

SISTEM PAKAR DIAGNOSA PENYAKIT AYAM MENGUNAKAN PERINTAH SUARA

Bambang Yuwono

Jurusan Teknik Informatika UPN "Veteran" Yogyakarta
Jl. Babarsari 2 Tambakbayan 55281 Telp (0274) 485323
e-mail : bambangy@gmail.com

Abstract

This research was conducted based on the need for a tool for farmers and extension workers in diagnosing the disease in chickens. The tools in the form of an expert system which in addition to diagnosis is also able to provide medical advice. This expert system in the consultation can be run using voice commands to answer every question with a yes or no vote. The process of speech recognition using SAPI (Speech Application Programming Interface) is a tool that is used to process the analog sound is then converted to text. Inference method used is backward chaining with the model of Depth First Search. The output of this system in the form of the name of the disease and treatment suggestions.

Keywords : Expert System, Disease of Chickens, Depth First Search

Penelitian ini dilakukan berdasarkan kebutuhan akan adanya alat bantu bagi peternak maupun penyuluh dalam mendiagnosis penyakit pada ayam. Alat bantu tersebut berupa sistem pakar yang selain untuk mendiagnosis juga mampu memberikan saran-saran pengobatannya. Sistem pakar ini dalam konsultasinya dapat dijalankan menggunakan perintah suara dengan menjawab setiap pertanyaan dengan suara yes atau no. Proses pengenalan suara menggunakan SAPI (Speech Application Programming Interface) yaitu sebuah tool yang digunakan untuk memproses suara analog kemudian dikonversi menjadi teks. Metode inferensi yang digunakan adalah backward chaining dengan model Depth First Search. Keluaran dari sistem ini berupa nama penyakit dan saran pengobatannya.

Kata kunci : Sistem Pakar, Penyakit Ayam, Depth First Search

1. PENDAHULUAN

Penyakit pada ternak tidaklah asing lagi bagi peternak, namun ini merupakan kendala yang sering dihadapi peternak. Para peternak kadang tahu kalau ternaknya terserang penyakit, tetapi peternak tidak tahu penyakit apa yang sedang menyerang ternaknya serta bagaimana mengobatinya. Sehingga untuk mengetahui penyakit yang sedang menyerang ternak dibutuhkan seorang penyuluh ternak atau dokter hewan. Namun pada kenyataannya penyuluh atau dokter tidak selalu ada ketika dibutuhkan karena terbatasnya tenaga ahli. Hal ini bila tidak segera ditangani besar kemungkinan ternak yang lain akan tertular sehingga berujung pada kematian.

Para penyuluh ternak juga kesulitan untuk mengidentifikasi jenis penyakit yang menyerang ternak, walaupun terlihat adanya perubahan pada hewan ternak. Bahkan kadang-kadang penyuluh tidak tahu cara pengobatannya untuk menyembuhkan penyakit ternak tersebut. Sehingga penyuluh kesulitan dalam memberi penjelasan kepada peternak tentang gejala-gejala (tanda-tanda) ternak sedang diserang penyakit.

Berdasarkan kemajuan dalam bidang komputer dan informatika, kerumitan dan kesulitan dapat ditanggulangi dengan menyediakan suatu perangkat lunak (sistem pakar) berupa program untuk mendiagnosis penyakit yang menyerang ternak ayam dan cara pengobatannya. Sistem pakar ini dapat membantu peternak dalam mendiagnosa penyakit dan memberi resep atau obat. Sistem pakar ini tidak berarti menggantikan kedudukan dokter, tetapi hanya membantu dalam mengkonfirmasi keputusannya, karena mungkin bisa terdapat banyak alternatif yang harus dipilih secara tepat.

Penelitian tentang pembuatan sistem pakar diagnosa penyakit ternak ayam menggunakan perintah suara ini sangatlah berguna untuk menghilangkan ketergantungan *user* terhadap piranti *mouse* dan *keyboard*. Disamping itu, untuk membantu *user* yang cacat tangan,

agar tidak perlu mencari pertolongan dalam menjalankan sistem. Sistem pakar ini selain dapat dijalankan menggunakan mouse dan keyboard juga dapat dijalankan menggunakan perintah suara. Disamping untuk mendiagnosis sistem ini diharapkan mampu memberikan saran-saran cara pengobatan.

Dalam menggunakan perintah suara, aplikasi ini membutuhkan SAPI (*Speech Application Programming Interface*) yang digunakan sebagai alat untuk memproses suara analog dan kemudian dikonversi menjadi teks. Perlengkapan minimum yang digunakan dalam menjalankan aplikasi ini adalah seperangkat komputer yang dilengkapi dengan *microphone*

2. TINJAUAN PUSTAKA

Beberapa penelitian yang menggunakan sistem pakar telah dilakukan, antara lain : (Dian, 2002) mengaplikasikan sistem pakar berbasis logic programming untuk simulasi seleksi ternak. Penelitian ini telah menghasilkan sebuah model program simulasi yang dapat digunakan sebagai alat bantu untuk melakukan seleksi ternak, dengan menerapkan logic programming dan menggunakan visual-prolog sebagai perangkat lunaknya.

