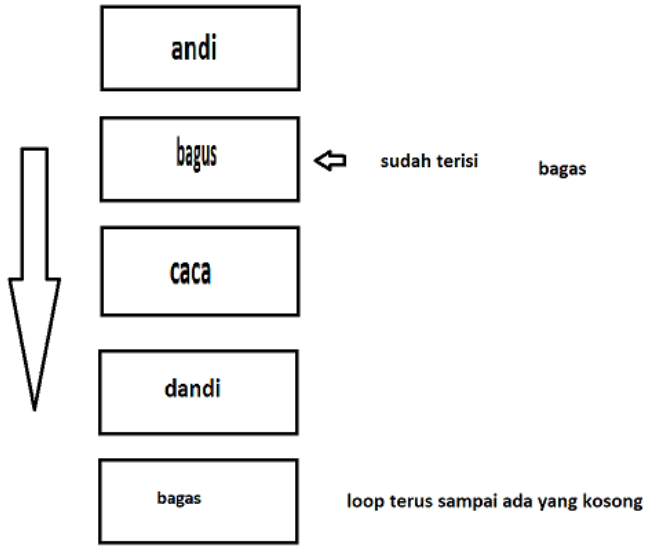
Hash Table

1. hashtable adalah tempat dimana kita menyimpan data dari sebuah string ataupun angka. Biasanya hashtable menyimpan data dari sebuah index string yang sudah di hashed.

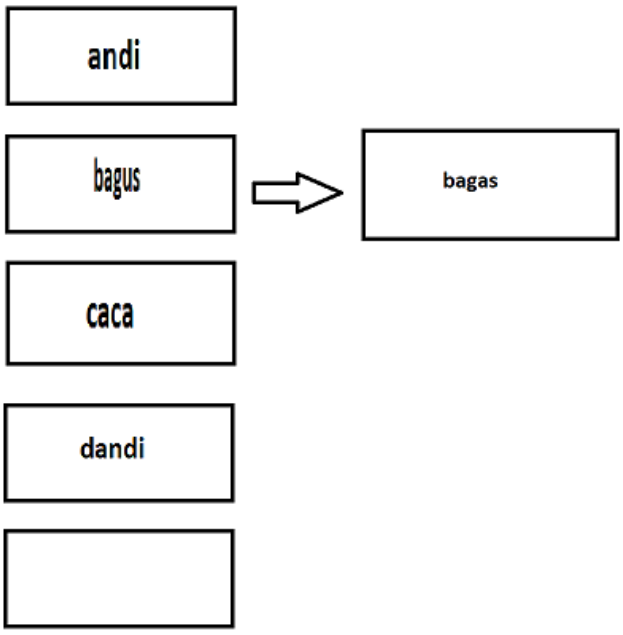
Hash function adalah rumus yang digunakan untuk memasukkan data ke dalam hashtable. Ada banyak cara contohnya seperti kita mendapatkan index sesuai abjad dari karakter pertama dari string

Collision adalah situasi dimana terdapat hal yang sama. contohnya adalah Ketika kita memiliki string andi, bagus, caca, dandi. Contoh hash function yang digunakan adalah mendapatkan index sesuai karakter pertama dari string. Dari ke 4 string tersebut, kita akan mendapatkan masing masing index 0 1 2 3. Ketika kita ingin memasukan string bagas, maka collision terjadi. String bagas memiliki index 2, tetapi pada index 2 sudah terisi oleh bagus, maka dari itu, bagas akan ditampung pada index yang baru.

