

BAB II

KAJIAN PUSTAKA

A. Kajian Teoritis

1. Rancang Bangun

Rancang bangun merupakan suatu proses yang meliputi perancangan (*design*) dan pembangunan (*development*) suatu sistem secara terstruktur untuk menghasilkan sistem sesuai dengan kebutuhan pengguna. Tahap perancangan berfokus pada penyusunan konsep, struktur, dan mekanisme kerja sistem berdasarkan hasil analisis kebutuhan. Tahap pembangunan dilakukan dengan mengimplementasikan rancangan tersebut menjadi sebuah sistem yang dapat dijalankan sesuai dengan fungsi yang telah ditetapkan. Hal ini sejalan dengan pendapat Maulana, (2023:22) yang menyatakan bahwa perancangan merupakan serangkaian prosedur yang dilakukan untuk menguraikan hasil analisis sebuah sistem ke dalam bentuk rancangan yang dapat diimplementasikan melalui bahasa pemrograman.

Menurut Ramelia et al., (2024:74) proses rancang bangun sistem berbasis *website* dilakukan secara sistematis melalui tahapan analisis kebutuhan, perancangan model sistem, hingga implementasi aplikasi agar sistem yang dikembangkan mampu memenuhi kebutuhan pengguna secara efektif. Proses tersebut menjadi dasar dalam pengembangan sistem berbasis *website*, karena setiap tahapan yang dilakukan secara sistematis dapat menghasilkan sistem yang mampu menjalankan fungsi sesuai dengan kebutuhan pengguna. Pendapat tersebut didukung oleh Surya & Kurniawan,

(2024:1249) yang menjelaskan bahwa konsep rancang bangun dapat diterapkan dalam pengembangan berbagai sistem informasi berbasis website melalui tahapan yang sistematis. Hal tersebut menunjukkan bahwa proses perancangan dan pembangunan merupakan bagian yang saling berkaitan dalam menghasilkan sistem yang efektif, mudah digunakan, serta mampu memenuhi kebutuhan pengguna pada berbagai bidang.

Berdasarkan uraian di atas, rancang bangun merupakan proses yang mengintegrasikan kegiatan perancangan dan pembangunan sistem secara sistematis untuk menghasilkan sistem yang sesuai dengan kebutuhan pengguna.

2. Sistem

Sistem merupakan sekumpulan elemen yang saling berkaitan, berinteraksi, dan membentuk suatu kesatuan untuk mencapai tujuan tertentu. Menurut Qomaruddin et al., (2025:52) sistem dibangun melalui keterpaduan berbagai komponen yang saling berhubungan dan terintegrasi untuk mencapai tujuan tertentu. Sejalan dengan pendapat tersebut, Kurniawati et al., (2024:1718) menjelaskan bahwa setiap komponen dalam sistem memiliki peran yang saling melengkapi sehingga seluruh proses dapat berlangsung secara terstruktur dan efektif.

Menurut Bratha, (2022:346) suatu sistem tersusun atas beberapa komponen yang saling berhubungan, meliputi masukan (*input*), proses (*process*), keluaran (*output*), dan pengendalian (*control*). Selain komponen tersebut, operasional sistem juga didukung oleh perangkat keras (*hardware*),

perangkat lunak (*software*), basis data (*database*), dan antarmuka pengguna (*user interface*) yang saling berinteraksi dalam menjalankan fungsi sistem. Keterkaitan antar komponen tersebut memungkinkan sistem bekerja secara terstruktur sehingga tujuan yang telah ditetapkan dapat tercapai.

Berdasarkan pendapat tersebut, sistem dapat disimpulkan sebagai kesatuan dari sejumlah komponen yang memiliki fungsi berbeda, tetapi bekerja dalam satu alur yang terkoordinasi. Data yang diterima sebagai masukan diolah melalui proses tertentu dengan dukungan komponen sistem, kemudian menghasilkan keluaran yang sesuai.

3. Sistem Pakar

Sistem pakar merupakan salah satu cabang kecerdasan buatan (*Artificial Intelligence*) yang dirancang untuk meniru cara berpikir seorang pakar dalam menyelesaikan suatu permasalahan pada bidang tertentu. Menurut Siregar & Hayadi, (2024:738) sistem pakar adalah sistem berbasis komputer yang memanfaatkan pengetahuan serta mekanisme penalaran untuk menghasilkan solusi atau rekomendasi yang menyerupai keputusan seorang ahli. Sistem pakar dikembangkan agar pengetahuan yang dimiliki pakar dapat diterapkan oleh pengguna melalui suatu sistem sehingga proses pengambilan keputusan dapat dilakukan secara lebih cepat dan konsisten.

