

BAB 2

LANDASAN TEORI

2.1 Sistem Pakar

Sistem Pakar adalah program komputer yang mengemulasikan proses pengambilan keputusan seorang ahli dalam bidang tertentu dengan menggunakan pengetahuan dan aturan-aturan yang telah ditentukan sebelumnya [10]. Tujuan utama dari sistem pakar adalah untuk menyediakan solusi yang cepat dan akurat terhadap masalah-masalah spesifik yang biasanya memerlukan intervensi seorang ahli. Sistem ini bekerja dengan mengumpulkan informasi dari pengguna melalui antarmuka interaktif, kemudian menerapkan aturan-aturan logis yang telah diprogramkan untuk menganalisis informasi tersebut dan menghasilkan kesimpulan atau rekomendasi.

Dalam praktiknya, Sistem Pakar dapat diterapkan di berbagai bidang seperti kedokteran, teknik, bisnis, keuangan, dan pendidikan. Misalnya, dalam bidang kedokteran, sistem ini dapat membantu dokter dalam membuat diagnosis penyakit atau merencanakan perawatan pasien. Salah satu keunggulan utama Sistem Pakar adalah kemampuannya untuk mengurangi ketergantungan pada keahlian manusia yang terbatas, sehingga memungkinkan penyebaran pengetahuan ahli ke berbagai tempat dan situasi. Sistem ini juga mampu bekerja secara konsisten tanpa kelelahan atau kesalahan manusia, serta dapat diakses kapan saja dan di mana saja. Selain itu, Sistem Pakar dapat diperbarui secara terus menerus untuk mencakup pengetahuan dan teknik terbaru dalam bidang yang diterapkan.

Secara keseluruhan, Sistem Pakar menawarkan solusi inovatif dalam berbagai bidang, meningkatkan efisiensi, akurasi, dan ketersediaan pengetahuan ahli, serta memberikan kontribusi signifikan dalam penyelesaian masalah-masalah kompleks yang memerlukan keahlian khusus.

2.2 Forward Chaining

Forward chaining adalah metode penalaran yang bergerak maju dari fakta-fakta atau kondisi yang diketahui menuju kesimpulan yang diinginkan. Dalam *forward chaining*, mesin inferensi memulai dengan sekumpulan fakta dan menerapkan aturan logika untuk menghasilkan kesimpulan baru hingga mencapai tujuan atau fakta baru yang diinginkan. Teknik ini banyak digunakan dalam sistem

berbasis aturan dan kecerdasan buatan, terutama dalam pengembangan sistem pakar untuk mengambil keputusan secara otomatis. Sistem pakar dengan *forward chaining* menggunakan pendekatan *data-driven*, di mana proses dimulai dengan data yang diketahui untuk memprediksi hasil tertentu berdasarkan aturan-aturan yang ada [11].

Salah satu keunggulan *forward chaining* adalah kesederhanaan implementasinya, terutama untuk aplikasi yang berfokus pada pengumpulan informasi bertahap hingga diperoleh kesimpulan. *Forward chaining* sangat cocok untuk aplikasi di mana semua data awal diketahui, seperti diagnosis penyakit, sistem navigasi, atau deteksi penipuan dalam transaksi keuangan. Proses ini berlangsung sampai tidak ada aturan lagi yang bisa diterapkan atau kesimpulan akhir tercapai. Pada penerapannya, *forward chaining* sering ditemukan dalam sistem manajemen basis pengetahuan dan aplikasi penelusuran peraturan dalam domain industri dan medis [12].

Pada sistem deteksi *FLUTD* (*Feline Lower Urinary Tract Disease*), pengguna akan memasukkan gejala-gejala yang diamati pada kucing. Sistem kemudian akan membandingkan gejala tersebut dengan basis aturan yang telah disimpan, lalu mencocokkan aturan yang relevan dengan fakta yang ada. Setiap kali sebuah aturan terpenuhi, sistem akan menambahkan kesimpulan atau informasi baru ke dalam basis fakta, dan proses ini berlanjut hingga tidak ada aturan lain yang bisa diterapkan, atau hingga kesimpulan akhir tercapai, sehingga akhirnya sistem dapat memberikan kesimpulan penyakit yang dialami kucing, hingga saran pengobatannya. Metode ini sangat cocok untuk sistem pakar diagnosis medis, karena dapat memberikan solusi berdasarkan gejala yang diberikan secara bertahap.

