

BAB II

LANDASAN TEORI

Landasan teori merupakan panduan untuk menemukan solusi pemecahan masalah yang sedang dihadapi. Pada bab ini akan dikemukakan landasan teori yang terkait dengan permasalahan untuk mendukung perancangan sistem. Adapun landasan teori yang digunakan sebagai berikut:

2.1 Sistem Pakar

2.1.1 Pengertian sistem pakar

Menurut Jusak (2007:1) sistem pakar dapat didefinisikan sebagai sebuah program komputer yang mencoba meniru atau mensimulasikan pengetahuan (*knowledge*) dan ketrampilan (*skill*) dari seorang pakar pada area tertentu. Selanjutnya sistem ini akan mencoba memecahkan suatu permasalahan sesuai dengan kepakarannya.

Sistem pakar dibuat pada wilayah pengetahuan tertentu untuk suatu kepakaran tertentu yang mendekati kemampuan manusia di satu bidang. Sistem pakar mencoba mencari solusi yang memuaskan sebagaimana yang dilakukan seorang pakar (Kusrini, 2006:12).

Adapun beberapa definisi sistem pakar dari beberapa ahli yang dikutip oleh Kusumadewi (2003), antara lain:

1. Menurut Giarratano dan Riley: Sistem pakar adalah suatu sistem komputer yang bisa menyamai atau meniru kemampuan seorang pakar.

2. Menurut Durkin: Sistem pakar adalah suatu program komputer yang dirancang untuk memodelkan kemampuan penyelesaian masalah yang dilakukan oleh seorang pakar.
3. Menurut Ignizio: Sistem pakar adalah suatu model dan prosedur berkaitan, dalam suatu domain tertentu, yang mana tingkat keahliannya dapat dibandingkan dengan seorang pakar.

2.1.2 Struktur sistem pakar

Menurut Jusak (2007:6) secara umum struktur sebuah sistem pakar terdiri atas tiga komponen utama, yaitu; *knowledge base*, *working memory* dan *inference engine*.

1. *Knowledge base* (basis pengetahuan) adalah bagian dari sebuah sistem pakar yang mengandung/menyimpan pengetahuan (*domain knowledge*). *Knowledge base* yang dikandung oleh sebuah sistem pakar berbeda antara satu dengan yang lain tergantung pada bidang kepakaran dari sistem yang dibangun. Misalnya, *medical expert system* akan memiliki basis pengetahuan tentang hal-hal yang berkaitan dengan medis. *Knowledge base* direpresentasikan dalam berbagai macam bentuk, salah satunya adalah dalam bentuk sistem berbasis aturan (*ruled-based system*).
2. *Working memory* mengandung/menyimpan fakta-fakta yang ditemukan selama proses konsultasi dengan sistem pakar. Selama proses konsultasi, *user* memasukkan fakta-fakta yang dibutuhkan. Kemudian sistem akan mencari padanan tentang fakta tersebut dengan informasi yang ada dalam *knowledge base* untuk menghasilkan fakta baru. Sistem akan memasukkan fakta baru ini ke dalam *working memory*. Jadi *working memory* akan menyimpan informasi

tentang fakta-fakta yang dimasukkan oleh *user* ataupun fakta baru hasil kesimpulan dari sistem.

3. *Inference engine* bertugas mencari padanan antara fakta yang ada di dalam *working memory* dengan fakta-fakta tentang *domain knowledge* tertentu yang ada di dalam *knowledge base*, selanjutnya *inference engine* akan menarik/mengambil kesimpulan dari problem yang diajukan kepada sistem.

2.1.3 Ciri-ciri sistem pakar

Menurut Kusrini (2006:14) sistem pakar memiliki beberapa ciri-ciri sebagai berikut :

1. Terbatas pada bidang yang spesifik.
2. Dapat memberikan penalaran untuk data-data yang tidak lengkap atau tidak pasti.
3. Dapat mengemukakan rangkaian alasan yang diberikannya dengan cara yang dapat dipahami.
4. Berdasarkan pada *rules* atau aturan tertentu.
5. Dirancang untuk dikembangkan secara bertahap.
6. *Output* bersifat nasihat atau anjuran.
7. *Output* tergantung dari dialog dengan *user*.
8. *Knowledge base* dan *inference engine* terpisah.

