

BAB 2 LANDASAN TEORI

2.1 Sistem Pakar

Sistem pakar adalah sistem komputer yang dirancang untuk meniru kemampuan penyelesaian masalah dan pengambilan keputusan seperti yang dilakukan oleh seorang pakar manusia dalam bidang tertentu. Sistem pakar biasanya terdiri dari dua komponen utama: basis pengetahuan dan mesin inferensi [8].

Sistem pakar ialah termasuk kedalam beberapa domain masalah dari Artificial Intelligence (AI). Menurut Profesor Edward Feigenbaum dalam Rosnelly (2012:2) menjelaskan definisi dari sistem pakar bahwa, "Sebuah program komputer pintar (intelligent computer program) yang memanfaatkan pengetahuan (knowledge) dan prosedur inferensi (inference procedure) untuk memecahkan masalah yang cukup sulit hingga membutuhkan keahlian khusus dari manusia" (Feigenbaum, dikutip dalam Rosnelly, 2012:2). Rosnelly (2012:2) menyimpulkan bahwa, "Sistem pakar adalah sistem komputer yang ditujukan untuk meniru semua aspek kemampuan pengambilan keputusan dari seorang pakar. Sistem pakar memanfaatkan secara maksimal pengetahuan khusus selayaknya pakar dalam memecahkan suatu masalah" (Rosnelly, 2012:2).

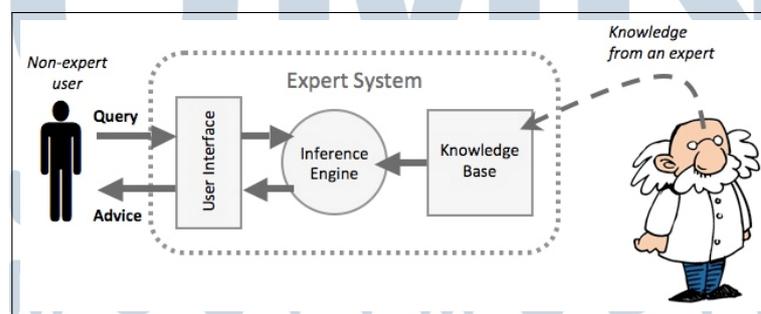
Menurut Rosnelly (2012:3), pengetahuan pakar adalah bahwa pakar adalah seseorang dengan pengetahuan dan keahlian khusus yang tidak dimiliki kebanyakan orang. Pengetahuan yang dapat dimuat kedalam sistem pakar berasal dari seorang pakar, atau pengetahuan yang berasal dari buku, jurnal, majalah, dan dokumentasi yang dipublikasikan lainnya, serta seorang yang memiliki pengetahuan meskipun bukan ahli. Menurut Rosnelly (2012:14), ada beberapa komponen yang termasuk dalam struktur dasar sistem pakar, yaitu sebagai berikut:

1. Basis Pengetahuan Berisi pengetahuan mengenai suatu pemahaman, formulasi, dan melakukan penyelesaian masalah berdasarkan suatu fakta maupun aturan yang sudah ada. Pada struktur ini, sistem melakukan penyimpanan pengetahuan dari pakar berupa rule atau aturan (if(kondisi) then(aksi) atau dapat dikatakan sebagai condition-action rules).

2. Mesin Inferensi Mesin inferensi ini ialah otak dari sistem pakar atau disebut juga sebagai control structure (struktur kontrol) atau interpretasi dari aturan

yang mengandung mekanisme pola pikir dan penalaran yang digunakan oleh pakar dalam melakukan penyelesaian pada suatu masalah dengan menyamakan bagian kondisi dari rule atau aturan yang tersimpan pada knowledge base.

3. **Working Memory** Melakukan penyimpanan yang dihasilkan oleh inference engine dengan menambahkan parameter berupa derajat kepercayaan dari fakta yang digunakan oleh aturan yang ada.
4. **Explanation Facility** Memberikan suatu kebenaran dari solusi yang telah dihasilkan kepada user (reasoning chain)
5. **Knowledge Acquisition Facility** Meliputi proses pengumpulan, pemindahan dan perubahan dari kemampuan yang meliputi pemecahan masalah dari seorang pakar atau sumber pengetahuan yang sudah terdokumentasi ke program komputer, bertujuan untuk memperbaiki maupun mengembangkan basis pengetahuan.
6. **User Interface** Mekanisme yang dilakukan untuk memberikan kesempatan kepada user atau pengguna dan sistem pakar untuk berkomunikasi, dimana antar muka menerima informasi dari pengguna dan mengubahnya dalam bentuk yang bisa diterima oleh sistem. Selain hal itu antarmuka juga dapat menerima informasi dari sistem dan melakukan penyajian ke dalam bentuk yang bisa dimengerti oleh pengguna.



