

# Aplikasi Pohon pada Pohon Binatang (*Animal Tree*)

Cilvia Sianora Putri (13512027)  
Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia  
13512027@std.stei.itb.ac.id

**Abstrak**—Makalah ini membahas tentang studi Pohon dan aplikasinya terhadap pohon binatang (*Animal Tree*). Pohon binatang (*Animal Tree*) merupakan penerapan dari pohon biner teratur. Pohon binatang adalah pohon yang dibentuk sebuah program dari hasil interaksinya dengan user.

**Kata Kunci**—pohon, animal tree.

## I. PENDAHULUAN

Studi pohon (*tree*) adalah salah satu bab dalam Matematika Diskrit. Pohon adalah suatu graf yang memiliki sifat-sifat tertentu. Pohon sudah sering diaplikasikan di kehidupan sehari-hari, seperti untuk membuat struktur organisasi, silsilah keluarga, bagan pertandingan sepak bola, klasifikasi makhluk hidup, dan sebagainya.

Di dunia ini ada berbagai macam hewan. Anak-anak banyak yang menyukai hewan-hewan tersebut. Untuk mengasah ke-kreatif-an anak-anak tersebut, terdapat sebuah program yang didesain untuk permainan anak-anak. Program tersebut berinteraksi dengan user untuk membentuk sebuah pohon yang berisi dengan berbagai macam pertanyaan dan nama binatang yang disebut dengan pohon binatang (*animal tree*).

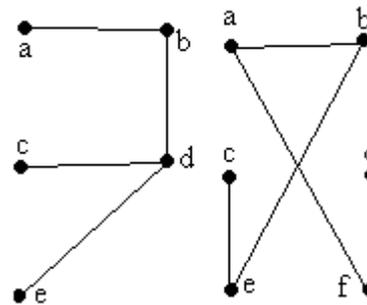
## II. POHON

### II.1 Definisi Pohon

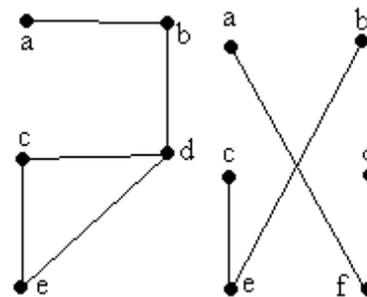
Pohon terdefinisi sebagai graf tak-berarah terhubung yang tidak memiliki sirkuit.

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

1.  $G$  adalah pohon.
2. Setiap pasang simpul di  $G$  terhubung dengan lintasan tunggal.
3.  $G$  terhubung dan memiliki  $n-1$  lintasan.
4.  $G$  tidak memiliki sirkuit dan memiliki  $n-1$  sisi.
5.  $G$  tidak memiliki sirkuit dan penambahan satu sisi akan membuat satu sirkuit saja.
6.  $G$  terhubung dan semua sisinya adalah jembatan. Jembatan adalah sebuah sisi bila sisi tersebut dihapus mengakibatkan graf tersebut terpecah menjadi dua komponen.



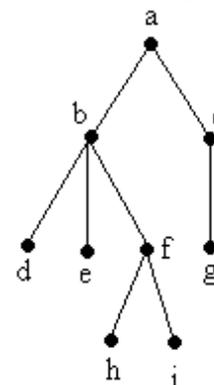
Gambar 1. Contoh pohon



Gambar 2. Contoh graf bukan pohon

### II.2. Pohon berakar

Pohon berakar adalah pohon yang salah satu simpulnya dianggap sebagai akar dan sisi-sisinya diberi arah sehingga menjadi graf berarah. Dan sebagai perjanjian gambar anak panahnya diubah menjadi garis biasa.



