

## Pohon (Tree)

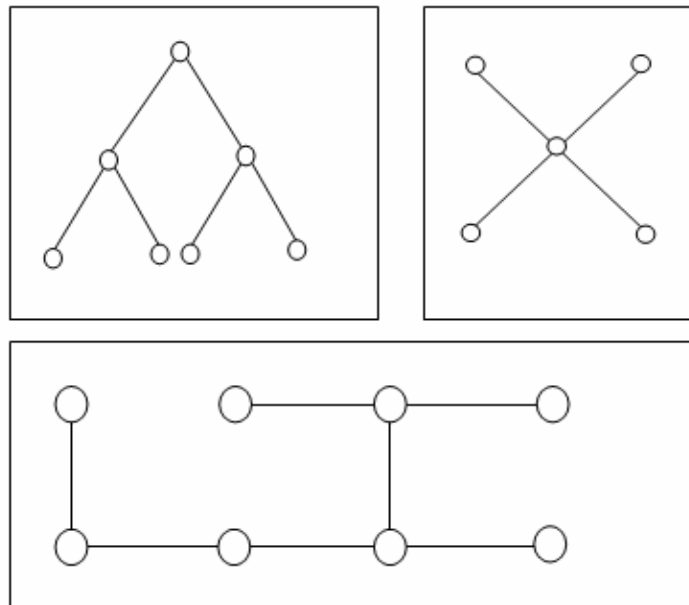
### POHON (TREE)

**Pohon (Tree)** didefinisikan sebagai graph terhubung yang tidak mengandung sirkuit. Sedangkan **Hutan (Forest)** adalah graph yang tidak mengandung sirkuit. Jadi pohon adalah hutan yang terhubung.

Untuk itu perlu diingat kembali bahwa :

- Suatu Graf  $G$  disebut terhubung apabila untuk setiap dua simpul dari graf  $G$  selalu terdapat jalur yang menghubungkan kedua simpul tersebut.
- Sirkuit atau *cycle* adalah suatu lintasan tertutup dengan derajat setiap simpul dua.

Contoh :



Sifat :

Suatu Graf  $G$  adalah Pohon jika dan hanya jika terdapat satu dan hanya satu jalur diantara setiap pasang simpul dari Graf  $G$ .

Teorema :

Suatu Graf G dengan n buah simpul adalah Pohon jika :

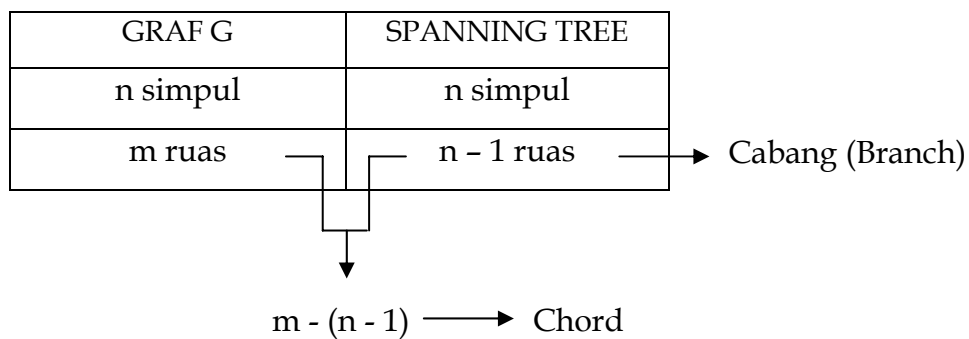
- (1) G terhubung dan tak mengandung sirkuit, atau
- (2) G tidak mengandung sirkuit dan mempunyai n - 1 buah ruas, atau
- (3) G mempunyai n - 1 buah ruas dan terhubung

Teorema :