Meskipun *forward chaining* memiliki banyak keunggulan, ia juga memiliki beberapa keterbatasan. Salah satu tantangan utama adalah efisiensi, terutama jika terdapat banyak aturan atau data yang harus diproses. Karena *forward chaining* akan terus mengaplikasikan aturan-aturan sampai tidak ada lagi yang berlaku, ada potensi untuk menghasilkan banyak kesimpulan yang tidak relevan atau redundan [13]. Meskipun demikian, *forward chaining* lebih tepat untuk diterapkan dalam sistem ini karena metode ini memanfaatkan gejala yang diketahui dan memberikan hasil diagnosis berupa penyakit apa yang dialami kucing. Lain hal apabila menggunakan *backward chaining*, dimana sistem akan memproses hasil penyakitnya terlebih dahulu, dimana kemudian akan ditelusuri satu per satu gejala yang dialami kucing [14].

2.2.1 Horn Clauses dan Definite Clauses

Dalam implementasi *forward chaining*, sering digunakan representasi logika berupa *Horn clauses* dan *definite clauses*. Kedua jenis klausa ini memiliki peran penting dalam menyederhanakan proses inferensi logika.

Horn clause adalah sebuah klausa dalam logika proposisional atau logika predikat yang terdiri dari paling banyak satu literal positif dan sejumlah literal negatif. Dalam bentuk normalnya, *Horn clause* dapat ditulis sebagai berikut:

$$\neg A_1 \vee \neg A_2 \vee \dots \vee \neg A_n \vee B$$

yang dapat ditransformasikan ke bentuk implikasi:

$$(A_1 \wedge A_2 \wedge \dots \wedge A_n) \implies B$$

Pada bentuk ini, A_1, A_2, \dots, A_n adalah fakta-fakta atau kondisi, sedangkan B adalah kesimpulan yang dapat diambil jika semua kondisi terpenuhi.

Definite clause adalah bentuk khusus dari *Horn clause* yang memiliki tepat satu literal positif, yaitu kesimpulan, dan satu atau lebih literal negatif, yaitu premis. *Definite clause* merupakan dasar dari banyak sistem berbasis aturan karena bentuknya yang sederhana dan mudah untuk diproses oleh mesin inferensi. Contoh *definite clause* dapat dituliskan sebagai:

$$(A \wedge B) \implies C$$

Dalam *forward chaining*, proses inferensi menggunakan *Horn clauses* atau *definite clauses* untuk mencocokkan fakta-fakta yang diketahui dengan premis-premis aturan. Ketika premis suatu aturan terpenuhi, kesimpulan dari aturan tersebut akan ditambahkan ke dalam basis fakta, sehingga memicu aplikasi aturan lainnya hingga kesimpulan akhir tercapai. Penggunaan *Horn clauses* dan *definite clauses* membantu menyederhanakan struktur aturan, sehingga sistem inferensi dapat bekerja secara efisien meskipun terdapat banyak aturan yang harus diproses.

2.3 Penyakit Saluran Kemih Pada Kucing

Penyakit saluran kemih pada kucing, terutama yang dikenal sebagai *Feline Lower Urinary Tract Disease (FLUTD)*, adalah gangguan yang sering terjadi pada kucing, terutama yang berusia dewasa. *FLUTD* mencakup berbagai kondisi yang

memengaruhi kandung kemih dan uretra, termasuk sistitis idiopatik, batu kandung kemih, infeksi saluran kemih, dan obstruksi uretra . Gejala yang umum meliputi sering buang air kecil, buang air kecil dalam jumlah sedikit, darah dalam urin (hematuria), hingga ketidakmampuan buang air kecil yang dapat menyebabkan kondisi darurat medis. Faktor risiko *FLUTD* meliputi obesitas, pola makan yang salah, stres, dan gaya hidup kucing yang kurang aktif [15]

Penanganan penyakit saluran kemih pada kucing sering kali bergantung pada penyebab yang mendasari. Untuk kasus yang ringan, perubahan diet yang mengurangi pembentukan kristal dalam urin sering direkomendasikan. Namun, untuk kondisi yang lebih parah, seperti obstruksi uretra, tindakan medis seperti *urethral catheterization* diperlukan untuk mengosongkan kandung kemih dan mencegah komplikasi lebih lanjut. Pencegahan jangka panjang termasuk manajemen stres, pemeliharaan berat badan ideal, dan diet yang tepat untuk menjaga pH urin yang sehat [16].

2.3.1 Jenis Penyakit Saluran Kemih Kucing

1. Sistitis Idiopatik Feline (*Feline Idiopathic Cystitis - FIC*)

Sistitis idiopatik pada kucing adalah kondisi yang menyebabkan peradangan pada kandung kemih tanpa adanya penyebab yang jelas, seperti infeksi atau batu. *FIC* sering dikaitkan dengan faktor-faktor lingkungan, stres, dan pola hidup kucing yang kurang aktif, terutama pada kucing indoor. Gejala umum termasuk sering buang air kecil, ketidaknyamanan saat buang air, dan bahkan darah dalam urin. *FIC* adalah salah satu penyebab paling umum dari *FLUTD*, tetapi sering kali tidak memerlukan antibiotik, melainkan penanganan berbasis manajemen stres dan modifikasi diet [17].