2.1.4 Keuntungan dan kelemahan sistem pakar

Menurut Pradika (2012:12) terdapat beberapa keuntungan yang diperoleh dengan mengembangkan sistem pakar, antara lain:

1. Membuat seorang awam dapat bekerja seperti layaknya seorang pakar.
2. Dapat bekerja dengan informasi yang tidak lengkap atau tidak pasti.

3. Meningkatkan *output* dan produktivitas.
4. Meningkatkan kualitas.
5. Menyediakan nasihat atau solusi yang konsisten dan dapat mengurangi tingkat kesalahan.
6. Membuat peralatan yang kompleks dan mudah dioperasikan karena sistem pakar dapat melatih pekerja yang tidak berpengalaman.
7. Sistem tidak dapat lelah atau bosan.
8. Memungkinkan pemindahan pengetahuan ke lokasi yang jauh serta memperluas jangkauan seorang pakar, dan dapat diperoleh atau dipakai dimana saja.

Selain memiliki keuntungan, sistem pakar juga memiliki beberapa kelemahan di dalam penerapannya, antara lain :

1. Daya kerja dan produktivitas manusia menjadi berkurang karena semuanya dilakukan secara otomatis oleh sistem.
2. Pengembangan perangkat lunak sistem pakar lebih sulit dibandingkan dengan perangkat lunak konvensional.
3. Biaya pembuatan mahal, karena seorang pakar membutuhkan pembuat aplikasi untuk membuat sistem pakar yang diinginkan.

2.2 Kecerdasan Buatan

Menurut Sutojo, dkk (2010:1) Kecerdasan Buatan atau yang lebih dikenal dengan *Artificial Intelligence* (AI) merujuk pada mesin yang mampu berpikir, menimbang tindakan yang akan diambil dan mampu mengambil keputusan seperti yang dilakukan oleh manusia. AI dikembangkan pertama kali pada tahun 1960-an ketika John McCarthy dari *Massachusetts Institute of*

Technology (MIT) menciptakan bahasa pemrograman LISP. Kemudian berkembang dengan dibuatnya program komputer yang “berpikir” seperti permainan catur dan pembuktian perhitungan matematis secara komputasi. Pada tahun 1964, Joseph Weizenbaum juga dari MIT membuat ELIZA, sebuah program yang menggambarkan konsultasi seorang psikiater dengan pasiennya. Pada Era 70-an perkembangan AI menghasilkan beberapa terobosan dan satu diantaranya yang paling populer adalah *Expert System* (ES). Salah satu ES yang pertama kali dibuat oleh MYCIN-nya Universitas Stamford yang membantu para ahli medis untuk mendiagnosis dan menganalisis sakit yang diderita oleh para pasien.

2.3 *Certainty Factor*

2.3.1 Definisi *certainty factor*

Menurut Sutojo, dkk (2010:194) awal mula Teori *certainty factor* (CF) diusulkan oleh Shortliffe dan Buchanan pada 1975 untuk mengakomodasi ketidakpastian pemikiran seorang pakar. Seorang pakar/ahli dalam hal ini biasanya dokter sering kali menganalisis informasi yang ada dengan ungkapan seperti “mungkin”, “kemungkinan besar”, “hampir pasti”. Untuk mengakomodasi hal ini kita menggunakan *certainty factor* guna menggambarkan tingkat keyakinan pakar terhadap masalah yang sedang dihadapi.

Ada dua cara dalam mendapatkan *certainty factor* (CF) dari sebuah *rule*, yaitu :

1. Metode “*Net Belief*” yang diusulkan oleh E.H. Shortliffe dan B.G. Buchanan

$$CF (Rule) = MB(H,E) - MD(H,E)$$

$$MB(H,E) = \begin{cases} \frac{\max[P(H|E),P(H)]-P(H)}{\max[1,0]-P(H)} & P(H) = 1, \text{ lainnya} \end{cases}$$

$$MD(H,E) = \begin{cases} \frac{\min[P(H|E),P(H)]-P(H)}{\min[1,0]-P(H)} & P(H) = 0, \text{ lainnya} \end{cases}$$

Dimana :

$CF(Rule)$ = Faktor Kepastian

$MB(H,E)$ = *Measure of Belief* (ukuran kepercayaan) terhadap hipotesis H, jika diberikan *evidence* E (antara 0 dan 1)

$MD(H,E)$ = *Measure of Disbelief* (ukuran ketidakpercayaan) terhadap *evidence* H, jika diberikan *evidence* E (antara 0 dan 1)

$P(H)$ = Probabilitas kebenaran hipotesis H

$P(H|E)$ = Probabilitas bahwa H benar karena fakta E

2. Dengan cara mewawancarai seorang pakar/ahli

Nilai $CF(Rule)$ didapat dari interpretasi "*term*" dari pakar, yang dirubah menjadi nilai CF tertentu. Sebagai contoh dapat dilihat pada Tabel 2.1, yakni *uncertain term* dari seorang pakar dikonversi menjadi sebuah nilai CF.

