

# PERTEMUAN 13

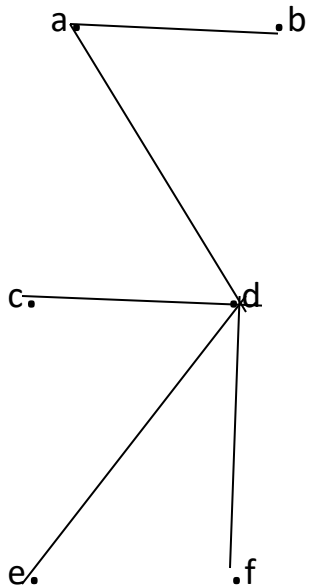
## POHON (TREE)

# Definisi Pohon

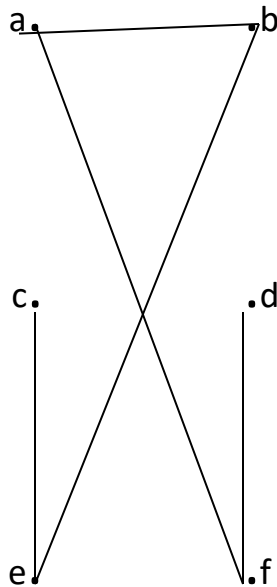
## Definisi Pohon

Adalah bentuk khusus dari graf, atau graf tak berarah terhubung yang tidak mengandung sirkuit.

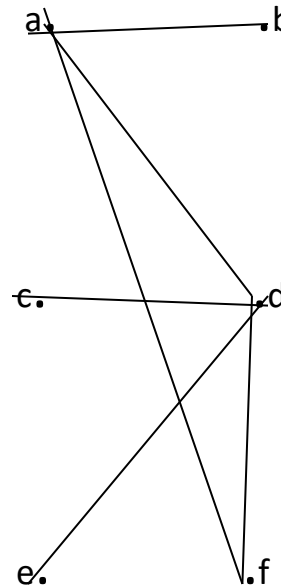
Contoh: bedakan mana yang pohon dan yang bukan pohon



