

Aplikasi Pohon dalam Sistem Pakar/ *Expert System* Klasifikasi Hewan dengan Bahasa Prolog

Mochamad Try Yulyanto (13513067)
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
mochamadtry@students.itb.ac.id

Abstract— Di alam semesta ini banyak sekali hewan dan tumbuhan dengan ciri yang berbeda – beda antar satu spesies dengan spesies lainnya. Banyak diantara kita yang tidak mengetahui ciri khusus dari setiap binatang. Para ahli / pakar telah membuat pengklasifikasian terhadap berbagai hewan yang ada di alam berdasarkan ciri khusus dari setiap hewan tersebut.

Memang sangat sulit bagi kita untuk mengingat satu per satu dari jenis hewan dan klasifikasi hewan tersebut tanpa bantuan para ahli/pakar. Karena ahli/pakarlh yang dapat mengetahui secara pasti mengenai ciri dan pengklasifikasian kelas hewan tersebut. Apalagi bagi para orang awam yang tidak tahu sama sekali mengenai ciri dan klasifikasi kelas suatu hewan akan sangat sulit jika harus mengenali cirri khusu suatu hewan dan memasukkan dalam kategori suatu kelas hewan .

Namun, pada era yang serba *modern* ini, banyak digunakan berbagai sistem dalam bentuk aplikasi yang bertujuan untuk memudahkan suatu proses klasifikasi atau untuk memudahkan mendapatkan solusi dari suatu permasalahan . Salah satunya adalah sistem pakar. Dengan sistem pakar ini kita dapat dengan mudah mengetahui solusi suatu permasalahan dan mengetahui kategori dari suatu benda atau penyakit tanpa kita harus menanyakan langsung pada ahli dari bidang tersebut. Misalnya saja ketika kita akan mengetahui atau menentukan kelas dari suatu binatang, apakah binatang tersebut termasuk kedalam mamalia, aves, reptil, *pisces*, *amphibi*, serangga ataupun invertebrata. Kita dapat mengetahui kelas binatang tersebut dengan menggunakan sistem pakar.

Berdasarkan latar belakang diatas, dalam makalah ini akan dibahas mengenai pembuatan sistem pakar klasifikasi suatu hewan. Sistem pakar dibuat dengan basis pengetahuan dari suatu pohon keputusan dan bahasa prolog.

Keywords—Bahasa Prolog, List, Pohon Keputusan, Sistem Pakar.

I. Pendahuluan

Sistem pakar merupakan suatu program komputer intelegensi yang menggunakan basis pengetahuan dan prosedur inferensi untuk memecahkan masalah yang cukup sulit atau untuk mengklasifikasikan suatu kategori yang biasanya hanya dapat diselesaikan oleh seorang ahli atau pakar yang memiliki keahlian pada bidang tersebut.

Dengan sistem pakar ini, setiap lapisan masyarakat dapat menyelesaikan masalahnya atau hanya sekedar mencari suatu informasi berkualitas yang sebenarnya hanya dapat diperoleh dengan bantuan para ahli di bidangnya. Sistem pakar ini juga dapat membantu aktivitas para pakar dengan menjadikan sistem pakar tersbut sebagai asisten yang berpengalaman karena memiliki pengetahuan yang dibutuhkan. Sehingga kesalahan/*human error* dapat dikurangi dengan adanya sitem pakar ini. Dalam penyusunannya, sistem pakar mengkombinasikan kaidah-kaidah penarikan kesimpulan (*inference rules*) dengan basis pengetahuan tertentu yang diberikan oleh satu atau lebih pakar dalam bidang tertentu. Kombinasi dari kedua hal tersebut disimpan dalam komputer, yang selanjutnya digunakan dalam proses pengambilan keputusan untuk penyelesaian masalah tertentu.

Tentu saja sistem pakar ini dibuat dengan banyak variasi sesuai dengan masalah apa yang diselesaikan dengan suatu sistem pakar spesifik dan unik. Karena setiap sistem pakar menyimpan basis pengetahuan tertentu yang berbeda.

Misalnya saja yang akan dibahas didalam makalah ini, yaitu sistem pakar klasifikasi suatu hewan. Sistem pakar ini tentu saja dapat mengetahui berbagai klasifikasi kelas suatu hewan. Sistem pakar tersebut dapat kita buat dengan mengimplementasikan bahasa prolog dengan memasukkan beberapa pengetahuan (*query*) terkait dengan ciri-ciri spesifik dari suatu binatang yang akan dikenali. Tentu saja pengetahuan yang digunakan dapat dipakai untuk membedakan antara satu kelas binatang dengan kelas hewan lainnya. Sehingga sitem pakar yang dihasilkan dapat mengklaisifikasikan kelas suatu hewan dengan akurat.