Tabel 2.1 Nilai *evidence* tingkat keyakinan pakar

Uncertain Term	CF
Definitely Not (Pasti Tidak)	-1.0
Almost Certainly Not (Hampir Pasti Tidak)	-0.8
Probably Not (Kemungkinan Besar Tidak)	-0.6
Maybe Not (Mungkin Tidak)	-0.4
Unknown (Tidak Tahu)	-0.2 to 0.2
Maybe (Mungkin)	0.4
Probably (Kemungkinan Besar)	0.6
Almost Certainly (Hampir Pasti)	0.8
Definitely (Pasti)	1

Sumber : Buku Kecerdasan Buatan (Sutojo, dkk. 2010:195-196)

2.3.2 Perhitungan *certainty factor* gabungan

Secara umum, *rule* dipresentasikan dalam bentuk sebagai berikut (Sutojo, dkk. 2010:196).

IF E_1 AND E_2 AND E_n THEN H (CF Rule)

Atau

IF E_1 AND E_2 OR E_n THEN H (CF Rule)

Dimana :

$E_1 \dots E_2$: Fakta – fakta (*Evidence*) yang ada

H : Hipotesis atau konklusi yang dihasilkan

CF Rule : Tingkat keyakinan terjadinya hipotesis H akibat adanya fakta –
fakta

$E_1 \dots E_n$

1. Rule dengan *evidence* E tunggal dan Hipotesis H Tunggal (*Certainty Factor Sequential*)

IF E THEN H (CF Rule)

CF (H,E) = CF(E) X CF(Rule)

2. Rule dengan *evidence* E ganda dan Hipotesis H Tunggal (*Certainty Factor Paralel*)

IF E_1 AND E_2 AND E_n THEN H (CF Rule)

CF (H,E) = min[CF(E_1), CF(E_2),, CF(E_n)] x CF(Rule)

IF E_1 OR E_2 OR E_n THEN H (CF Rule)

CF (H,E) = max[CF(E_1), CF(E_2),, CF(E_n)] x CF(Rule)

3. Kombinasi dua buah *rule* dengan *evidence* berbeda (E_1 dan E_2), tetapi hipotesis sama

$$\begin{array}{lll}
 \text{IF } E_1 \text{ THEN H} & \text{Rule 1} & \text{CF(H, } E_1) = \text{CF}_1 = \text{C}(E_1) \times \text{CF}(\text{Rule1}) \\
 \text{IF } E_2 \text{ THEN H} & \text{Rule 2} & \text{CF(H, } E_2) = \text{CF}_2 = \text{C}(E_2) \times \text{CF}(\text{Rule2}) \\
 \text{CF}(\text{CF}_1, \text{CF}_2) & \left\{ \begin{array}{ll}
 \text{CF}_1 + \text{CF}_2 (1 - \text{CF}_1) & \text{jika } \text{CF}_1 > 0 \text{ dan } \text{CF}_2 > 0 \\
 (\text{CF}_1 + \text{CF}_2) / 1 - (\min[|\text{CF}_1|, |\text{CF}_2|]) & \text{jika } \text{CF}_1 < 0 \text{ atau } \text{CF}_2 < 0 \\
 \text{CF}_1 + \text{CF}_2 (1 + \text{CF}_1) & \text{jika } \text{CF}_1 < 0 \text{ dan } \text{CF}_2 < 0
 \end{array} \right.
 \end{array}$$

Kelebihan dan kekurangan dari metode *certainty factor*

Kelebihan metode *certainty factor* adalah :

1. Metode ini cocok dipakai dalam sistem pakar yang mengandung ketidakpastian.
2. Dalam sekali proses perhitungan hanya dapat mengolah 2 data saja sehingga keakuratan data tetap terjaga.

