

SISTEM PAKAR DIAGNOSA PENYAKIT PADA KELINCI MENGGUNAKAN METODE FORWARD CHAIN

Adityaz Shofyan Fajri Al Hafiizh
S1 Teknik Informatika
Universitas Negeri Malang
Malang, Indonesia
Dyazsfa@gmail.com

Abstrak:

Sistem pakar yang diperuntukkan kepada mereka yang suka memelihara hewan lucu kelinci yang terkena wabah penyakit namun tidak mengetahui penyakitnya, system pakar ini diharapkan dapat membantu untuk menentukan diagnosa dari penyakit kelinci yang umum terjadi. Sistem ini juga menggunakan bahasa pemrograman logika, yaitu prolog dimana keakuratannya mencapai 100%. Metode yang digunakan adalah forward chaining yaitu melakukan pencarian dari fakta dari input user yang berupa jawaban dari pertanyaan yang sudah dibuat didalam sistem. Fakta yang didapatkan oleh sistem akan dicocokkan dengan peraturan – peraturan yang tersimpan pada sistem yang nantinya jika sudah mendapatkan sebuah aturan yang cocok, maka sistem akan mengambil kesimpulan.

Keywords— sistem pakar, penyakit, kelinci, gejala, forward chaining

I. PENDAHULUAN

Sistem pakar adalah suatu sistem yang di program pada komputer agar komputer dapat menyelesaikan masalah seperti yang dilakukan oleh para ahli dibidangnya. Sistem pakar juga suatu teknologi yang sekarang banyak digunakan untuk mempermudah masalah dan membantu pekerjaan manusia. Sistem pakar dirancang dan memiliki logika layaknya seorang pakar berdasarkan fakta dan aturan yang ada. Sistem pakar hanya sebatas sebagai konsultan saja, tetapi tidak bisa menggantikan pakar sesungguhnya. Sistem pakar bisa digunakan untuk banyak hal salah satunya adalah untuk mendiagnosa penyakit yang sedang dialami oleh hewan peliharaan, contohnya seperti kelinci.

Penyakit umum yang sering terjadi dan menyerang kelinci ini dapat diketahui dengan menganalisa gejala-gejala yang timbul pada kelinci tersebut. Semua gejala yang ada akan dikumpulkan dan dicocokkan dengan data data pada system pakar ini. Selanjutnya program akan mencocokkan dan menampilkan hasil yang paling tepat dan paling cocok dengan gejala tersebut.

II. DASAR TEORI

A. Knowledge Base.

Knowledge base merupakan inti dari program system pakar, knowledge base merupakan sebuah basis data pengetahuan yang berisi dari sekumpulan objek dan beserta atributnya, semua atribut dari gejala penyakit yang ada akan tertulis didalam knowledge base ini, knowledge base

dapat menyimpan lebih dari satu atribut, karena pada system pakar ini gejala yang timbul tidak hanya satu atau dua atribut namun juga dapat lebih.

B. Working Memory

Working memory adalah bagian yang mengandung semua fakta-fakta baik fakta awal pada saat system beroperasi maupun fakta-fakta pada saat pengambilan kesimpulan sedang dilaksanakan selama system pakar beroperasi basis data berada di dalam memori kerja.

C. Inference Engine

Mesin inferensia adalah bagian yang menyediakan mekanisme fungsi berfikir yang biasanya digunakan oleh pakar. Mekanisme ini akan menganalisa masalah dan akan mencari jawaban dan kesimpulan terbaik. Mesin ini akan mencocokkan kaidah-kaidah dengan fakta-fakta yang ada.

D. Forward Chain

Metode *forward chaining* merupakan salah satu metode inferensi yang pencariannya dimulai dari fakta – fakta yang dimiliki. Fakta – fakta yang didapat dari user dicocokkan dengan aturan – aturan dan fakta pada *knowledge base*. Kemudian didapat kesimpulan berdasarkan data dan hipotesa yang ada pada *knowledge base*.

III. METODOLOGI

Cara yang digunakan untuk memperoleh kebenaran menggunakan metode Forward chaining sebagai berikut:

A. Persiapan Software

Software yang diperlukan dalam membangun sistem meliputi :

1. Sebuah komputer untuk menampung semua aplikasi yang ada.
2. SWI-Prolog 8.0.1 untuk membuat aplikasi console sederhana
3. MS Excel untuk memetakan data
4. MS Word untuk menyusun laporan