G1



G2

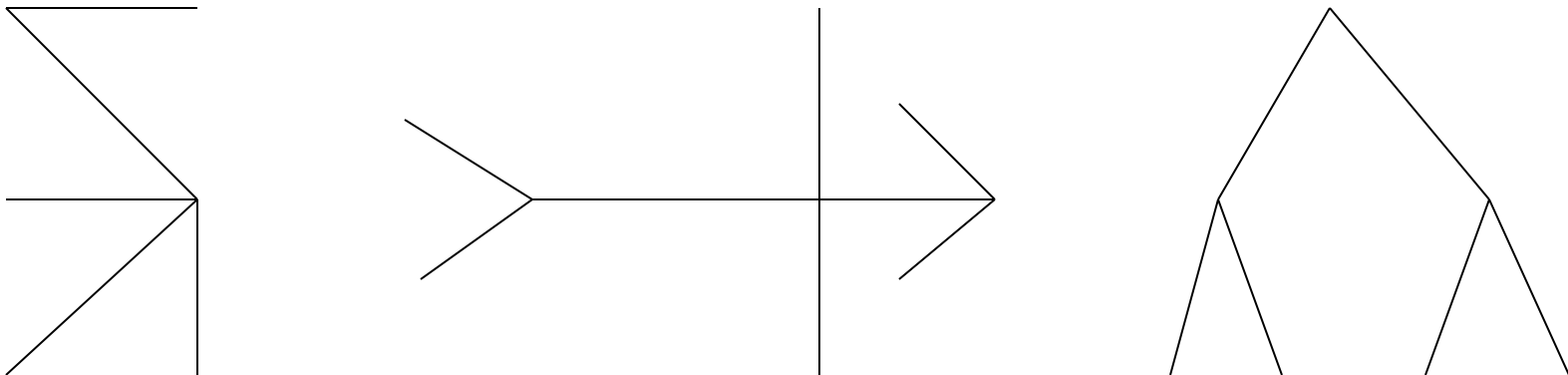


G3

# Definisi Hutan

Contoh lain: misalkan himpunan  $V = \{a, A, b, B, c, C, d, D\}$  adalah empat pasangan suami-istri tukang gosip, dengan  $a, b, c$  dan  $d$  para suami dan  $A, B, C$  dan  $D$  para istri. Misalkan  $a$  menceritakan gosip lewat telpon ke istri  $A$ , yang kemudian  $A$  menelpon para istri lainnya untuk menyebarkan gosip itu, dan setiap istri itu menelpon dan menceritakan gosip kepada suami-suami masing2, bagaimana bentuk pohonnya?

Hutan (*forest*) adalah kumpulan pohon yang saling lepas. Atau graf tidak terhubung yang tidak mengandung sirkuit.



# Sifat-sifat Pohon

## 1.2 Sifat-sifat pohon

### Teorema 1.

Misalkan  $T = (V, E)$  adalah graf tak-berarah sederhana dan jumlah simpulnya  $n$ . Maka, semua pernyataan di bawah ini adalah ekuivalen:

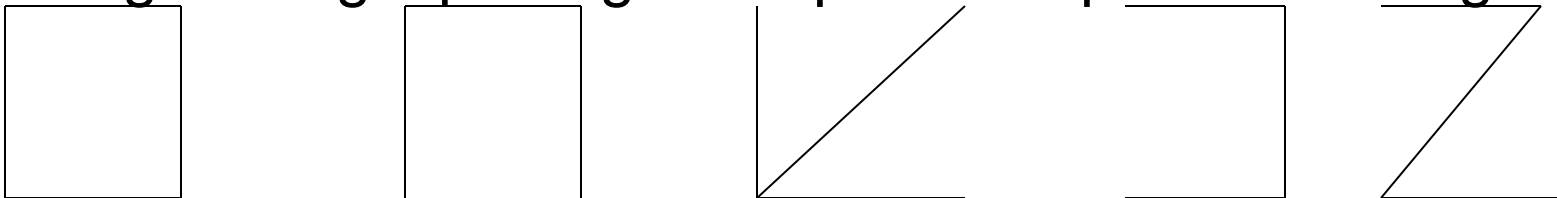
1.  $T$  adalah pohon
2. Setiap pasang simpul di dalam  $T$  terhubung dengan lintasan tunggal.
3.  $T$  terhubung dan memiliki  $m = n-1$  buah sisi.
4.  $T$  tidak mengandung sirkuit dan memiliki  $m = n-1$  buah sisi.
5.  $T$  tidak mengandung sirkuit dan penambahan satu sisi pada graf akan membuat hanya satu sirkuit.
6.  $G$  terhubung dan semua sisinya adalah jembatan (jembatan adalah sisi yang dihapus menyebabkan graf terpecah menjadi dua komponen)

# Pohon Rentang

## 1.3. Pohon Rentang

Misalkan  $G=(V,E)$  adalah graf tak-berarah terhubung yang bukan pohon, yang berarti di  $G$  terdapat beberapa sirkuit.  $G$  dapat diubah menjadi pohon  $T=(v_1,E_1)$  dengan cara memutuskan sirkuit-sirkuit yang ada. Secara berulang-ulang disebut pohon rentang (spanning tree).

Contoh: graf lengkap dengan empat buah pohon rentang.



penerapan pohon rentang misalnya pada pemeliharaan jalan raya. Karena keterbatasan dana pemeliharaan, pemerintah daerah mempertimbangkan hanya jalan-jalan sesedikit mungkin sehingga ke empat kota masih tetap terhubung satu sama lain.

# Istilah-istilah pada pohon rentang

Teorema 2: setiap graf terhubung mempunyai paling sedikit satu buah pohon rentang.

