

Graf Pohon dan Implementasinya dalam beberapa persoalan

Amir Muntaha NIM 13505041

Program Studi Teknik Informatika,
Sekolah Teknik Elektro dan Informatika,
Institut Teknologi Bandung
Jl. Ganesha 10, Bandung

E-mail : if115041@students.if.itb.ac.id

Abstrak

Makalah ini membahas tentang studi Graf Pohon dan implementasi dalam berbagai contoh persoalan sederhana. Graf Pohon merupakan Graf tak-berarah terhubung yang tidak mengandung sirkuit. Dalam makalah ini graf pohon di titik beratkan pada pohon biner yaitu pohon yang berderajat maksimal 2.

Kata kunci: pohon, tree, anak, child, akar, parent, pohon biner, level, daun, akar

1. Pendahuluan

1.1. Konsep Pohon

Pohon didefinisikan sebagai suatu graf tak berarah terhubung (connected undirected graph) yang tidak mengandung rangkaian sederhana. Pohon adalah bentuk khusus dari suatu graf yang banyak diterapkan untuk berbagai keperluan. Misalnya struktur organisasi suatu perusahaan, silsilah suatu keluarga, skema sistem gugur suatu pertandingan, dan ikatan kimia suatu molekul adalah jenis graf yang tergolong sebagai pohon. Pada pohon, simpul-simpul yang berderajat satu dinamakan daun (leaf), sedangkan simpul yang derajatnya lebih besar daripada satu dinamakan simpul cabang (branch node) atau simpul internal (internal node) dan kumpulan pohon-pohon yang terpisahkan satu sama lain disebut hutan (forest).

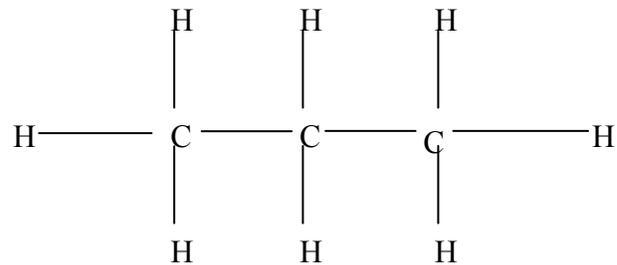
1.2. Sifat-sifat Pohon

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

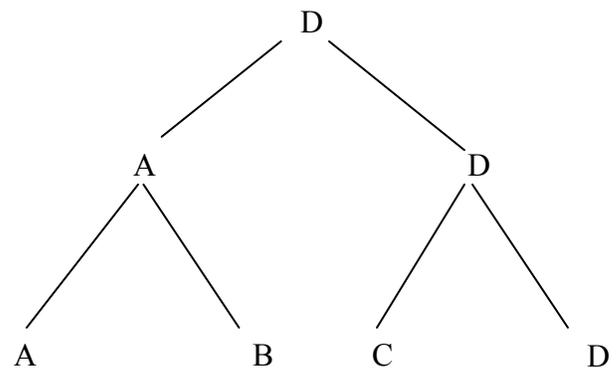
1. G adalah pohon.
2. Setiap pasang simpul di dalam G terhubung dengan lintasan tunggal.
3. G terhubung dan memiliki $m = n - 1$ buah sisi.
4. G tidak mengandung sirkuit dan memiliki $m = n - 1$ buah sisi.

5. G tidak mengandung sirkuit dan penambahan satu sisi pada graf akan membuat hanya satu sirkuit.
6. G terhubung dan semua sisinya adalah jembatan.

Contoh pohon :

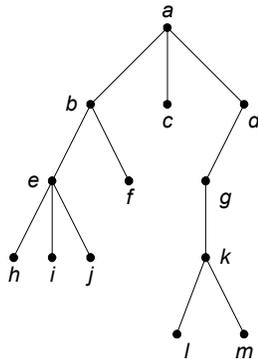


(a) ikatan kimia



(b) pertandingan sistem gugur

1.3 Terminologi pada Pohon Berakar



1.3.1. Anak (*child* atau *children*) dan Orangtua (*parent*)

anak adalah simpul setelah suatu simpul dalam pohon, jadi dari gambar diatas diketahui *b*, *c*, dan *d* adalah anak-anak simpul *a*, dan orang tua adalah simpul sebelum suatu simpul lain dalam satu pohon jadi *a* adalah orangtua dari anak-anak itu

1.3.2. Lintasan (*path*)

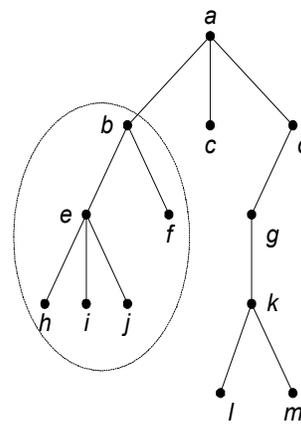
Lintasan adalah jalur yang harus ditempuh dari suatu simpul ke simpul lainnya, panjang lintasan ditentukan dari simpul asal. jadi dari gambar pohon diatas diketahui lintasan dari *a* ke *j* adalah *a*, *b*, *e*, *j* dan panjang lintasan dari *a* ke *j* adalah 3.

1.3.3. Saudara kandung (*sibling*)

Saudara kandung adalah simpul yang mempunyai orangtua yang sama. Jadi dari gambar diatas diketahui *f* adalah saudara kandung *e*, tetapi, *g* bukan saudara kandung *e*, karena orangtua mereka berbeda.

1.3.4. Upapohon (*subtree*)

Upapohon atau subtree adalah pohon yang dibentuk dengan memotong pohon yang sudah ada

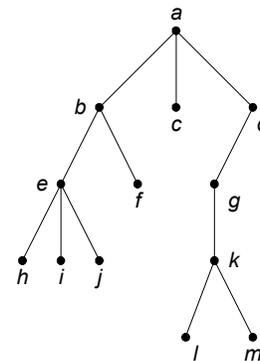


1.3.5. Derajat (*degree*)

Derajat sebuah simpul adalah jumlah upapohon atau jumlah anak pada simpul tersebut, jadi yang dimaksudkan disini adalah derajat ke bawah atau keluar. Jadi pada gambar diatas derajat *a* adalah 3, derajat *b* adalah 2, derajat *d* adalah satu dan derajat *c* adalah 0.