Gambar 3. Contoh pohon berakar

### II.3 Terminologi Pohon Berakar

#### 1. Anak (*child/children*) dan Orangtua (*parent*)

Misalkan ada suatu simpul  $x$  dan  $y$  pada suatu pohon. Simpul  $y$  disebut anak dari simpul  $x$  jika terdapat sisi dari simpul  $y$  ke  $x$ . Begitu juga untuk simpul  $x$  sebagai orangtua dari simpul  $y$ . Pada gambar 3,  $b$  dan  $c$  adalah anak-anak dari simpul  $a$ , dan  $a$  adalah orangtua dari simpul  $b$  dan  $c$ .  $d, e, f$  adalah anak-anak dari simpul  $b$ , dan  $b$  adalah orangtua dari simpul  $d, e, f$ .  $g$  adalah anak dari simpul  $c$ , dan  $c$  adalah orangtua dari simpul  $g$ .  $h$  dan  $i$  adalah anak dari simpul  $f$ , dan  $f$  adalah orangtua dari simpul  $h$  dan  $i$ . Simpul  $d, e, g, h, i$  tidak memiliki anak, dan simpul  $a$  tidak memiliki orangtua.

#### 2. Lintasan (*path*)

Lintasan dari simpul  $v_1$  ke simpul  $v_k$  sehingga  $v_1$  adalah orangtua dari simpul  $v_2$  hingga  $v_k$ . Dalam gambar 3, lintasan dari simpul  $a$  ke  $h$  adalah  $a, b, f, h$ . Panjang lintasan adalah banyaknya sisi yang dilalui lintasan tersebut. Panjang lintasan dari simpul  $a$  ke  $h$  adalah 3.

#### 3. Keturunan (*descendant*) dan Leluhur (*ancestor*)

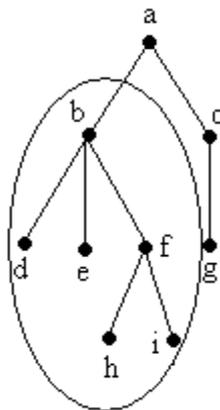
Jika ada lintasan pada pohon yang menghubungkan simpul  $x$  dan  $y$ , simpul  $x$  disebut sebagai leluhur dari simpul  $y$  dan simpul  $y$  disebut sebagai keturunan simpul  $x$ . Dalam gambar 3, simpul  $a$  adalah leluhur simpul  $h$  dan simpul  $h$  adalah keturunan simpul  $a$ .

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

Simpul-simpul yang memiliki orangtua yang sama disebut sebagai saudara kandung satu sama lain. Dalam gambar 3, simpul  $b$  adalah saudara kandung simpul  $c$ , dan begitu pula sebaliknya.

#### 5. Upapohon (*subtree*)

Misalkan  $x$  adalah salah satu simpul dalam suatu pohon. Bila simpul  $x$  tersebut dianggap sebagai akar, pohon yang terbentuk dari akar  $x$  tersebut disebut upapohon dari pohon semula.



Gambar 4. Upapohon dengan simpul  $b$  sebagai akar

#### 6. Derajat (*degree*)

Derajat suatu simpul adalah banyaknya anak yang dimiliki simpul tersebut. Dalam gambar 3, derajat dari simpul  $a$  adalah 2.

#### 7. Daun (*leaf*)

Daun adalah simpul yang berderajat nol atau tidak memiliki anak. Dalam gambar 3, simpul  $d, e, g, h, i$  adalah daun.

#### 8. Simpul dalam (*internal nodes*)

Simpul dalam adalah simpul yang memiliki anak. Dalam gambar 3, simpul  $b, c, f$  adalah simpul dalam.

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

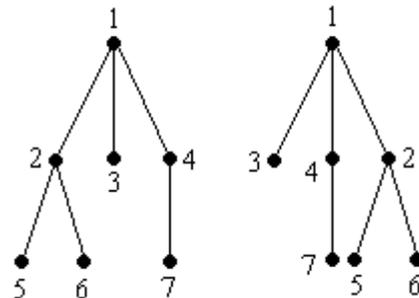
Akar dianggap memiliki aras 0. Simpul lain memiliki aras sebesar jumlah lintasan dari akar ke simpul tersebut. Dalam gambar 3, simpul  $d, e, f$  memiliki aras 2.

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

Tinggi dari suatu pohon adalah aras maksimum dari pohon tersebut. Pohon pada gambar 3 memiliki tinggi 3.

