**Konsep Dasar Queue (Antrian)**

 Perbedaan antara stack dan queue terdapat pada aturan penambahan dan penghapusan elemen. Pada stack, operasi penambahan dan penghapusan elemen dilakukan di satu tempat atau satu ujung (LIFO).

 Sementara pada queue operasi penambahan dapat dilakukan di tempat yang berbeda. Penambahan elemen dilakukan di salah satu ujung, biasa disebut dengan elemen depan, dan ujung lainnya, yang biasa disebut dengan ujung belakang digunakan untuk penghapusan data (FIFO).

**Konsep Dasar Queue (Antrian)**

 Queue disebut juga antrian dimana data masuk di satu sisi dan keluar di sisi yang lain.

 Konsep penyimpanan data, terdiri dari beberapa data, dengan aturan data yang baru masuk akan diletakkan di data sebelumnya, data yang keluar adalah data sebelumnya, data yang keluar adalah data yang masuk pertama kali (FIFO, First in – First Out).

 Antrian (Queue) merupakan suatu kumpulan data yang penambahan elemennya (masuk antrian) hanya bisa dilakukan pada suatu ujung (disebut dengan sisi belakang/rear) atau disebut juga enqueue yaitu apabila data masuk ke dalam sebuah antrian.

 Jika data keluar dari antrian/penghapusan (pengambilan elemen) dilakukan lewat ujung yang lain (disebut dengan sisi depan/front) atau disebut juga dequeue yaitu apabila seseorang keluar dari antrian.

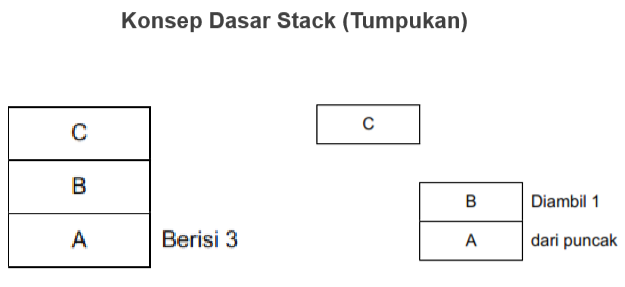
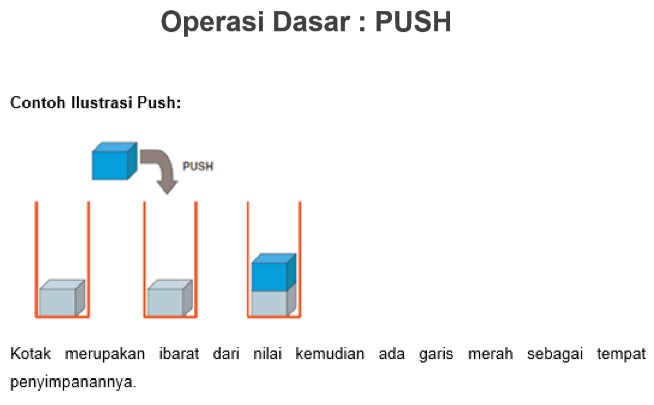
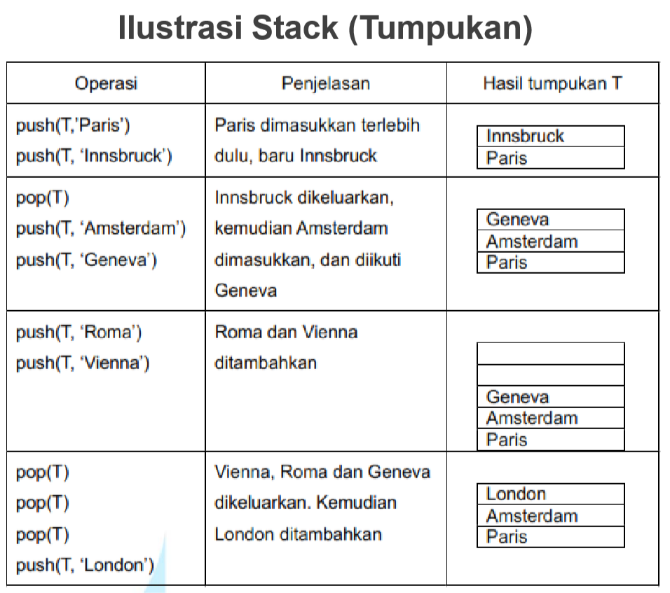
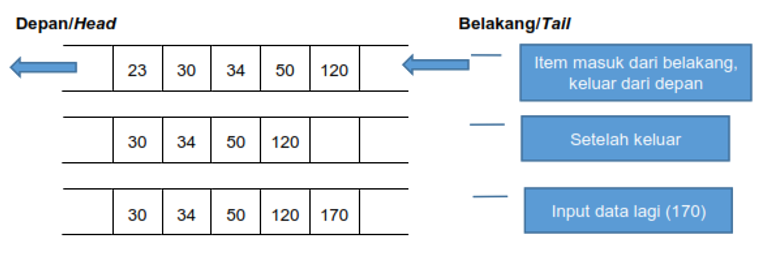
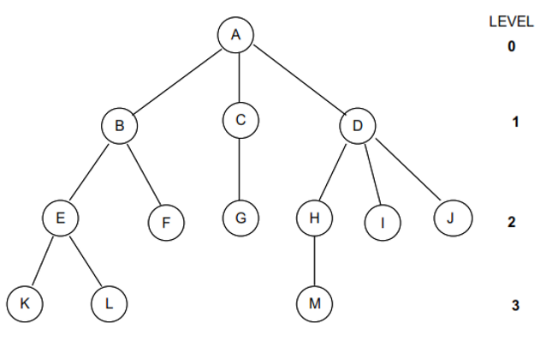
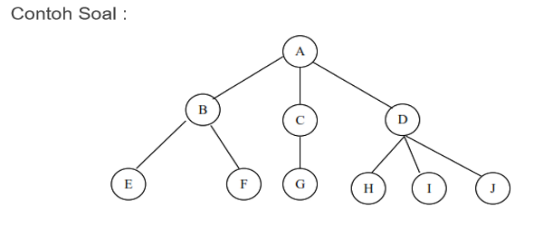
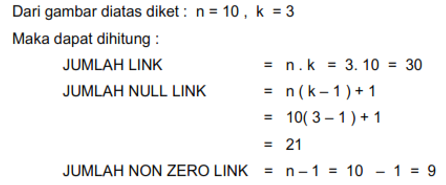
**Konsep Dasar Stack**