II. Dasar – Dasar Teori

2.1 Definisi Pohon

Pohon adalah graf tak berarah terhubung yang tidak mengandung sirkuit. Menurut definisi tersebut, maka ada dua sifat penting pada pohon yaitu terhubung dan tidak mengandung sirkuit. Karena definisi pohon diacu dari teori graf, maka sebuah pohon dapat mempunyai hanya sebuah simpul tanpa sebuah sisi. Dengan kata lain, jika $G = (V, E)$ adalah pohon, maka V tidak boleh berupa himpunan kosong, namun E boleh merupakan himpunan kosong. Dengan sifat khusus tersebut, maka sifat – sifat (*properties*) dari suatu pohon dapat dinyatakan sebagai berikut :

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

2.2 Pohon Berakar

Pada kebanyakan aplikasi pohon, simpul diperlakukan sebagai akar (*root*). Sekali sebuah simpul ditetapkan sebagai akar, maka simpul–simpul lainnya dapat dicapai dari akar dengan memberi arah dan sisi–sisi pohon yang mengikutinya. Pohon yang sebuah simpulnya diperlakukan sebagai akar dan sisi –sisinya diberi arah sehingga menjadi graf berarah dinamakan pohon berakar. 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. Berikut ini beberapa jenis dari Pohon Berakar :

1. Pohon Terurut

Pohon berakar yang urutan anak – anaknya penting disebut pohon terurut.

2. Pohon Biner

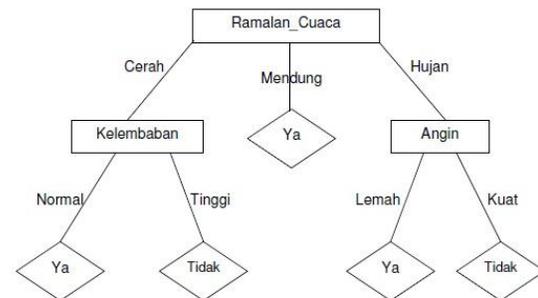
Pohon biner merupakan kasus khusus pohon n -ary jika $n = 2$. Pohon biner adalah pohon yang setiap simpul cabangnya mempunyai paling banyak dua buah anak. Alih – alih menyebutnya anak pertama dan anak kedua dari suatu simpul dalam, kita menyebutnya anak kiri (*left child*) dan anak kanan

(*right child*). Pohon yang akarnya adalah anak kiri disebut upapohon kiri (*left subtree*), sedangkan pohon yang akarnya merupakan anak kanan disebut upapohon kanan (*right subtree*). Karena adanya perbedaan anak/upapohon kiri dan anak/upapohon kanan, maka pohon biner adalah pohon terurut.

2.3 Pohon Keputusan

Pohon keputusan merupakan salah satu terapan pohon biner. Pohon keputusan ini digunakan untuk memodelkan persoalan yang terdiri dari serangkaian keputusan yang mengarah ke solusi. Tiap simpul dalam menyatakan keputusan , sedangkan daun menyatakan solusi.

Pada pemakaian dalam sistem pakar di bahasa prolog, metode pohon keputusan mengubah fakta yang sangat besar menjadi pohon keputusan yang merepresentasikan rule. Pohon keputusan ini salah satu metode klasifikasi yang paling populer karena mudah untuk diinterpretasi oleh manusia. Konsep dari pohon keputusan adalah mengubah data menjadi pohon keputusan dan aturan-aturan keputusan.



Gambar 1 Contoh Pohon Keputusan Ramalan Cuaca

Sumber : <https://aif01.wordpress.com/>

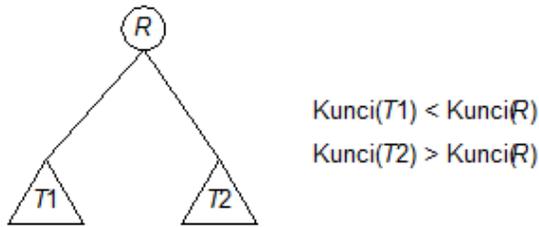
2.4 Pohon Pencarian Biner

Pohon pencarian biner merupakan pohon biner yang setiap kuncinya diatur dalam suatu urutan tertentu. Pohon biner jenis ini dapat dikategorikan sebagai salah satu jenis pohon biner yang penting karena banyak permasalahan yang dapat diselesaikan dengan pohon pencarian biner ini.

Simpul pada pohon pencarian dapat berupa *field* kunci pada data *record*, atau data itu sendiri (dengan catatan setiap data adalah unik). Kunci adalah nilai yang membedakan setiap simpul. Dengan simpul yang lainnya. Kunci harus unik, karena itu tidak ada dua buah simpul atau lebih yang mempunyai kunci yang sama. Dengan ketentuan pengaturan kunci adalah sebagai berikut :

Jika S adalah akar, dan semua kunci yang tersimpan pada setiap simpul tidak ada yang sama, maka :

- a) Semua simpul pada pohon bagian upapohon kiri mempunyai kunci lebih kecil dari kunci(S)
- b) Semua simpul di upapohon kanan mempunyai kunci nilai lebih besar dari kunci (S)



Gambar 2 Skema Pohon Pencarian Biner

Sumber : Diktat Kuliah Matematika Diskrit , Rinaldi

2.5 Sistem Pakar (*Expert System*)

Secara umum, sistem pakar merupakan suatu program komputer intelegensi yang menggunakan pengetahuan dan prosedur inferensi untuk memecahkan masalah yang cukup sulit atau untuk mengklasifikasikan suatu kategori yang biasanya hanya dapat diselesaikan oleh seorang ahli atau pakar yang memiliki keahlian pada bidang tersebut. Dengan sistem pakar ini, orang awam dapat menyelesaikan masalahnya atau hanya sekedar mendapatkan informasi berkualitas yang sebenarnya hanya dapat diketahui melalui para ahli / pakar.