Dari teorema diatas graf yang tidak mengandung sirkuit adalah pohon rentang itu sendiri, sedangkan graf yang mengandung sirkuit, pohon rentangnya diperoleh dengan cara memutuskan sirkuit yang ada.

Sisi pada pohon rentang disebut **cabang** (branch) adalah sisi dari graf semula sedangkan **tali-hubung** (chord atau link) dari pohon adalah sisi yang tidak terdapat di dalam pohon rentang. Himpunan tali-hubung beserta simpul yang bersisian dengannya disebut **komplemen pohon**.

Graf  $G$  dengan  $n$  buah simpul dan  $m$  buah sisi, dapat dicari jumlah cabang dan tali-hubung dengan rumus:

$$\text{jumlah cabang} = n-1$$

$$\text{jumlah tali-hubung} = m-n+1$$

# Pohon Rentang Minimum

Graf tidak terhubung dengan  $k$  komponen,  $m$  buah sisi dan  $n$  buah simpul

jumlah cabang =  $n - k$

jumlah tali-hubung =  $m - n + k$

Jumlah cabang pada pohon rentang daribsebuah graf  $G$  disebut *rank* graf  $G$  dan jumlah tali-hubung  $G$  disebut *nullity* graf  $G$ . Sehingga :

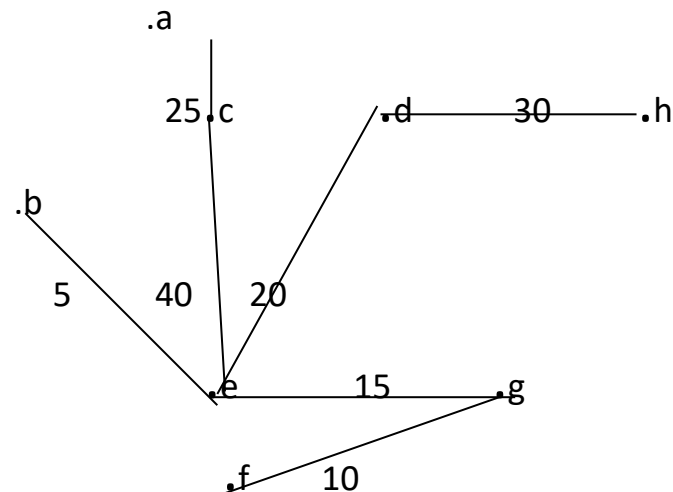
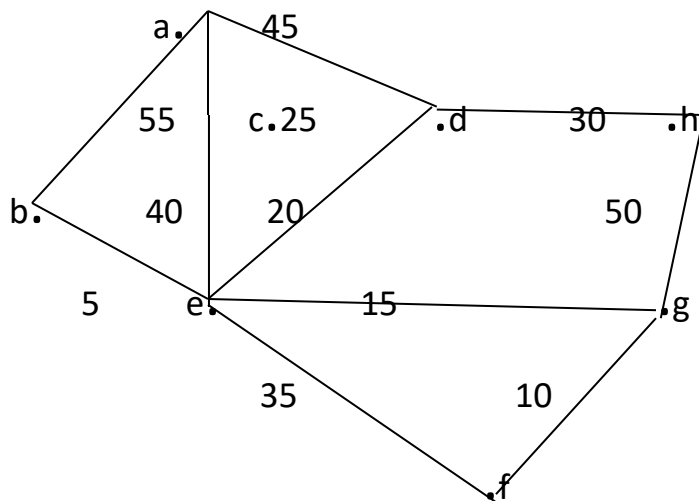
$rank + nullity = \text{jumlah sisi graf } G$

## 1.3.1 Pohon Rentang Minimum (*minimum spanning tree*)

Jika  $G$  adalah graf berbobot, maka bobot pohon rentang  $T$  dari  $G$  didefinisikan sebagai jumlah bobot semua sisi di  $T$ . Semua pohon rentang di  $G$ , pohon rentang yang berbobot minimum dinamakan pohon rentang minimum (*minimum spanning tree*). Jenis ini mempunyai terapan yang luas.