Derajat sebuah pohon adalah derajat maksimum dari semua simpul derajat pohon itu sendiri. Pohon di atas berderajat 3

1.3.6. Daun (*leaf*)



Daun adalah simpul paling ujung dalam sebuah pohon. Jadi daun tidak mempunyai anak dan berderajat 0. Simpul *h*, *i*, *j*, *f*, *c*, *l*, dan *m* adalah daun.

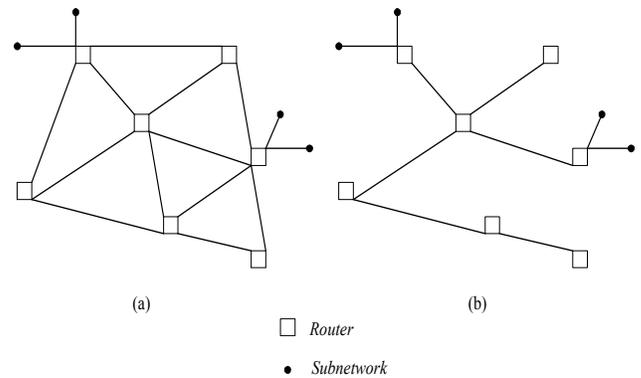
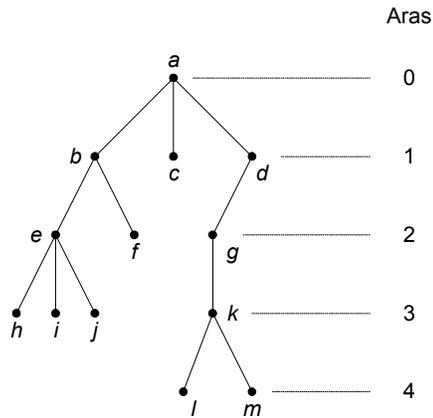
1.3.7. Simpul Dalam (*internal nodes*)

Simpul yang mempunyai upapohon atau anak disebut simpul dalam. Simpul *b*, *d*, *e*, *g*, dan *k* adalah simpul dalam.

1.3.8. Aras (*level*) dan Tingkat ketinggian (*height*) atau Kedalaman (*depth*)

Aras maksimum dari suatu pohon disebut **tinggi** atau kedalaman pohon tersebut. Pohon di bawah mempunyai tinggi 4.

- Perutean (*routing*) pesan pada jaringan komputer.
- Multicast

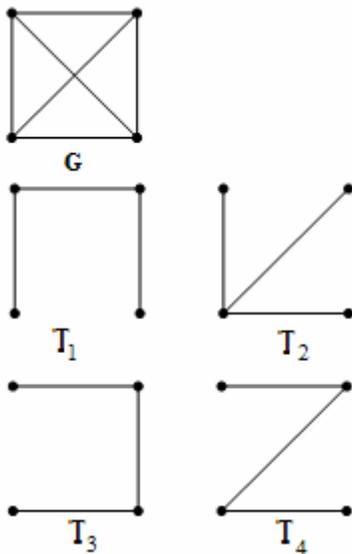


2. Tipe Pohon

2.1. Pohon Merentang

Pohon merentang dari graf terhubung adalah upagraf merentang yang berupa pohon. Pohon merentang diperoleh dengan memutus sirkuit di dalam graf. Sehingga setiap graf terhubung mempunyai paling sedikit satu buah pohon merentang dan graf tak-terhubung dengan k komponen mempunyai k buah hutan merentang yang disebut hutan merentang (*spanning forest*).

contoh :

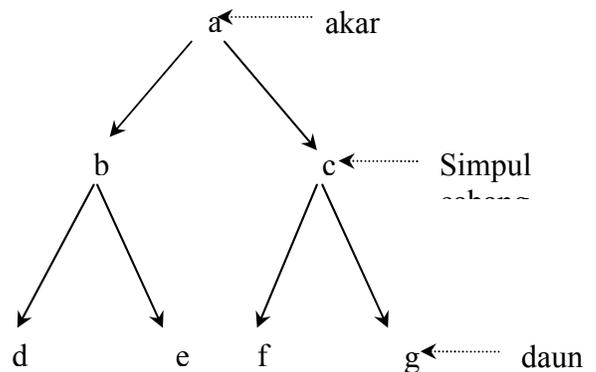


Aplikasi Pohon Merentang

- Jalan terminimum yang menghubungkan semua kota

(a) Jaringan komputer,
(b) Pohon merentang *multicast*

2.2. Pohon Berakar



Suatu graf dinamakan pohon berarah bila arah rusuknya diabaikan dan suatu pohon berarah dinamakan pohon berakar (*rooted tree*) bila ada tepat satu simpul yang berderajat masuk 0, dan semua simpul lain berderajat masuk 1. simpul berderajat masuk 0 dinamakan akar, simpul berderajat keluar 0 dinamakan daun, sedangkan simpul yang berderajat masuk 1 tetapi derajat keluaranya tidak 0 disebut simpul cabang. pada gambar di atas simpul a adalah akar, simpul-simpul b, c adalah simpul cabang sedangkan simpul-simpul d, e, f, dan g adalah daun.