### II.4 Pohon Terurut

Pohon terurut adalah pohon yang pengurutan anak-



anakanya penting.

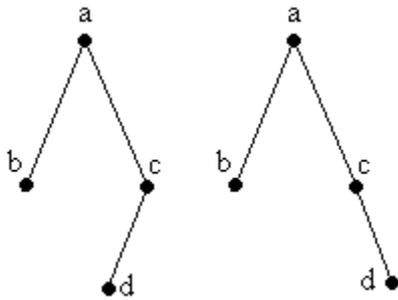
Gambar 5. Dua pohon terurut yang berbeda

### II.5 Pohon $n$ -ary

Pohon  $n$ -ary adalah pohon yang tiap simpulnya memiliki anak maksimum sebanyak  $n$ . Pohon  $n$ -ary penuh atau teratur adalah pohon  $n$ -ary yang setiap simpulnya memiliki anak simpul sebanyak  $n$ .

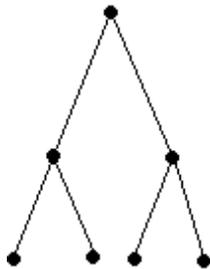
### II.6 Pohon Biner

Pohon biner adalah pohon  $n$ -ary yang memiliki  $n=2$ , yang berarti setiap simpulnya memiliki anak maksimum sebanyak 2. Di dalam pohon biner, anak-anak dari simpulnya disebut anak kiri (*left child*) dan anak kanan (*right child*). Selain itu pohon yang terbentuk dengan akarnya adalah anak kiri disebut sebagai upapohon kiri (*left subtree*) dan pohon yang terbentuk dengan akarnya adalah anak kanan disebut sebagai upapohon kanan (*right subtree*). Karena anak/upapohon kanan berbeda dengan anak/upapohon kiri, pohon biner adalah pohon terurut.



Gambar 6. Dua pohon biner yang berbeda

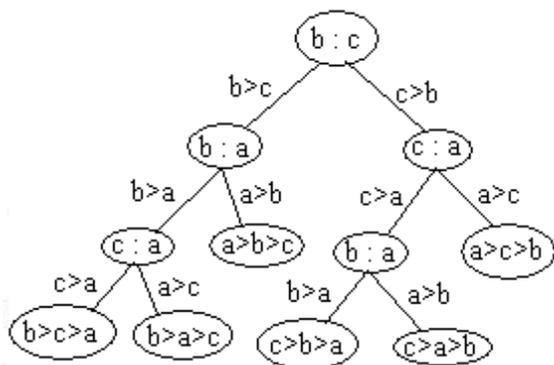
Pohon biner teratur adalah pohon biner yang setiap simpulnya memiliki anak tepat sebanyak 2.



Gambar 7. Contoh pohon biner teratur

### II.7 Pohon Keputusan

Pohon keputusan adalah pohon yang digunakan untuk menggambarkan suatu persoalan yang terdiri dari beberapa keputusan yang akhirnya menuju ke solusi. Tiap simpul dalamnya berisi keputusan dan di simpul daun berisi solusi dari persoalan tersebut. Contohnya pada persoalan mencari maksimum dari dua bilangan  $a$  dan  $b$  dengan pohon keputusan seperti gambar di bawah ini.



Gambar 8. Pohon keputusan dalam pengurutan 3 bilangan

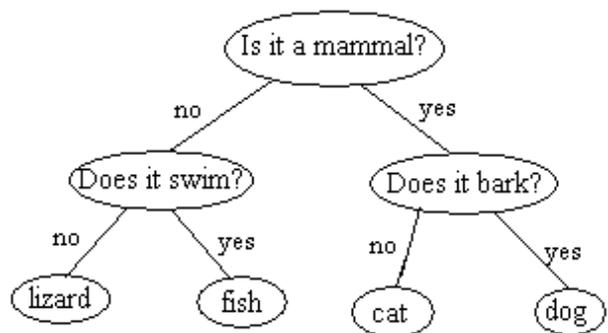
### III. POHON BINATANG (ANIMAL TREE)

Di dunia ini terdapat berbagai macam hewan. Di samping itu, setiap individunya berbeda. Setiap spesies pasti memiliki ciri khasnya masing-masing.