Sistem pakar ini akan menyediakan solusi untuk menyelesaikan masalah – masalah ataupun klasifikasi suatu keanggotaan. Misalnya saja diagnosa suatu penyakit influenza yang biasanya hanya dapat dilakukan oleh ahlinya yaitu dokter. Namun dengan sistem pakar , diagnosa tersebut diselesaikan. Hal ini karena cara kerja sistem pakar memakai model dari pengetahuan manusia yang digunakannya. Selain menyelesaikan suatu masalah, sistem pakar juga dapat membantu mengklasifikasikan suatu keanggotaan. Misalnya saja sistem pakar klasifikasi binatang yang akan dibahas dalam studi kasus laporan ini.

Model tersebut dapat dikenali dalam bentuk yang dapat diproses oleh komputer. Pemodelan dasar ahli terdiri atas *dasar aturan (rule base)* yaitu sebuah program intelegensia yang mempunyai pernyataan berbentuk *if – then* atau aturan yang terhubung dan berulang dalam jumlah besar sebagai dasar pengetahuan dalam sistem dan pemodelan *knowledge frames* yaitu sebuah metode untuk mengorganisasikan pengetahuan dari sistem pakar kedalam potongan besar, hubungannya berdasarkan karakteristik yang diberikan yang ditentukan oleh pengguna.

2.6 Bahasa Prolog

Prolog adalah singkatan dari *PROgramming in LOGic*. Bahasa ini pertama kali dikembangkan oleh Alain Colmerauer dan P. Roussel di Universitas Marseilles Perancis pada tahun 1972. Bahasa ini dikembangkan berdasarkan Kalkulus Predikat Orde Pertama (Relational Logic). Bahasa prolog ini biasanya dikaitkan dengan kemampuan berpikir secara logika dan merupakan bahas pemograman untuk perhitungan simbolik tak berangka rumit. Bahasa Prolog paling sesuai untuk menyelesaikan masalah yang berkaitan dengan objek dan hubungan antara objek, masalah persamaan corak/*pattern*, masalah penjejakan *flash back* dan masalah yang informasinya tidak lengkap.

Algoritma dalam Bahasa Prolog terdiri dari logika dan kontrol. Logika merupakan fakta dan peraturan yang menerangkan apa yang seharusnya dilakukan oleh algoritma. Sedangkan kontrol merupakan algoritma yang diimplementasikan dengan menggunakan peraturan. Sintaks yang dibentuk dalam Prolog adalah dalam bentuk klausa atau formula *First Order Predicate Logic*.

Dalam pembuatan sistem pakar kali ini akan digunakan sistem pemodelan *rule base* untuk menyelesaikan sistem pakar klasifikasi kelas hewan tersebut. Sistem pemodelan tersebut digunakan karena disini kita menggunakan pohon keputusan dimana program nantinya akan mengeluarkan beberapa aturan/pengetahuan yang menyatakan kevalidan kepemilikan organ / sistem organ binatang tersebut.

Pengetahuan tersebut dikodekan menjadi aturan analisis kasus **if ... then**. Hal ini mengindikasikan kesimpulan didapat dari aturan yang terpenuhi lebih dulu. Misalnya saja kita akan mengetahui kelas suatu binatang X. Mekan kita akan mengecek beberapa aturan yang telah disediakan oleh sistem pakar melalui pohon keputusan. **if** (kondisi) **then** (kelas), namun sistem pakar bisa saja mengani dua kasus/aturan yang harus dipenuhi binatang tersebut agar binatang tersebut dapat masuk dalam sistem klasifikasi hewan. misalnya :

if (kondisi) **then**
if (kondisi) **then** (kelas)

Bahasa prolog memiliki beberapa bagian khusus yaitu :

- a. Fakta dan Relasi
Prolog terdiri dari kumpulan data-data objek yang merupakan suatu fakta. Fakta dibedakan menjadi dua macam yaitu menunjukkan relasi dan menunjukkan sifat

Fakta	Prolog
Slamet adalah ayah Amin	ayah(slamet, amin)
Ani adalah seorang wanita	wanita(ani).

Tabel 1 Contoh Fakta dalam Prolog

b. Aturan / rules

Aturan merupakan suatu pernyataan yang menunjukkan bagaimana fakta-fakta berinteraksi satu dengan yang lain untuk membuat suatu kesimpulan.