(Mutaqin, 2002) mengimplementasikan sistem pakar dalam dunia medis : suatu pengembangan sistem diagnosis kesehatan gigi dan mulut. Sistem ini dapat memberikan informasi tentang hasil diagnosis jenis penyakit gigi dan mulut berdasarkan gejala-gejala yang diderita oleh pasien yang dimasukkan sebagai entri data melalui disain dialog yang diberikan ke sistem.

(Handayani, 2008) membuat Sistem Pakar untuk Diagnosis Penyakit THT Berbasis web dengan "*e2gLite Expert System Shell*". Sistem ini menginformasi diagnosis Penyakit THT berbasis Web sehingga user dapat mengakses dari website

(Yuwono, 2007) membuat sistem pakar untuk diagnosa penyakit ayam menggunakan program prolog.

Penelitian-penelitian tersebut diatas, tidak sama dengan penelitian yang akan dipakai dalam pengembangan sistem pakar diagnosa penyakit ternak ayam menggunakan perintah suara. Perbedaannya pada penelitian ini memanfaatkan suara untuk menjalankan sistem sedangkan penelitian-penelitian sebelumnya masih terbatas pada mouse dan keyboard. Tapi secara umum aspek-aspek yang diperoleh peneliti-peneliti terdahulu memberi dukungan informasi yang diperlukan.

2.1 Sistem Pakar

Secara umum, sistem pakar (*Expert system*) adalah sistem yang berusaha mengadopsi pengetahuan manusia ke komputer, agar komputer dapat menyelesaikan masalah seperti yang biasa dilakukan oleh para ahli (Kusumadewi, 2003). Sistem pakar yang baik dirancang agar dapat menyelesaikan suatu permasalahan tertentu dengan meniru kerja dari para ahli. Dengan sistem pakar ini, orang awampun dapat menyelesaikan masalah yang cukup rumit yang sebenarnya hanya dapat diselesaikan dengan bantuan para ahli. Bagi para ahli, sistem pakar ini juga akan membantu aktivitasnya sebagai asisten yang sangat berpengalaman.

Menurut Turban (1995) konsep dasar sistem pakar mengandung keahlian (*expertise*), pakar (*expert*), pengalihan keahlian (*transferring expertise*), inferensi (*inferencing*), aturan (*rules*) dan kemampuan menjelaskan (*explanation capability*).

Keahlian (*expertise*) adalah suatu kelebihan penguasaan pengetahuan di bidang tertentu yang diperoleh dari pelatihan, membaca atau pengalaman. Pengetahuan tersebut memungkinkan para ahli untuk dapat mengambil keputusan lebih cepat dan lebih baik daripada seseorang yang bukan ahli.

Pakar (*Expert*) adalah seseorang yang mampu menjelaskan suatu tanggapan, mempelajari hal-hal baru seputar topik permasalahan (*domain*), menyusun kembali pengetahuan jika dipandang perlu, memecah aturan-aturan jika dibutuhkan, dan menentukan relevan tidaknya keahlian mereka.

Pengalihan keahlian (*transferring expertise*) dari para ahli ke komputer untuk kemudian dialihkan lagi ke orang lain yang bukan ahli, hal inilah yang merupakan tujuan utama dari sistem pakar. Proses ini membutuhkan 4 aktivitas yaitu :

1. Tambahan pengetahuan (dari para ahli atau sumber-sumber lainnya)
 2. Representasi pengetahuan (ke komputer)
 3. Inferensi pengetahuan
 4. dan pengalihan pengetahuan ke user.
-

Pengetahuan yang disimpan di komputer disebut dengan nama basis pengetahuan. Ada dua tipe pengetahuan, yaitu fakta dan prosedur (biasanya berupa aturan).

Salah satu fitur yang harus dimiliki oleh sistem pakar adalah kemampuan untuk menalar. Jika keahlian-keahlian sudah tersimpan sebagai basis pengetahuan dan sudah tersedia program yang mampu mengakses basisdata, maka komputer harus dapat diprogram untuk membuat inferensi. Proses inferensi ini dikemas dalam bentuk motor inferensi (*inference engine*)

Sebagian besar sistem pakar komersial dibuat dalam bentuk *rule based systems*, yang mana pengetahuan disimpan dalam bentuk aturan-aturan. Aturan tersebut biasanya berbentuk IF-THEN.

Fitur lainnya dari sistem pakar adalah kemampuan untuk memberikan nasehat atau merekomendasi. Kemampuan inilah yang membedakan sistem pakar dengan sistem konvensional.

2.2 Motor Inferensi

Mesin inferensi adalah bagian yang mengandung mekanisme fungsi berpikir dan pola-pola penalaran sistem yang digunakan oleh seorang pakar (Turban, 1995). Mekanisme ini akan menganalisa suatu masalah tertentu dan selanjutnya akan mencari jawaban atau kesimpulan terbaik.

Ada dua teknik yang dapat dikerjakan dalam melakukan inferensi, yaitu :

1. *Forward Chaining*

Pencocokan fakta atau pernyataan dimulai dari bagian sebelah kiri (IF dulu). Dengan kata lain, penalaran dimulai dari fakta terlebih dahulu untuk menguji kebenaran hipotesis.