Menurut Rusito & Putra, (2022:74) sistem pakar terdiri atas beberapa komponen utama, yaitu basis pengetahuan (*knowledge base*), mesin inferensi (*inference engine*), basis fakta (*working memory*), dan antarmuka pengguna (*user interface*). Basis pengetahuan menyimpan pengetahuan yang

diperoleh dari pakar, sedangkan mesin inferensi melakukan proses penalaran untuk menghasilkan suatu kesimpulan. Selain itu, basis fakta digunakan untuk menyimpan data dari pengguna, sementara antarmuka pengguna berfungsi sebagai media komunikasi antara pengguna dan sistem pakar.

Dengan demikian sistem pakar dapat disimpulkan sebagai sistem komputer yang mengolah data pengguna dengan menggunakan pengetahuan dan aturan dari seorang pakar. Proses pengolahan dilakukan melalui mesin inferensi hingga diperoleh kesimpulan berupa solusi atau rekomendasi yang dapat digunakan sebagai bahan pertimbangan dalam pengambilan keputusan.

4. Kebutuhan Kalori dan Menu Diet

Kebutuhan kalori merupakan jumlah energi yang diperlukan tubuh untuk menjalankan fungsi-fungsi vital serta mendukung aktivitas sehari-hari. Menurut Ronitawati et al., (2022:118) kebutuhan kalori setiap individu berbeda-beda dan dipengaruhi oleh berbagai faktor, seperti usia, jenis kelamin, berat badan, tinggi badan, serta tingkat aktivitas fisik. Menurut Khasanah et al., (2023:21) menyatakan bahwa kalori merupakan satuan energi yang diperoleh dari makanan dan minuman yang dikonsumsi, kemudian digunakan tubuh untuk menjalankan berbagai aktivitas, mulai dari bernapas, mencerna makanan, hingga melakukan aktivitas fisik.

Menu diet yang baik harus memperhatikan keseimbangan zat gizi makro, yaitu karbohidrat sebagai sumber energi utama, protein untuk pertumbuhan dan perbaikan jaringan tubuh, serta lemak sebagai sumber cadangan energi dan pelarut vitamin. Menurut Yuda et al., (2026:580)

penyusunan menu diet perlu mempertimbangkan kebutuhan kalori, indeks massa tubuh, dan tingkat aktivitas fisik agar rekomendasi yang diberikan sesuai dengan kondisi masing-masing individu.

Kebutuhan kalori menunjukkan jumlah energi yang diperlukan tubuh, sedangkan menu diet menjadi sarana untuk memenuhi kebutuhan tersebut melalui susunan makanan yang terencana. Oleh karena itu, penentuan menu perlu disesuaikan dengan karakteristik dan aktivitas setiap individu agar jumlah energi yang dikonsumsi tidak terlalu rendah maupun berlebihan.

5. Metode Harris-Benedict

Metode Harris-Benedict merupakan salah satu metode yang digunakan untuk memperkirakan kebutuhan kalori harian berdasarkan nilai *Basal Metabolic Rate* (BMR). Menurut Agustin et al., (2026:275) BMR merupakan jumlah energi minimum yang dibutuhkan tubuh untuk menjalankan fungsi-fungsi vital saat beristirahat, sedangkan perhitungannya didasarkan pada usia, jenis kelamin, berat badan, dan tinggi badan. Umar et al., (2023:150) menjelaskan bahwa metode Harris-Benedict digunakan untuk memperkirakan kebutuhan kalori tubuh dengan menyesuaikan nilai BMR terhadap tingkat aktivitas fisik setiap individu.

Nilai *Basal Metabolic Rate* (BMR) yang telah diperoleh selanjutnya disesuaikan dengan tingkat aktivitas fisik setiap individu untuk menghitung kebutuhan energi total harian atau *Total Daily Energy Expenditure* (TDEE), sehingga estimasi kebutuhan kalori harian menjadi lebih sesuai dengan

aktivitas yang dilakukan. Semakin tinggi tingkat aktivitas fisik, semakin besar kebutuhan energi yang diperlukan tubuh sehingga estimasi kebutuhan kalori harian juga akan meningkat Fahanani et al., (2022:103).