2. linear probing adalah cara dimana kita akan loop sampai menemukan tempat kosong



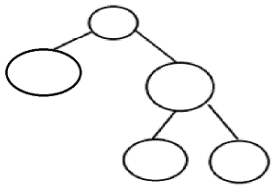
Chaining adalah cara dimana kita membuat sebuah linked list untuk menampung alphabet yang lebih dari 1



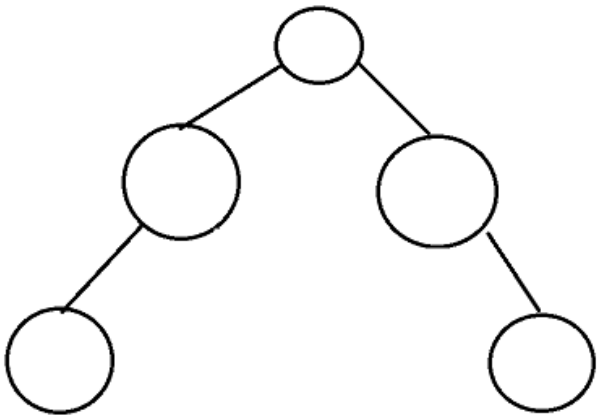
Jadi pada chaining , bagas akan di linked ke next dari si bagus, jadi bagas tidak di isi pada index 4

Binary Search Tree

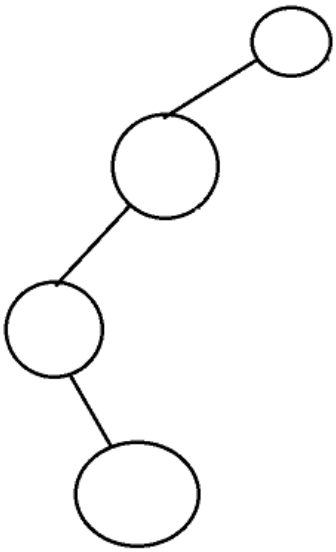
1. Full -> tree yang hanya boleh memiliki 2 atau 0 anak



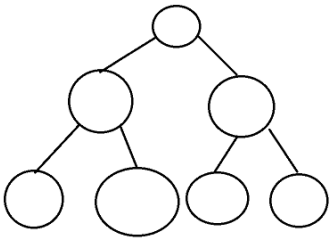
complete -> tree dimana node harus memiliki anak Kecuali leaf nya.



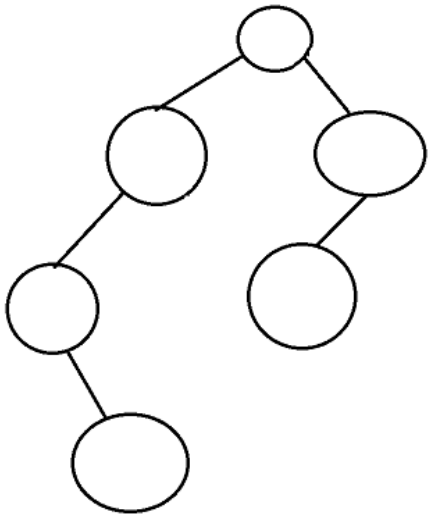
degenerate -> tree yang searah, jika ada anak ke kiri ,maka kiri semua, jika ada anak kanan, kanan semua.



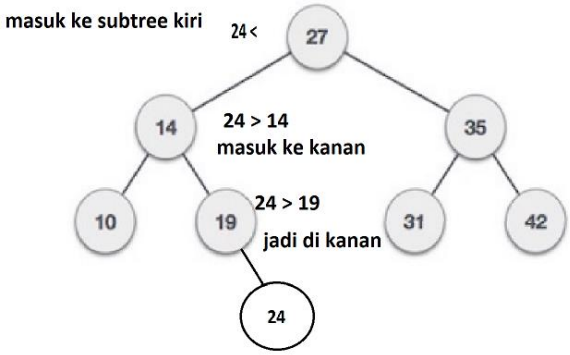
perfect -> tree yang node nya hanya memiliki 2 anak