Sedangkan kekurangan metode *certainty factor* adalah :

1. Pemodelan ketidakpastian proses perhitungan yang menggunakan perhitungan metode *certainty factor* biasanya masih diperdebatkan.
2. Untuk data lebih dari 2 buah, harus dilakukan beberapa kali pengolahan data.

2.4 Pengertian Kulit pada Kucing

Kulit merupakan organ terbesar pada tubuh hewan yang membatasi tubuh dengan dunia luar, selain itu kondisi kulit merupakan refleksi kesehatan hewan secara umum serta dapat menjadi sebuah indikator terhadap adanya penyakit dalam tubuh hewan. Kulit juga merupakan organ yang aktif terlibat dalam proses reaksi kekebalan tubuh. Berbagai zat yang terkandung dalam lapisan emulsi tersebut antara lain sel *langerhan* pada *epidermis*, *dendrosit*, *keratinosit*, *limfosit T*, sel *mast*, endotel kapiler darah, sitokin E, komplemen dan *imunoglobulin* yang memiliki efek *antimicrobial* (Kusumawati, 2011:1).

2.5 Penyakit Kulit

2.5.1 Pengertian penyakit kulit

Penyakit kulit adalah suatu keadaan dimana terdapat gangguan terhadap bentuk dan fungsi kulit sehingga berada dalam keadaan yang tidak normal. Penyakit kulit pada kucing dapat disebabkan oleh beberapa faktor, antara lain : parasit, jamur, bakteri, virus, pengaruh nutrisi hingga faktor lingkungan yang kurang baik (Kusumawati, 2011:34). Beberapa penyakit kulit pada kucing juga dapat menular kepada manusia, antara lain : seperti *scabies*, infestasi *cheyletiella* dan *dermatophytosis*.

2.5.2 Jenis penyakit kulit

Berikut ini merupakan jenis-jenis penyakit kulit yang dapat dialami oleh kucing :

1. Defisiensi Zinc

Defisiensi *zinc* hampir selalu disebabkan oleh rendahnya kandungan *zinc* dalam makanan yang dikonsumsi oleh kucing. Kekurangan *zinc* dapat mengganggu fase penyembuhan dan perbaikan luka, selain itu juga berdampak pada keindahan kulit. Selain faktor rendahnya kadar nutrisi pada makanan, kucing yang hamil dan menyusui juga rentan terhadap penyakit ini. Gejala yang nampak pada kucing yang mengalami defisiensi *zinc* adalah pengerasan (*crusta*) *periorbital*, kekurusan dan *pododermatitis* (Dhamojono, 2001:57).

2. Defisiensi Asam Lemak

Defisiensi asam lemak dapat terjadi karena kehilangan bahan ini pada saat penyimpanan atau karena tengik/bau. Dapat juga terjadi pada makanan yang

sebenarnya sudah cukup kandungannya tetapi hanya sedikit kandungan antioksidannya seperti vitamin E. Kucing yang menderita defisiensi asam lemak biasanya juga akan mengalami *intestinal malabsorption*, penyakit pankreas, dan penyakit hepatoikronis. Gejala defisiensi asam lemak yang terlihat pada kulit adalah abnormalitas seperti *keratinisasi* dan *hipergranulosis* (Dhamojono, 2001:58).

3. Defisiensi Protein

Defisiensi protein dapat terjadi karena kucing mengonsumsi makanan yang kadar proteinnya rendah. Tetapi kucing yang diberi makanan khusus tidak akan mengalami penyakit ini, karena banyak *pet food* komersial yang kandungan proteinnya sudah cukup tinggi. Kucing yang mengalami defisiensi protein akan menunjukkan gejala *keratinisasi*, *hiperpigmentasi*, bila ada lesi biasanya simetris dan terjadi di kepala, punggung, toraks, abdomen, dan kaki (Dhamojono, 2001:60).

4. Defisiensi Vitamin A

Defisiensi vitamin A dapat terjadi karena kucing mengonsumsi makanan yang kadar vitamin A rendah. Vitamin A berfungsi mempertahankan kesehatan kulit serta sel epitel pada kucing. Kucing yang mengalami defisiensi vitamin A akan menunjukkan gejala *keratinisasi* serta gangguan pada kelenjar *sebaceous* meliputi sekresi serta hambatan pada salurannya (Dhamojono, 2001:61).