θ Stack bisa diartikan sebagai suatu kumpulan data yang seolaholah ada data yang diletakkan diatas data yang lain.

θ Perlu diingat, didalam stack kita bisa menambahkan (menyisipkan) dan mengambil (menghapus) data melalui ujung yang sama yang disebut sebagai ujung atas stack (top of stack).

 Stack mempunyai sifat LIFO (Last In First Out), yaitu yang terakhir masuk adalah yang pertama keluar.

 Konsep ini melakukan penyimpanan dengan data yang pertama masuk akan keluar terakhir dan data yang terakhir masuk akan keluar pertama



Pernyataan Tree (Pohon)

Ada 2 cara untuk menyatakan struktur pohon yaitu :

 Gambar

 Daftar(List)

Proses dalam struktur data non linier, bentuk pohon akan lebih mudah digambarkan bila diketahui :

n ( Jumlah Simpul atau Node ) k ( Derajat Pohon)

Dari data n dan k maka dapat dihitung :

JUMLAH LINK = n . k

JUMLAH NULL-LINK = n (k - 1)+1

JUMLAH NON ZERO LINK = n - 1

 Degree adalah derajat maksimum dari suatu simpul dalam pohon. Contoh derajat pohon dalam gambar diatas adalah 3.

 Nenek Moyang dari suatu simpul adalah seluruh simpul-simpul yang ada sepanjang lintasan dari akar sampai simpul tersebut. Contoh nenek moyang dari M adalahA, D dan H.

 Kedalaman (HEIGHT atau DEPTH) suatu pohon ditentukan oleh level maksimum dari simpul dalam pohon. Contoh kedalaman pohon dari gambar diatas adalah A.

Forest adalah susunan dari beberapa pohon.

Bila akar Adihilangkan maka akan diperoleh hutan dengan 3 pohon yaitu :

 B(E(K,L),F)

 C(G)

 D(H(M),I,J)

 Akar dari struktur pohon ini adalah A

 Simpul A sebagai akar mempunyai 3 Link yang membentuk SUBTREE B,C, D.

 Jumlah SUBTREE dari satu simpul disebut DERAJAT / DEGREE. Derajat Simpul / Degree :

A = 3 B = 2 C = 1 G = 0

 Simpul yang mempunyai derajat = 0 disebut SIMPUL TERMINAL atau DAUN (LEAF) Contoh simpul daun gambar diatas adalah : K , L, F, G, M, I , J

**Konsep Dasar Tree (Pohon)**

Susunan dari satu atau lebih simpul (node) yang terdiri dari satu simpul khusus yang disebut akar (root) sedang sisanya membentuk subtree dari akar.

Satu simpul akan berisi :

 Informasi ( misal,A, B, dst)

 Cabang-cabang (Link) yang menghubungkan ke simpul yang lain

**Operasi Dasar Queue (Antrean)**

 Init : Membuat Queue (Inisialisasi)

 InQueue : Memasukan elemen ke dalam Queue (Insert Queue)

 DeQueue : Menghapus Elemen dari Queue (Delete Queue)

 Full : Mengecek apakah Queue penuh

 Empty : Mengecek apakah Queue kosong

Dalam struktur Tree (Pohon) dikenal istilah geneologi (silsilah). Dalam struktur tersebut adanya simpul anak (children) dan orang tua (parent).

 DERAJAT (DEGREE) SUATU POHON

 KEDALAMAN (HEIGHT atau DEPTH)

 HUTAN (FOREST)

 Init, merupakan prosedur untuk membuat queue pada kondisi awal, yaitu queue yang masih kosong (belum mempunyai elemen).

 InQueue, Insert Queue merupakan prosedur untuk memasukkan sebuah elemen baru pada queue. Jumlah elemen Queue akan bertambah satu dan elemen tersebut merupakan elemen belakang.

 DeQueue, Delete Queue merupakan prosedur untuk menghapus/mengambil sebuah elemen dari queue. Elemen yang diambil adalah elemen depan dan jumlah elemen queue akan berkurang satu.

 Full, merupakan prosedur untuk mengetahui apakah Queue penuh atau tidak. Prosedur ini hanya berlaku untuk queue yang jumlahnya terbatas.

 Empty, yaitu prosedur untuk mengetahui apakah queue dalam keadaan kosong atau tidak. Dengan status ini maka dapat dicegah dilakukannya operasi Dequeue dari suatu queue yang kosong.

**Operasi Dasar Pada Stack**

Push adalah operasi untuk memasukkan data ke dalam tumpukan. Operasi ini biasa dinyatakan dengan push(T,d). T menyatakan tumpukan dan d menyatakan item data yang disisipkan ke dalam tumpukan T.

Pop adalah operasi untuk mengambil data dari tumpukan. Operasi ini biasa dinyatakan dengan pop(T). Dalam hal ini data teratas dari tumpukan T akan dikeluarkan dan menjadi nilai balik pop.