Fakta dan aturan	Prolog
F : Try suka apel	suka :- (try,apel).
A : Yuli suka sesuatu yang disukai Try	suka(yuli,x) :- suka(try,x).

Tabel 2 Contoh Aturan dalam Prolog

2.7 Struktur Program Prolog

Ada dua cara terkait tata letak struktur program bahasa prolog yaitu :

- a. Organisasi cara pertama, fakta dan aturan dikelompokkan secara **terpisah**. Seluruh fakta dikelompokkan tersendiri, demikian juga aturan dikelompokkan tersendiri, tetapi penulisan fakta mendahului aturan.
- b. Organisasi cara kedua, fakta dan aturan dikelompokkan secara **tidak terpisah**. Pengelompokan didasarkan hanya atas nama predikat yang sama, atau lebih luasnya: didasarkan atas sekelompok fakta dan aturan yang mewakili suatu persoalan. Tetapi dalam penulisannya tetap fakta mendahului aturan.

Cara Pertama	Cara Kedua
/* FAKTA */	/* FAKTA */
/* Spesifikasi Fakta ke-1 */	/* Spesifikasi Fakta ke-1 */
.....
/* Spesifikasi Fakta ke-n */	/* Spesifikasi Fakta ke-n */
.....
/* ATURAN */	/* ATURAN */
/* Spesifikasi Aturan ke-1 */	/* Spesifikasi Aturan ke-1 */
.....
/* Spesifikasi Aturan ke-n */	/* Spesifikasi Aturan ke-n */
.....
	/* FAKTA DAN ATURAN */
	/*Spesifikasi Fakta & Aturan ke-1
	*/

	/*Spesifikasi Fakta & Aturan ke-n
	*/

Tabel 3 Struktur Program Prolog

2.8 Hubungan Sistem Pakar dengan Prolog

Salah satu aplikasi dari prolog adalah sistem pakar. Prolog dapat dipergunakan untuk membuat sistem pakar karena bahasa prolog akan menyimpan berbagai fakta dan rule. Hal ini sesuai dengan definisi sistem pakar yang akan menyimpan beberapa informasi berupa fakta mengenai identitas suatu ilmu pengetahuan dari ahli /pakar agar dapat menjawab persoalan atau identifikasi suatu persoalan tanpa bantuan dari ahli tersebut.

III. Program Sistem Pakar
Klasifikasi Hewan

3.1 Pohon Keputusan Klasifikasi Hewan

Pohon keputusan ini akan digunakan dalam basis pengetahuan pada pembuatan sistem pakar penentuan klasifikasi binatang kali ini. Di dalam program, pohon keputusan ini akan digunakan sebagai penentu suatu hewan termasuk ke dalam klasifikasi hewan mamalia, bird, insect, invertebrata, fish, amphibian ataupun reptil.

Pohon keputusan ini diimplementasikan secara berurut artinya program akan mengecek dari aras paling kecil dalam pohon dan akan terurut sampai aras paling tinggi (simpul daun). Berikut pohon keputusan klasifikasi hewan :

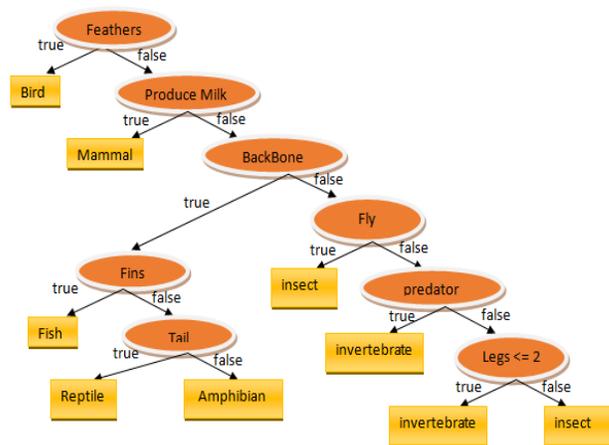


Diagram Pohon Keputusan

Penjelasan : Pohon tersebut nantinya akan dipakai di program sebagai penentu klasifikasi suatu hewan. Program akan mengecek dari beberapa fakta yang telah dibuat sesuai pohon keputusan dan akan mengecek dari akar sampai simpul daun dari pohon keputusan tersebut. Misalnya fakta pertama yang dicek adalah feathers. Program akan mengeluarkan rule yang menanyakan pada pengguna apakah hewan tersebut memiliki bulu atau tidak. Jika benar, maka program akan mengeksekusi dan mengambil keputusan langsung bahwa hewan tersebut termasuk ke dalam klasifikasi hewan bird. Jika bernilai salah maka program akan terus cek ke anak pohon kanan selanjutnya untuk dicek. Proses ini terus berulang dan akan berakhir di simpul daun dari sub pohon kanan.

