

BAB I

PENDAHULUAN

1.1 Latar Belakang

Gizi buruk merupakan kondisi serius yang dapat memengaruhi pertumbuhan fisik dan perkembangan kognitif anak. Masalah ini menjadi perhatian global karena dampaknya yang dapat berlangsung dalam jangka panjang, bahkan hingga usia dewasa. Berdasarkan Survei Studi Status Gizi Indonesia tahun 2022, prevalensi gizi buruk di Indonesia mencapai 7,7%, hal ini mengalami peningkatan dari tahun sebelumnya sebesar 7,1% pada tahun 2021 (Ramli et al., 2025). Menurut *World Health Organization* (WHO), prevalensi gizi buruk dikategorikan sebagai masalah kesehatan masyarakat yang mengkhawatirkan apabila melebihi angka 5%.

Balita atau anak usia di bawah lima tahun merupakan kelompok yang paling rentan mengalami gangguan gizi. Pada usia 0–5 tahun, anak berada dalam masa tumbuh kembang yang optimal (*golden age*), sehingga gangguan gizi pada fase ini dapat memengaruhi kesehatan, perkembangan dan kualitas generasi penerus. Jika tidak segera ditangani dengan cepat dan tepat, gizi buruk dapat menimbulkan dampak serius termasuk meningkatnya risiko kematian pada usia muda (Lubis & Irawan, 2023).

Kurangnya pemahaman orang tua terhadap status dan kebutuhan gizi balita menyebabkan keterlambatan dalam mengenali serta menangani gejala gizi buruk. Kondisi ini menjadi hambatan dalam upaya menurunkan prevalensi gizi buruk, terutama di daerah dengan akses terbatas terhadap tenaga ahli gizi, seperti wilayah pedesaan dan terpencil. Untuk mengatasi masalah tersebut, diperlukan peran

teknologi informasi guna mengatasi keterlambatan penanganan dan membantu dalam proses diagnosis awal gejala gizi buruk. Salah satu solusi yang dapat diterapkan adalah penggunaan sistem pakar, yaitu sistem yang dapat membantu masyarakat mengenali gejala gizi buruk secara cepat dan memberikan rekomendasi penanganan yang sesuai.

Menurut Abbrar et al.. (2025) Sistem pakar adalah sistem yang mengadopsi pengetahuan manusia ke dalam komputer untuk menyelesaikan masalah seperti yang dilakukan oleh para ahli. Pada penelitian ini, sistem pakar dibangun menggunakan metode *Certainty Factor* (CF) untuk menghitung tingkat keyakinan diagnosis penyakit gizi buruk berdasarkan gejala yang dimasukkan disertai tingkat keyakinannya. Metode *Certainty Factor* memungkinkan pakar utnuk dapat merepresentasikan keyakinannya melalui pemberian bobot pada setiap gejala, berdasarkan pengalaman dan pengetahuan yang dimiliki.

Metode ini terbukti efektif dalam menangani permasalahan dengan tingkat ketidakpastian, seperti dalam proses diagnosis penyakit. Berbagai penelitian sebelumnya menunjukkan bahwa penggunaan *Certainty Factor* dapat menghasilkan sistem pakar yang akurat. Salah satu contohnya adalah penelitian oleh Ajisari et al. (2024) yang berjudul “*Sistem Pakar Diagnosa Penyakit Kardiovaskular Menggunakan Metode Certainty Factor*”. Hasil penelitian tersebut menunjukkan bahwa sistem yang dikembangkan berhasil memperoleh akurasi sebesar 100%. Selain itu, *usability testing* juga menunjukkan bahwa antarmuka sistem, termasuk halaman utama, halaman konsultasi, dan halaman hasil, mudah digunakan serta informatif bagi pengguna.

Berdasarkan uraian di atas, maka penelitian ini mengangkat judul: “**Sistem Pakar Diagnosis Gizi Buruk pada Balita Menggunakan Metode Certainty Factor Berbasis Framework Laravel**”. Penelitian ini diharapkan dapat membantu dalam proses penanganan gizi buruk pada balita.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, rumusan masalah dalam penelitian ini adalah:

1. Bagaimana membantu masyarakat atau orang tua dalam mendeteksi gizi buruk pada balita?
2. Bagaimana menerapkan metode *Certainty Factor* dalam mendiagnosis gizi buruk pada balita?
3. Bagaimana menghasilkan aplikasi sistem pakar gizi buruk pada balita menggunakan metode *Certainty Factor* berbasis *framework laravel*?

1.3 Tujuan dan Manfaat Penelitian

1.3.1 Tujuan Penelitian

Adapun tujuan dari penelitian sistem pakar diagnosa gizi buruk pada balita ini adalah sebagai berikut:

1. Membantu masyarakat dan orang tua dalam mendeteksi gizi buruk pada balita.
2. Menerapkan metode *Certainty Factor* dalam mendiagnosis gizi buruk pada balita.
3. Menghasilkan aplikasi sistem pakar gizi buruk pada balita menggunakan metode *Certainty Factor* berbasis *framework laravel*.

1.3.2 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Mempermudah masyarakat dalam mendeteksi gizi buruk pada balita.
2. Mengetahui penerapan metode *Certainty Factor* dalam mendiagnosis gizi buruk pada balita.
3. Menambah wawasan dalam pembuatan perangkat lunak sistem pakar gizi buruk pada balita menggunakan metode *Certainty Factor* berbasis *framework laravel*.

1.4 Batasan Masalah

1. Metode yang digunakan pada sistem ini adalah Metode *Certainty Factor* dan berbasis *framework laravel*.
2. Penelitian hanya berfokus pada anak usia 0-5 tahun.
3. Penyakit dan gejala yang digunakan dalam sistem ini didasarkan pada data dan wawancara dengan ahli gizi di Rumah Sakit Surya Insani, Ibu Dwi Astuti, AMd.Gz.
4. Penelitian ini hanya menggunakan dua jenis penyakit gizi buruk, yaitu *marasmus* dan *kwashiorkor*.
5. Sistem menerima *input* berupa data gejala, data penyakit dan basis pengetahuan.
6. *Output* dari penelitian ini adalah sistem pakar berbasis web yang dapat membantu pengguna dalam mendiagnosis gizi buruk pada balita. Sistem menghasilkan diagnosis jenis penyakit gizi buruk dengan tingkat keyakinan dan memberikan solusi penanganan yang dapat dilakukan.

1.5 Metode Pengumpulan Data

Metode pengumpulan data untuk memperoleh data yang relevan dan mendukung penelitian ini, metode pengumpulan data yang digunakan meliputi:

1. Observasi, yaitu pengamatan yang dilakukan selama proses wawancara dengan ahli gizi. Peneliti mengamati cara ahli gizi menjelaskan kondisi balita dengan gizi buruk, termasuk ciri-ciri fisik, kebiasaan makan, dan indikator lainnya. Observasi ini bertujuan untuk memahami bagaimana pakar mengenali gejala-gejala gizi buruk pada balita.
2. Data, yaitu mengumpulkan informasi terkait gizi buruk, seperti jenis penyakit gizi buruk, gejala-gejala gizi buruk, dan penanganan yang tepat, proses diagnosis yang biasa dilakukan, serta data pendukung lainnya yang relevan.
3. Wawancara, bertujuan untuk memperoleh informasi yang relevan mengenai gizi buruk pada balita. Peneliti melakukan wawancara dengan Ahli Gizi Rumah Sakit Surya Insani, Ibu Dwi Astuti, AMd.Gz.

1.6 Sistematika Penulisan

Sistematika penulisan dari skripsi ini terdiri dari enam bagian utama sebagai berikut :

BAB I PENDAHULUAN

Bab ini berisi latar belakang, rumusan masalah, tujuan penelitian, batasan masalah, manfaat penelitian, metodologi penelitian dan sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini berisi teori-teori yang digunakan pada penelitian ini. Teori-teori yang berhubungan dengan aplikasi diagnose gizi buruk pada balita berdasarkan gejala menggunakan metode *Certainty Factor*.