Ada sebuah program yang didesain untuk permainan anak-anak, sebut saja Tebak Binatang (*Animal Guessing*). Program tersebut berinteraksi dengan user untuk

membentuk sebuah pohon yang berisi dengan berbagai macam pertanyaan dan nama binatang yang disebut dengan pohon binatang (*animal tree*). Pohon binatang (*animal tree*) adalah salah satu aplikasi pohon biner. Pohon binatang merupakan pohon biner teratur karena di tiap simpulnya pasti memiliki dua anak simpul. Pohon binatang ini juga merupakan implementasi terhadap pohon keputusan karena untuk menuju ke solusi yang berada di simpul bergantung pada jawaban yes/no dari user. Setiap simpul dalamnya berisi pertanyaan mengenai hewan dan setiap daunnya berisi nama hewan.

Program ini tergolong sederhana. Program dimulai dengan menanyakan pertanyaan yang pertama atau di akar pohon. Program akan menanyakan pertanyaan lagi berdasarkan jawaban dari user. Bila user menjawab iya, program akan menuju ke simpul anak kanan. Sedangkan bila user menjawab tidak, program akan menuju ke simpul anak kiri. Setelah itu, bila masih berada di simpul dalam, program akan menanyakan isi dari simpul tersebut. Selain itu, bila sudah mencapai daun, program akan menebak hewan yang dimaksud dengan nama hewan yang berada pada simpul daun tersebut. Bila program ternyata salah menebak, program akan meminta nama dan pertanyaan untuk hewan yang dimaksud. Pertanyaan tersebut bermaksud untuk membedakan hewan yang dimaksud dengan hewan yang ditebak oleh program. Jadi, pertanyaan tersebut mengandung ciri khas dari hewan yang dimaksud. Lalu, program akan menambahkan pertanyaan dan nama tersebut ke dalam pohon binatang yang sudah ada.



Gambar 9. Contoh pohon binatang

Berikut adalah contoh interaksi antara program dan user dengan pohon binatang pada gambar 9.

Misalkan hewan yang dimaksud adalah ikan (*fish*). Pertama-tama program akan menanyakan pertanyaan yang berada di akar, yaitu “*Is it a mammal?* (Apakah dia mamalia?)”.

Is it a mammal?

User menjawab tidak karena ikan bukanlah mamalia.

Is it a mammal? NO

Lalu, program menuju ke simpul anak kiri karena user menjawab tidak dan menanyakan “Does it swim? (Apakah dia berenang?)”.

Does it swim?

User menjawab iya karena ikan bisa berenang.

Does it swim? YES

Lalu, program menuju ke simpul anak kanan karena user menjawab iya. Saat ini program berada di simpul daun yang berisi *fish*, sehingga program menebak apakah hewan yang dimaksud adalah ikan (*fish*) dengan menanyakan “Is it fish? (Apakah dia ikan?)”.

Is it fish?

User menjawab iya karena hewan yang dimaksud adalah ikan.

Is it fish? YES

Pada akhirnya, program dapat menebak hewan yang dimaksud, sehingga dalam kasus ini program dinyatakan menang. Secara keseluruhan interaksi yang berlangsung adalah sebagai berikut.

Is it a mammal? NO  
Does it swim? YES  
Is it fish? YES

Berikut adalah contoh interaksi lain antara program dan user dengan pohon binatang pada gambar 9.

Kali ini, hewan yang dimaksud adalah *kangaroo*. Seperti sebelumnya, program menanyakan pertanyaan yang berada di akar, yaitu “Is it a mammal? (Apakah dia mamalia?)”.

Is it a mammal?

User menjawab iya karena *kangaroo* adalah mamalia.

Is it a mammal? YES

Lalu, program menuju ke simpul anak kanan karena user menjawab iya dan menanyakan “Does it bark? (Apakah dia menggonggong?)”.

Does it bark?

User menjawab tidak karena *kangaroo* tidak menggonggong.