# Contoh soal pohon rentang minimum

Contohnya: Pemerintah akan membangun jalur rel kereta api yang menghubungkan sejumlah kota. Karena biayanya mahal, pembangunan jalur ini tidak perlu menghubungkan langsung dua buah kota, tetapi cukup membangun jalur kereta seperti pohon rentang. Karena dalam sebuah graf mungkin saja terdapat lebih dari satu pohon rentang, maka harus dicari pohon rentang yang mempunyai jumlah jarak terpendek, dengan kata lain harus dicari pohon rentang minimum.





# Algoritma penyelesaian pohon rentang minimum

Dua algoritma untuk menyelesaikan pohon rentang minimum yaitu : 1. Algoritma Prim

2. Algoritma Kruskal

## 1. Algoritma Prim

membentuk pohon rentang minimum langkah demi langkah. Setiap langkah kita ambil sisi dari graf  $G$  yang mempunyai bobot minimum namun terhubung dengan pohon rentang minimum yang telah terbentuk.

### Algoritma Prim

langkah 1: ambil sisi dari graf  $G$  yang berbobot minimum, masukkan ke dalam  $T$

langkah 2: pilih sisi  $(u,v)$  yang mempunyai bobot minimum dan bersisian dengan simpul di  $T$ , tetapi  $(u,v)$  tidak membentuk sirkuit di  $T$ . tambahkan  $(u,v)$  ke dalam  $T$

Langkah 3: ulangi langkah 2 sebanyak  $n-2$  kali

# Algoritma Kruskal

## 2. Algoritma Kruskal

Sisi-sisi graf diurut terlebih dahulu berdasarkan bobotnya yang meenaik ( dari kecil ke besar). Sisi yang dimasukkan ke dalam himpunan  $T$  adalah sisi graf  $G$  sehingga  $T$  adalah pohon. Pada keadaan awal, sisi-sisi yang sudah diurut berdasarkan bobot membentuk hutan. Masing-masing pohon di hutan hanya berupa satu simpul. Hutan tersebut dinamakan hutan pohon rentang.

### Algoritma Kruskal

( langkah 0: sisi-sisi dari graf sudah diurut menaik berdasarkan bobotnya dari bobot kecil ke bobot besar)

Langkah 1 :  $T$  masih kosong

Langkah 2 : Pilih sisi  $(u,v)$  dengan bobot minimum yang tidak membentuk sirkuit di  $T$ . Tambahkan  $(u,v)$  ke dalam  $T$ .

Langkah 3 : ulang langkah 2 sebanyak  $n-1$  kali

# Pohon Berakar

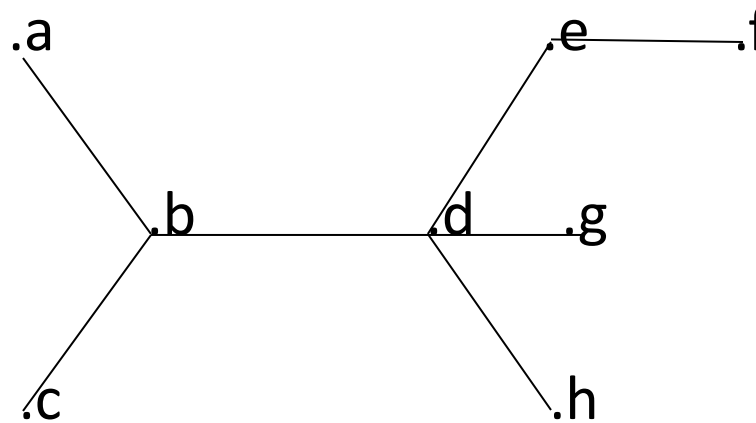
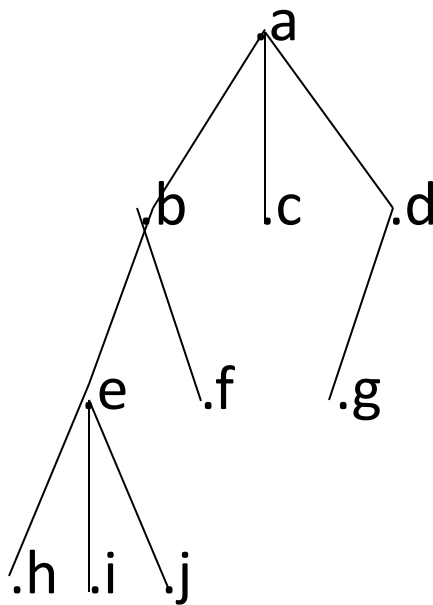
## 1.4 Pohon Berakar