2. *Backward Chaining*

Pencocokan fakta atau pernyataan dimulai dari bagian sebelah kanan (THEN dulu). Dengan kata lain, penalaran dimulai dari hipotesis terlebih dahulu, dan untuk menguji kebenaran hipotesis tersebut harus dicari fakta-fakta yang ada dalam basis pengetahuan.

Kedua metode inferensi tersebut dipengaruhi oleh tiga macam penelusuran, yaitu *Depth-first search*, *Breadth-first search* dan *Best-first search*.

- a. *Breadth-first search*, Pencarian dimulai dari simpul akar terus ke level 1 dari kiri ke kanan dalam 1 level sebelum berpindah ke level berikutnya.
- b. *Depth-first search*, Pencarian dimulai dari simpul akar ke level yang lebih tinggi. Proses ini dilakukan terus hingga solusinya ditemukan atau jika menemui jalan buntu.
- c. *Best-first search*, bekerja berdasarkan kombinasi kedua metode sebelumnya.

2.3 Teknologi SAPI (*Speech Application Programming Interface*)

SAPI (*Speech Application Program Interface*) 5.1 adalah salah satu API (*Application Programming Interface*) yang disediakan sebagai pengantar untuk pemrograman dengan menggunakan metode *speech recognition* dan *Text To Speech* (*help, overview SAPI 5.1*). Dengan adanya SAPI memungkinkan pembuat untuk mengimplementasikan sistem pengenalan suara dengan menggunakan *engine* yang berbeda tanpa merubah aplikasi yang telah dibuat (Adipranata, 2003).

Terdapat dua macam mode pada pengenalan suara atau wicara, yaitu (Adipranata, 2003) :

1. Mode Diktasi

Pada metode ini pengguna komputer dapat mengucapkan kata atau kalimat, yang selanjutnya akan dikenali oleh komputer dan diubah menjadi data *text*. Kemungkinan jumlah kata yang dikenali dibatasi oleh jumlah kata yang telah terdapat pada *database*. Pengenalan Mode Diktasi merupakan *Speaker Dependent*. Keakuratan pengenalan mode ini bergantung pada pola suara dan aksan pembicara serta pelatihan yang telah dilakukan.

2. *Mode Command and Control*

Pada metode ini, pengguna komputer mengucapkan kata atau kalimat yang sudah didefinisikan terlebih dahulu pada *database* dan selanjutnya akan digunakan untuk menjalankan perintah tertentu pada aplikasi komputer. Jumlah perintah yang dapat

dikenali tergantung dari aplikasi yang telah didefinisikan terlebih dahulu pada *database* jenis-jenis perintah yang dapat dieksekusikan. Mode ini merupakan *Speaker Independent* karena jumlah kata yang dikenali biasanya terbatas sekali dan ada kemungkinan pembicara tidak perlu melakukan pelatihan pada sistem sebelumnya.

Terdapat empat proses utama pada sistem pengenalan suara, baik pada mode *Diktasi* ataupun *Command and Control*, yaitu (Adipranata, 2003) :

1. Pemisahan kata

Pemisahan kata adalah proses untuk memisahkan suara yang diucapkan oleh pengguna menjadi beberapa bagian. Masing-masing bagian bisa berupa kalimat ataupun hanya sebuah kata. Terdapat tiga macam metode yang dapat digunakan pada proses pemisahan kata, yaitu (Adipranata, 2003) :

a. *Discrete Speech*

Pada metode ini, pengguna diharuskan mengucapkan kalimat secara terpecah dengan adanya jeda sejenak diantara kata. Jeda tersebut digunakan oleh sistem untuk mendeteksi awal dan akhir sebuah kata. *Discrete Speech* mempunyai kelebihan yaitu sedikit *resource* (memori komputer waktu proses) yang digunakan oleh sistem untuk mendeteksi suara, tetapi mempunyai kelemahan yaitu ketidaknyamanan pengguna dalam mengucapkan sebuah kalimat.

b. *Word Spolling*

Dalam sebuah kalimat yang diucapkan pengguna, sistem hanya mendeteksi kata yang terdapat didalam perbendaharaan yang dimilikinya, dan mengabaikan kata-kata lain yang tidak dimilikinya. Sehingga walau pengguna mengucapkan kalimat yang berbeda tetapi didalam kalimat tersebut terdapat sebuah kata yang sama dan terdapat diperbendaharaan sistem, maka hasil pengenalan akan sama. Kelemahan metode ini adalah besar kemungkinan sistem akan melakukan kesalahan arti pengenalan dalam bentuk kalimat. Tetapi metode ini mempunyai kelebihan yaitu pengguna dapat mengucapkan kalimat secara normal tanpa harus berhenti diantara kata.

c. *Continuous Speech*

Sistem akan mengenali dan memproses setiap kata yang diucapkan. Metode ini akan menghasilkan keakuratan dalam mengenali ucapan pengguna. Tetapi disamping itu, metode ini memerlukan *resource* (memori komputer waktu proses) yang besar dalam prosesnya. Pada metode ini sistem harus bisa mendeteksi awal dan akhir setiap kata dalam kalimat tanpa adanya jeda diantara kata-kata tersebut, dan setelah berhasil memisahkan kata, langkah selanjutnya adalah mencocokkan dengan perbendaharaan kata yang dipunyai.