BAB III METODOLOGI PENELITIAN

Bab ini berisi tahapan-tahapan dalam pengumpulan data, perancangan sistem, perumusan masalah dan analisa.

BAB IV ANALISA DAN PERANCANGAN

Bab ini berisi Analisa dan perancangan sistem diagnosa gizi buruk pada balita berdasarkan gejala menggunakan metode *Certainty Factor*.

BAB V IMPLEMENTASI DAN PENGUJIAN

Bab ini berisi implementasi dari analisa dan perancangan serta pengujian pada sistem yang berhasil dibangun.

BAB VI PENUTUP

Bab ini berisi kesimpulan dari hasil penelitian yang telah dilakukan dan saran-saran untuk pengembangan sistem atau penelitian selanjutnya.

BAB II

LANDASAN TEORI

2.1 Sistem

Sistem berasal dari bahasa Latin *systēma* atau bahasa Yunani *sustēma* yang berarti suatu kesatuan yang terdiri dari komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi, atau energi (Suawah, 2021). Pengertian sistem adalah suatu kesatuan yang terdiri komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi untuk mencapai suatu tujuan. Atau dapat juga dikatakan bahwa pengertian sistem adalah sekumpulan unsur elemen yang saling berkaitan dan saling mempengaruhi dalam melakukan kegiatan bersama untuk mencapai suatu tujuan (Effendy et al., 2023).

Menurut Azhar Susanto sistem adalah pengelompokan atau kumpulan subsistem, bagian, atau komponen, atau apa pun, baik fisik maupun non fisik, yang saling terhubung satu sama lain dan dapat bekerja sama untuk mencapai suatu tujuan (Anika et al., 2023). Sistem adalah serangkaian komponen yang saling berinteraksi dan bergantung satu sama lain untuk mencapai tujuan tertentu (Sitorus & Sakban, 2021).

Murdick mendefinisikan sistem sebagai perangkat elemen atau pengolahan berbentuk kegiatan maupun prosedur untuk mencari tujuan yang sama dalam menjalankan data dalam waktu yang ditentukan sehingga menghasilkan sebuah informasi, energi maupun barang. Secara sederhana, menurut Murdick sistem yaitu

sekumpulan elemen yang membentuk prosedur-prosedur yang mencari suatu tujuan tertentu (Efendi et al., 2023).

Anjeli et al. (2022) menyatakan bahwa sistem adalah suatu kesatuan yang terdiri dari dua atau lebih komponen atau subsistem yang berinteraksi untuk mencapai suatu tujuan yang sama. Sistem dapat membantu dan mempermudah suatu pekerjaan yang berbasis komputer atau *online*. Sistem juga dapat dipahami sebagai gabungan dari elemen-elemen yang mencakup (*input*), pengolahan (*processing*), dan pengeluaran (*output*). Secara mudah, sistem dapat diartikan sebagai sekumpulan elemen ataupun variabel yang saling terkait, terstruktur, dan berinteraksi satu sama lain (Anwar et al., 2024).

2.2 Pakar

Pakar adalah seorang yang mempunyai pengetahuan, pengalaman dan metode khusus, serta mampu menerapkannya untuk memecahkan masalah atau memberi nasehat. Seorang pakar harus mampu menjelaskan dan mempelajari hal-hal yang berkaitan dengan topic permasalahan (Dona et al., 2021).

Palelleng et al. (2025) mendefinisikan bahwa pakar adalah orang yang secara luas diakui sebagai sumber yang dapat diandalkan dari suatu bidang atau keahlian tertentu, yang bakatnya dalam menyelesaikan sesuatu dengan benar menurut aturan tertentu atau seseorang dalam bidang keahlian tertentu. Sebutan lain dari sistem pakar adalah *knowledge-based expert system* yang memiliki kemampuan mengadopsi cara kerja seorang ahli atau pakar dalam menyelesaikan permasalahan. Pakar merupakan seseorang yang memiliki kemampuan dalam menjelaskan dan menganalisis topik-topik yang sesuai dengan bidang keahliannya (Pigi et al., 2022).

Konsultasi terhadap seseorang yang memiliki *expertise* dibidang tertentu dalam menyelesaikan suatu permasalahan merupakan pilihan tepat guna mendapatkan jawaban, saran, solusi, keputusan atau kesimpulan terbaik. Jawaban seorang *expert* atas sebuah konsultasi tentunya sangat dapat dipercaya atau dipertanggungjawabkan serta dapat berpengaruh terhadap mutu serta kualitas hasil dari suatu permasalahan, ini dikarenakan seorang *expert* selalu menguasai terhadap bidang yang ditekuninya berdasarkan keilmuan dan pengalamannya (Gusmita et al., 2023).

2.3 Sistem pakar

Menurut E. Fraim Turban Sistem pakar adalah sebuah sistem yang ada pada komputer yang dapat menirukan tindakan manusia dalam menyelesaikan masalah, sedangkan menurut Garratano dan Riley menyatakan bahwa sistem pakar merupakan sistem yang mampu meniru atau menyamai kemampuan dari seorang pakar pada bidangnya. Sistem pakar dimanfaatkan sebagai sistem yang berkemampuan baik untuk membantu memecahkan masalah dan memindahkan pengetahuan dari seorang pakar ke dalam bentuk sebuah sistem komputer (Asnawati & Simanjuntak, 2025).

Menurut Hartati sistem pakar (*expert system*) adalah sistem berbasis pengetahuan, salah satu cabang keilmuan dalam kecerdasan buatan dan dapat didefinisikan sebagai perangkat lunak berpengetahuan intensif yang dapat melakukan beberapa tugas yang biasanya membutuhkan keahlian manusia. Sistem pakar digunakan untuk memecahkan masalah domain spesifik. Setiap langkah

penalaran untuk masalah yang spesifik ditentukan oleh pakar manusia secara profesional (Febriyanto et al., 2024).

Sistem pakar adalah salah satu bagian dari kecerdasan buatan yang menggabungkan pengetahuan dan penelusuran data untuk memecahkan masalah yang memerlukan keahlian manusia (Kevin et al., 2022). Sistem pakar atau juga dikenal sebagai sistem basis pengetahuan, adalah aplikasi komputer yang dirancang untuk mendukung pengambilan keputusan atau pemecahan masalah di bidang tertentu. Sistem ini disebut sistem pakar karena fungsi dan perannya seperti profesional yang mengharuskannya memiliki pengetahuan dan pengalaman pemecahan masalah (Nashiruddin & Hidayat, 2022). Sistem pakar adalah suatu sistem informasi yang berusaha mengadopsi pengetahuan dari manusia ke komputer, agar komputer dapat menyelesaikan masalah layaknya seorang pakar (Ayu et al., 2023).

Sistem pakar merupakan sistem berbasis komputer yang menggunakan ilmu pengetahuan, fakta dan teknik penalaran dalam menyelesaikan bentuk permasalahan yang biasanya dilakukan oleh seorang pakar atau seorang yang seorang yang ahli pada bidang tersebut. Sistem pakar dimanfaatkan untuk dapat mencari solusi sebagaimana pakar melakukannya (Refany & Fajrin, 2024). Sistem pakar hadir menjadi pembantu atau asisten yang akan menuntun seseorang menyelesaikan permasalahan dengan dukungan data kepakaran yang disimpan dalam komputer. Dengan bantuan kepakaran, informasi dirangkum dalam database sebagai sumber penanganan diagnosis kerusakan sampai solusi yang akan dilakukan sebagai langkah penyelesaian permasalahan (Sagala et al., 2021).