5. Flea

Flea atau pinjal merupakan ektoparasit yang sering terdapat pada kucing atau sering juga disebut kutu kucing. Penyebab kucing terjangkit *flea* adalah

karena penularan dari kucing lain dan lingkungan kandang yang kotor. Kucing yang terserang oleh parasit ini akan menunjukkan gejala sering melakukan garukan pada tubuh dikarenakan gatal yang ditimbulkan oleh gigitan kutu. Biasanya terdapat bekas gigitan dan terlihatnya parasit ini pada area yang digaruk (Dhamojono, 2001:63).

6. *Scabies*

Scabies adalah penyakit kulit yang disebabkan tungau (sejenis kutu) *scabies/sarcoptes*. Penyakit ini sering menyerang anjing, kucing, kelinci dan dapat juga menular ke manusia. *Scabiesis* pada kucing lebih sering disebabkan *notoedres cati*. Tungau ini berukuran sangat kecil (0.2-0.4 mm), hanya bisa dilihat dengan mikroskop atau kaca pembesar. Seluruh siklus hidup tungau ini berada di tubuh induk semangnya. Tungau betina menggali dan melubangi kulit kemudian bertelur beberapa kali sambil terus menggali saluran-saluran dalam kulit induk semangnya. Lubang-lubang dalam kulit yang digali seekor tungau betina dapat mencapai panjang beberapa centimeter (Dhamojono, 2001:65).

7. *Impetigo*

Impetigo pada kucing disebabkan oleh jilatan induk yang membersihkan anaknya. Bakteri yang terlibat *Pasteurella multocida* dan β -hemolytic streptococci. Lesinya berupa pustula superfisial kecil yang jarang sekali melibatkan folikel bulu dan tidak menimbulkan rasa nyeri atau gatal. Pustula yang ditimbulkan mudah pecah dan meninggalkan lesi berupa *epidermal collarette* atau krusta berwarna kecoklatan. Biasanya penyakit ini menyerang bagian leher, kepala dan tengkuk (Dhamojono, 2001:67).

8. *Superficial Folliculitis*

Superficial Folliculitis adalah penyakit yang disebabkan oleh bakteri *Staph intermedius*, tetapi juga dapat disebabkan oleh penularan spesies lain dari *Staphylococcus*. Bakteri yang masuk melalui trauma lokal atau infeksi akibat kontaminasi bulu/kulit yang kotor, seborrea, infestasi parasit, hormonal, iritasi lokal atau alergi. Walaupun begitu, hanya tiga macam etiologi utama dari penyakit ini yaitu : *staphylococci*, *dermatophytes* dan *demodex*. Penyakit ini terlihat dari gejala adanya lesi berupa kebotakan, terbentuknya sisik, dan kerak di daerah kepala dan leher yang menyerupai lesi *dermatophytosis* (Dhamojono, 2001:68).

9. *Deep Pyoderma*

Deep pyoderma merupakan infeksi kulit yang serius karena menyerang bagian kulit yang lebih dalam dari folikel bulu, yaitu dermis dan subkutan. Infeksinya dapat menimbulkan jejas luka (*cicatrix*). Faktor predisposisi untuk penyakit ini adalah gangguan kekebalan tubuh, lesi kulit dan folikel yang hebat, trauma gigitan atau garukan dan sebagainya), pengobatan dengan antibiotika yang salah dan pemberian *kortikosteroid* (Soedarto,2003:45).

10. *Folliculitis*

Folliculitis adalah infeksi folikel yang merusak dinding folikel bulu dan menimbulkan furunkulosis dan selulitis. Hal ini biasanya disebabkan oleh bakteri *Staph intermedius*, tetapi dapat pula disebabkan oleh *Proteus sp.*, *Pseudomonas sp.*, dan *E. Coli*. Penyakit ini biasanya ditandai oleh gejala adanya papula dan pustula pada folikel rambut dan terjadi di bagian wajah,

kepala dan punggung sebagai akibat sekunder dari gigitan kutu (Soedarto,2003:46).

11. *Dermatophytosis*

Dermatophytosis adalah infeksi kulit yang pada umumnya disebabkan oleh salah satu dari tiga spesies *Dermatophytes* yaitu *Microsporum*, *Trichophyton*, *Epidermophyton*. Walaupun organisme ini bersifat keratinofilik serta sering disertai *alopesia*, namun sebagian besar tidak menimbulkan pruritus. Jamur kulit yang sering menyerang anjing dan kucing antara lain adalah *Microsporum canis*, *Microsporum gypseum* dan *Trychophyton mentagrophytes*. Penyakit ini cukup berbahaya karena bersifat *zoonosis* (Soedarto,2003:48).