2. Ketergantungan terhadap pengguna

Sistem pengenalan suara mempunyai beberapa sifat, dilihat dari ketergantungan terhadap pengguna, yaitu (Adipranata, 2003) :

a. *Speaker Dependent*

Sistem membutuhkan pelatihan untuk setiap pengguna yang akan menggunakan sistem tersebut. Sistem tidak akan bisa mengenali pengguna yang belum pernah melakukan pelatihan.

b. *Speaker Independent*

Pengguna tidak perlu melakukan pelatihan sebelum dapat menggunakan sistem, karena sistem mampu mengenali suara pengguna tidak tergantung warna suara dan dialek yang digunakan.

3. *Speaker Adaptive*

Speaker Adaptive merupakan perpaduan dari *Speaker Dependent* dan *Speaker Independent*, dimana pengguna tidak perlu melakukan pelatihan dan keakuratan pengenalan sistem akan makin meningkat jika pengguna yang sama bekerja terus menerus selama beberapa waktu tertentu

4. Pencocokan kata

Pencocokan kata adalah proses untuk mencocokkan kata ucapan yang berhasil diidentifikasi dengan basis data yang dipunyai oleh sistem. Terdapat dua metode yang dapat dipakai pada proses pencocokan kata ini, yaitu (Adipranata, 2003) :

a. *Whole-Word Matching*

Sistem akan mencari dibasis data kata yang sama persis dengan kata hasil ucapan pengguna.

b. *Phoneme Matching*

Sistem memiliki kamus fonem. Fonem ialah bagian terkecil dan unik dari suara yang berbentuk sebuah kata.

5. Perbendaharaan kata

Perbendaharaan kata adalah bagian terakhir dalam sebuah sistem pengenalan suara. Terdapat dua hal yang perlu diperhatikan pada perbendaharaan kata, yaitu ukuran dan keakuratan. Jika perbendaharaan kata berjumlah banyak maka sebuah sistem akan mudah dalam melakukan pencocokan kata, tetapi dengan meningkatnya jumlah perbendaharaan kata, maka jumlah kata yang mempunyai ucapan hampir sama juga meningkat, dimana hal ini menurunkan keakuratan pengenalan. Dan begitu juga sebaliknya, jika sebuah sistem mempunyai perbendaharaan kata yang sedikit, maka keakuratan akan tinggi karena sedikitnya kata yang hampir sama, tetapi akan semakin banyak kata yang tidak dikenali (Adipranata, 2003).

Untuk sistem pengenalan pembicaraan dengan mode *Command and Control*, akan lebih baik jika menggunakan jumlah perbendaharaan kata sedikit (kurang dari 100 kata), tetapi untuk mode *Diktansi* akan lebih membutuhkan jumlah perbendaharaan kata yang banyak.

Pada penelitian ini, aplikasi yang dikembangkan adalah aplikasi berbasis *text* untuk menerima diktansi bahasa Indonesia. Sehubungan dengan aplikasi tersebut, maka mode yang digunakan adalah *command and control* dengan proses *continuous speech* dan *speaker adaptive* pada penggunaannya.

Selanjutnya SAPI (*Speech Application Program Interface*) 5.1 secara garis besar dapat dibagi menjadi dua struktur utama yang berfungsi untuk proses *Speech to Text* ataupun *Text To Speech* yaitu : (Adipranata, 2003)

Pada aplikasi pengenalan suara pada Microsoft Word, API berfungsi menghubungkan antara aplikasi Microsoft Word dengan SAPI *Runtime*.

1. *Device Drive Interface* (DDI)

DDI menyediakan fungsi untuk menerima data suara dari SAPI *Runtime* dengan mengembalikan pengenalan frase pada level SAPI paling dasar. Terdapat dua *engine* yang digunakan oleh DDI yaitu *recognition engine* yang berfungsi sebagai pengubah suara menjadi *text* atau tulisan dan TTS *engine* (*Text to Speech Engine*). Dalam perancangan aplikasi ini hanya menggunakan *recognition engine* saja, karena TTS hanya berfungsi untuk penerapan aplikasi yang berbasis *Text To Speech*.

2. *Device Drive Interface* (DDI)

DDI menyediakan fungsi untuk menerima data suara dari SAPI *Runtime* dengan mengembalikan pengenalan frase pada level SAPI paling dasar. Terdapat dua *engine* yang digunakan oleh DDI yaitu *recognition engine* yang berfungsi sebagai pengubah suara menjadi *text* atau tulisan dan TTS *engine* (*Text to Speech Engine*). Dalam perancangan aplikasi ini hanya menggunakan *recognition engine* saja, karena TTS hanya berfungsi untuk penerapan aplikasi yang berbasis *Text To Speech*.

2.4 Diagnosis penyakit ayam

Diagnosis suatu penyakit pada ternak ayam yang telah mati, dapat dilakukan dengan pemeriksaan secara sistematis dan cermat, sehingga dapat diketahui perubahan-perubahan yang terjadi pada ayam. Kita dapat mengetahui secara langsung jenis penyakitnya dan jenis pengobatan terhadap ayam yang masih sehat dengan obat-obatan yang tepat. Cara mendiagnosis penyakit ayam demikian disebut *nekropsis* ayam, khususnya pada penyakit-penyakit yang mengakibatkan perubahan karakteristik di dalam tubuh ayam.