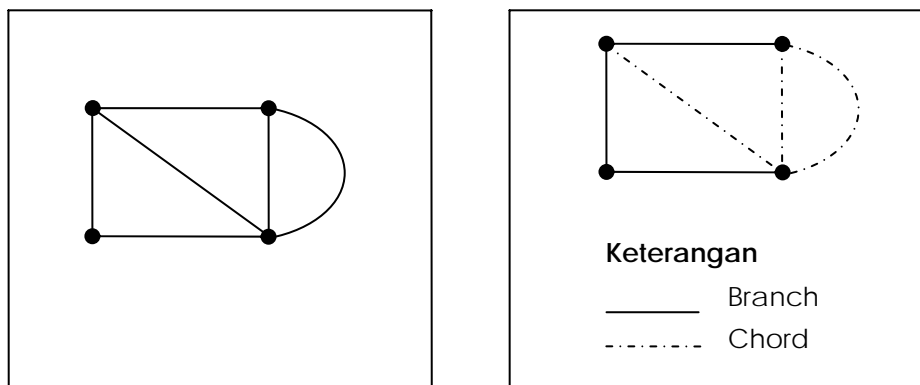
Pohon (dan hutan) adalah berwarna 2.

**POHON RENTANGAN (SPANNING TREE)**

Suatu pohon rentangan atau *spanning tree* adalah suatu subgraf dari graf G yang mengandung semua simpul dari G dan merupakan suatu pohon.

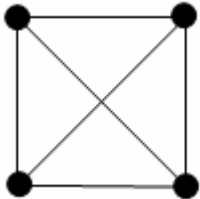


Contoh :

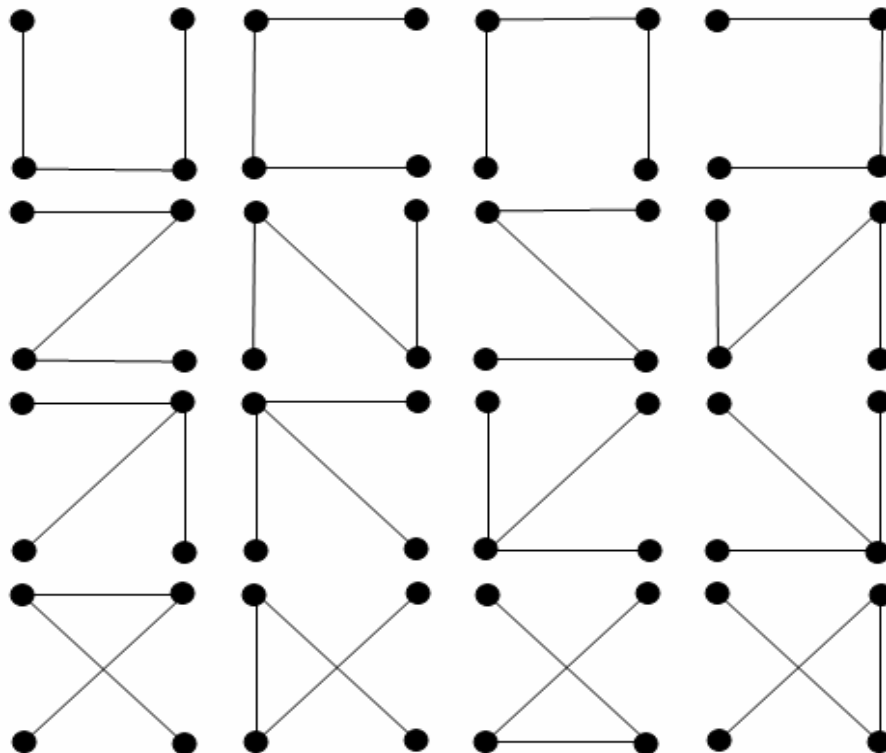


Contoh :

**Graf G**



**Spanning tree dari graf G**



**POHON RENTANGAN MINIMAL (MINIMAL SPANNING TREE)**

Apabila G suatu Graf berbobot (Suatu Network), maka pohon rentangan minimal dari graf adalah pohon rentangan dengan jumlah bobot terkecil.