12. Luka Bakar

Lesi akibat panas baik superfisial ataupun di dalam (*deep*), sering disertai komplikasi infeksi bakteri dan sepsis. Penyebab utama adalah air panas, sentuhan dengan benda panas dan api. Kondisinya dapat diklasifikasikan menjadi dua jenis, yaitu sebagian dan menyeluruh. Bila gejala klinis yang ada melebihi 30% bagian tubuh, maka pada umumnya akan timbul manifestasi sistemik. Terjadi septikemia, syok, gagal ginjal, anemia dan kesulitan respirasi (Kusumawati, 2011:54).

2.6 Aplikasi Web

Pada awalnya aplikasi web dibangun dengan hanya menggunakan bahasa yang disebut HTML (*HyperText Markup Language*). Pada perkembangan berikutnya, sejumlah skrip dan objek dikembangkan untuk memperluas kemampuan HTML seperti PHP dan ASP pada *script* dan *Applet* pada objek.

Aplikasi Web dapat dibagi menjadi dua jenis yaitu aplikasi web statis dan dinamis (Firdaus, 2007:8).

Web statis dibentuk dengan menggunakan HTML. Kekurangan aplikasi seperti ini terletak pada keharusan untuk memelihara program secara terus menerus untuk mengikuti setiap perkembangan yang terjadi. Kelemahan ini diatasi oleh model aplikasi web dinamis. Pada aplikasi web dinamis, perubahan informasi dalam halaman web dilakukan tanpa perubahan program tetapi melalui perubahan data. Sebagai implementasi, aplikasi web dapat dikoneksikan ke basis data sehingga perubahan informasi dapat dilakukan oleh operator dan tidak menjadi tanggung jawab dari *webmaster*.

Arsitektur aplikasi web meliputi *client*, *web server*, *middleware* dan basis data. *Client* berinteraksi dengan *web server*. Secara *internal*, *web server* berkomunikasi dengan *middleware* dan *middleware* yang berkomunikasi dengan basis data. Contoh *middleware* adalah PHP dan ASP. Pada mekanisme aplikasi web dinamis, terjadi tambahan proses yaitu *server* menerjemahkan kode PHP menjadi kode HTML. Kode PHP yang diterjemahkan oleh mesin PHP yang akan diterima oleh *client*.

2.7 Testing dan Implementasi Sistem

2.7.1 White Box Testing

White box testing terkadang disebut juga *glass box testing* atau *clear box testing*, adalah suatu metode desain *test case* yang menggunakan struktur kendali dari desain prosedural (Romeo, 2003:34).

Metode desain *test case* dari *white box testing* digunakan agar dapat menjamin :

1. Semua jalur yang independen / terpisah dapat dites setidaknya sekali test.
2. Semua logika keputusan dapat di tes dengan jalur yang salah atau jalur yang benar.
3. Semua *loop* dapat dites terhadap batasannya dan ikatan operasional.
4. Semua struktur internal data dapat dites untuk memastikan validitasnya.

Desain uji coba *white box testing* pada sistem pakar ini digunakan untuk menguji setiap fungsi yang terdapat pada *class-class* yang ada. Hal tersebut bertujuan untuk mengetahui kesesuaian keluaran dari sistem pakar dengan keluaran yang dihasilkan logika dan asumsi pada desain sistem pakar.

2.7.2 *Black Box Testing*

Black box testing sering juga disebut sebagai *behavioral testing*, atau *functional testing* adalah sebuah metode *testing* yang dilakukan tanpa pengetahuan detail struktur internal dari sistem atau komponen yang dites. *Black box testing* berfokus pada kebutuhan fungsional pada software, berdasarkan pada spesifikasi kebutuhan dari *software* (Romeo, 2003:52).

Dengan adanya *black box testing*, perancang *software* dapat menggunakan sekumpulan kondisi masukan yang dapat secara penuh memeriksa kebutuhan fungsional pada suatu program. Desain uji coba *black box testing* pada sistem pakar ini digunakan untuk mengontrol dan memberikan gambaran fungsional dari sistem pakar yang dibuat. Pengujian sistem ini dengan melakukan uji coba pada tiap fitur-fitur utama yang ada pada sistem pakar.