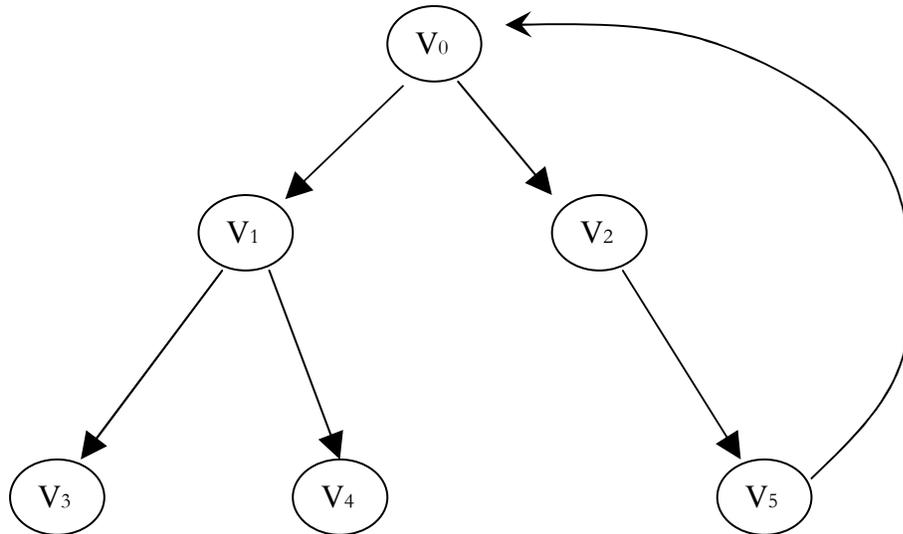
Simpul d disebut anak (*child*) dari simpul b bila ada rusuk dari b ke d, dalam hal ini simpul b disebut ayah (*parent*) dari simpul d. Bila simpul

d memiliki anak lagi maka anak dari simpul d merupakan keturunan (descendent) dari simpul a, b, d, karena ada lintasan berarah dari simpul-simpul tersebut ke simpul anak dari d. Sebaliknya, simpul-simpul a, b, dan d disebut leluhur (ancestor) dari simpul anak dari d. Bila dalam menggambar suatu pohon berakar, anak suatu simpul cabang selalu ditempatkan di bawahnya, maka tanda panah rusuk dapat diabaikan saja.

Teorema 1
geometrik pohon

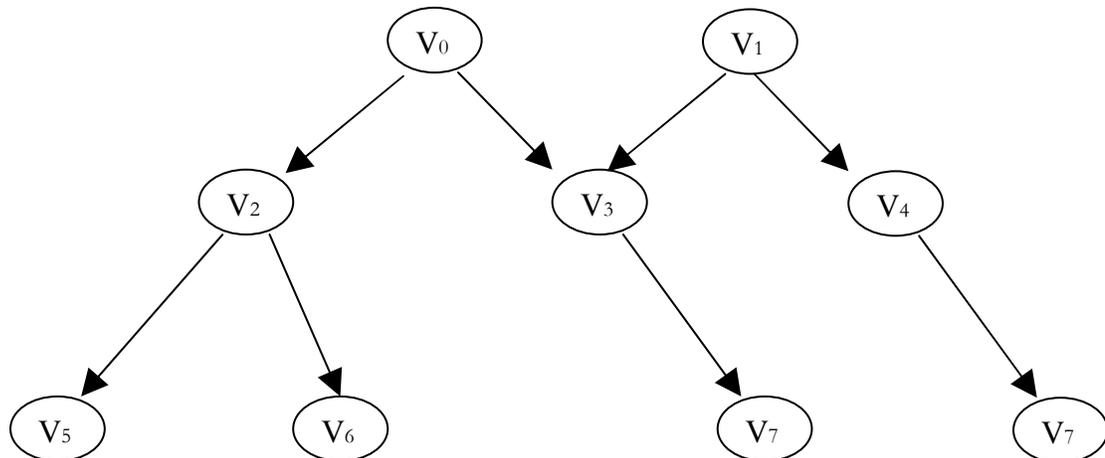
Bila (T, V_0) adalah pohon berakar (T adalah relasi dan V_0 adalah akar) maka:

- Tidak ada siklus dalam T
Gambar di bawah bukanlah suatu pohon berakar karena ada suatu siklus dari V_0 -



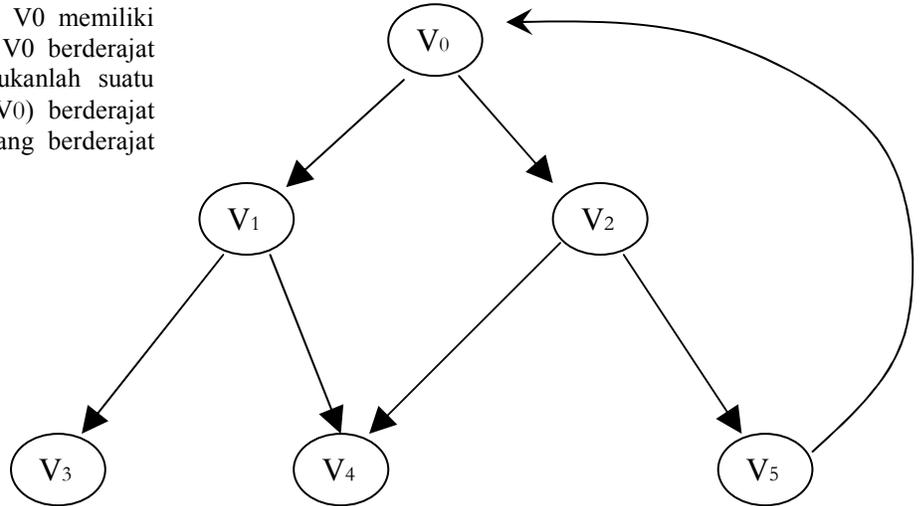
$V_2 - V_3$ kembali ke V_0 .

- V_0 merupakan satu-satunya akar dari T
Tidak ada akar selain V_0 pada suatu pohon.



Gambar di atas bukanlah pohon berakar karena mempunyai 2 akar pohon yaitu V_0 dan V_1 .