Sistem pakar menyimpan informasi yang diperoleh dari ahli/pakar akan tetapi fungsi yang dimiliki tidak sepenuhnya dapat menggantikan kemampuan pakar tersebut, melainkan mewakili pengetahuan, pengalaman serta kemampuan seorang ahli. Sistem pakar dapat bekerja dengan baik apabila diimplementasikan dengan algoritma/metode yang mampu mendeskripsikan kepercayaan terhadap suatu aturan (*rule*) ataupun fakta berdasarkan intensitas keyakinan yang dimiliki dimana dalam penelitian ini sistem pakar yang dibangun mengimplementasikan metode *Certainty Factor* (Suarnatha & Gunawan, 2022).

Konsep Sistem Pakar.

Konsep sistem pakar dapat meliputi enam hal berikut (Dona et al., 2021):

1. Kepakaran (*Expertise*)

Kepakaran merupakan suatu pengetahuan yang diperoleh dari pelatihan, membaca, dan pengalaman. Kepakaran inilah yang memungkinkan para ahli dapat mengambil keputusan lebih cepat dan lebih baik daripada seseorang yang bukan pakar.

2. Pakar (*Expert*)

Pakar adalah seorang yang mempunyai pengetahuan, pengalaman dan metode khusus, serta mampu menerapkannya untuk memecahkan masalah atau memberi nasehat. Seorang pakar harus mampu menjelaskan dan mempelajari hal-hal yang berkaitan dengan topic permasalahan.

3. Pemindahan kepakaran (*Transferring Expertise*)

Tujuan dari sistem pakar adalah memindahkan kepakaran dari seseorang pakar ke dalam *computer*, kemudian ditransfer kepada orang lain yang bukan pakar.

4. Inferensi (*Inferencing*)

Inferensi adalah sebuah prosedur (program) yang mempunyai kemampuan dalam melakukan penalaran. Semua pengetahuan yang dimiliki oleh seorang pakar disimpan pada basis pengetahuan oleh sistem pakar. Tugas mesin inferensi adalah mengambil kesimpulan berdasarkan basis pengetahuan yang dimilikinya.

5. Aturan-aturan (*Rule*)

Kebanyakan software sistem pakar komersional adalah sistem yang berbasis *rule* (*rule-based system*), yaitu pengetahuan disimpan terutama dalam bentuk *rule*, sebagai prosedur-prosedur pemecahan masalah.

6. Kemampuan menjelaskan (*Explanation Capability*)

Fasilitas lain dari sistem pakar adalah kemampuan untuk menjelaskan saran atau rekomendasi yang diberikannya. Penjelasan dilakukannya dalam subsistem yang disebut subsistem penjelasan (*explanation*).

2.4 Gizi Buruk

Menurut Kemenkes RI Gizi buruk adalah kondisi yang terjadi akibat tubuh anak tidak mendapatkan asupan nutrisi yang cukup untuk memenuhi kebutuhannya, baik karena kekurangan maupun kelebihan. Kondisi ini bisa berdampak pada pertumbuhan fisik dan perkembangan kognitif anak (Ramlil et al., 2025). Gizi buruk dapat dijelaskan sebagai keadaan di mana seseorang mengalami kekurangan asupan karbohidrat, protein, dan kalori. Kondisi ini menyebabkan individu tidak mampu memenuhi kebutuhan gizi yang diperlukan untuk menjalankan aktivitas sehari-hari dan mengalami pertumbuhan yang optimal (Lubis & Irawan, 2023).

Gizi buruk atau yang sering disebut sebagai *Malnutrition* merupakan kondisi serius di mana berat badan seorang balita jauh lebih rendah dibandingkan tinggi badannya akibat asupan nutrisi yang kurang memadai untuk pertumbuhannya. Kekurangan gizi dapat terjadi saat tubuh kekurangan kalori, protein, atau keduanya. Kondisi gizi buruk pada anak-anak kecil terjadi ketika berat badan dan tinggi badan, seperti yang diukur dengan *Z-score* kurang dari -3 Standar Deviasi (SD), mengakibatkan masalah gizi klinis seperti *marasmus*, *kwashiorkor*, dan *marasmus-kwashiorkor* (Sihite et al., 2024).

Marasmus adalah bentuk gizi buruk yang umum terjadi pada balita, ditandai dengan kekurangan kalori yang parah yang menyebabkan tubuh menjadi sangat kurus sehingga tulang-tulang terlihat jelas dan rentan terhadap infeksi serta dehidrasi. *Marasmus* disebabkan oleh asupan makanan yang tidak mencukupi kalori, yang memaksa tubuh untuk terus-menerus menggunakan cadangan energinya. Gejalanya meliputi: tubuh kurus dan kecil, wajah terlihat tua, kepala terlihat besar tidak sesuai dengan usia, anak menjadi apatis dan cengeng, rentan terhadap infeksi, diare, dan dehidrasi (Fitriani et al., 2024).

Kwashiorkor adalah kekurangan protein yang parah, yang sering terjadi meskipun asupan energi (karbohidrat) cukup. Gejala khas *kwashiorkor* termasuk edema (pembengkakan tubuh), perut buncit, perubahan warna rambut, dan lesu. Kondisi ini sering terjadi Ketika pola makan anak sangat miskin protein namun cukup kalori. *Kwashiorkor* dapat menyebabkan kerusakan permanen pada organ dan perkembangan mental jika tidak segera ditangani (Jones et al., 2023).

Kwashiorkor umum terjadi di negara berkembang, terutama pada anak-anak yang lebih tua dan bayi yang telah disapih. Kondisi ini sering muncul di daerah rawan pangan atau kelaparan, terutama di negara-negara yang makanan pokoknya adalah jagung, nasi, dan kacang-kacangan. Edema merupakan ciri pembeda utama dari *kwashiorkor*, yang tidak ditemukan pada *marasmus*. Edema disebabkan oleh kombinasi rendahnya kadar albumin serum, meningkatnya hormon kortisol, dan ketidakmampuan mengaktifkan hormon antidiuretik. *Marasmik-kwashiorkor* adalah bentuk gabungan dari *marasmus* dan *kwashiorkor*. Anak-anak dengan kondisi ini menunjukkan penyusutan tubuh yang parah (*wasting*) disertai edema. Biasanya, manifestasi kulit dan rambutnya lebih ringan dibandingkan *kwashiorkor* murni, tetapi hati membesar (*hepatomegali*) tetap ditemukan (Dipasquale et al., 2020).

2.5 Certainty Factor

Metode CF telah dikembangkan oleh Shortliffe dan Buchanan pada pertengahan tahun 1970-an untuk MYCIN, yaitu sebuah sistem pakar yang digunakan dalam mengobati dan mendiagnosa penyakit infeksi darah dan meningitis (Sukiakhy et al., 2022). Metode *Certainty Factor* dapat menampilkan nilai kepercayaan terhadap fakta yang diidentifikasi, selain itu dengan metode ini pengguna dapat memberikan nilai berdasarkan tingkat keparahan gejala yang dirasakan (Ariawan & Putra, 2023). Menurut Marbun et al. (2022) metode *Certainty Factor* menggunakan suatu nilai untuk mengasumsikan derajat keyakinan seseorang pakar terhadap suatu nilai. Metode ini menggunakan perhitungan

berdasarkan kemiripan yang dibagi dengan bobot yang telah ditentukan. Metode CF menunjukkan ukuran kepastian terhadap suatu fakta atau aturan.

Certainty Factor digunakan untuk mengetahui sebuah permasalahan yang belum pasti jawabannya kedalam bentuk hasil yang mungkin atau hampir pasti hasilnya/hipotesa. *Certainty Factor* atau CF merupakan nilai untuk mengukur keyakinan pakar (Dahria et al., 2023). Metode *Certainty Factor* memiliki kelebihan yaitu metode ini sangat cocok digunakan untuk sistem pakar karena keakuratan data yang diolah dapat terjaga karena dalam sekali proses perhitungan keakuratan hanya dapat mengolah dua data saja. Metode *Certainty Factor* (Faktor Kepastian) dapat menjadi jawaban keyakinan kepada pakar atas ketidakpastian dalam menerjemahkan informasi atau pengetahuan melalui analisis yang membentuk sebuah matrik (Putra & Yunus, 2021).