Does it bark? NO

Lalu, program menuju ke simpul anak kiri karena user

menjawab tidak. Saat ini program berada di simpul daun yang berisi *cat*, sehingga program menebak apakah hewan yang dimaksud adalah kucing (*cat*) dengan menanyakan “Is it cat? (Apakah dia kucing?)”.

Is it cat?

User menjawab tidak karena hewan yang dimaksud adalah *kangaroo*.

Is it fish? NO

Dalam hal ini, program salah menebak hewan yang dimaksud, sehingga program menanyakan nama hewan yang dimaksud.

What is your animal?

Setelah user memberitau nama hewan yang dimaksud, program menanyakan pertanyaan apa yang menjadi ciri khas hewan yang dimaksud.

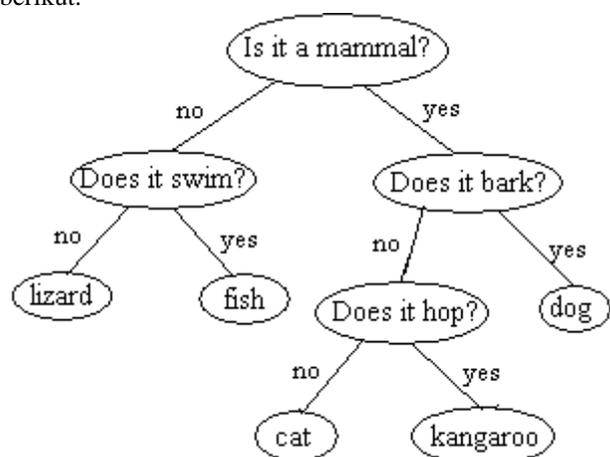
What is your animal? KANGAROO

What is the question about your animal that has a yes answer?

Ciri khas dari *kangaroo* adalah cara Bergeraknya yang tergolong berbeda dari hewan pada umumnya, yaitu berloncat. Jadi, pertanyaan yang cocok untuk *kangaroo* adalah “Does it hop?”.

What is the question about your animal that has a yes answer? DOES IT HOP?

Lalu pertanyaan dan nama tersebut ditambahkan ke dalam pohon binatang yang ada, menjadi seperti gambar berikut.



**Gambar 10. Pohon binatang yang sudah ditambah pertanyaan dan nama baru**

Secara keseluruhan interaksi yang berlangsung adalah

sebagai berikut.

Is it a mammal? YES  
Does it bark? NO  
Is it cat? NO  
What is your animal? KANGAROO  
What is the question about your animal that has a yes  
answer? DOES IT HOP?

## V. KESIMPULAN

Teori pohon ini banyak diterapkan untuk kemudahan dalam kehidupan sehari-hari. Salah satunya adalah sebagai basis bagi permainan *Animal Guessing* yang membantu dalam membangun kreativitas anak-anak. Dalam permainan ini, terjadi interaksi antara user dengan program. Bila program berhasil menebak hewan yang dimaksud, program dinyatakan menang. Di samping itu, bila program salah menebak hewan yang dimaksud, program akan meminta nama dan pertanyaan dari hewan yang dimaksud lalu menambahkannya ke dalam pohon binatang yang dimiliki.

## DAFTAR PUSTAKA

- [1] Munir, Rinaldi. 2006. Diktat Kuliah IF2153 Matematika Diskrit. Departemen Teknik Informatika, Institut Teknologi Bandung.
- [2] <http://www.openbookproject.net/py4fun/animal/animal.htm> diakses tanggal 15 Desember 2013, pukul 13.21 WIB.
- [3] <http://www.smccd.net/accounts/hasson/C++2Notes/Trees.html> diakses tanggal 15 Desember 2013, pukul 13.21 WIB.
- [4] <http://www.openbookproject.net/thinkcs/python/english2e/ch21.html> diakses tanggal 15 Desember 2013, pukul 13.22 WIB
- [5] [http://efsa.sourceforge.net/archive/bielak/animal\\_guess.htm](http://efsa.sourceforge.net/archive/bielak/animal_guess.htm) diakses tanggal 15 Desember 2013, pukul 13.23 WIB

## PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 15 Desember 2013



Cilvia Sianora Putri 13512027