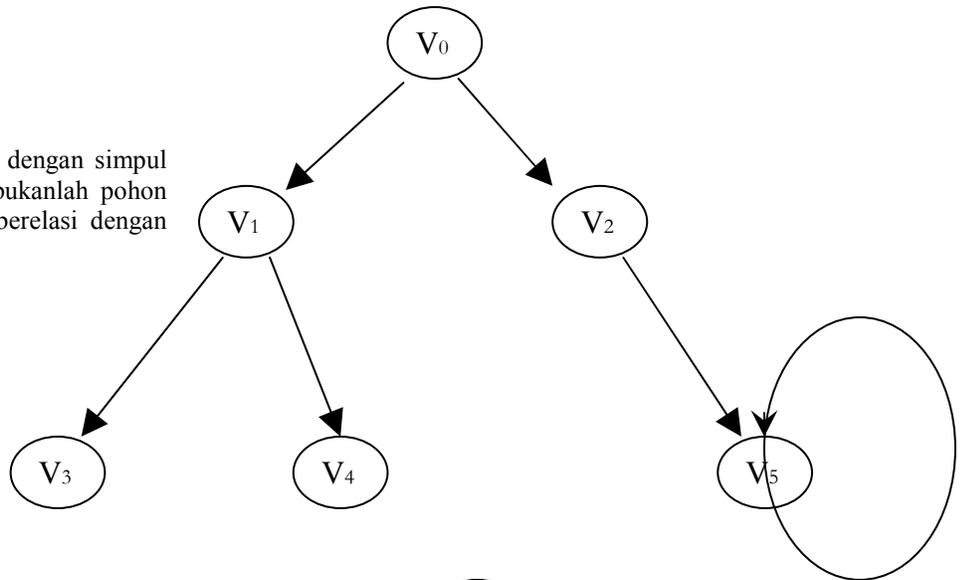
- Tiap simpul di T kecuali V_0 memiliki derajat masuk satu sedangkan V_0 berderajat masuk 0. Gambar di samping bukanlah suatu pohon berakar karena akarnya (V_0) berderajat masuk 1 dan ada simpul lain yang berderajat masuk 2 yaitu V_4 .



Teorema 2

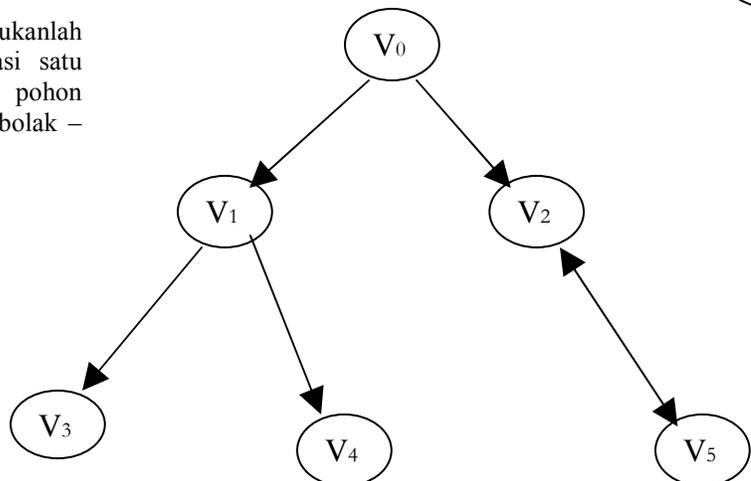
- Irreflexive

Setiap simpul tidak berelasi dengan simpul itu sendiri. Gambar di atas bukanlah pohon berakar karena simpul V_5 berelasi dengan dirinya sendiri.



- Asymmetric

Relasi yang terjadi antar simpul bukanlah merupakan relasi bolak-balik (relasi satu arah). Gambar di samping bukan pohon berakar karena V_2 dan V_5 berelasi bolak-balik.

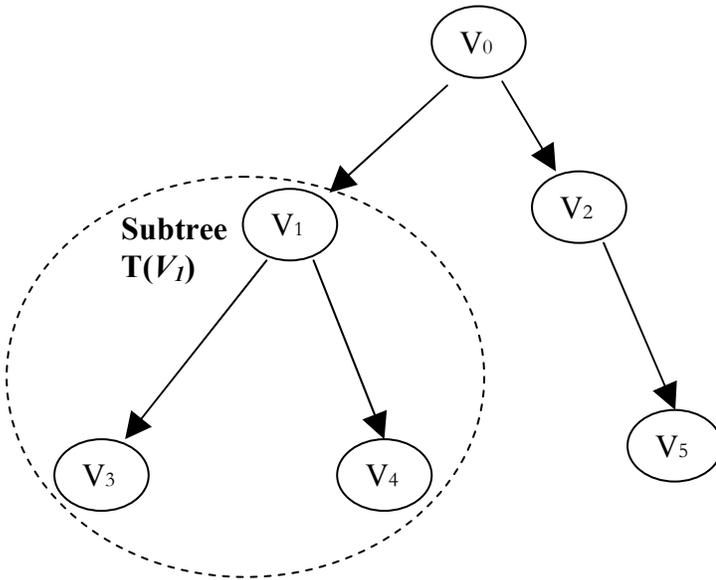


- Jika $(a, b) \in T$ dan $(b, c) \in T$, maka $(a, c) \notin T$
 Bila b berelasi dengan a dan bila c berelasi dengan b, maka c tidak memiliki relasi dengan a.

Teorema 3

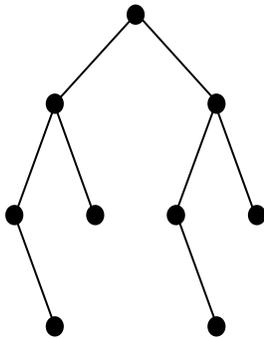
Bila (T, v_0) adalah pohon berakar dan $v \in T$
 maka :

- ♣ $T(v)$ juga pohon berakar dengan akar v .
- $T(v)$ juga subtree dari T dengan awal v .