3.2 Basis Pengetahuan Sistem Pakar

Basis pengetahuan yang akan digunakan dalam sistem pakar dibatasi dalam 6 klasifikasi hewan dengan cirri sebagai berikut :

- a. Bird (Aves)
Untuk mengecek suatu hewan jenis ini, akan digunakan ciri yaitu jika hewan tersebut memiliki bulu (*feathers*)
- b. Mammal (Mamalia)
Seperti yang diketahui bahwa dalam sistem pakar akan digunakan beberapa fakta yang mengacu dari keterangan para ahli. Untuk membuktikan hewan tersebut termasuk mamalia, digunakan fakta yaitu hewan mamalia tersebut harus bisa menghasilkan susu.
- c. Fish
Untuk mengecek suatu hewan termasuk ke dalam kelas fish, akan digunakan 2 fakta yaitu memiliki tulang belakang (*Backbone*) dan memiliki sirip (*fins*).
- d. Reptile
Untuk kategori hewan yang satu ini akan di cek fakta yaitu harus bernilai true memiliki tulang belakang, bernilai false memiliki sirip dan bernilai true untuk fakta memiliki buntut
- e. *Amphibian*
Untuk kategori hewan yang satu ini akan di cek fakta yaitu harus bernilai true memiliki tulang belakang, namun bernilai false dalam fakta memiliki sirip (*fins*) dan fakta memiliki buntut (*tail*).
- f. *Insect*
Untuk klasifikasi hewan serangga/*insect* fakta yang digunakan adalah jika ketika ditanyakan dari berbagai basis pengetahuan, fakta tersebut bernilai true untuk bisa terbang (*fly*).
- g. Invertebrate
Untuk klasifikasi Invertebrate, bernilai true untuk fakta predator atau kaki ≤ 2

Untuk mempermudah mengenali basis pengetahuan yang digunakan, berikut tabel dari basis pengetahuan dari klasifikasi hewan :

Klasifikasi Hewan	Fakta bernilai true	Fakta bernilai false
<i>Bird (Aves)</i>	<i>Has_Feathers</i>	-
<i>Mammal</i>	<i>Produce_Milk</i>	<i>Has_Feathers</i>
<i>Fish</i>	<i>Backbone</i> <i>Has_Fins</i>	<i>Has_Feathers</i> <i>Produce_Milk</i>
<i>Reptile</i>	<i>Backbone</i> <i>Has_Tail</i>	<i>Has_Feathers</i> <i>Produce_Milk</i> <i>Has_Fins</i>
<i>Amphibian</i>	<i>Backbone</i>	<i>Has_Feathers</i> <i>Produce_Milk</i> <i>Has_Fins</i> <i>Has_Tail</i>
<i>Insecta</i>	<i>Can_Fly</i>	<i>Has_Feathers</i> <i>Produce_Milk</i> <i>Backbone</i>

		<i>Predator</i> <i>Has_Legs<2</i>
<i>Invertebrate</i>	<i>Predator</i> <i>Has_Legs<2</i>	<i>Has_Feathers</i> <i>Produce_Milk</i> <i>Backbone</i>

Tabel 4 Basis Pengetahuan Sistem Pakar

3.3 Pemakaian Prolog dalam Prolog

Seperti yang telah dijelaskan sebelumnya, dalam pembuatan sistem pakar digunakan program yang menggunakan bahasa prolog. Sintaks program akan diperlihatkan seperti pada potongan program untuk sistem pakar klasifikasi hewan berikut ini yang dibuat dengan menggunakan gnu prolog :

```

/* expert system */

mulai :- write('Mengecek Klasifikasi Hewan') ,nl, nl,
nama (Peliharaan),

/* nama peliharaan */
nama(Pet) :- write('nama hewan peliharaan :
'),read(Pet),nl.

/* Klasifikasi Hewan */
klasifikasi(burung)           :- burung, !.
klasifikasi(mammalia)        :- mammalia, !.
klasifikasi(X)                :- vertebrata(X),!.
klasifikasi(X)                :-
invertebrata(X), !.
klasifikasi(unclassified).

/* Rule */
burung :- verifikasi(punya_bulu),!.
mammalia :- verifikasi(menghasilkan_susu),!.
vertebrata(fish) :- ver(P) ,verifikasi(P), verifikasi
(punya_sirip),!.
vertebrata(reptil) :- ver(P),verifikasi(P), verifikasi
(punya_ekor),!.
vertebrata(amphibi) :- ver(P),verifikasi(P),!.
invertebrata(serangga) :- verifikasi(bisa_terbang),!.
invertebrata(invertebrata) :- verifikasi(predator),!.
invertebrata(invertebrata) :- verifikasi (kaki_2 ) , !.
invertebrata(serangga).

/* LIST */

concat([],List,List).
concat([H|T], List, [H|L]) :- concat(T,List,L).

printList([]).
printList([X|Tail]) :- write('-'), write(X),nl,
printList(Tail).

printtidakList([]).
printtidakList([X|Tail]) :- write('-'), write('tidak_'),