Dalam aplikasi pohon, sering simpul tertentu diperlukan sebagai akar (*root*). Sekali sebuah simpul ditetapkan sebagai akar, maka simpul-simpul lainnya dapat dicapai dari akar dengan memberi arah pada sisi-sisi pohon yang mengikutinya. Pohon yang satu buah simpulnya diperlakukan sebagai akar dan sisi-sisinya diberi arah sehingga menjadi graf berarah dinamakan pohon berakar (*rooted tree*).

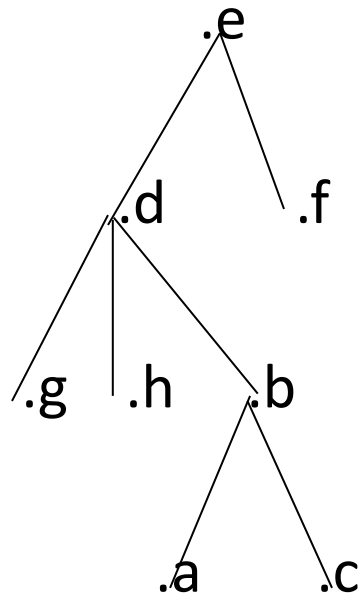
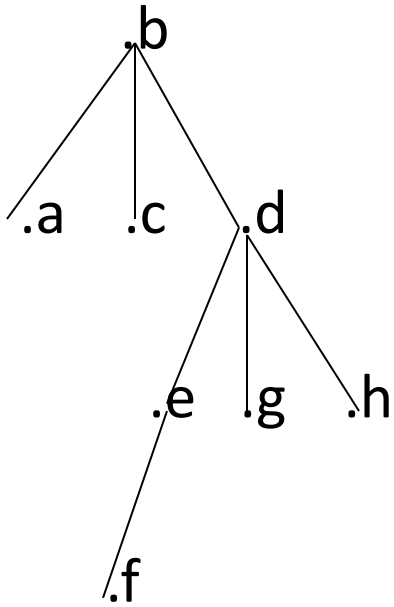
Akar mempunyai derajat masuk sama dengan nol dan simpul-simpul lainnya berderajat masuk sama dengan satu. Simpul yang mempunyai derajat keluar sama dengan nol disebut daun atau simpul terminal. Simpul yang mempunyai derajat keluar tidak sama dengan nol disebut simpul dalam atau simpul cabang. Setiap simpul di pohon dapat dicapai dari akar dengan sebuah lintasan tunggal (unik)

# Perubahan dari pohon tak berakar menjadi pohon berakar

Sembarang pohon tak berakar dapat diubah menjadi pohon berakar dengan memilih sebuah simpul sebagai akar. Pemilihan simpul yang berbeda menjadi akar menghasilkan pohon berakar yang berbeda pula.