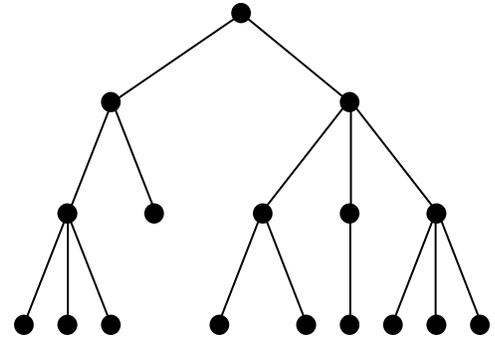


Sebuah pohon berakar yang simpul cabangnya memiliki paling banyak m anak (maksimal), disebut dengan pohon m -er (*m-ary tree*). Dan sebuah pohon m -er dikatakan teratur bila setiap simpul cabangnya tepat memiliki m anak.

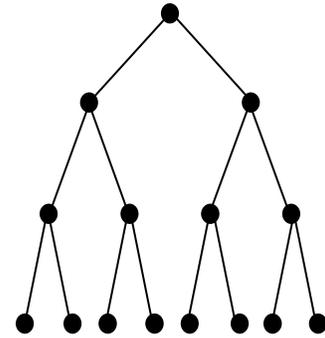
contoh:



(a) Pohon Biner



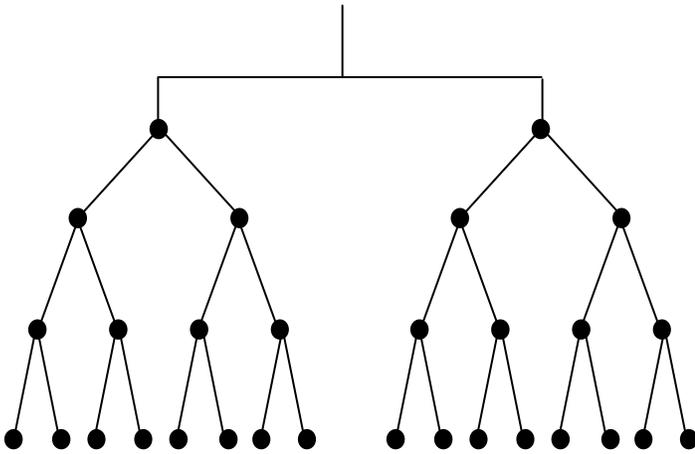
(b) Pohon Ternier



(c) Pohon Biner teratur

Hubungan antara banyaknya simpul cabang dengan banyaknya daun pada suatu pohon m -er teratur bisa kita lihat pada contoh berikut. Misalkan ada sebuah turnamen, pada setiap pertandingan menggunakan sistem gugur. Ada 16 klub peserta turnamen, sehingga pada akhir turnamen hanya tersisa satu tim yang menjadi juara. Bila kita tuangkan jadwal pertandingannya dalam bentuk grafik, ini merupakan contoh sebuah pohon biner teratur dimana setiap simpul cabang tepat memiliki 2 anak. Maka kita dapat menemukan bahwa jumlah pertandingan yang dilangsungkan adalah 15 pertandingan (satu lebih sedikit daripada jumlah klub peserta).

JUARA



Bila i menyatakan banyak simpul cabang, dan t menyatakan banyaknya daun, maka diperoleh hubungan :

$$i = t - 1$$

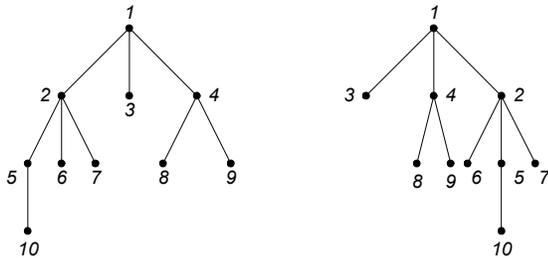
hasil ini dapat diperluas untuk pohon m-er teratur lainnya menjadi :

$$(m-1) i = t - 1$$

2.3. Pohon teratur

Pohon teratur (*ordered tree*), adalah pohon berakar yang urutan anak-anaknya penting, dalam hal ini urutan bisa di definisikan sendiri.

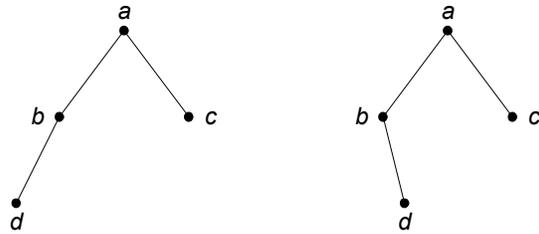
contoh :



2.4. Pohon Biner

Pohon biner merupakan jenis pohon m-er yang simpul cabangnya memiliki maksimal dua anak ($m = 2$). Karena anak dari suatu cabang maksimal hanya dua, maka anak cabang ini dinamakan anak cabang kiri atau anak cabang kanan. Dalam pohon biner, cabang kiri dan kanan ini dibedakan (untuk pohon secara umum tidak). Bukan hanya cabangnya saja, bahkan

urutan cabang kiri atau cabang kanan pun dibedakan. Perhatikan gambar dibawah ini, kedua contoh ini merupakan pohon biner yang berbeda.

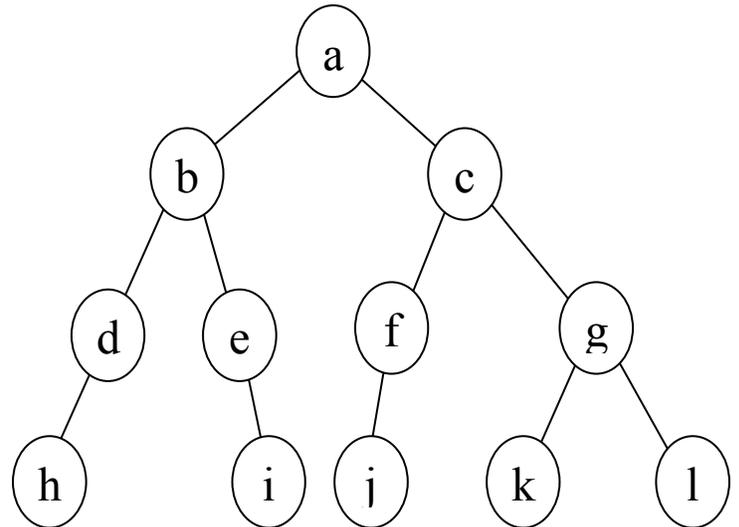


2.4.1. Penelusuran Pohon Biner

Penelusuran pohon biner ada 3 macam :