Secara umum, metode *Harris-Benedict* dapat disimpulkan sebagai metode perhitungan kebutuhan energi harian yang menggabungkan nilai BMR dan tingkat aktivitas fisik. BMR menggambarkan kebutuhan energi dasar tubuh, sedangkan faktor aktivitas digunakan untuk menyesuaikan perhitungan dengan kegiatan sehari-hari sehingga diperoleh estimasi TDEE setiap individu.

6. *Forward Chaining*

Menurut Sufi et al., (2023:10) *Forward Chaining* merupakan metode penalaran yang dimulai dari sekumpulan fakta yang telah diketahui, kemudian dilakukan proses pencocokan terhadap aturan-aturan yang terdapat dalam basis pengetahuan hingga diperoleh suatu kesimpulan. Sejalan dengan pendapat tersebut, Sajida et al., (2024:101) menjelaskan bahwa *Forward Chaining* bekerja dengan memulai proses inferensi dari fakta awal yang diberikan pengguna, kemudian bergerak secara bertahap melalui aturan-aturan yang sesuai hingga menghasilkan suatu keputusan atau tujuan tertentu. Maka dari itu, metode *Forward Chaining* banyak digunakan pada sistem pakar yang memerlukan proses pengambilan keputusan berdasarkan fakta atau data yang diberikan oleh pengguna.

Proses kerja *Forward Chaining* diawali dengan memasukkan fakta awal ke dalam *working memory*. Selanjutnya, sistem mencocokkan fakta

tersebut dengan bagian *IF* pada setiap aturan yang terdapat dalam basis pengetahuan. Apabila kondisi pada aturan terpenuhi, maka bagian *THEN* akan dijalankan sehingga menghasilkan fakta baru atau suatu kesimpulan. Proses inferensi tersebut berlangsung secara bertahap hingga tidak ditemukan lagi aturan yang dapat dijalankan atau tujuan yang diharapkan telah tercapai. Karakteristik tersebut menjadikan metode *Forward Chaining* mudah diterapkan, memiliki alur penalaran yang sistematis, serta sesuai digunakan dalam sistem pakar yang menghasilkan rekomendasi maupun keputusan berdasarkan data yang diberikan oleh pengguna (Fuad et al., 2022:730).

Berdasarkan uraian tersebut, *forward chaining* merupakan metode penalaran berbasis fakta yang bekerja dengan mencocokkan data pengguna terhadap aturan-aturan dalam basis pengetahuan. Proses pencocokan dilakukan melalui kondisi *IF-THEN* secara bertahap hingga diperoleh kesimpulan yang menjadi dasar pemberian keputusan atau rekomendasi.

7. Website

Menurut Bariah & Pradina, (2024:87) website merupakan fasilitas internet yang menghubungkan berbagai dokumen sehingga informasi dapat diakses oleh pengguna dari berbagai lokasi selama terhubung dengan jaringan internet. Hal ini sependapat dengan, Caniago et al., (2025:162) yang menyatakan bahwa website tidak hanya berfungsi sebagai media penyampaian informasi, tetapi juga memungkinkan terjadinya interaksi antara pengguna dan sistem secara interaktif.

Menurut Alfiana, (2025:14) Website dapat diklasifikasikan menjadi dua jenis, yaitu website statis dan website dinamis. Website statis menyajikan informasi yang bersifat tetap sehingga perubahan isi halaman hanya dapat dilakukan oleh pengembang. Sebaliknya, website dinamis memungkinkan pengguna berinteraksi dengan sistem melalui proses pengolahan data secara langsung sehingga informasi yang ditampilkan dapat berubah sesuai dengan masukan pengguna.

Secara umum, website berfungsi sebagai media untuk menyampaikan informasi sekaligus menyediakan sarana interaksi bagi pengguna. Situs statis lebih sesuai untuk informasi yang jarang mengalami perubahan, sedangkan situs dinamis digunakan ketika sistem perlu mengolah masukan dan menghasilkan informasi yang menyesuaikan kebutuhan pengguna.