balanced -> selisih dari subtree kanan dan subtree kiri maksimal 1

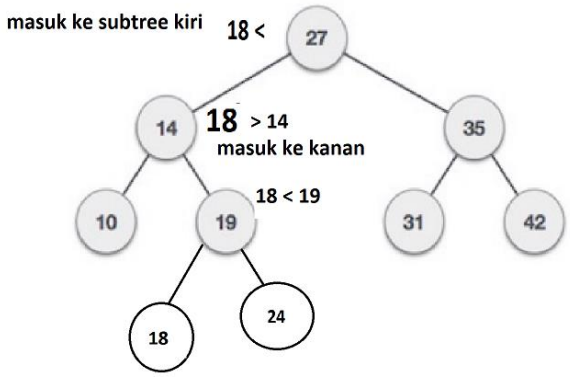


2. insert 24



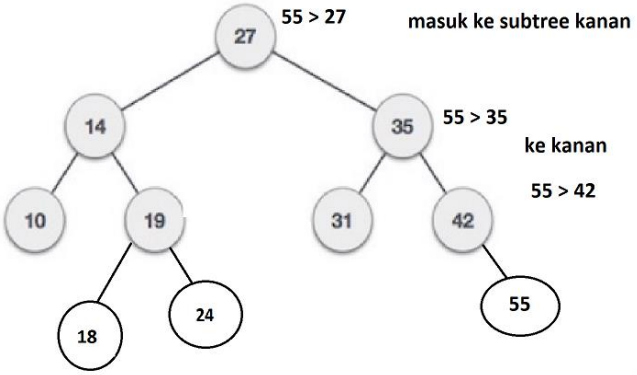
Pada awalnya , 24 lebih kecil dari 27 maka masuk ke subtree kiri. Lalu dibandingkan dengan 14, 24 lebih besar dari 14 maka ke kanan, dibandingkan lah dengan 19, 24 lebih besar dari 19, jadi anak kanan.

Insert 18

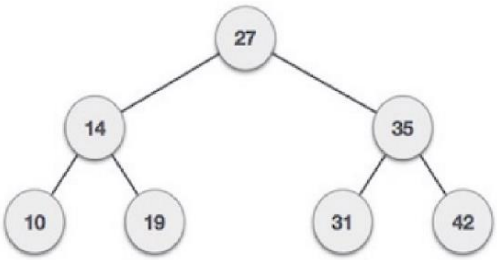


Pada awalnya , 18 lebih kecil dari 27 maka masuk ke subtree kiri. Lalu dibandingkan dengan 14, 18 lebih besar dari 14 maka ke kanan, dibandingkan lah dengan 19, 18 lebih kecil dari 19, jadi anak kiri.

Insert 55

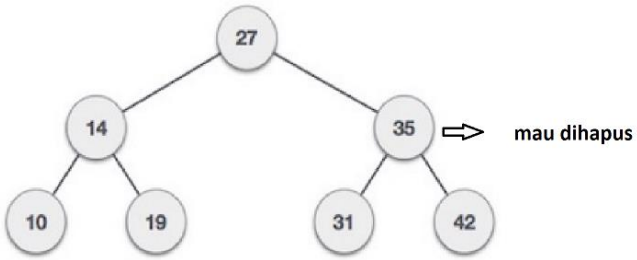


3. deletion 27



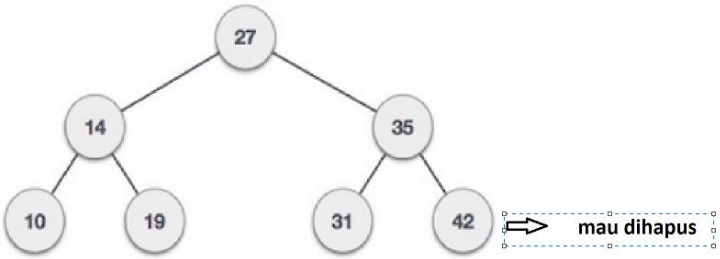
Ketika kita ingin menghapus root maka kita akan memilih angka subtree kanan yang terkecil atau memilih angka subtree kiri yang terbesar. Jadi kita bisa mengganti 27 dengan 31 atau 19.

Deletion 35



Ketika kita ingin menghapus parent yang memiliki 2 anak, maka untuk menggantinya kita harus memilih anak terkecil sebagai parent pengganti. Jadi anak terkecil nya adalah 31.

Deletion 42



Karena 42 adalah anak atau leaf, maka jika mau dihapus tinggal dihapus saja. Hapus nilai 42 dan hapus link terhadap angka 35