Prinsip dasar dari metode CF ialah dengan mendefinisikan keyakinan/kepastian terhadap suatu fakta atau aturan berdasarkan tingkat keyakinan dari seorang pakar/ahli (*rule base*). Perhitungan metode CF dilakukan dengan menghitung nilai perkalian antara CFuser dengan nilai CFpakar, sehingga menghasilkan nilai CFkombinasi. (Palelleng et al., 2025). Team pengembang MYCIN mencatat bahwa dokter sering kali menganalisis informasi yang ada dengan ungkapan seperti misalnya : mungkin, kemungkinan besar, hampir pasti (Dwi Meilani et al., 2022).

Menurut penelitian oleh Kevin, 2022, berikut langkah-langkah metode *Certainty Factor* :

1. Penentuan gejala penyakit gizi buruk pada balita

2. Penentuan data gejala
3. Penentuan data gabungan antara data gejala dan data penyakit
4. Penentuan nilai CF pakar dan CF pasien
5. Pemilihan data gejala oleh user/pakar
6. Perhitungan nilai CF dari gejala user/pakar
7. Hasil diagnosis penyakit gizi buruk

Untuk mengekspresikan tingkat keyakinan, seorang ahli menggunakan sebuah nilai yang dikenal sebagai *Certainty Factor* (CF) untuk menentukan seberapa yakin ia terhadap suatu data. Berikut adalah rumusan dasar dari Faktor Keyakinan:

$$CF[H,E] = MB[H,E] - MD[H,E]$$

Keterangan:

CF: *Certainty Factor* (faktor kepastian) dalam hipotesis H yang dipengaruhi oleh fakta E

MB: *Measure of Belief* (tingkat keyakinan), ialah skala peningkatan keyakinan pada hipotesis H yang dipengaruhi oleh fakta E.

MD: *Measure of Disbelief* (tingkat ketidakkeyakinan), merupakan peningkatan dari ketidakpercayaan pada hipotesis H yang dipengaruhi oleh fakta E.

E : *Evidence* (peristiwa atau fakta)

H : *Hipotesis* (dugaan)

Tabel 2.1 Bobot *Certainty Factor*

No.	Keterangan	Bobot
1	Tidak Tahu/Tidak Yakin	0
2	Hampir Mungkin	0,2

3	Mungkin	0,4
4	Cukup Yakin	0,6
5	Yakin	0,8
6	Sangat yakin	1

(Sumber: (Ajisari & Prasetyaningrum, 2024)

Dalam *certainly theory*, data-data kualitatif dipresentasikan sebagai derajat keyakinan (*degree of belief*). Ada dua langkah dalam perepresentasian data-data kualitatif. Langkah pertama adalah kemampuan untuk mengekspresikan derajat keyakinan sesuai dengan metodenya. Langkah kedua adalah mampu untuk menempatkan data mengkombinasikan derajat keyakinan tersebut dalam sistem pakar. Konsep ini kemudian diformulasikan dalam rumus dasar *Certainty Factor*.

Certainty Factor untuk kaidah dengan gejala tunggal :

$$CF [H, E] = CF[user] * CF[pakar] \quad (1)$$

Selanjutnya adalah melakukan kombinasi nilai CF dari masing-masing gejala.

Dengan Rumus ditunjukkan pada persamaan 2 dan 3:

$$CF \text{ Combine } [H,E]1,2 = CF[H,E]1 + CF[H,E]2 * (1-CF[H,E]1) \quad (2)$$

$$CFCombine [H,E]old,3 = CF[H,E]old + CF[H,E]3 * (1 -CF[H,E]old) \quad (3)$$

Keterangan:

CFCombine [H,E]1,2 : Kombinasi dari *CF[H,E]1* dan *CF[H,E]2*

CFCombine [H,E]*old*,3 : Kombinasi dari perhitungan *CF[H,E]old* atau *CF[H,E]1,2* dengan *CF[H,E]3*

Perhitungan kombinasi antar CF ini akan terus dilakukan sampai gejala yang diinputkan oleh *user* berakhir. Setelah melakukan kombinasi antar CF dan

menemukan nilai keyakinan, berikutnya adalah mengubah nilai keyakinan menjadi persentase keyakinan. Dengan Rumus ditunjukkan pada persamaan 4:

$$CF = CF_{old} \times 100\% \quad (4)$$

Keterangan:

CF_{old} : $CF_{Combine}$ terakhir dari kemungkinan gejala yang ada (Fikri et al., 2021).

2.6 Balita

Balita adalah kelompok anak yang berada pada rentang usia 0-5 tahun. Masa balita merupakan periode penting dalam proses tumbuh kembang manusia dikarenakan perkembangannya berlangsung lebih cepat. Tumbuh kembang tersebut merupakan suatu proses yang berkelanjutan dari konsepsi (pembuahan benih) sampai dewasa yang dipengaruhi oleh faktor *genetic* dan lingkungan (Ula et al., 2021). Balita adalah anak usia di bawah lima tahun, dimana pada usia ini merupakan periode yang sangat penting dalam kehidupan manusia. Apabila terjadi permasalahan kesehatan pada periode ini, maka besar kemungkinan terjadi permasalahan kesehatan pada periode kehidupan selanjutnya (Susanti et al., 2023).

2.7 Rumah Sakit

Rumah sakit adalah institusi pelayanan kesehatan yang menyelenggarakan pelayanan kesehatan perorangan secara paripurna yang menyediakan pelayanan rawat inap, rawat jalan dan gawat darurat (Harmoni et al., 2022). Rumah sakit adalah sarana pelayanan kesehatan yang menyelenggarakan pelayanan medis perorangan secara menyeluruh yang menyediakan pelayanan rawat inap dan jalan, serta pelayanan gawat darurat. Salah satu fungsi rumah sakit adalah memberikan

elayanan pengobatan dan rehabilitasi yang sesuai standar pelayanan rumah sakit (Ananda, 2023).

2.8 *Draw.io*

Draw.io merupakan sebuah situs yang didesain khusus untuk menggambar diagram secara *online*. Untuk mengaksesnya hanya diperlukan *browser* yang mendukung HTML dan juga koneksi internet. *Draw.io* sudah terintegrasi dengan *Google Drive* untuk penyimpanan file selain mengekspor dalam bentuk JPG/PNG/SVG/XML (Nasir & Yuniawati, 2023).

2.9 *Unified Modelling Language (UML)*

UML (*Unified Modelling Language*) disebut sebagai sebuah teknik pengembangan sistem yang menggunakan bahasa grafis sebagai alat untuk pendokumentasian dan melakukan spesifikasi pada sistem. UML pertama kali dipopulerkan oleh Grady Booch dan James Rumbaugh pada tahun 1994 untuk mengkombinasikan dua metodologi terkenal yaitu Booch dan OMT, kemudian Ivar Jacobson, yang menciptakan *Object Oriented Software Engineering* (OOSE) ikut bergabung. Standar UML dikelola oleh *Object Management Group* (OMG) (Okmayura et al., 2023).

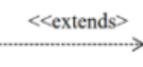
Menurut Sukamto dan Shalahuddin UML (*Unified Modeling Language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (Ramadhan & Gusrianto, 2021). UML juga dapat diartikan sebagai bahasa permodelan perangkat lunak yang biasa dipakai dalam perancangan sistem berorientasi objek. UML juga dapat memberikan

gambaran software yang dirancang secara visual, mendokumentasi dan menspesifikasikannya agar lebih mudah dimengerti oleh programmer serta pengguna (Febryanda et al., 2024).