```

```

write(X),nl, printtidakList
(Tail).

:- dynamic(bukankarakter/1).
bukankarakter ([]).
:- dynamic(karakter/1).
karakter([]).

addkarakter(X) :- karakter(L), concat([X],L,A),
retractall(karakter(L)), asserta(karakter(A)).
addbukankarakter(X) :- bukkarakter(L),

:- dynamic(benar/1,tidak/1).

/* verifikasi hewan */
ver(punya_tulang_belakang).
tanya(X) :- write('apakah hewan ini '),
write(X),
write(' (y/n) ? '),
read(Jawaban),
nl,
((Jawaban == yes ; Jawaban == y )->
asserta(benar(X)),addkarakter(X);

verifikasi(H) :- (benar(H)->true; tidak(H)->fail ;
tanya(H)).
/* set up */
undo :- retract(benar(_)),fail.
undo :- retract(tidak(_)),fail.
undo :- retract(salahmasukan(_)),fail.
undo :- retractall(karakter(_)), asserta
(karakter([])),fail.
undo :- retractall(bukankarakter(_)), asserta
(bukankarakter([])),fail.
undo.

```

3.4 Penjelasan Sintaks Program

- **klasifikasi(burung) :- burung, !.**
 Dalam program prolog akan ada aturan yang berfungsi memverifikasi klasifikasi burung yang diambil dari pohon keputusan. Output dari program :
 mulai :- klasifikasi(Hewan),write('klasifikasi hewan tersebut : '),nl,write(Hewan),nl
- **klasifikasi(X) :- vertebrata (X),!.**
 Aturan tersebut dipakai pada hewan vertebrata dengan X dapat bernilai fish, reptile ataupun amphi. Sama seperti bird dan mamalia, hewan dengan klasifikasi ini akan dicek berdasarkan aturan yang telah disediakan pada pohon keputusan. Output pada program penggunaan rule tersebut :
 mulai :- klasifikasi(Hewan),write('klasifikasi hewan tersebut : '),nl,write(Hewan),nl

- **burung :- verifikasi(punya_bulu),!.**
 Rule tersebut digunakan untuk memverifikasi apakah hewan tersebut burung atau bukan berdasarkan pohon keputusan dengan aturan verifikasi(punya_bulu). Untuk verifikasi tersebut benar atau salah melalui input user digunakan :
 verifikasi(H) :- (benar(H) ->true; tidak(H)->fail ;tanya(H)).
 Perhatikan bahwa dalam rule tersebut setelah cek verifikasi, akan ada tanda cut sehingga ketika verifikasi bernilai true maka akan langsung diberi kesimpulan hewan tersebut adalah burung.
- **invertebrata(invertebrata) :- verifikasi (predator) ,!.**
 Rule tersebut digunakan untuk memverifikasi apakah hewan tersebut invertebrata atau bukan berdasarkan pohon keputusan dengan aturan verifikasi (predator).Verifikasi ini dilakukan ketika pengguna menyatakan false di 4 tahap verifikasi sebelumnya Perhatikan bahwa dalam rule tersebut setelah cek verifikasi, akan ada tanda cut sehingga ketika verifikasi bernilai true maka akan langsung diberi kesimpulan hewan tersebut adalah invertebrata.
- **Pengecekan Basis Pengetahuan**

Rule	Penjelasan
<pre> :- dynamic (benar/1,tidak/1). ... tanya(X) :- write('apakah hewan ini '), write(X), write(' (y/n) ? '), read(Jawaban), nl, ((Jawaban == yes ; Jawaban == y)-> asserta(benar(X)),addkara kter(X) ; asserta(tidak(X)), addbukankarakter(X),fail) ,! verifikasi(H) :- (benar(H)- >true; tidak(H)->fail ;tanya(H)). undo :- retract(benar(_)),fail. undo :- retract(tidak(_)),fail. </pre>	<p>Rule tersebut digunakan untuk memverifikasi seluruh hal yang harus diverifikasi. Kami set secara dynamic, hal ini bertujuan agar dapat diisi secara dinamis (berubah-ubah) ketika program dijalankan. Verifikasi pada awalnya tidak bernilai true maupun false, namun dia akan menggunakan tanya(H) untuk memanggil rule verifikasi tanya (X). Lalu akan dicek jika bernilai benar, akan dimasukkan ke asserta (benar(X)) jika false, akan dimasukkan ke asserta (tidak(X)). Barulah verifikasi(H) memiliki nilai true atau false.</p>

Tabel 3 Basis Pengetahuan

- Penyimpanan ke List

Rule	Penjelasan
concat([],List,List). concat([H T], List, [H L]) :- concat(T,List,L).	Rule tersebut digunakan untuk menyimpan klasifikasi kedalam suatu list. Terdiri atas dua penyimpanan yaitu bukan karakter dan karakter dari hewan tersebut yang sudah diuji dengan sistem verifikasi sebelumnya.
printList([]). printList([X Tail]) :- :- dynamic(karakter/1). karakter([]).	Lalu akan digunakan sebagaimana prosedur dari list yaitu pada add karakter(L) dan bukannya karakter(L).
addkarakter(X) :- karakter(L), concat([X],L,A), retractall(karakter(L)), asserta(karakter(A)). addbukankarakter(X) :- bukankarakter(L), concat([X],L,A), retractall(bukankarakter(L)), asserta(bukankarakter(A)).	