Gambar 2.1. Ilustrasi Sistem Pakar [16]

Sistem pakar dapat digunakan dalam berbagai aplikasi, seperti diagnostik medis, analisis keuangan, pengelolaan sumber daya alam, dan sebagainya. Keuntungan dari sistem pakar adalah dapat menghemat waktu dan biaya yang

diperlukan untuk mempelajari bidang yang kompleks, meningkatkan efisiensi dalam pengambilan keputusan, dan mengurangi risiko kesalahan atau keputusan yang tidak tepat.

2.2 Stunting

Stunting adalah kondisi ketika anak memiliki tinggi badan lebih pendek dari standar usianya, yang diukur dalam satuan Z-score. Standar tersebut ditetapkan oleh World Health Organization (WHO) dan dikenal sebagai standar pertumbuhan anak. Anak dikatakan mengalami stunting jika tinggi badannya lebih pendek dari 2 standar deviasi di bawah standar usianya [9].

Stunting dapat terjadi pada anak yang mengalami kekurangan gizi dalam jangka waktu yang panjang, terutama pada masa awal kehidupannya (1.000 hari pertama kehidupan). Kekurangan gizi pada anak dapat disebabkan oleh berbagai faktor, seperti kurangnya asupan nutrisi, kurangnya akses ke layanan kesehatan, sanitasi yang buruk, dan kemiskinan [10].

Stunting dapat berdampak pada perkembangan fisik dan kognitif anak, serta berdampak pada kesehatan dan kualitas hidupnya di masa depan. Anak yang mengalami stunting lebih rentan terhadap penyakit infeksi, memiliki kemampuan belajar yang lebih rendah, dan kemungkinan lebih besar untuk mengalami obesitas dan penyakit tidak menular di kemudian hari. Oleh karena itu, pencegahan dan pengobatan stunting pada anak sangat penting untuk memastikan tumbuh kembang dan kualitas hidup yang optimal [11].

2.3 Algoritma Certainty Factor

Algoritma Certainty Factor adalah salah satu metode inferensi yang digunakan dalam sistem pakar untuk menentukan tingkat keyakinan atau kepastian terhadap suatu pernyataan atau hipotesis. Metode ini mengukur tingkat kepercayaan suatu aturan dengan mempertimbangkan dua faktor, yaitu faktor keyakinan dan ketidakkeyakinan terhadap suatu pernyataan [16].

Tahapan dalam menggunakan algoritma certainty factor adalah sebagai berikut:

1. Identifikasi gejala atau fakta, tahap pertama adalah mengidentifikasi gejala atau fakta yang sesuai. Gejala atau fakta ini digunakan sebagai masukan untuk proses diagnosa.

2. Membangun basis pengetahuan, basis pengetahuan sistem pakar dibangun dengan mengumpulkan pengetahuan dari pakar manusia dalam bentuk aturan yang menyatakan hubungan antara gejala dengan kondisi atau keputusan yang berkaitan. Aturan ini akan membentuk dasar penggunaan algoritma certainty factor.
3. Menghitung faktor keyakinan, digunakan untuk menghitung faktor keyakinan atau tingkat kepercayaan terhadap suatu hipotesis atau keputusan berdasarkan aturan yang ada.
4. Inferensi dan pengambilan keputusan, setelah faktor keyakinan telah dibuat tahap selanjutnya ialah melakukan inferensi atau penarikan kesimpulan berdasarkan faktor keyakinan tersebut.

Cara kerja algoritma certainty factor adalah dengan mengalikan nilai keyakinan dan ketidakkeyakinan dari setiap aturan yang terlibat dalam inferensi, kemudian menghitung nilai faktor kepastian (CF) dari setiap aturan tersebut. Terdapat dua cara dalam mendapatkan tingkat keyakinan (CF) dari sebuah rule atau aturan [18], yaitu:

1. Metode Net Belief yang diusulkan oleh E.H. Shortliffe dan B.G. Buchannan

$$CF(Rule) = MB(H, E) - MD(H, E) \quad (2.1)$$

$$MB(H, E) = \frac{Max[P(H|E)] - P(H)}{Max(1, 0) - P(H)} \quad (2.2)$$

$$MD(H, E) = \frac{Max[P(H|E)] - P(H)}{Min(1, 0) - P(H)} \quad (2.3)$$

Keterangan :

- CF (Rule) adalah Faktor Kepastian
- MB (H,E) adalah Measure of Belief(ukuran kepercayaan) terhadap hipotesis H, jika diberikan evidence E (antara 0 dan 1)
- MD (H,E) adalah Measure of Disbelief(ukuran ketidakpercayaan) terhadap evidence H, jika diberikan evidence E (antara 0 dan 1)
- P(H) adalah probabilitas kebenaran hipotesis H
- P(H—E) adalah probabilitas bahwa H benar karena fakta E

2. Dengan cara mewawancarai seorang pakar, Nilai CF(Rule) didapat dari interpretasi "term" dari pakar, yang diubah menjadi nilai CF tertentu sesuai dengan tabel kepastian.