1. Preorder
 - Kunjungi akar
 - Telusuri cabang kiri
 - Telusuri cabang kanan
2. Inorder
 - Telusuri cabang kiri
 - Kunjungi akar
 - Telusuri cabang kanan
3. Postorder
 - Telusuri cabang kiri
 - Telusuri cabang kanan
 - Kunjungi akar

contoh :

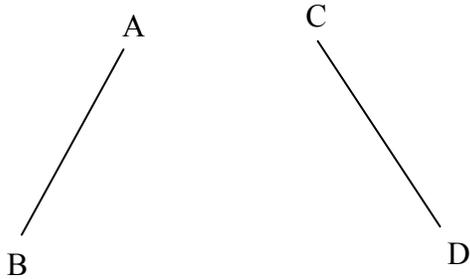


Preorder : a-b-d-h-e-i-c-f-j-g-k-l

Inorder : h-d-b-e-i-a-j-f-c-k-g-l

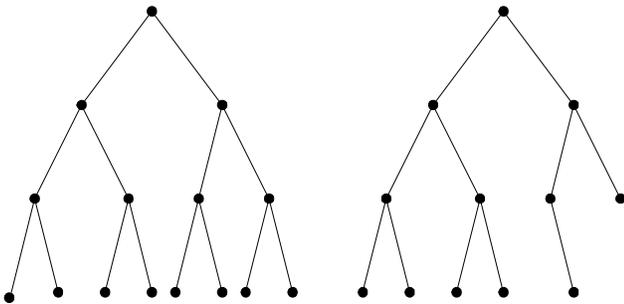
Postorder : h-d-i-e-b-j-f-k-l-g-c-a

2.4.2. Pohon biner condong kiri atau kanan
 pohon condong kiri (uner-left) atau pohon condong kanan adalah pohon yang anaknya hanya menuju salah satu arah. contoh :

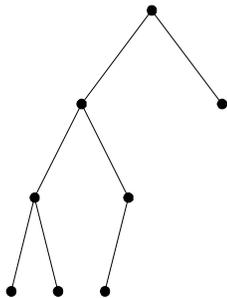


2.4.3. Pohon Biner Seimbang

Pohon biner seimbang adalah pohon dengan tinggi upapohon kiri dan tinggi upapohon kanan yang seimbang, yaitu berbeda maksimal 1. contoh yang benar :



contoh yang salah:



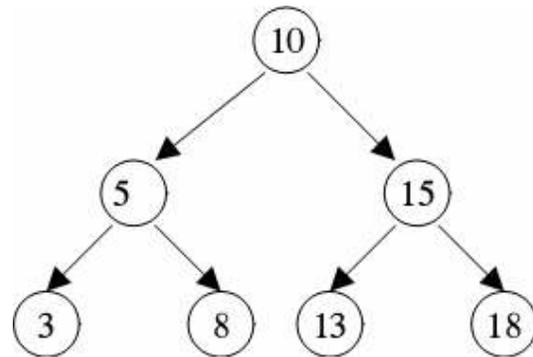
2.4.4. Pohon Biner Terurut

Pohon biner terurut (*binary search tree*) adalah pohon biner dengan aturan dimana bobot anak kiri dari suatu simpul harus lebih kecil dari bobot simpul tersebut dan bobot anak kanan dari suatu simpul harus lebih besar dari bobot simpul tersebut.

Graf biner terurut ini mempunyai kelemahan yaitu tidak bisa mempunyai dua buah simpul yang sama. Kelemahan ini bisa diatasi dengan memberi pencacah (*counter*) di setiap simpulnya.

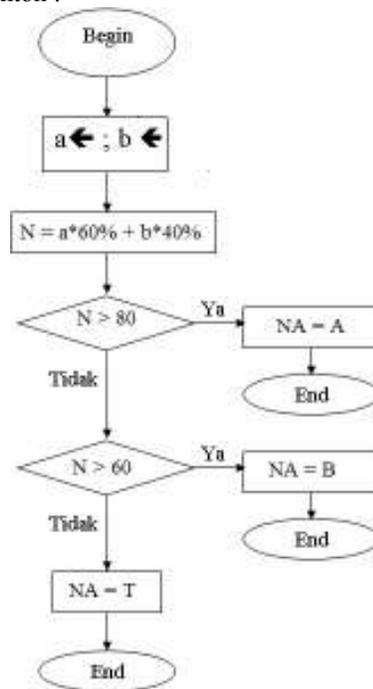
Dalam pemograman software dengan struktur data pohon biner terurut merupakan salah satu struktur untuk mempercepat pencarian karena tidak perlu mengecek semua data.

contoh:



2.4.5 Pohon Berlabel

Pohon berlabel adalah Pohon yang disetiap simpul mempunyai keterangan yang digunakan untuk mengindikasikan bahwa diagram tersebut digunakan untuk tujuan tertentu. Misalnya digunakan untuk merepresentasikan flowchart pada program, digunakan pada penggunaan senarai berantai pada C++, penyelesaian permasalahan sehari – hari dan sebagainya. Label digunakan untuk memberi keterangan pada simpul, simpul tersebut mempunyai fungsi tertentu. contoh :