B. Menyusun Data yang diperlukan

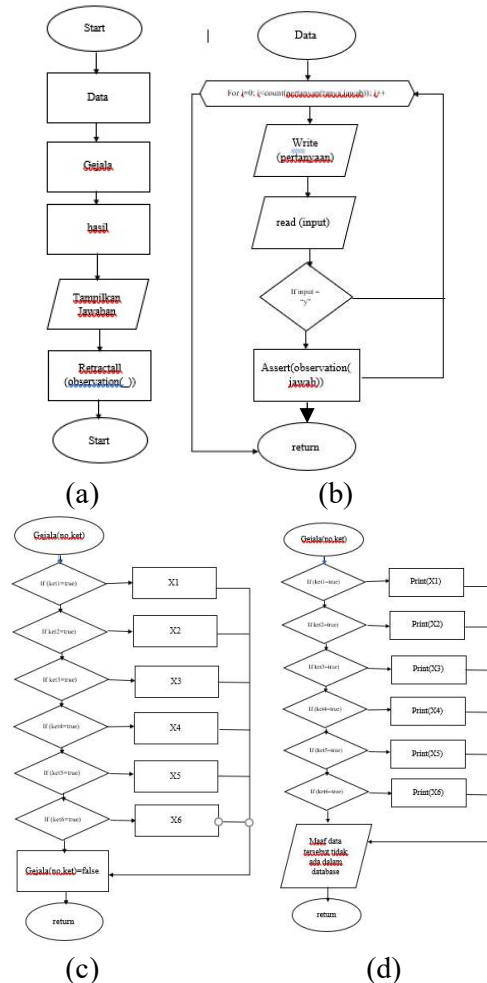
Dilansir dari website Mitalom.com beberapa penyakit dan gejala yang ada yang dapat digunakan untuk memprediksi penyakit disusun dalam tabel berikut :

No	Kode penyakit	keterangan
1	X1	Entritis Kompleks
2	X2	Young Doe Syndrome
3	X3	Sembelit
4	X4	Radang Paru
5	X5	Cacingan
6	X6	Pilek

No	Kode Gejala	keterangan
1	Y1	Kotoran tidak kelar normal
2	Y2	Kotoran berwarna hijau gelap
3	Y3	Lendir bergantung pada dubur
4	Y4	Kelinci tampak kurus
5	Y5	Suhu tubuh panas
6	Y6	Nafsu makan berkurang
7	Y7	Selalu merasa haus
8	Y8	Menggaruk dubur
9	Y9	Mengeluarkan lendir pada hidung

No	Nama Penyakit	Gejala Penyakit								
		Y1	Y3	Y3	Y4	Y5	Y6	Y7	Y8	Y9
1	X1	*	*	*	*					
2	X2				*	*	*			
3	X3	*					*			
4	X4	*		*	*		*	*		
5	X5						*			
6	X6					*	*			*

C. Menyusun Flowchart



Gambar 1.Flowchart (a)Rule Mulai (b) Rule data(c)Rule Penyakit (d)Rule Hasil

Flowchart ini nantinya menjadi basis algoritma pembuatan sistem pakar diagnosa penyakit yang dialami oleh kelinci. Konsep kerjanya, ketika aturan mulai pada gambar (a) dipanggil, secara runtut aturan pada gambar (b),(c), dan(d) dijalankan dan nantinya kembali lagi pada aturan awal untuk menampilkan *goal*.

Pada sistem pakar diagnose penyakit hewan kelinci, ketika aturan pertama mulai dipanggil, proses akan berlanjut ke aturan data. Pada aturan data, dilakukan pemanggilan fakta pertanyaan satu persatu sebanyak jumlah pertanyaan yang ada. Ketika *input* terhadap pertanyaan adalah “Y” , variabel Jawab di tegaskan ke fakta observasi yang ada. Ketika semua pertanyaan sudah ditampilkan, proses akan kembali ke aturan mulai dan lanjut ke rule berikutnya.

Pada *rule* penyakit, dilakukan penentuan fakta berdasarkan keterangan yang ada. Ketika fakta sesuai dengan keterangan yang ada, misalnya Ket 1 = true, berarti fakta pada aturan penyakit, misalnya **penyakit**(2,cacingan) bernilai true dan *rule* penyakit selesai dilakukan. Hal yang sama terjadi untuk keterangan lainnya, namun jika tidak ada yang sesuai, berarti *rule* pemyakit mengembalikan nilai *false*. Setelah *rule* selesai dijalankan, proses kembali ke *rule* mulai dan lanjut ke *rule* berikutnya.

Setelah *rule*-rule sebelumnya selesai, dicari *goal*/jawaban yang dibutuhkan dengan memanggil *rule* hasil. variabel Ket disini didasarkan pada fakta yang didapat dari *rule* sebelumnya yaitu **penyakit**(No,Ket). Jika nilai variabel Ket sesuai dengan data yang ada pada fakta hasil, nilai variabel Jawaban akan disesuaikan dengan fakta yang ada. Jika tidak ada yang sesuai maka **hasil**(Ket,Jawaban) akan memberikan nilai *false*. Setelah *rule* selesai dijalankan, kembali ke *rule* **mulai** dan variabel Jawaban ditampilkan menggunakan sintaks 'write(Jawaban)', dan apabila **hasil**(Ket,Jawaban) bernilai *false* yang ditampilkan adalah keterangan 'Data tidak ada pada *database*'. Setelah *goal* dicapai, dipanggil perintah '**retractall**(observasi(_))' untuk menghapus seluruh fakta observasi dan kemudian proses pada *rule* **mulai** selesai.