Tabel 4 Penyimpanan List

3.5 Analisis dan Pembatasan Eksekusi Program

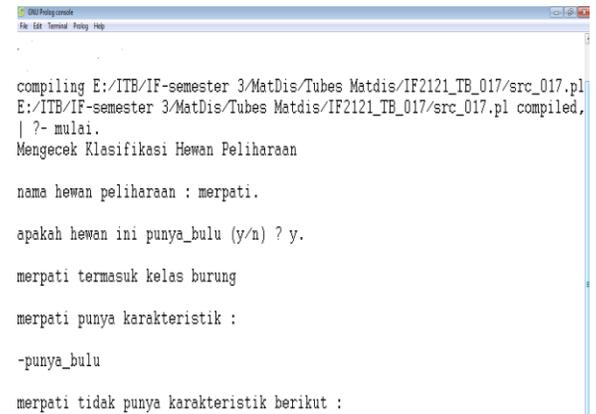
Program prolog yang dibuat harus dapat menampilkan pertanyaan yang sesuai dengan alur pohon keputusan yang telah diilustrasikan pada bagian pembahasan sebelumnya. Lalu pengguna akan memberikan masukan / jawaban dari pertanyaan yang diberikan program. Pada akhirnya (*Goal*) harus dapat menentukan kelas dari suatu hewan meliputi : *bird, mammal, reptile, fish, amphibian, insect*, ataupun *invertebrate*. Selain itu, pada (*Goal*) sistem pakar yang dibangun juga memberikan penjelasan bagaimana bisa didapat solusi yang diberikan. Penjelasan ini diberikan pada saat sitem pakar tersebut mendapatkan solusi kelas hewan dari persoalan pengguna. Penjelasan solusi tersebut terdiri atas ciri yang dimiliki hewan tersebut sehingga hewan tersebut dapat ditetapkan merupakan klasifikasi tertentu yang terdapat pada pohon keputusan. Dengan penjelasan sistem pakar disimpan dalam suatu list.

Sistem pakar yang dibuat dapat membantu masyarakat untuk mengetahui klasifikasi beberapa binatang. Pengguna mendapat tambahan ilmu pengetahuan tanpa harus bertanya terlebih dahulu kepada pakar yang bersangkutan terkait klasifikasi hewan. Batasan Persoalan sistem pakar hanya akan menangani beberapa ciri hewan yang disediakan di pohon keputusan tidak memenuhi semua ciri secara spesifik.

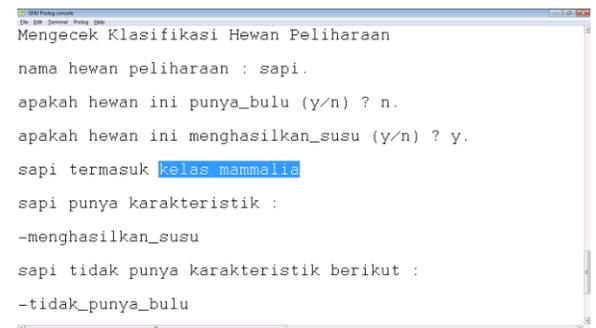
3.6 EKSEKUSI PROGRAM

Program akan dimulai dengan ketik kata “mulai.”. Lalu program akan minta input jenis hewan yang akan dicek klasifikasinya. Lalu akan memulai verifikasi dengan menanyakan beberapa pertanyaan yang sudah di set pada prolog sesuai basis pengetahuan dari pohon keputusan. Setelah melakukan beberapa verifikasi, program akan menunjukkan hewan tersebut termasuk kelas hewan apa dan alasan mengapa hewan tersebut termasuk dalam klasifikasi tersebut yang diambil dari list.

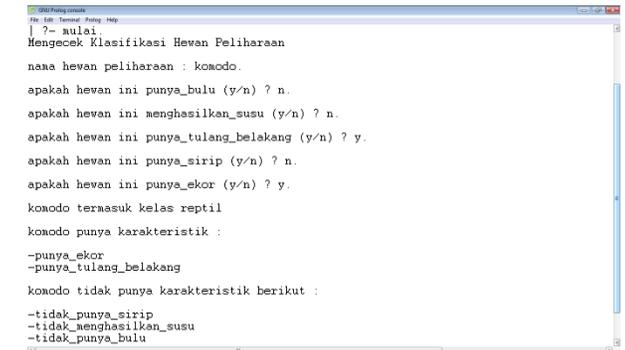
Berikut beberapa contoh hasil eksekusi program



Gambar 3 Klasifikasi Burung / Bird



Gambar 4 Klasifikasi Mamalia



Gambar 5 Klasifikasi Reptil

```

GNU Prolog console
File Edit Terminal Prolog Help
|-- mulai.
Mengecek Klasifikasi Hewan Peliharaan

nama hewan peliharaan : lalat.

apakah hewan ini punya_bulu (y/n) ? n.
apakah hewan ini menghasilkan_susu (y/n) ? n.
apakah hewan ini punya_tulang_belakang (y/n) ? n.
apakah hewan ini bisa_terbang (y/n) ? t.
salah masukan
apakah hewan ini bisa_terbang (y/n) ? y.

lalat termasuk kelas serangga

lalat punya karakteristik :
-bisa_terbang

lalat tidak punya karakteristik berikut :
-tidak_punya_tulang_belakang
-tidak_menghasilkan_susu
-tidak_punya_bulu