2.9.1 Use Case Diagram

Use case diagram adalah gambaran interaksi antara sistem dengan lingkungannya, untuk memperlihatkan proses aktivitas secara urut dalam suatu sistem (Okmayura et al., 2023).

Tabel 2.2 Simbol-simbol Use Case Diagram

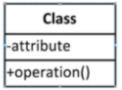
Simbol	Deskripsi
<i>Use Case</i> 	Unit-unit yang saling bertukar pesan antar aktor.
<i>Actor</i> 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi.
<i>Association</i> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi.
<i>Ekstensi/extend</i> 	Relasi <i>use case</i> tambahan.
<i>Generalisasi/Generalized</i> 	Hubungan generalisasi dan spesialisasi (umum-khusus).
<i>Included</i> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankannya <i>use case</i> ini.

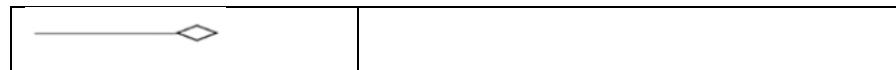
Sumber : (Nurlita & Anggraini, 2023)

2.9.2 Class Diagram

Class diagram merupakan hubungan antar *class* sudah terdapat nama *class*, *attribute* dan *method* (Setiaji & Sastra, 2021).

Tabel 2.3 Simbol-simbol Class Diagram

Simbol	Deskripsi
<p>Kelas/class</p> 	Kelas merupakan Gambaran dari struktur sistem. Atribut adalah penggambaran tentang keadaan dari suatu objek. Operasi adalah penggambaran tentang fungsi.
<p>Antar muka / <i>Interface</i></p> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
<p>Asosiasi / <i>Association</i></p> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
<p>Asosiasi berarah / <i>Direct association</i></p> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
<p>Generalisasi</p> 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus).
<p>Kebergantungan/ <i>Dependency</i></p> 	Relasi antar kelas dengan makna kebergantungan antar kelas.
<p>Aggregasi / <i>Aggregation</i></p>	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>)



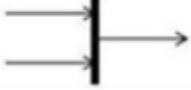
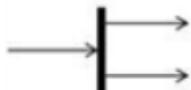
Sumber: (Nurlita & Anggraini, 2023)

2.9.3 *Activity Diagram*

Activity diagram merupakan alur dari sistem atau perangkat lunak (Setiaji & Ricki Sastra 2021). *Activity diagram* berfungsi untuk menunjukkan alur program yang terjadi antara user dan sistem yang berjalan. Setiap proses pada sistem akan digambarkan dengan diagram aktifitas. *Activity Diagram* dirancang berdasarkan *use case* yang telah dibuat (Nugraha & Rosmeida, 2021).

Tabel 2.4 Simbol-simbol Activity Diagram

NO	BENTUK SIMBOL	NAMA SIMBOL	FUNGSI SIMBOL
1.		<i>Activity</i>	Menyatakan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
2.		<i>Control Flow</i>	Menunjukkan urutan eksekusi.
3.		<i>Object Flow</i>	Menunjukkan aliran objek dari sebuah <i>action</i> atau <i>activity</i> ke <i>action</i> .
4.		<i>Start Point</i>	Menyatakan bahwa sebuah objek dibentuk atau diawali.
5.		<i>End Point</i>	Menyatakan bahwa sebuah objek dibentuk atau diakhiri.

6.		<i>Join/Penggabungan</i>	Menyatakan untuk menggabungkan kembali <i>activity</i> atau <i>action</i> yang parallel.
7.		<i>Fork</i>	Menyatakan untuk memecah <i>behavior</i> menjadi <i>activity</i> atau <i>action</i> yang parallel.
8.		<i>Decision</i>	Menunjukkan penggambaran suatu keputusan/tindakan yang harus diambil pada kondisi tertentu.

Sumber : (Ramdany, 2024)

2.9.4 *Sequence Diagram*

Sequence diagram mengilustrasikan objek-objek yang berpartisipasi di dalam suatu use case. *Sequence diagram* atau *interaction diagram* digunakan untuk memodelkan interaksi objek di dalam sebuah *use case* (proses). *Sequence diagram* memperlihatkan interaksi yang memuat himpunan dari objek dan relasi yang terjadi antar objek tersebut, termasuk juga bagaimana *message* (pesan) mengalir di antara objek (Hidayati et al., 2023).

Tabel 2.5 Simbol-simbol Sequence Diagram

Simbol	Deskripsi
<i>Actor</i>  Actor	Menggambarkan orang yang sedang berinteraksi dengan sistem
<i>Entity Class</i>	Menggambarkan hubungan dan kegiatan yang akan dilakukan

<i>Boundary Class</i> 	Menggambarkan hubungan suatu elemen yang berbeda.
<i>Control Class</i> 	Menggambarkan penghubung antara <i>boundary</i> dengan tabel.
<i>A focus of control & a life line</i> 	Menggambarkan tempat dimulainya dan berakhirnya sebuah pesan.
<i>Message</i> 	Menggambarkan spesifikasi dari komunikasi antara objek yang memuat informasi-informasi tentang aktivitas yang terjadi.

Sumber: (Nurlita & Anggraini, 2023)

2.10 Bahasa Pemrograman

Pengertian dari bahasa pemrograman yaitu sejumlah simbol, dan kata-kata kode yang membuat seorang programmer mampu menyampaikan perintah atau instruksi kepada komputer. Dalam menciptakan sebuah program, seorang programmer menggunakan sebuah perangkat pemrograman. Dalam pembuatan sebuah aplikasi web yang dapat dikembangkan lebih lanjut, maka dibutuhkan suatu bahasa *markup*, bahasa pemrograman, bahasa *script* yang fungsinya agar *web* terlihat lebih dinamis dan juga terdapat bahasa *stylesheet* (Sukiakhy et al., 2022).

2.10.1 Hypertext Markup Language (HTML)

HTML, singkatan dari *Hypertext Markup Language*, adalah bahasa markup yang digunakan untuk menentukan format dan gaya teks yang ditandai (Manurung & Sembiring, 2024). *Hypertext Markup Language* (HTML) merupakan sebuah bahasa markah yang dipergunakan buat menghasilkan sebuah laman *web*, menampilkan aneka macam berita didalam sebuah penjelajah *web* internet serta performatan hiperteks sederhana yang ditulis dalam berkas format ASCII supaya bisa menghasilkan tampilan wujud yang terintegrasi. *Hypertext Markup Language* merupakan kepanjangan dari bahasa pemrograman HTML, yaitu bahasa pemrograman baku yang dipergunakan untuk menghasilkan sebuah halaman *web* (Rahmasari et al., 2024).

2.10.2 *Cascading Style Sheet* (CSS)

Cascading Style Sheet (CSS) merupakan aturan untuk mengatur beberapa komponen dalam sebuah *web* sehingga akan lebih terstruktur dan seragam (Kumalasari et al., 2023). Andy Antonius Setiawan, Arie S.M. Lumenta & Sherwin R.U.A. Sompie (2016) Menyatakan bahwa *Cascading Style Sheets* (CSS) adalah suatu bahasa pemrograman yang digunakan untuk mendukung pembuatan website agar memiliki tampilan yang lebih menarik dan terstruktur. CSS dikembangkan oleh W3C. organisasi yang mengembangkan teknologi internet. Tujuannya tak lain untuk mempermudah proses penataan halaman *web* (Risaldy & Hardinata, 2023).