Untuk mendapatkan pohon rentangan minimal dapat digunakan Algoritma berikut :

**SOLIN**

1. Urutkan ruas dari G menurut bobotnya, dari besar ke kecil.
2. Lakukan penghapusan ruas berdasarkan urutan yang sudah dilakukan, dengan ketentuan bahwa penghapusan ruas tersebut tidak menyebabkan graf menjadi tidak terhubung.

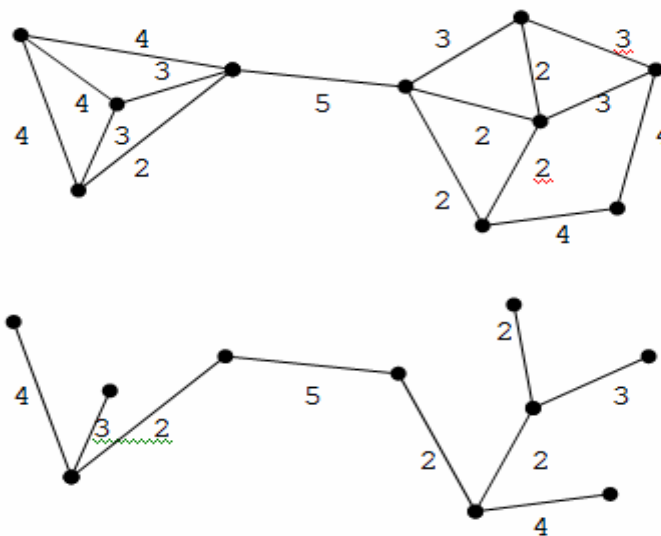
**KRUSKAL**

1. Urutkan ruas dari G menurut bobotnya, dari kecil ke besar.
2. Lakukan penambahan ruas berdasarkan urutan yang sudah dilakukan, dengan ketentuan bahwa penambahan ruas tersebut tidak menyebabkan adanya sirkuit.

**PRIM'S**

= Kruskal + menjaga graf tetap terhubung

Contoh :



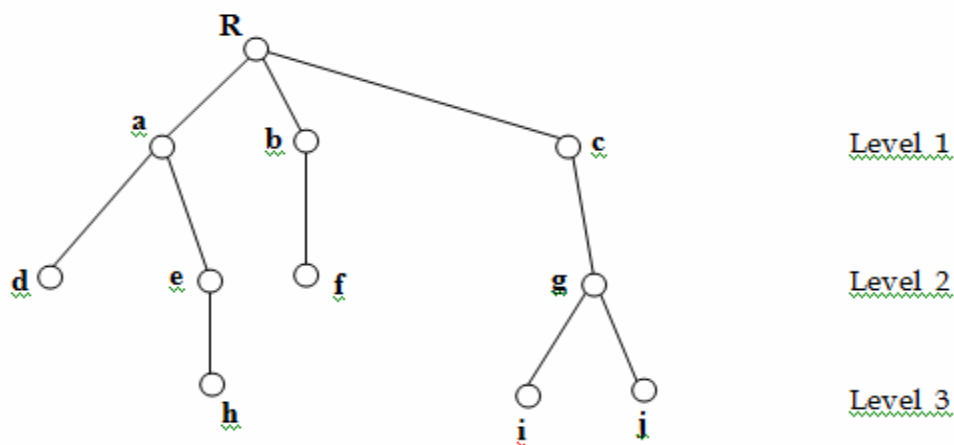
$$\text{Total bobot} = 4 + 3 + 2 + 5 + 2 + 4 + 2 + 2 + 3 = 27$$

**POHON BERAKAR (ROOTED TREE)**

Suatu pohon berakar  $R$  adalah suatu pohon bersama dengan suatu simpul  $r$  yang dirancang/ ditunjuk sebagai akar (root) dari  $R$ .