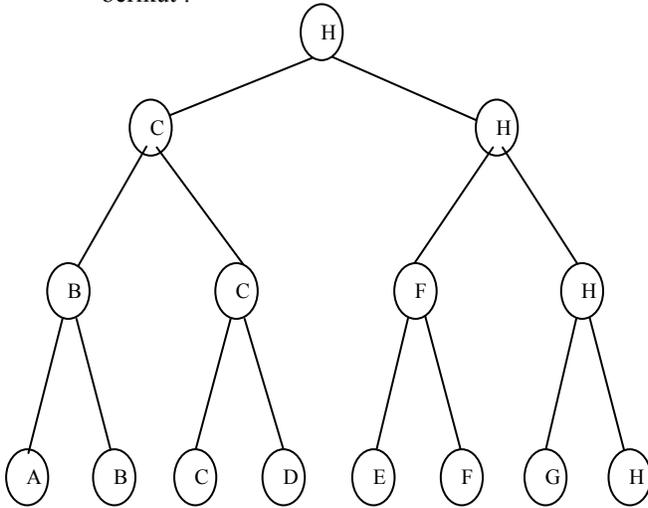


3. Implementasi Pohon

1. Buatlah contoh bagan pertandingan sepakbola dengan sistem gugur yang diikuti oleh 10 klub dimana pada akhir tournament tersisa 1 peserta sebagai juara ! , dan tentukan jumlah pertandingan yang terjadi !

Jawaban :

Bagan pertandingan sistem gugur adalah contoh pohon biner teratur, sehingga kita dapat membuat contoh bagannya sebagai berikut :



Dalam pohon biner berlaku $i = t - 1$, dimana i menyatakan banyak simpul cabang dan t menyatakan banyaknya daun. Dengan rumus di atas maka dapat diketahui banyaknya pertandingan = $8 - 1 = 7$ pertandingan.

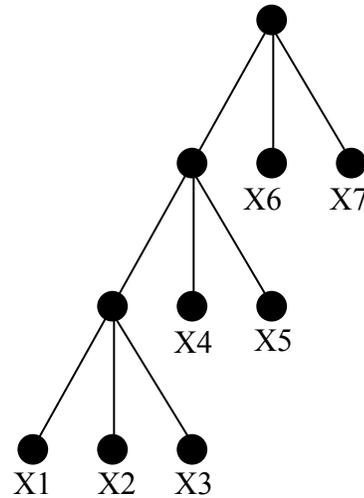
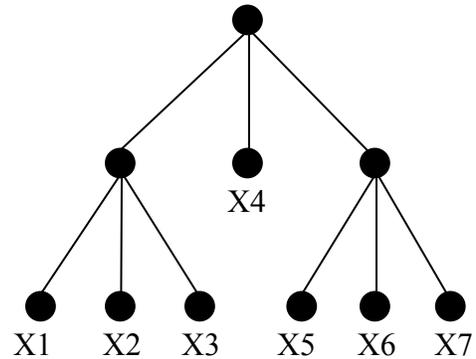
2. Bila sebuah komputer dapat menghitung 3 buah bilangan sekaligus dengan sebuah instruksi , berapa instruksilah yang dibutuhkan untuk menjumlahkan 7 buah bilangan ?

Jawaban :

Untuk pohon m-er berlaku $(m - 1)i = t - 1$ Sehingga bila diterapkan dalam soal ini :

$$\begin{aligned} (m - 1)i &= t - 1 \\ (3 - 1)i &= 7 - 1 \\ i &= 3 \end{aligned}$$

ini berarti ketujuh bilangan tersebut harus dijumlahkan 3 kali dan pohon penjumlahannya dapat digambarkan sebagai berikut :



3. Bila R dianggap suatu pohon dan tiap relasi pada R ditetapkan A maka tentukanlah akar pohon itu , ditentukan

- * $A = \{ a, b, c, d, e, f \}$
- * $B = \{(a,d), (b,c), (c,a), (d,e)\}$

Jawaban :

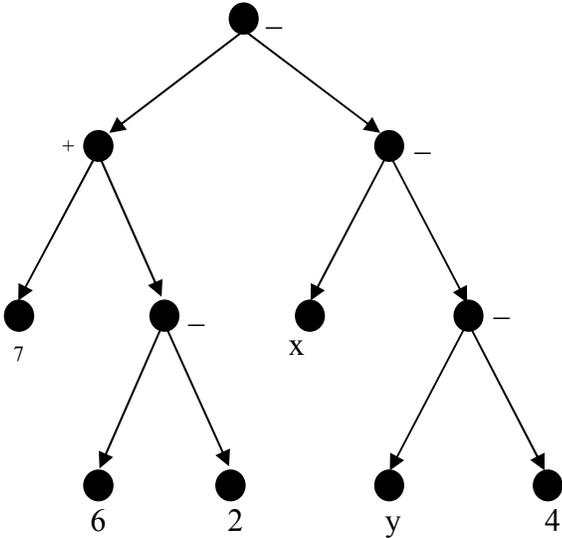
$b \rightarrow c \rightarrow a \rightarrow d \rightarrow e$
berarti akar pohon itu adalah b

4. Buatlah sebuah pohon ekspresi berdasarkan ekspresi prefix di bawah ini :

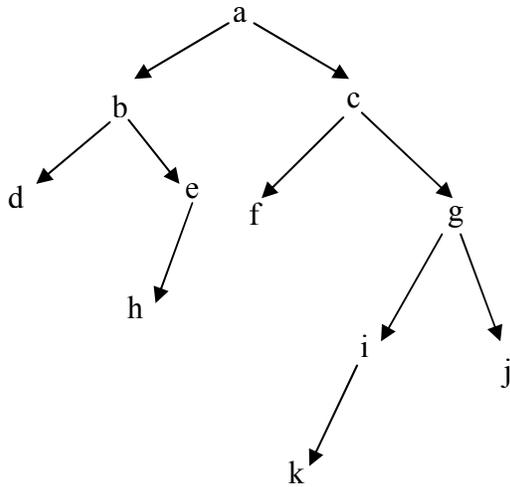
- * $(7 + (6 - 2)) - (x - (y - 4))$

pohon) : d - h - e - b - f -
k - i - j - g - c - a

Jawaban :



5. Telusurilah pohon biner berikut secara preorder, inorder, postorder !



Jawaban :

- * Preorder (akar → cabang kiri → cabang kanan) : a - b - d - e - h - c - f - g - i - k - j
- * Inorder (cabang kiri → akar pohon → cabang kanan) : d - b - h - e - a - f - c - k - i - g - j
- * Postorder (cabang kiri → cabang kanan → akar

6. Gambarkan pohon berlabel untuk mencari akar - akar dari $ax^2 + bx + c = 0$ dengan a, b, dan c merupakan masukan dari user!