D. Pembuatan Aplikasi Pada SWI-Prolog 8.0.1

1. Menulis Fakta

Penulisan fakta pada SWI-Prolog bisa dilihat pada penggalan penulisan fakta berikut.

```
pertanyaan('Apakah awan terlihat
           pertanyaan('apakah
           kelinci tampak kurus?',
           kurus).
pertanyaan('apakah kelinci sering
           menggaruk dubur?',
           garuk).
pertanyaan('apakah kotoran kelinci
           tidak keluar normal?',
           kotoran).
```

```
pertanyaan('apakah kotoran berwarna
           hijau gelap?', gelap).
pertanyaan('apakah lendir bergantung
           pada dubur?', lendir).
pertanyaan('apakah nafsu makan
           kelinci berkurang?',
           nafsu).
pertanyaan('apakah kelinci selalu
           haus?', haus).
pertanyaan('apakah suhu kelinci
           panas?', panas).
pertanyaan('apakah kelinci
           mengeluarkan lendir
           dari hidung?', flu).
```

2. Menulis Aturan

Penulisan aturan pada SWI-Prolog bisa dilihat pada penggalan penulisan aturan berikut.

```
main:-
    init,
    pengamatan,
    aturan(Nomor,Alasan),
    jawaban(Alasan,Jawab),nl,
    write('=====
====='),nl,
    write('Jenis Penyakit Adalah :
           '(Jawab)),nl,
    write('=====
====='),nl,
    retractall(observasi( )).
```

3. Membuat Goal

Penulisan aturan pada SWI-Prolog bisa dilihat pada penggalan penulisan aturan berikut.

```
jawaban(a,'Entritis Kompleks').
jawaban(b,'Young De Syndrome').
jawaban(c,'Sembelit').
jawaban(d,'Radang Paru').
```

```
jawaban(e,'Cacingan').
```

```
jawaban(f,'Pilek').
```

4. Melakukan Pengujian

Pengujian dilakukan dengan menggunakan aplikasi SWI-Prolog. Bentuk aplikasi berupa aplikasi *console* prolog.

```
6 ?- main.
=====
PROGRAM KLASIFIKASI PENYAKIT PADA KELINCI
=====

Jawablah pertanyaan ini dengan "y" atau "t"

apakah kelinci tampak kurus?
|: t
apakah kelinci sering menggaruk dubur?
|: y
apakah kotoran kelinci tidak keluar normal?
|: t
apakah kotoran berwarna hijau gelap?
|: t
apakah lendir bergantung pada dubur?
|: t
apakah nafsu makan kelinci berkurang?
|: y
apakah kelinci selalu haus?
|: t
apakah suhu kelinci panas?
|: t
apakah kelinci mengeluarkan lendir dari hidung?
|: t
=====
Jenis Penyakit Adalah : (Cacingan)
=====
true
```

IV. ANALISA DAN TESTING

Proses testing dilakukan berdasarkan inputan user secara realtime. *Testing* dilakukan untuk memprediksi penyakit. Tabel dibawah menampilkan hasil akurasi prediksi penyakit dari serangkaian tes yang sudah dilakukan.

	X1	X2	X3	X4	X5	X6	Err
Jumlah							70
Prediksi	8%	8%	3%	3%	4%	4%	%
i							
Akurasi	90	90	90	90	90	90	95
i	%	%	%	%	%	%	%

Tabel Testing Diagnosa Penyakit

Proses *testing* dilakukan sebanyak tiga puluh kali berdasarkan variabel data. Sistem mampu memprediksi sembilan *goal* dari tiga puluh percobaan (30%) dan sisanya (70%) gagal memberi hasil prediksi. Hal ini disebabkan keterbatasan sistem dalam mengolah fakta dari user untuk mencapai *goal* tertentu dikarenakan *knowledge base* masih terbatas pada beberapa kondisi saja. Untuk memberikan hasil dengan akurasi yang tinggi, *knowledge base* harus lebih detail dan memuat segala kemungkinan yang bisa terjadi dalam memprediksi cuaca.

V. PENUTUP

Berdasarkan data dari Analisa dan testing dapat disimpulkan bahwa system pakar diagnosa penyakit kelinci hanya memberikan respon 30% dari jumlah percobaan, yang artinya masih banyak kekurangan pada sistem pakar ini, hal ini diakibatkan keterbatasan data yang ada. Selain itu, diperlukan juga *knowledge base* yang lebih detail untuk mendapatkan sistem dengan akurasi yang lebih tinggi terhadap beragam user input .

REFERENSI

- Azzami. 2018. *18 penyakit penting pada hewan ternak kelinc*, (Online) ,
(<https://mitalom.com/18-penyakit-penting-pada-ternak-kelinci-penyebab-gejala-dan-cara-pengendaliannya/>), diakses 10 April 2019.
- Wordpress. 2019. *sistem pakar*, (Online) ,
(<https://3onoikom.wordpress.com/materi-kuliah/sistem-pakar/>), diakses 4 Mei 2019.
- Uti, D. 2019. *perbedaan dan pengertian forward chaining dan backward chaining*, (Online),
(<https://garudacyber.co.id/artikel/526-perbedaan-dan-pengertian-forward-chaining-dan-backward-chaining>), diakses 4 Mei 2019.