- Seperti diketahui bahwa hanya terdapat satu jalur antara  $r$  dengan simpul lain  $v$  pada pohon pohon tersebut.
- Panjang jalur antara  $r$  dengan simpul  $v$  disebut level atau kedalaman simpul  $v$ .
- Simpul bukan akar, yang berderajat satu disebut daun. Jalur antara suatu simpul dengan suatu daun disebut cabang (branch).

Contoh :



- Simpul  $u$  dikatakan mendahului simpul  $v$  jika jalur dari akar  $r$  ke  $v$  melalui  $u$ .
- Dikatakan  $u$  mendahului langsung  $v$  bila  $u$  mendahului  $v$  serta simpul  $u$  dan  $v$  berdampingan.
- Pada contoh di atas,  $a$  mendahului  $d$ , mendahului  $e$ , dan mendahului  $h$ .

Suatu pohon berakar dapat digunakan untuk menelusuri semua kemungkinan dari kejadian, dengan masing-masing kejadian dapat muncul dalam sejumlah hingga cara.

Beberapa contoh lain yang penting dari pohon berakar adalah pohon binar (binary tree) dan pohon sintaks (syntax tree) atau pohon derivasi (derivation tree).

## POHON BINAR (BINARY TREE)

Dalam struktur data, pohon memegang peranan yang cukup penting. Struktur ini biasanya digunakan terutama untuk menyajikan data yang mengandung hubungan hierarkikal antara elemen-elemen mereka.

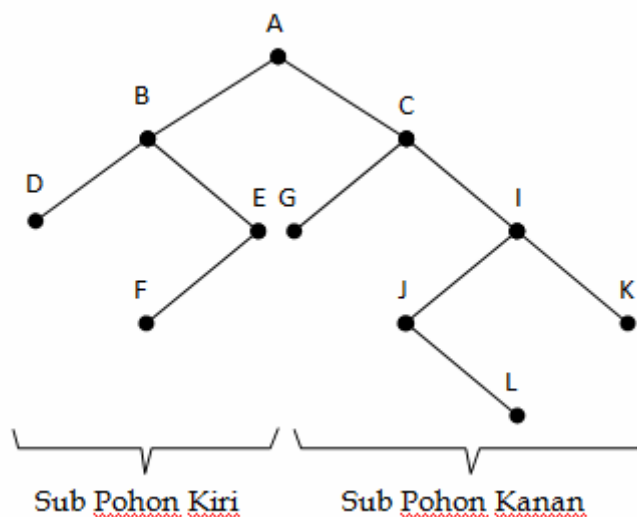
Bentuk pohon khusus yang lebih mudah dikelola dalam komputer adalah pohon binary. Bentuk ini merupakan bentuk pohon yang umum. Sebuah pohon binar  $T$  didefinisikan terdiri dari sebuah himpunan hingga elemen yang disebut simpul, sedemikian sehingga :

- $T$  adalah hampa (disebut pohon null) atau
- $T$  mengandung simpul  $R$  yang dipilih disebut akar (root) dari  $T$ , dan simpul sisanya membentuk 2 pohon binary  $T_1$  dan  $T_2$  yang saling lepas.

Setiap simpul didalam pohon binar hanya dapat mempunyai 0, 1 atau 2 *successor* (turunan langsung).

Untuk menyajikan pohon binar, simpul akar adalah simpul yang digambar pada bagian paling atas. Sedangkan suksesor kiri (*left successor*) digambarkan sebagai garis ke kiri bawah dan suksesor kanan (*right successor*) sebagai garis ke kanan bawah.