Tabel 2.1. Interpretasi Nilai CF

Uncertainty Term	CF
Unknow (tidak tahu)	-0.2 to 0.2
Maybe (mungkin)	0.4
Probably (kemungkinan benar)	0.6
Almost certainly (hamper pasti)	0.8
Definitely (pasti)	1.0

Pada tabel 2.1 merupakan nilai untuk mengukur keyakinan dari seorang pakar. CF menunjukkan ukuran kepastian terhadap suatu fakta atau aturan nilai tertinggi dalam CF mempresentasikan nilai pasti benar sedangkan untuk nilai CF dengan aturan nilai terendah dalam CF mempresentasikan ketidakkeyakinan.

Tabel 2.2. Nilai Bobot User

No	Keterangan	Nilai User
1	Sangat Yakin	1
2	Yakin	0.8
3	Cukup Yakin	0.6
4	Sedikit Yakin	0.4
5	Tidak Tahu	0.2
6	Tidak	0

Faktor kepastian adalah selisih antara nilai keyakinan dan nilai ketidakkeyakinan, yang kemudian digunakan untuk menghitung tingkat kepercayaan pada suatu hipotesis [12].

Secara matematis, rumus Certainty Factor dapat dirumuskan sebagai berikut:

$$CF(X, Y) = CF(Y) * CF[Rule] \quad (2.4)$$

$$CF(X,Y) = \frac{P(X|Y) - P(\neg X|Y)}{P(X|Y) + P(\neg X|Y)} \quad (2.5)$$

$$CF(X,Y) = CF(lama) + CF(baru) * (1 - CFlama) \quad (2.6)$$

Keterangan singkat :

- CF adalah singkatan dari certainty factor.
- X adalah suatu pernyataan.
- Y adalah suatu fakta atau kondisi.
- $P(X|Y)$ adalah probabilitas bahwa pernyataan X benar jika fakta/kondisi Y terpenuhi.
- $P(\neg X|Y)$ adalah probabilitas bahwa pernyataan X salah jika fakta/kondisi Y terpenuhi.
- Rumus tersebut menghitung nilai certainty factor untuk pernyataan X berdasarkan fakta/kondisi Y.

Formula dasar yang digunakan untuk mendiagnosa penyakit sebagai berikut:

- Certainty factor dengan gejala tunggal

$$CF_{Gejala} = CF(user) * CF(pakar) \quad (2.7)$$

- Certainty factor dengan gejala lebih dari satu

$$CF_{combine} = CFlama + CF_{gejala} * (1 - CFlama) \quad (2.8)$$

- Untuk menghitung presentase terhadap penyakit

$$CF_{presentase} = CF_{combine} * 100 \quad (2.9)$$

2.4 Black box testing

Blackbox testing ialah suatu metode testing untuk memeriksa apakah aplikasi yang dibuat berfungsi dengan baik. Metode black box testing berguna

untuk melakukan pengujian fungsional pada aplikasi, dengan tujuan untuk memeriksa apakah input, output dan fungsi lainnya dapat sesuai dengan spesifikasi yang diperlukan, tanpa memeriksa tampilan antarmuka dan kode program [13]. Blackbox testing juga digunakan untuk menguji hasil keluaran sistem pakar sesuai dengan aturan yang ditetapkan oleh pakar sebenarnya. Dengan blackbox testing, hasilnya adalah semua fitur dalam aplikasi berjalan dengan baik berdasarkan fungsinya dan aturan yang ditetapkan oleh pakar [14].