Cascading Style Sheets (CSS) merupakan Bahasa pemrograman *mark-up* (HTML) yang biasa digunakan untuk membuat tampilan halaman *web* lebih menarik. *Cascading Style Sheets* (CSS) mengontrol format tampilan pada halaman HTML dan XHTML dan bisa juga diaplikasikan pada segala dokumen XML,

SVG.dan XUL. CSS digunakan untuk memisahkan kontek utama dengan tampilan dokumen seperti layout, warna dan font (Orisa et al., 2023).

2.10.3 *Javascript*

JavaScript merupakan bahasa pemrograman yang dirancang khusus untuk pengembangan *web*. Bahasa ini berjalan di sisi klien (*client-side*), dengan kemampuan interaktif yang unggul, *JavaScript* memungkinkan pembuatan aplikasi *web* yang *responsif*. Selain itu, *JavaScript* sering dimanfaatkan dalam pengembangan *web*, baik pada bagian *front-end* maupun *back-end* (Kynta et al., 2024).

A. Yani, B. Saputra, & R. T (2018) Menyatakan bahwa *JavaScript* adalah bahasa yang digunakan untuk membuat program yang digunakan agar dokument HTML yang ditampilkan pada sebuah Browser menjadi lebih interaktif, tidak sekedar indah saja. *JavaScript* memberikan beberapa fungsionalitas ke dalam halaman *web*, sehingga dapat menjadi sebuah program yang disajikan dengan menggunakan antar muka *web* (Risaldy & Hardinata, 2023).

2.10.4 *Structured Query Language (SQL)*

SQL adalah Bahasa standar yang digunakan untuk berkomunikasi dengan basis data relasional, dan juga merupakan Bahasa yang digunakan oleh banyak aplikasi atau tool berintegrasikan dengan server basis data (Sitorus & Sakban, 2021). SQL mempunyai kemampuan untuk mendefinisikan struktur data, modifikasi data dalam basis data dan menentukan konstrain sekuriti. SQL (*Structured Query Language*) merupakan bahasa yang digunakan untuk merepresentasikan kueri terhadap sistem manajemen basis data (DBMS) (Alghawazi et al., 2022).

2.10.5 *Hypertext Preprocessor (PHP)*

PHP (PHP: *Hypertext Preprocessor*) merupakan sebuah bahasa pemrograman yang berfungsi untuk mengonversi baris kode program menjadi kode mesin yang dapat dipahami oleh computer, khususnya dalam konteks *server-side*, dan dapat diintegrasikan ke dalam HTML (Adensa et al., 2023). PHP diciptakan untuk pemrograman bahasa C oleh Rasmus Lerdorf, PHP adalah bahasa yang mudah dibuat, notepad merupakan editor teks yang biasa digunakan, dan fungsi-fungsi yang telah ada didalam PHP tidak membedakan huruf besar dan huruf kecil (Arianto Pradana & Ibnu Hardi, 2021).

Model kerja PHP diawali dengan permintaan suatu halaman *web* oleh browser. Berdasarkan URL (*Uniform Resource Locator*) atau dikenal dengan sebutan alamat Internet, Browser mendapatkan alamat dari *web server*, mengidentifikasi halaman yang dikehendaki, dan menyampaikan segala informasi yang dibutuhkan oleh *web server*. Selanjutnya *Web Server* akan mencariakan berkas yang diminta dan isinya segera dikirim kemesin PHP dan mesin inilah yang memproses dan memberikan hasilnya (berupa kode HTML) ke *Web server*, selanjutnya web server menyampaikan ke klien (Sitorus & Sakban, 2021).

2.11 Alat Bantu Pemrograman

2.11.1 *Visual Studio Code*

Visual Studio Code merupakan aplikasi editor yang digunakan untuk kode dan teks sumber yang kuat tetapi ringan yang berjalan pada desktop serta tersedia untuk berbagai sistem operasi, antara lain Windows, macOS dan Linux. Muncul dengan dukungan bawaan untuk, Node.js, *JavaScript* dan *TypeScript* mempunyai

ekosistem ekstensi yang kaya untuk bahasa lain (seperti C#, C++, Python, Java, Go, PHP) dan *runtime* (seperti *Unity* dan *.NET*) (Sukiakhy et al., 2022).

2.11.2 *TablePlus*

TablePlus adalah aplikasi manajemen basis data modern yang mendukung berbagai sistem manajemen basis data seperti MySQL, PostgreSQL, SQLite, dan beberapa lainnya. Aplikasi ini dirancang dengan antarmuka pengguna yang sederhana dan responsif, sehingga mempermudah proses manajemen data. *TablePlus* memungkinkan pengembang untuk menjalankan kueri SQL, memvisualisasikan tabel, serta memodifikasi data dan struktur basisdata dengan cepat (Prastyia et al., 2024).

2.11.3 *MySQL*

MySQL adalah sistem manajemen basis data relasional yang bersifat *open-source* dan banyak digunakan dalam pengembangan aplikasi *web*. *MySQL* memungkinkan pengguna untuk menyimpan, mengatur, dan mengakses data dengan efisien menggunakan bahasa SQL (*Structured Query Language*). Dengan struktur data yang terorganisir dalam bentuk tabel, *MySQL* memberikan kemudahan dalam pengelolaan data terstruktur, yang sangat penting dalam berbagai aplikasi modern (Adwiyah & Saptari, 2024).

MySQL adalah sebuah perangkat lunak iasm manajemen basis data SQL (bahasa Inggris: *database management system*) atau DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. *MySQL* AB membuat *MySQL* tersedia sebagai perangkat lunak gratis dibawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk

kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL (Usman, 2023).

2.11.4 *Web Browser*

Web browser atau sering juga disebut *Internet Browser* berfungsi sebagai jembatan bagi pengguna computer dalam menjelajah dunia maya. Internet browser merupakan sebuah aplikasi atau software yang digunakan untuk mengolah data yang ditransfer dari *world wide web* (lebih dikenal dengan istilah www) ke computer dan menampilkannya secara visual agar mudah dimengerti oleh pengguna internet (Uminingsih et al., 2022).

2.12 *Web*

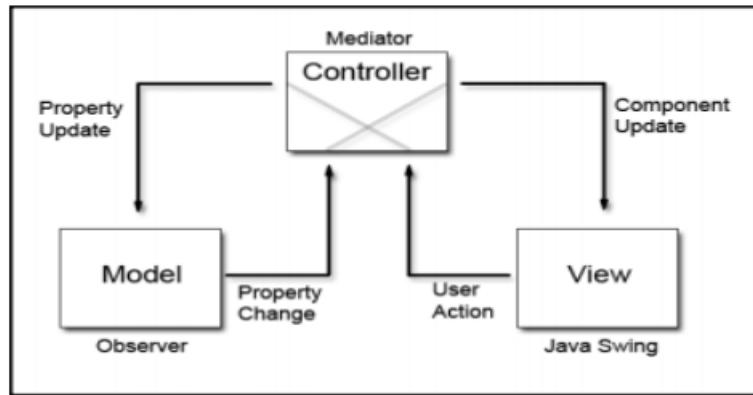
Website adalah media yang digunakan untuk menampung data teks, gambar, suara dan animasi yang dapat ditampilkan di internet dan dapat diakses oleh computer yang terhubung dengan internet secara global. *Website* merupakan media informasi berbasis jaringan computer yang dapat diakses dimana saja dengan biaya relatif murah. *Website* merupakan bentuk implementasi dari bahasa pemrograman *web* (Kumalasari et al., 2023).

Website merupakan kumpulan halaman yang menyajikan berbagai jenis informasi digital seperti teks, gambar, animasi, suara, dan video yang dapat diakses melalui internet serta dapat diakses dan oleh siapa saja di seluruh dunia. Halaman-halaman *website* tersebut dibuat menggunakan bahasa standar HTML yang diterjemahkan oleh *web browser* untuk ditampilkan dalam bentuk informasi yang dapat dibaca oleh semua orang (Manurung & Sembiring, 2024).