Contoh :



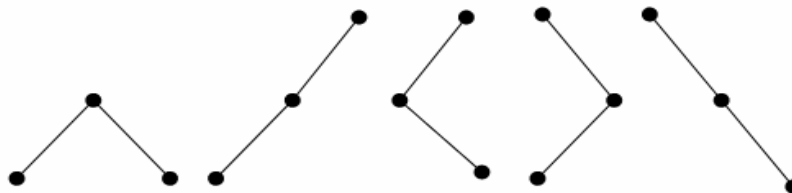
- B adalah *left successor* dan C adalah *right successor* dari simpul A
- *Left subtree* dari root A terdiri dari simpul B, D, E, F
- *Right subtree* dari root A terdiri dari simpul C, G, H, J, K, L
- F adalah left successor dari simpul E
- L adalah right successor dari simpul J
- Kedalaman atau ketinggian (depth/height) dari pohon binar T adalah banyak maksimum simpul dari cabang di T. Untuk pohon binar di atas, ketinggiannya adalah 5.

Pohon biner T dan T' disebut *similar* jika strukturnya (bentuknya) sama.

Pohon biner T dan T' disebut salinan (*copy*) jika strukturnya (bentuknya) sama dan nama simpulnya sama.

Contoh :

Gambar semua kemungkinan pohon biner yang non-similar dengan 3 simpul

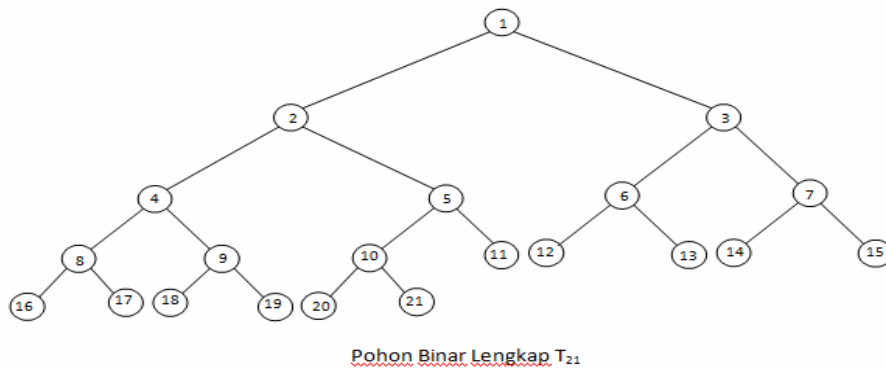
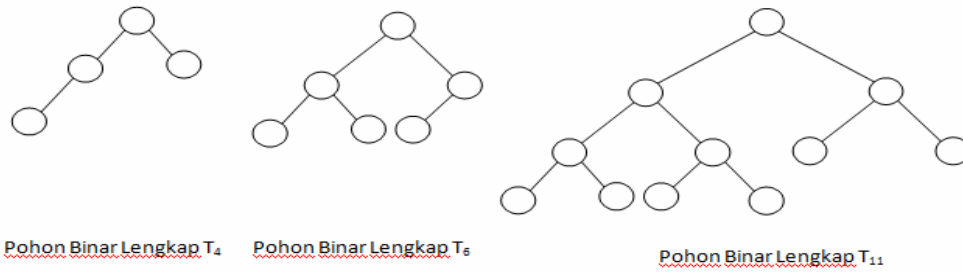


### POHON BINAR LENGKAP

Suatu pohon binar T dikatakan **lengkap** bila setiap tingkatnya, kecuali mungkin tingkat yang terakhir, mempunyai semua simpul yang mungkin, yaitu  $2^r$  simpul untuk tingkat ke- $r$ , dan bila semua simpul pada tingkat terakhir muncul di bagian kiri pohon.

Kedalaman atau ketinggian pohon binar lengkap T dengan n simpul :  $\lceil \log_2 n \rceil + 1$

Contoh :

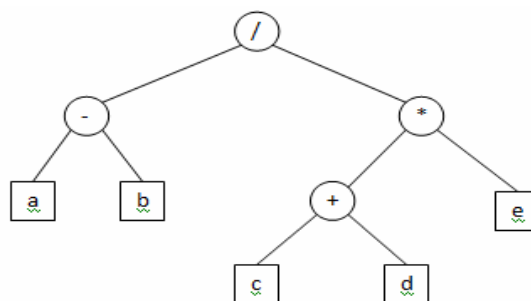


## POHON - 2

Pohon binar  $T$  dikatakan *pohon-2* atau pohon binar yang diperluas (*extended binary tree*) bila setiap simpul mempunyai 0 atau 2 anak.