Cara nekropsis ayam sering dibutuhkan oleh peternak, sebab diagnosis ini adalah diagnosis lapangan. Bila kurang yakin akan hasil diagnosis, dapat dikuatkan dengan bantuan diagnosis laboratoris. Sifat pemeriksaan nekropsis berdasarkan perubahan anatomis histologis, yang untuk penyakit tertentu diagnosis dapat ditentukan dengan hanya melihat perubahan makroskopis.

Nekropsis harus dilakukan secepat mungkin setelah ayam mati, untuk mencegah terjadinya perubahan setelah kematian. Apabila tidak dapat dilakukan oleh peternak, bangkai ayam dapat disimpan dalam tumpukan es atau tempat yang dingin, untuk mencegah pencemaran.

3. METODE PENELITIAN

Metodologi yang digunakan pada penelitian yaitu ; Studi Literatur dan SDLC (*System Development Life Cycle*) yang meliputi tahap *Analysis, Design, Implementation, Testing* dan *Maintenance*, (Pressman,2002).

1. Studi Literatur

Tahap ini merupakan tahap pengumpulan informasi dan literatur yang diperlukan untuk pembuatan sistem. Adapun informasi dan literatur yang dipergunakan diantaranya mengenai diagnosis penyakit ayam, sistem pakar, *Voice Recognizer*, dan SAPI 5.1

2. Analisis dan perancangan

Pada tahap ini dilakukan analisis serta desain yang diperlukan dalam membuat sistem, diantaranya akuisisi pengetahuan, representasi pengetahuan, mekanisme inferensi, perancangan basisdata dan perancangan user interface

Akuisisi pengetahuan adalah proses pengumpulan pengetahuan. Pada penelitian ini informasi mengenai diagnosis penyakit ayam ini diperoleh dari seorang pakar (dokter hewan) yang dilengkapi dengan buku-buku mengenai penyakit dan kesehatan ayam. Pengetahuan yang diperoleh meliputi : Gejala-gejala yang diderita ayam, Jenis penyakit dan Cara pengobatannya.

Setelah akuisisi pengetahuan diperoleh, selanjutnya dilakukan representasi pengetahuan yang dikumpulkan. Tujuan **representasi pengetahuan** adalah untuk mengembangkan suatu struktur yang akan membantu pengkodean pengetahuan ke dalam program. Dalam penelitian ini basis pengetahuan direpresentasikan dengan menggunakan kaidah produksi, yaitu berupa IF – THEN.

IF Kondisi1 (**AND** Kondisi2 ...) **THEN** Kesimpulan

Kaidah produksi merupakan statemen dua bagian yang disatukan menjadi sepenggal kecil pengetahuan. Kaidah bagian pertama IF yang menyatakan premis, kondisi atau antecedent, dan kaidah bagian kedua THEN yang menyatakan suatu kesimpulan atau konklusi. Pada contoh berikut diberikan beberapa inputan *antecedent* dan memberikan satu kesimpulan berdasarkan premis yang ada untuk menentukan jenis atau nama penyakit yang diderita pada ayam.

```
IF produksi telur turun
  AND batuk
  AND ayam lesu serta lemah
  AND nafsu makan berkurang
  AND kelumpuhan pada sayap dan kaki
  AND bersih
  AND berat telur turun dan ukuran tidak seragam
  AND sulit bernafas
  AND diare berwarna kehijau-hijauan
  THEN penyakit Tetelo
```

Setelah representasi selesai dilakukan, langkah selanjutnya adalah menentukan **mekanisme inferensi** atau sistem pelacakan. Dalam penelitian ini sistem pelacakan yang dilakukan adalah menggunakan *Backward chaining* dengan metode penelusuran *Depth First Search*. Proses pelacakan ini bermula dari simpul akar dan bergerak ke bawah ke tingkat dalam yang berurutan. Proses ini berlangsung terus sampai kesimpulan ditemukan, atau jika menemui jalan buntu akan melacak ke belakang (*backtracking*).

3. Implementasi

Pada tahap ini, rancangan sistem yang telah dibuat diimplementasikan menggunakan Visual Basic dan Microsoft Access serta program pendukung SAPI 5.1 untuk pemrograman suaranya.

4. Uji coba dan evaluasi

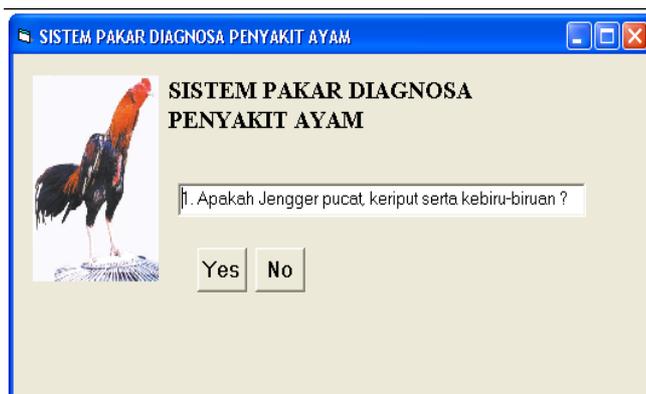
Pada tahap ini, akan dilakukan uji coba dan evaluasi terhadap sistem serta akan dilakukan perbaikan-perbaikan yang diperlukan.