```

Gambar 6 Klasifikasi Insecta / serangga

```

GNU Prolog console
File Edit Terminal Prolog Help
|-- mulai.
Mengecek Klasifikasi Hewan Peliharaan

nama hewan peliharaan : katak.

apakah hewan ini punya_bulu (y/n) ? n.
apakah hewan ini menghasilkan_susu (y/n) ? n.
apakah hewan ini punya_tulang_belakang (y/n) ? y.
apakah hewan ini punya_sirip (y/n) ? n.
apakah hewan ini punya_ekor (y/n) ? n.

katak termasuk kelas anphibi

katak punya karakteristik :
-punya_tulang_belakang

katak tidak punya karakteristik berikut :
-tidak_punya_ekor
-tidak_punya_sirip
-tidak_menghasilkan_susu
-tidak_punya_bulu

```

Gambar 7 Klasifikasi Amphibi

```

GNU Prolog console
File Edit Terminal Prolog Help
|-- mulai.
Mengecek Klasifikasi Hewan Peliharaan

nama hewan peliharaan : lohan.

apakah hewan ini punya_bulu (y/n) ? n.
apakah hewan ini menghasilkan_susu (y/n) ? n.
apakah hewan ini punya_tulang_belakang (y/n) ? y.
apakah hewan ini punya_sirip (y/n) ? y.

lohan termasuk kelas fish

lohan punya karakteristik :
-punya_sirip
-punya_tulang_belakang

lohan tidak punya karakteristik berikut :
-tidak_menghasilkan_susu
-tidak_punya_bulu

(47 ms) yes
|--

```

Gambar 8 Klasifikasi Fish

IV. Kesimpulan

Salah satu aplikasi dari pohon khususnya pohon keputusan adalah untuk digunakan dalam menyelesaikan suatu permasalahan atau mengidentifikasi suatu hal lalu mengelompokkannya. Salah satu implementasi terhadap pohon keputusan adalah sistem pakar.

Sistem Pakar / *expert system* dibuat dengan tujuan untuk mempermudah klasifikasi suatu binatang tanpa bantuan para ahli. Selain itu, dengan sistem pakar ini kita bisa melakukan proses secara berulang mengklasifikasikan binatang secara otomatis. Adapun kita dapat menyimpan pengetahuan dan keahlian para pakar.

Sistem pakar ini pun tidak memerlukan biaya saat digunakan dan diperbanyak sistemnya sesuai kebutuhan dengan waktu yang minimal dan biaya yang sedikit. Sistem pakar ini pun bertujuan untuk memecahkan masalah pengklasifikasian suatu binatang lebih cepat daripada kemampuan manusia dengan catatan *menggunakan data yang sama* sesuai dengan pohon keputusan yang disediakan.

Sistem pakar ini pun dibuat bertujuan untuk meningkatkan kualitas dan produktivitas karena dapat memberi rule yang konsisten dan mengurangi kesalahan. Serta meningkatkan kapabilitas sistem terkomputerisasi yang lain. Yang lebih utama adalah memasyarakatkan sistem pakar ini bagi orang awam atau pemula agar pengetahuan lebih berkembang.

V. Ucapan Terima Kasih

Saya mengucapkan terima kasih kepada Ibu Harlili dan Bapak Rinaldi Munir atas bimbingannya dalam mata pelajaran Matematika Diskrit (IF2120). Saya pun memohon maaf jika ada kesalahan baik lisan maupun perbuatan selama proses pembelajaran.

Ucapan terima kasih pun saya ucapkan kepada teman seangkatan dan keluarga yang telah memberikan semangat dan dorongan moril selama proses pembelajaran.

REFERENSI

- [1] Munir, Renaldi "Diktat Kuliah IF2110 Matematika Diskrit", Informatika Bandung : Bandung, 2007
- [2] K.H. Rosen, Discrete Mathematics and Its Applications 7th. New York: McGraw - Hill, 2012
- [3] <http://pohonkeputusan.com/konsep-pohon-keputusan-id3-dan-c4-5/> diakses : 4 Desember 2014 pukul, 5:28 PM
- [4] <https://aif01.wordpress.com/> diakses : 4 Desember 2014 pukul, 5 : 30 PM
- [5] <http://3onoikom.wordpress.com/materi->

[kuliah/sistem-pakar/](#) diakses : 4 Desember 2014
pukul , 5 :10 PM

[6] <http://kuliah.itb.ac.id/app245/course/view.php?id=40> diakses : 5 Desember 2014 pukul 08.00 AM

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, 8 Desember 2014



Mochamad Try Yulyanto (13513067)