Website adalah kumpulan halaman dalam suatu domain yang memuat tentang berbagai informasi agar dapat dibaca dan dilihat oleh pengguna atau pemakai internet melalui sebuah mesin pencari atau search engine. Informasi yang dapat dimuat pada website biasanya berisi mengenai konten gambar, ilustrasi, video, dan teks untuk berbagai macam kepentingan. Umumnya untuk tampilan awal suatu website dapat diakses melalui halaman utama atau disebut juga dengan homepage dengan menggunakan browser dengan cara memasukkan dan menuliskan alamat URL secara lengkap dan tepat (Fitriani et al., 2022).

2.13 *Model View Controller* (MVC)

MVC merupakan sebuah konsep yang diperkenalkan oleh penemu *Smalltalk* (Trygve Reenskaug) untuk membuat satu jenis data jaringan menjadi jenis data lainnya bersama dengan pemrosesan (model), mengisolasi dari proses manipulasi (*controller*) dan tampilan (*view*) untuk direpresentasikan pada sebuah *user interface* (Wijaya et al., 2022). MVC merupakan teknik perancangan aplikasi yang memisahkan antara *layer model* (data), *layer view* (tampilan) dan *layer controller* (cara proses). MVC mengacu pada logika membagi kode ke dalam fungsi di kelas yang berbeda. Pemakaian kembali (*reusability*) kode menjadi keuntungan besar dalam pendekatan ini. Keterkaitan antara *layer model*, *layer view* dan *layer controller* sangat erat, karena itu masing-masing *layer* harus saling merujuk satu dengan lainnya, seperti pada gambar berikut (Atmaja et al., 2021).



Gambar 2. 1 Interaksi MVC

Sumber: (Atmaja et al., 2021)

Model akan berinteraksi langsung dengan *database* untuk melakukan pengelolaan data, melakukan validasi data yang diterima dari *controller*, namun tidak bisa secara langsung berinteraksi dengan *layer view*. *View* akan menerima dan menampilkan data sesuai aturan yang ditetapkan oleh *controller*, serta mengatur *logic* tampilan sebuah aplikasi. *Controller* akan bertugas mengelola interaksi model dengan *view*, serta menerima data maupun *request* dari *user* kemudian memutuskan apakah aplikasi akan memprosesnya atau menolaknya. Dengan konsep MVC, maka pengembangan aplikasi akan mudah dilakukan oleh setiap orang yang berbeda spesifikasi pekerjaan dalam sebuah tim. Basis data pada bagian model dapat dikerjakan oleh *database administrator*, tampilan antar muka dapat dikerjakan oleh *web designer*, dan proses bisnis aplikasi dapat dikerjakan oleh *web programmer* (Atmaja et al., 2021).

2.14 Framework Laravel

Laravel merupakan *framework web* berbasis PHP yang *open-source* dan tidak berbayar yang diciptakan oleh Taylor Otwell digunakan untuk mengembangkan aplikasi web dengan menggunakan pola MVC. Laravel merupakan sebuah MVC

(*ModelView-Controller*) *web development* yang berfungsi untuk meningkatkan kualitas dari perangkat lunak dengan mengurangi biaya pengembangan dan perbaikan serta meningkatkan produktifitas pekerjaan dengan sintak yang bersih dan fungsional (Wijaya et al., 2022).

Laravel merupakan salah satu *framework* berbasis PHP bersifat *open source* (terbuka), dan menggunakan konsep MVC (*model-view-controller*). Laravel berada di bawah lisensi MIT License dengan menggunakan Github sebagai tempat berbagi kode menjalankannya (Gibran et al., 2024). Laravel adalah framework PHP modern yang dirancang untuk mempermudah, mempercepat, dan membuat pengembangan PHP lebih intuitif. Laravel menerapkan pola arsitektur Model-View-Controller (MVC) dalam pengembangan aplikasi web. Untuk desain tampilan, Vue.js dipilih sebagai *framework frontend* karena kemampuannya dalam membangun antarmuka pengguna yang interaktif dan *responsive* (Richard & Santoso, 2025).

2.15 *Blackbox Testing*

Metode *blackbox testing* adalah metode pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak, tester dapat mendefinisikan umpan balik kondisi input dan melakukan pengetesan pada spesifikasi fungsional program. Proses *blackbox testing* dengan cara mencoba program yang telah dibuat dengan mencoba memasukkan data pada setiap formnya. Penggunaan metode ini dinilai sangat efektif dalam mengevaluasi perilaku perangkat lunak, terutama untuk menemukan kesalahan yang terdapat pada fungsi, antarmuka, model data, hingga akses terhadap data eksternal (Shadiq et al., 2021). Metode *blackbox testing* berfungsi untuk menguji aplikasi secara fungsional yang bertujuan untuk

mengetahui apakah *input*, *output*, dan fungsi-fungsi lainnya sesuai dengan spesifikasi yang dibutuhkan, tanpa memeriksa tampilan antarmuka dan kode program (Fatimah et al., 2022).

2.16 Penelitian Terdahulu

Tabel 2.6 Penelitian Terdahulu

No.	Penulis	Judul	Metode	Hasil
1	Ajisari & Prasetyaningrum 2024	Sistem Pakar Diagnosa Penyakit Kardiovaskular Menggunakan Metode <i>Certainty Factor</i>)	<i>Certainty Factor</i>	Berdasarkan hasil validasi pakar dengan data uji akurasi system menghasilkan tingkat kecocokan yang sesuai antara gejala dan penyakit dan didapat tingkat akurasi sebesar 100% dari 20 data periksa.
2	Efiyanti et al., 2024	Deteksi dini penyakit kekurangan gizi (Stunting) berbasis <i>web</i>	<i>Certainty Factor</i>	Menghasilkan akurasi yang cukup tinggi dalam mencerminkan

No.	Penulis	Judul	Metode	Hasil
		menggunakan metode <i>Certainty Factor</i>		preferensi pakar, yaitu 90,91%.
3	Kevin et al., 2022	Diagnosa penyakit jantung menggunakan metode <i>Certainty Factor</i>	<i>Certainty Factor</i>	Mampu memberikan hasil diagnosis secara mudah dan cepat berdasarkan pengetahuan pakar dan data gejala yang dipilih oleh pengguna. Diperoleh hasil 97% dari pengujian <i>blackbox testing</i> oleh admin atau pakar..
4	Patria et al., 2021	Sistem Pakar Menggunakan Metode <i>Certainty Factor</i> Untuk Mendiagnosa	<i>Certainty Factor</i>	Sistem berhasil mendiagnosis penyakit kulit kucing secara akurat. Contoh

No.	Penulis	Judul	Metode	Hasil
		Penyakit Kulit Pada Hewan Kucing		kasus Scabies menunjukkan nilai CF 98,2%, menunjukkan keakuratan yang tinggi berdasarkan gejala yang diinput <i>user</i>
5	Mulyani et al., 2023	Penerapan Metode <i>Certainty Factor</i> pada Sistem Pakar Diagnosis Penyakit Difteri berbasis <i>Web</i>	<i>Certainty Factor</i>	Hasil dari penelitian ini adalah aplikasi sistem pakar untuk diagnosis penyakit difteri dengan pengimplementasian metode <i>certainty factor</i> . Selain itu, berdasar pada uji akurasi sistem pakar, tingkat akurasi dari sistem pakar ini yaitu sebesar 90%

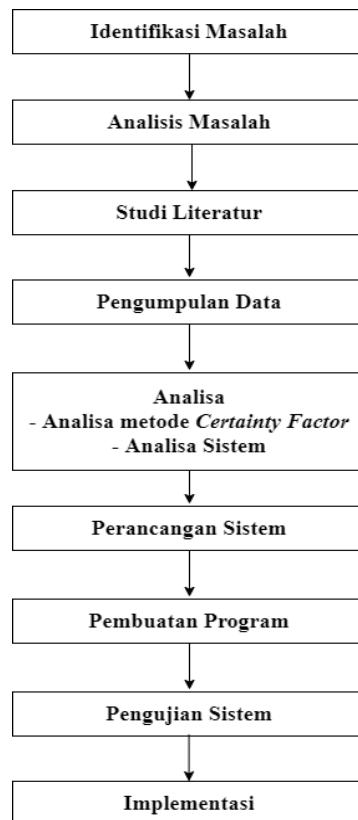
No.	Penulis	Judul	Metode	Hasil
				<p>berdasarkan akurasi yang telah dilakukan terhadap 10 data uji. Hasil tersebut menunjukkan bahwa keluaran dari sistem pakar memiliki hasil yang konsisten dengan diagnosis yang dilakukan oleh pakar.</p>

BAB III

METODOLOGI PENELITIAN

3.1 Kerangka Kerja Penelitian

Kerangka kerja penelitian merupakan gambaran alur sistematis dari tahapan-tahapan yang dilakukan dalam penelitian, mulai dari proses identifikasi masalah hingga implementasi sistem. Tahapan-tahapan penelitian tersebut digambarkan pada Gambar 3.1.