4. HASIL DAN PEMBAHASAN

Aplikasi program telah diujicobakan dengan cara memasukkan beberapa data atau jawaban berdasarkan pertanyaan sistem yang ada. Sistem dapat menjalankan fungsinya sebagaimana yang diharapkan. Dengan berbagai variasi jawaban yang diperlihatkan pada masukan data berupa jawaban pendek “ya” atau “tidak” ke sistem, ternyata sistem telah dapat bekerja dengan baik. Jika jawaban pertanyaan yang diberikan ke sistem dengan memasukkan jawaban ya atau tidak yang disesuaikan dengan kondisi atau gejala yang ada memenuhi syarat terhadap salah satu jenis penyakit tertentu, maka sistem akan memberikan kesimpulan tentang jenis penyakit tertentu dan dibagian bawahnya disertai dengan saran cara pengobatannya. Demikian sebaliknya, jika jawaban tidak memenuhi syarat atas gejala yang ditanyakan sistem kepada pemakai, maka sistem akan memberikan kesimpulan, “penyakit tidak ditemukan”

Pemakai bisa melakukan konsultasi dengan sistem untuk mengetahui kondisi berdasarkan pengamatan pada ayam yang terlihat adanya perubahan akibat terserang penyakit. Pemakai cukup menjawab dengan mengatakan *yes* atau *No* atau memilih tombol ‘*yes*’ atau ‘*No*’. Penelusuran untuk mencapai tujuan sistem akan menampilkan beberapa pertanyaan, jika user menjawab ‘*yes*’ berarti kondisi true pelacakan akan dilanjutkan ke node yang paling dalam. Jika user menjawab ‘*No*’, oleh sistem menemui jalan buntu maka solusinya adalah melakukan lacak runut balik ke level node di atasnya, kemudian pencarian dilanjutkan ke anak cabang node berikutnya, sampai mencapai kesimpulan. Kemudian masing-masing kesimpulan disertakan informasi cara pengobatannya.

Gambar 4.1 menampilkan form konsultasi sistem. Pengguna cukup menjawabnya dengan mengatakan *Yes* atau *No*, bisa juga dengan menekan tombol *Yes* atau *No*. Sedangkan Gambar 4.2 menunjukkan hasil diagnosa dan saran cara pengobatannya.



Gambar 4.1 Form Konsultasi



Gambar 4.2 Hasil Diagnosa

Pengujian untuk penyakit Tetelo

Misalnya terdapat kondisi (3,5,6,7,8,11,12,18,49) benar, maka mesin inferensi mengambil kesimpulan penyakit **Tetelo**

Berikut urutan dialog yang terjadi antara sistem dan user untuk penyakit **Tetelo**

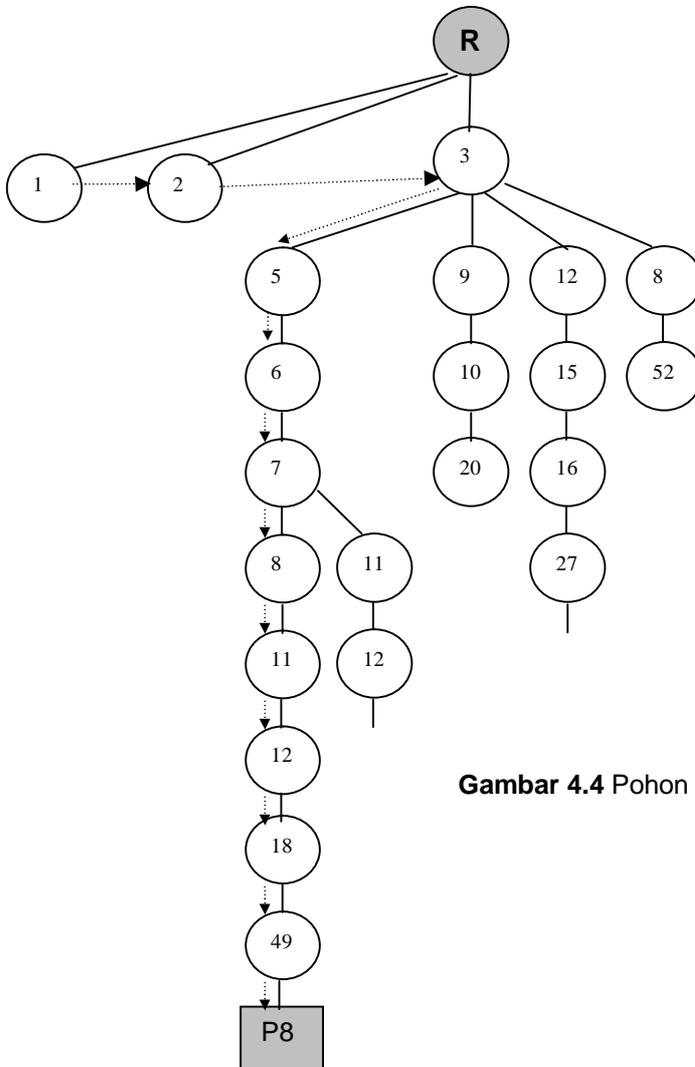
```
Sistem 1. Apakah jengger pucat, keriput serta kebiru-biruan (y/n)?
User : n
Sistem 2. Apakah pertumbuhan terhambat (y/n)?
User : n
Sistem 3. Apakah produksi telur menurun (y/n)?
User : y
Sistem 5. Apakah Batuk (y/n)?
User : y
Sistem 6. Apakah ayam lesu serta lemah (y/n)?
User : y
Sistem 7. Apakah Nafsu makan berkurang (y/n)?
User : y
Sistem 8. Apakah Kelumpuhan pada sayap dan kaki (y/n)?
User : y
```

```

Sistem 11. Apakah Bersin (y/n)?
User : y
Sistem 12. Apakah berat telur turun dan ukuran tidak seragam (y/n)?
User : y
Sistem 18. Apakah sulit bernafas (y/n)?
User : y
Sistem 49. Apakah Diare berwarna kehijau-hijauan (y/n)?
User : y
    
```