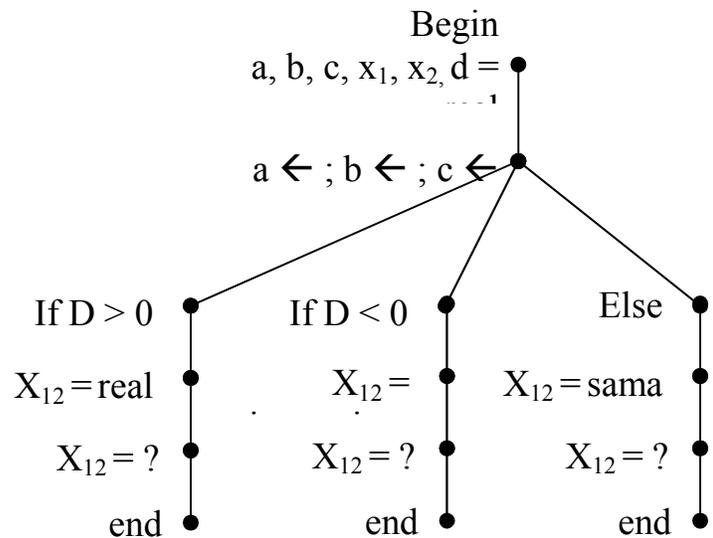
Jawaban:

Pada flowchart dari program pencarian dari akar - akar persamaan kuadrat, yang mendapat input a, b, c dimana ketiga variabel tersebut digunakan pada persamaan $ax^2 + bx + c = 0$. Pertama dilakukan pencarian parameter D untuk menentukan apakah hasilnya berupa bilangan real, imajiner atau real dan sama nilainya. Ketiga pilihan tersebut digunakan sebagai simpul.

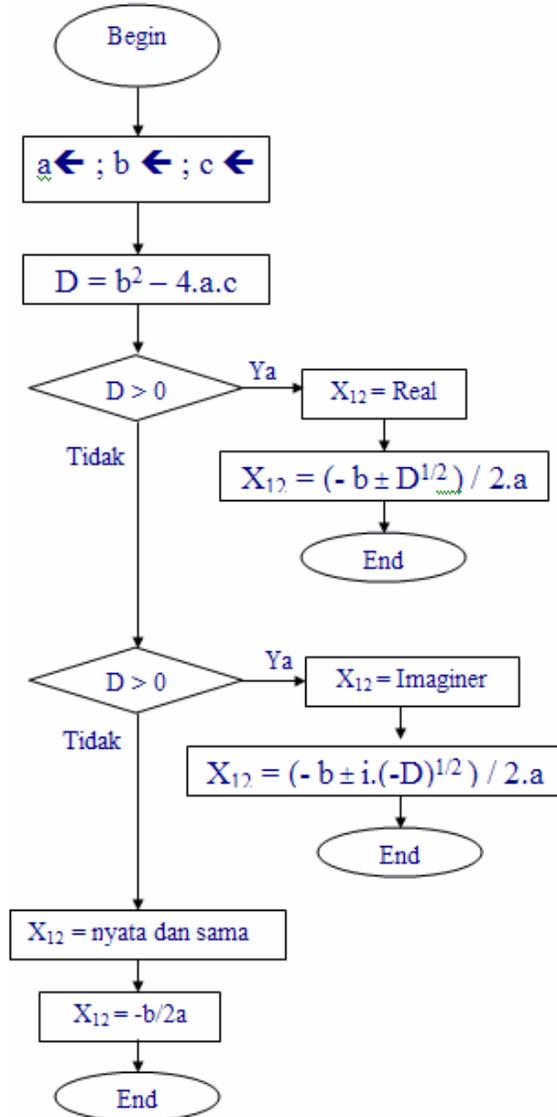
Programnya :

- Mulai
- Pendeklarasian variabel
- $a \leftarrow ; b \leftarrow ; c \leftarrow$ (input dari user)
- $D = b^2 - 4.a.c$
- if $D > 0$ then
 - akar akarnya nyata
 - $X_{1,2} = (-b \pm D^{1/2}) / 2.a$
 - selesai
- if $D < 0$ then
 - akar akarnya imajiner
 - $X_{1,2} = (-b \pm i.(-D)^{1/2}) / 2.a$
 - selesai
- else
 - akarnya sama
 - $X_{1,2} = -b / 2a$
 - selesai

Pohon berlabel sederhana



Pohon berlabel lengkap



5. Kesimpulan

Kesimpulan yang dapat diambil dari studi dan implementasi Pohon adalah graf pohon merupakan graf dengan prinsip untuk menyederhanakan masalah – masalah yang biasa muncul pada graf pada umumnya. Graf pohon juga berfungsi sebagai salah satu struktur yang sangat bagus bagi suatu program.

DAFTAR PUSTAKA

- [1] Munir, Rinaldi. (2006). Diktat Kuliah IF2153 Edisi Keempat. Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung.
- [2] Liem, Inggriani. (2006). Diktat Kuliah IF2182 Struktur Data. Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung.
- [3] <http://www.animal.ahrgr.de/en/Animation7.html>. Diakses pada tanggal 22 Desember 2006 jam 13.00 WIB
- [4] <http://www.dafferiato.web.ugm.ac.id/mystudy>. Diakses pada tanggal 22 Desember 2006 jam 13.00 WIB
- [5] <http://www.mti.ugm.ac.id/~adji/courses/resources/lectures/DiscMathmystudy>. Diakses pada tanggal 22 Desember 2006 jam 13.15 WIB