Gambar 3.1 Kerangka Kerja Penelitian

Berikut merupakan uraian dari tiap tahapan yang terdapat dalam kerangka kerja penelitian ini :

3.1.1 Identifikasi masalah

Setelah melakukan identifikasi masalah, masih banyak masyarakat yang minim pengetahuan mengenai jenis penyakit gizi buruk pada balita. Hal ini menyebabkan keterlambatan dalam penanganannya. Jika gizi buruk sudah ditingkat yang parah maka akan sulit dalam penyembuhannya dan beresiko kematian. Ketersediaan pakar/ahi gizi ada bidangnya pun menjadi salah satu faktor, karena dibutuhkan waktu yang lama dan biaya yang lumayan mahal untuk melakukan konsultasi. Masalah lain yang ditemukan yaitu belum tersedianya media berbasis teknologi yang dapat digunakan secara mandiri untuk mengenali gejala gizi buruk. Oleh karena itu, diperlukan sebuah sistem yang dapat membantu masyarakat dalam mendeteksi gizi buruk secara cepat, tepat, dan akurat berdasarkan gejala-gejala yang dialami oleh balita.

3.1.2 Analisis Masalah

Tahap selanjutnya yaitu analisis permasalahan yang akan diselesaikan melalui penelitian ini. Permasalahan yang diangkat adalah bagaimana membangun sistem pakar yang dapat mendiagnosis gizi buruk pada balita menggunakan metode *Certainty Factor*. Metode *Certainty Factor* memungkinkan sistem untuk mengolah tingkat keyakinan dari setiap gejala yang dimasukkan, berdasarkan pengetahuan pakar dan pilihan pengguna. Nilai-nilai tersebut kemudian dikombinasikan untuk menghasilkan tingkat kepastian terhadap diagnosis penyakit. Sistem pakar ini mampu memberikan hasil diagnosis dan solusi seperti pengetahuan pakar. Jadi, sistem ini dapat memudahkan masyarakat, khususnya orang tua balita dalam mendeteksi gejala gizi buruk pada balita.

3.1.3 Studi literatur

Tahapan ini dilakukan untuk mempelajari berbagai literatur yang berkaitan dengan masalah yang diteliti. Literatur yang dipilih bertujuan untuk memastikan bahwa penelitian yang dilakukan sesuai dan relevan dengan permasalahan yang sedang dibahas. Adapun sumber literatur diperoleh dari jurnal-jurnal ilmiah, buku, serta referensi terpercaya lainnya yang mendukung topik penelitian ini, khususnya tentang sistem pakar, metode *Certainty Factor*, dan gizi buruk pada balita.

3.1.4 Pengumpulan data

Pada tahap ini merupakan pengumpulan berbagai informasi yang diperlukan dalam penelitian. Pengumpulan data yang digunakan dalam penelitian ini adalah sebagai berikut:

1. Observasi, yaitu pengamatan yang dilakukan selama proses wawancara dengan ahli gizi. Peneliti mengamati cara ahli gizi menjelaskan kondisi balita dengan gizi buruk, termasuk ciri-ciri fisik, kebiasaan makan, dan indikator lainnya. Observasi ini bertujuan untuk memahami bagaimana pakar mengenali gejala-gejala gizi buruk pada balita.
2. Data, yaitu mengumpulkan informasi terkait gizi buruk, seperti jenis penyakit gizi buruk, gejala-gejala gizi buruk, dan solusi penanganan yang tepat, serta data pendukung lainnya.
3. Wawancara, peneliti melakukan wawancara dengan ahli gizi di Rumah Sakit Surya Insani, Ibu Dwi Astuti, AMd,Gz. Wawancara bertujuan untuk mendapatkan informasi yang relevan mengenai gizi buruk pada balita.

3.1.5 Analisa

1. Analisa Metode *Certainty Factor*

Data gejala dan penyakit yang telah dikumpulkan kemudian dianalisis menggunakan metode *Certainty Factor*. Tahap-tahap yang dilakukan adalah:

- a. Penyusunan data gejala dan jenis penyakit gizi buruk dalam bentuk *rule*.
- b. Pemberian nilai keyakinan pada setiap gejala.
- c. *Input* data gejala dari *user* berdasarkan nilai kepastian.
- d. Menghitung berdasarkan rumus *Certainty Factor*.
- e. Memperoleh hasil persentase jenis penyakit gizi buruk yang dialami pada balita.

2. Analisa Sistem

Analisa sistem dilakukan untuk mengetahui kebutuhan dan alur kerja sistem yang akan dibangun. Tahapan ini bertujuan agar sistem yang dikembangkan benar-benar sesuai dengan kebutuhan pengguna dan dapat menyelesaikan permasalahan yang ada. Dalam analisa ini, peneliti mengidentifikasi fitur utama yang dibutuhkan dalam sistem, seperti input gejala, proses diagnosa, pengelolaan data penyakit, pengelolaan data gejala, pengelolaan basis pengetahuan, serta laporan hasil diagnosa.

3.1.6 Perancangan sistem

Tahap perancangan sistem dilakukan setelah proses analisa data selesai dilaksanakan. Pada tahap ini, peneliti mulai menyusun rancangan sistem

berdasarkan kebutuhan dan hasil analisa yang telah diperoleh sebelumnya. Perancangan ini mencakup pembuatan model sistem menggunakan notasi UML (*Unified Modeling Language*), yang terdiri dari beberapa diagram penting yaitu *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, dan *Class Diagram*.

3.1.7 Pembuatan program

Setelah proses perancangan sistem selesai dilakukan, tahap selanjutnya adalah pembuatan program atau pengkodean. Pada tahap ini, peneliti mulai membangun sistem pakar sesuai dengan desain dan logika yang sudah dirancang sebelumnya. Pembuatan program dilakukan dengan menggunakan bahasa pemrograman berbasis *web* yaitu HTML, CSS, *JavaScript*, dan PHP, serta dibantu dengan database MySQL sebagai media penyimpanan data.

3.1.8 Pengujian Sistem

Tahap selanjutnya adalah pengujian sistem untuk memastikan semua fungsi berjalan dengan baik dan menghasilkan output yang sesuai. Pengujian dilakukan menggunakan metode *Blackbox Testing* untuk menguji fungsionalitas sistem, serta untuk mengetahui apakah *input*, *output*, dan fungsi-fungsi lainnya sesuai dengan spesifikasi yang dibutuhkan.

3.1.9 Implementasi

Implementasi sistem merupakan tahap uji coba sistem sehingga siap untuk dioperasikan. Tahap ini bertujuan untuk mengkonfirmasi modul-modul perancangan telah berjalan sesuai dengan yang diharapkan.