Gambar 4.3 Dialog konsultasi penyakit tetelo

Pada gambar 4.3 telah diuji dengan memunculkan 11 pertanyaan, pertanyaan di awal kalimat diberi nomor node untuk memudahkan pembacaan sekaligus pengecekan kebenaran metode *Depth First Search*. Langkah penelusuran dimulai dari pertanyaan 1 dijawab n=salah, karena dijawab salah maka penelusuran diarahkan node 2 dijawab n=salah, karena node 2 dijawab salah maka sistem mencari solusi ke node 3 dijawab y=benar, diteruskan runut maju ke node dibawahnya yaitu 5 dijawab y=benar, dilanjutkan ke node 6 dijawab y=benar, diteruskan ke node 7 dijawab y=benar, dilanjutkan ke node 6 dijawab y=benar, diteruskan ke node 11 dijawab y=benar, dilanjutkan ke node 12 dijawab y=benar, diteruskan ke node 18 dijawab y=benar, dilanjutkan ke node 49 dijawab y=benar sampai mencapai kesimpulan penyakit **Tetelo**. Gambar 5.4 menunjukkan pohon penelusuran untuk penyakit Tetelo.



Gambar 4.4 Pohon Penelusuran penyakit Tetelo

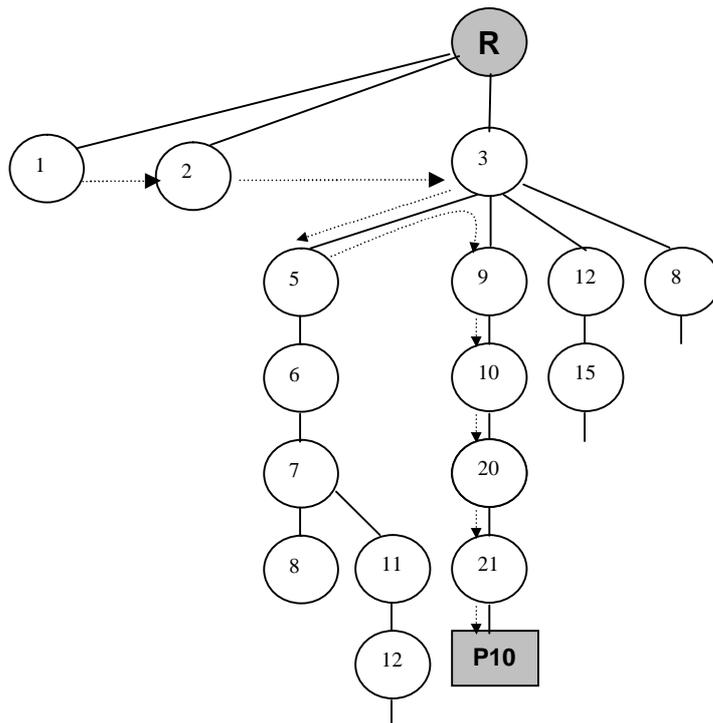
Pengujian kedua untuk penyakit syaraf leher

Mesin inferensi akan menyimpulkan Penyakit syaraf leher jika gejala gejala yang terpenuhi adalah node (3, 9, 10, 20, 21)

```
Sistem 1 Apakah jengger pucat, keriput serta kebiru-biruan (y/n)?
User : n
Sistem 2 Apakah pertumbuhan terhambat (y/n)?
User : n
Sistem 3. Apakah produksi telur menurun (y/n)?
User : y
Sistem 5. Apakah Batuk (y/n)?
User : n
Sistem 9. Apakah Malas bergerak dan sempoyongan (y/n)?
User : y
Sistem 10. Apakah mengalami gejala katarak bahkan kebutaan (y/n)?
User : y
Sistem 20. Apakah Bulu rontok (y/n)?
User : y
Sistem 21. Apakah kepala dan leher kaku (y/n)?
User : y
```