# Hasil Pohon berakar yang mungkin



# Terminologi pohon#1

Beberapa terminologi pada pohon

1. Anak (child atau children) dan orangtua (parent)

Misal X simpul pada pohon berakar, simpul Y dikatakan anak dari simpul X jika ada sisi yang menghubungkan X ke Y. dan X disebut orang tua Y

2. Lintasan (path)

Lintasan dari simpul  $v_1$  ke  $v_k$ , adalah runtunan simpul  $v_1, v_2, \dots, v_k$  sedemikian sehingga  $v_i$  orang tua dari  $v_{i+1}$  untuk  $1 \leq i \leq k$  contoh lintasan dari a ke h adalah a, b, e, h dengan **panjang lintasan** adalah jumlah sisi yang dilalui dalam suatu lintasan k-1 ada 3

3. Keturunan (descendant) dan leluhur (ancestor)

Jika terdapat lintasan dari simpul X ke Y di dalam pohon, maka X adalah leluhur Y dan Y adalah keturunan X

## Terminologi Pohon#2

### 4. Saudara kandung (*sibling*)

Simpul yang berorangtua yang sama adalah saudara sekandung.

### 5. Upapohon (*subtree*)

Misalkan  $X$  adalah simpul di dalam pohon  $T$ . Yang dimaksud dengan upapohon dengan  $X$  sebagai akarnya ialah upagraf  $T' = (V', E')$  sedemikian sehingga  $V'$  mengandung  $X$  dan semua keturunannya dan  $E'$  mengandung sisi-sisi dalam semua lintasan yang berasal dari  $X$ .

### 6. Derajat (*degree*)

Jumlah upapohon (atau jumlah anak) pada simpul tersebut.

## Terminologi Pohon#3

### 7. Daun (*leaf*)

Simpul yang berderajat nol (atau tidak mempunyai anak)

### 8. Simpul Dalam (*internal nodes*) adalah Simpul yang mempunyai anak.

### 9. Aras (*level*) atau tingkat

Akar mempunyai aras = 0, sedangkan aras simpul lainnya = 1 + panjang lintasan dari akar ke simpul tersebut.

### 10. Tinggi (*height*) atau kedalaman (*depth*)

Aras maksimum dari suatu pohon.



# Jenis Pohon yang lain

## Jenis Pohon yang lain

1.5 Pohon Terurut adalah pohon berakar yang urutan anak-anaknya penting.

1.6 Pohon  $m$ -ary adalah pohon berakar yang setiap simpul cabangnya mempunyai paling banyak  $m$  buah anak. Pohon  $m$ -ary dikatakan teratur atau penuh jika setiap simpul cabangnya mempunyai tepat  $m$  anak. Jika  $m=2$ , disebut pohon biner (*binary tree*).

Jumlah daun pada pohon  $m$ -ary teratur  $m$  pangkat  $h$ ,  $h$  tinggi pohon.

# Pohon biner

Hubungan antara jumlah daun dan simpul dalam pada pohon m-ary teratur

$$i = t - 1,$$

$i$  = banyaknya simpul dalam

$t$  = banyaknya simpul daun

## 1.7 Pohon biner

adalah pohon yang setiap simpul cabangnya mempunyai maksimum 2 buah anak. Anak pertama disebut anak kiri (*left child*) dan anak kedua disebut anak kanan (*right child*). Pohon yang akarnya adalah anak kiri disebut upapohon kiri (*left subtree*) dan sebaliknya disebut upapohon kanan (*right subtree*).

# Terapan Pohon Biner

Pohon biner seimbang adalah pohon yang memiliki tinggi upapohon kiri dan kanan seimbang. Dibuat dengan menyebarkan setengah dari jumlah simpul upapohon kiri dan setengah dari jumlah simpul upapohon kanan.

## Terapan Pohon Biner

1. Pohon ekspresi (*expression tree*)
2. Pohon keputusan (*decision tree*)
3. Kode Huffman (*Huffman code*)
4. Kode Prefiks (*Prefix code*)

Pohon pencarian biner (*binary search tree*)