- Simpul dengan 2 anak disebut **simpul internal**, digambarkan sebagai **lingkaran**. Biasanya berfungsi sebagai **operator**.
- Simpul dengan 0 anak disebut **simpul eksternal**, digambarkan sebagai **segi-empat**. Biasanya berfungsi sebagai **operand**.

Contoh : Pohon-2 yang menyajikan ekspresi  $(a-b)/((c+d)*e)$





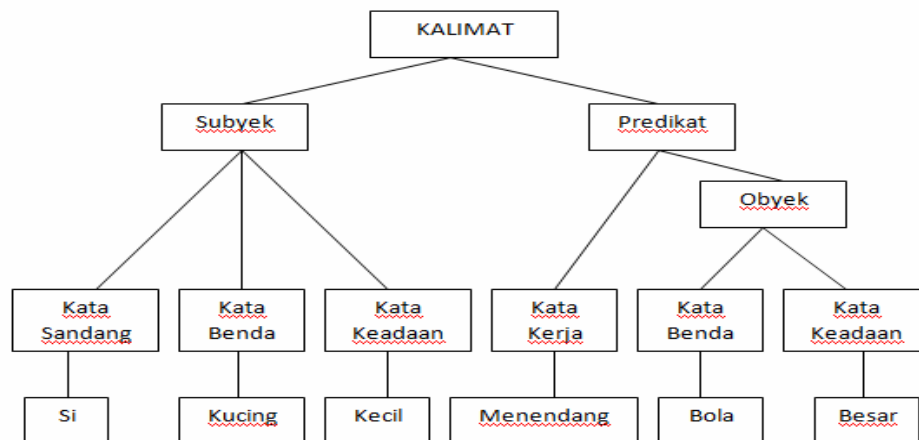
## POHON SINTAKS (SYNTAX TREE)

Untuk menjelaskan mengenai bahasa secara teoritis dan formal, kita lihat terlebih dahulu sebuah kalimat sehari-hari dalam bahasa Indonesia, yaitu :

### **Si kucing kecil menendang bola besar**

Gambar penguraian kalimat di atas membentuk struktur pohon, yang disebut pohon sintaks dari kalimat. Disini kalimat dibagi-bagi berdasar jenis dan fungsi kata. Dari pelajaran bahasa Indonesia kita tahu bahwa kalimat di atas telah benar susunannya, atau telah benar tata bahasanya.

Pohon sintaks dari kalimat di atas dapat dilihat sebagai berikut :



Derivasi adalah proses pembentukan untai terminal dengan melakukan sederetan produksi menggunakan himpunan produksi yang ada.

Himpunan produksi dari pohon sintaks diatas adalah :

1. <Kalimat> → <Subjek> <Predikat>
2. <Subjek> → <Kata Sandang> <Kata Benda> <Kata Keadaan>
3. <Predikat> → <Kata Kerja> <Objek>
4. <Objek> → <Kata Benda> <Kata Keadaan>

- 
5. <Kata Sandang> → si
  6. <Kata Benda> → kucing | bola
  7. <Kata Keadaan> → kecil | besar
  8. <Kata Kerja> → menendang

**Soal Latihan :**

Lakukan derivasi terhadap untai terminal  $x1x2x3$  dengan himpunan produksi sebagai berikut :

1. <OPERAND> → <ID> | <INTEGER>
2. <ID> → <LETTER> <LIST>
3. <LETTER> → x | y | z
4. <LIST> → <LETTER> <LIST> | <DIGIT> <LIST> | ^
5. <DIGIT> → 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
6. <INTEGER> → <SIGN> <DIGIT> <DIT> | <DIGIT> <DIT>
7. <SIGN> → - | +
8. <DIT> → <DIGIT> <DIT> | ^