2. Batu Kandung Kemih (*Bladder Stones*)

Batu kandung kemih adalah formasi kristal mineral dalam kandung kemih yang dapat mengakibatkan iritasi, infeksi, atau obstruksi saluran kemih. Batu ini terbentuk akibat ketidakseimbangan mineral dalam urin, sering kali karena pola makan yang kurang tepat atau dehidrasi. Batu kandung kemih bisa berukuran kecil hingga besar, dan dalam kasus yang parah, memerlukan intervensi bedah untuk menghilangkannya. Pencegahan jangka panjang biasanya melibatkan diet yang diatur untuk mengendalikan pH urin dan asupan air yang cukup [15].

3. Infeksi Saluran Kemih (*Urinary Tract Infection - UTI*)

Infeksi saluran kemih pada kucing relatif jarang, tetapi ketika terjadi, biasanya disebabkan oleh bakteri yang masuk ke kandung kemih melalui uretra. Kucing yang mengalami *UTI* sering menunjukkan gejala seperti sering buang air kecil, urin yang berdarah, dan buang air kecil di luar kotak pasir. *UTI* lebih umum terjadi pada kucing yang lebih tua atau yang memiliki kondisi kesehatan lain seperti diabetes. Penanganan *UTI* biasanya melibatkan penggunaan antibiotik, namun diagnosis yang tepat sangat penting sebelum meresepkan obat [15].

4. Obstruksi Uretra

Obstruksi uretra adalah kondisi yang serius di mana uretra (saluran yang membawa urin keluar dari kandung kemih) tersumbat, biasanya oleh batu, kristal, atau sumbatan lendir. Obstruksi ini sering ditemukan pada kucing jantan karena uretranya lebih sempit dibandingkan dengan kucing betina. Kondisi ini menyebabkan ketidakmampuan buang air kecil, yang dapat mengakibatkan kerusakan ginjal atau bahkan kematian jika tidak ditangani segera. Tindakan darurat seperti *urethral catheterization* diperlukan untuk mengatasi sumbatan ini [18].

2.4 End User Computing Satisfaction (EUCS)

Metode *End User Computing Satisfaction (EUCS)* adalah konsep yang digunakan untuk mengukur tingkat kepuasan pengguna akhir terhadap sistem informasi yang mereka gunakan. *EUCS* menilai sejauh mana sistem tersebut memenuhi kebutuhan dan harapan pengguna dalam aktivitas komputasi sehari-hari, baik dalam lingkungan kerja maupun dalam penggunaan pribadi. Konsep ini sering digunakan dalam evaluasi sistem informasi atau aplikasi, terutama dalam konteks di mana pengguna akhir memiliki peran penting dalam interaksi dengan teknologi. Ada lima aspek utama yang diukur dalam *EUCS*, yaitu konten, akurasi, format, kemudahan penggunaan, dan ketepatan waktu (*timeliness*) [19].

1. Konten (Content)

Konten mengacu pada relevansi dan kelengkapan informasi yang disediakan oleh sistem. Pengguna puas jika informasi tersebut memadai dan sesuai dengan kebutuhan mereka, mendukung pengambilan keputusan, dan tidak terlalu umum [19].

2. Akurasi (Accuracy)

Akurasi berkaitan dengan keandalan dan kebenaran data yang disajikan. Informasi yang akurat meningkatkan kepercayaan pengguna terhadap sistem dan mencegah kesalahan dalam pengambilan keputusan [19].

3. Format (Format)

Format merujuk pada cara informasi disajikan, termasuk tata letak dan visualisasi data. Format yang mudah dipahami dan user-friendly mempermudah akses informasi dan meningkatkan pengalaman pengguna [19].

4. Kemudahan Penggunaan (Ease of Use)

Kemudahan penggunaan mengacu pada seberapa mudah sistem digunakan tanpa memerlukan pelatihan yang rumit. Sistem yang intuitif dan mudah diakses meningkatkan kepuasan pengguna [20].

5. Ketepatan Waktu (Timeliness)

Ketepatan waktu mengacu pada kecepatan sistem dalam menyediakan informasi saat dibutuhkan. Pengguna puas jika informasi tersedia dengan cepat dan tepat waktu sesuai dengan kebutuhan operasional mereka [20].