Gambar 5.5 Dialog konsultasi penyakit syaraf leher

Pada dialog diatas (gambar 5.5) telah diuji dengan memunculkan 8 pertanyaan, pertanyaan di awal kalimat diberi nomor node untuk memudahkan pembacaan sekaligus pengecekan kebenaran metode *Depth First Search*. Pada gambar 5.6 menunjukkan pohon penelusuran penyakit syaraf leher. Penyakit syaraf leher terpenuhi jika gejala gejala yang terjadi adalah node (3, 9, 10, 20, 21) bernilai benar atau dijawab "ya". Pertama program selalu menanyakan pertanyaan pertama, karena pada ayam tidak nampak keriput dijengger, maka pertanyaan dilanjutkan ke pertanyaan ke-2 yaitu apakah pertumbuhan terhambat, ternyata dijawab "no". Kemudian dilanjutkan ke pertanyaan ke-3 apakah produksi telur menurun dijawab "ya", maka program akan merunut ke node dibawahnya yaitu pertanyaan ke-5 apakah ayam batuk dijawab "no", karena jawabannya tidak maka program kembali ke pertanyaan ke-3, tapi karena pertanyaan ke-3 sudah pernah ditanyakan maka tidak akan ditanyakan lagi. Hal ini disebabkan karena sistem pakar mampu mengingat pertanyaan sebelumnya, sehingga tidak mungkin ditanyakan lebih dari satu kali. Karena pertanyaan ke-5 dijawab tidak, maka pertanyaan dilanjutkan ke node sebelah kanan yang selevel yaitu pertanyaan ke-9, yaitu apakah ayam malas bergerak dan sempoyongan, dijawab "ya". Sistem akan merunut maju, disini program telah menyimpan 2 pertanyaan sebelumnya yang dijawab "ya" yaitu node ke-3 dan ke-9 untuk menuju kesimpulan. Sistem akan menanyakan pertanyaan ke-10, karena node ini ada dibawah node 9, yaitu apakah ayam mengalami katarak bahkan kebutaan, dijawab "ya". Karena node (3, 9, 10) benar, maka program akan menguji dengan mengajukan pertanyaan ke-20 yaitu apakah bulunya rontok, dijawab benar. Dilanjutkan ke pertanyaan terakhir node ke-21 yaitu apakah kepala dan leher kaku, dijawab "ya" yang berarti benar bahwa node yang terpenuhi adalah (3, 9, 10, 20, 21) maka sistem akan menyimpulkan ayam tersebut terserang penyakit Syaraf leher berikut saran cara pengobatannya.



Gambar 5.6 Pohon Penelusuran penyakit Syaraf Leher

5. KESIMPULAN

Berdasarkan hasil analisis, perancangan, dan implementasi, telah berhasil dibangun sistem pakar diagnosa penyakit ternak ayam menggunakan perintah suara. Untuk menjalankan sistem *user* dapat menggunakan suara maupun melalui *keyboard*. Proses Inferensi telah sesuai dengan yang diharapkan. Proses perintah suara menggunakan bahasa Inggris. Sensifitas dan keakuratan penangkapan suara *user* oleh program masih kurang, sehingga *user* perlu mengulang beberapa kali dalam pengucapannya, dengan demikian perlu adanya latihan berbicara (*English*).

6. UCAPAN TERIMAKASIH

Dalam kesempatan ini penulis mengucapkan terima kasih kepada DP2M Dikti yang telah membiayai penelitian ini pada tahun anggaran 2009.

7. DAFTAR PUSTAKA

- Adipranata, Rudy, 2003, *Implementasi Sistem Pengenalan Suara Menggunakan SAPI 5.1 dan Delphi 5*, Jurnal Informatika Petra Surabaya.
- Akoso Tri Budi, 1998, *Kesehatan Unggas Panduan bagi Petugas Teknik, Penyuluh dan Peternak*, Kanisius, Yogyakarta.
- Dian Retno Sawitri, 2002, *Sistem pakar berbasis logic programming untuk simulasi seleksi ternak*, Tesis, UGM, Yogyakarta.
- Giarratano, J. & Gary R., 1994, *Expert Systems Principles and Programming*, PWS Publishing Company, Boston.
- Microsoft, 2000, *Microsoft Speech SDK 5.1*, <http://www.microsoft.com>
- Murtidjo Agus Bambang, 1992, *Pengendalian Hama dan Penyakit Ayam*, Kanisius, Yogyakarta.
- Mutaqin, 2002, *Implementasi Sistem Pakar dalam Dunia Medis : Suatu Pengembangan Sistem Diagnosis Kesehatan Gigi dan Mulut*, Tesis, UGM, Yogyakarta.
- Priyan, 2007, *Voice Commander*, <[http:// www.planetsourcecode.com](http://www.planetsourcecode.com)>
- Ricky, 2006, *Tata bahasa atau Grammar*, <[http:// www.library.gunadarma.ac.id](http://www.library.gunadarma.ac.id)>
- Turban, Efraim, 1995, *Decision Support System and Expert System*, 4th ed., Prentice-Hall, Inc., New Jersey, pp 472-679
- Yuwono B, 2008, *Sistem Pakar Untuk Diagnosis Penyakit Ayam yang Disebabkan oleh Virus*, Jurnal "Telematika" ISSN : 1829-667X, Vol.3 No.1, Juli 2008