2.5 Confusion matrix

Confusion matrix dikenal sebagai matriks kesalahan yang berfungsi untuk memberikan informasi tentang perbandingan antara hasil klasifikasi yang dilakukan oleh sistem atau model dengan hasil klasifikasi yang sebenarnya [16]. Matriks ini berbentuk tabel yang menggambarkan performa model klasifikasi berdasarkan sejumlah data uji dengan nilai sebenarnya yang diketahui. Di bawah ini terdapat gambar yang menunjukkan confusion matrix dengan empat kombinasi nilai prediksi dan nilai aktual yang berbeda [15]. Silakan perhatikan pada gambar berikut ini:

		Actual Values	
		1 (Positive)	0 (Negative)
Predicted Values	1 (Positive)	TP (True Positive)	FP (False Positive) <i>Type I Error</i>
	0 (Negative)	FN (False Negative) <i>Type II Error</i>	TN (True Negative)

Gambar 2.2. Confusion Matrix [15]

Ada empat istilah yang digunakan dalam confusion matrix untuk merepresentasikan hasil proses klasifikasi. Keempat istilah tersebut adalah True Positive (TP), True Negative (TN), False Positive (FP), dan False Negative (FN) [15].

Keterangan singkat :

1. *True Positif* (TP)

True Positif adalah data positif yang dapat diprediksi secara benar, penjelasan singkatnya pasien menderita penyakit dan hasil prediksi menyatakan pasien menderita penyakit [15].

2. *True Negatif (TN)*

True Negatif adalah data negatif yang dapat diprediksi secara benar, penjelasan singkatnya pasien tidak menderita penyakit dan hasil prediksinya menyatakan pasien tidak menderita penyakit [15].

3. *False Positive (FP) - Type Error I*

False Positive adalah data negatif yang diprediksi secara positif, penjelasan singkatnya pasien tidak menderita penyakit namun hasil prediksinya menyatakan pasien menderita penyakit [15].

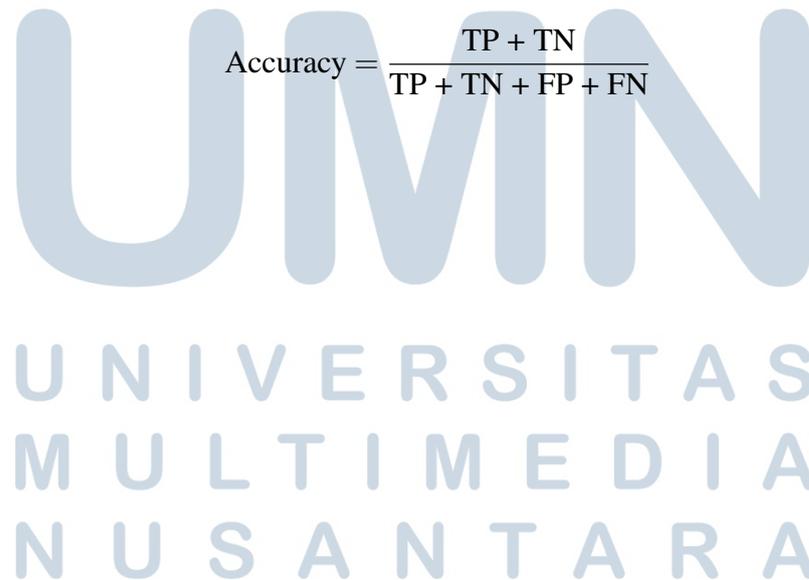
4. *False Negative (FN) - Type Error II*

False Negative adalah data positif namun diprediksi secara negatif, penjelasan singkatnya pasien menderita penyakit namun hasil prediksinya menyatakan pasien tidak menderita penyakit [15].

Confusion matrix digunakan untuk menghitung berbagai *performance metrics* untuk mengukur kinerja pada model yang dibuat. Pada penelitian ini dilakukan perhitungan pada model yang dibuat menggunakan perhitungan *accuracy*, *precision*, *recall*, dan *F1-Score*. Berikut penjelasannya :

- *Accuracy* adalah ukuran yang mengindikasikan sejauh mana model dapat melakukan klasifikasi dengan tepat, berikut rumusnya:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (2.10)$$



- *Precision* adalah metrik yang menggambarkan sejauh mana data yang diminta atau yang dianggap positif sesuai dengan hasil prediksi yang diberikan oleh model, berikut rumusnya:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (2.11)$$

- *Recall* adalah metrik yang menggambarkan sejauh mana model berhasil menemukan kembali informasi yang relevan atau penting dalam data yang ada, berikut rumusnya.

$$\text{Accuracy} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (2.12)$$

- *F1-Score* merupakan metrik evaluasi yang digunakan untuk mengevaluasi kinerja suatu sistem atau model dalam tugas klasifikasi. *F1-Score* menggabungkan presisi (*precision*) dan *recall* menjadi satu angka tunggal yang memberikan gambaran menyeluruh tentang performa sistem tersebut, berikut rumusnya:

$$\text{F1-Score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \quad (2.13)$$