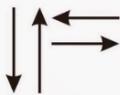
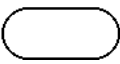
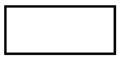
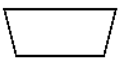
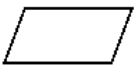
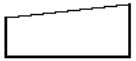
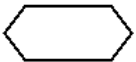
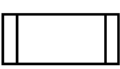
8. *Flowchart*

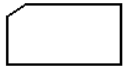
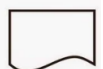
Menurut Listyoningrum et al., (2023:103) bahwa *Flowchart* adalah bagan yang menunjukkan alir di dalam program atau prosedur sistem secara logika, digunakan terutama sebagai alat bantu komunikasi dan untuk dokumentasi. *Flowchart* terdiri dari berbagai simbol standar yang masing-masing memiliki makna berbeda, seperti simbol proses, keputusan, terminal, dan konektor.

Menurut Cahyaningtyas et al., (2025:1537) menjelaskan *Flowchart* merupakan representasi grafis dari langkah-langkah yang harus dilakukan dalam menyelesaikan suatu masalah dan merupakan cara penyajian dari suatu

algoritma. *Flowchart* digunakan untuk menggambarkan alur kerja sistem pakar, mulai dari proses input data pengguna, perhitungan kebutuhan kalori, proses inferensi, hingga output berupa rekomendasi menu diet. Adapun simbol-simbol yang umum digunakan dalam flowchart disajikan pada Tabel 2.1

Tabel 2.1 Simbol simbol Flowchart

Simbol	Nama Simbol	Keterangan
	<i>Flow Direction Symbol</i>	Yaitu simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga connecting line.
	<i>Terminator Symbol</i>	Yaitu simbol untuk permulaan (start) atau akhir (stop) dari suatu kegiatan.
	<i>Connector Symbol (halaman sama)</i>	Yaitu simbol untuk keluar-masuk atau penyambungan proses dalam lembar/halaman yang sama.
	<i>Connector Symbol (beda halaman)</i>	Yaitu simbol untuk keluar-masuk atau penyambungan proses pada lembar/halaman yang berbeda.
	<i>Processing Symbol</i>	Simbol yang menunjukkan pengolahan yang dilakukan oleh komputer.
	<i>Simbol Manual Operation</i>	Simbol yang menunjukkan pengolahan yang tidak dilakukan oleh komputer.
	<i>Simbol Decision</i>	Simbol pemilihan proses berdasarkan kondisi yang ada.
	<i>Simbol Input-Output</i>	Simbol yang menyatakan proses input dan output tanpa tergantung dengan jenis peralatannya.
	<i>Simbol Manual Input</i>	Simbol untuk pemasukan data secara manual on-line keyboard.
	<i>Simbol Preparation</i>	Simbol yang mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan di dalam storage.
	<i>Simbol Predefined Process</i>	Simbol untuk pelaksanaan suatu bagian (sub-program)/prosedure.

	<i>Simbol Display</i>	Simbol yang menyatakan peralatan output yang digunakan yaitu layer, plotter, printer dan sebagainya.
	<i>Simbol Disk and On-line Storage</i>	Simbol yang menyatakan input yang berasal dari disk atau disimpan ke disk.
	<i>Simbol Magnetik Tape Unit</i>	Simbol yang menyatakan input berasal dari pita magnetik atau output disimpan ke pita magnetik.
	<i>Simbol Punch Card</i>	Simbol yang menyatakan bahwa input berasal dari kartu atau output ditulis ke kartu.
	Simbol Dokumen	Simbol yang menyatakan input berasal dari dokumen dalam bentuk kertas atau output dicetak ke kertas.

Sumber: (Rasiban et al., 2024:283)

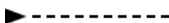
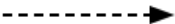
9. Unified Modeling Language (UML)

Unified Modeling Language (UML) merupakan bahasa pemodelan standar yang digunakan untuk menggambarkan, menjelaskan, dan membangun perangkat lunak secara visual. Menurut Hidayati et al., (2023:89) menjelaskan bahwa UML adalah bahasa standar yang digunakan sebagai metode dalam pengembangan sistem berbasis objek sekaligus berfungsi sebagai alat bantu dalam proses pengembangan sistem, di mana beberapa alat bantu yang digunakan dalam perancangan berbasis objek dengan menggunakan UML meliputi *use case diagram*, *activity diagram*, *sequence diagram*, dan *class diagram*. Penelitian ini memanfaatkan UML untuk memodelkan sistem secara terstruktur sehingga hubungan antar komponen sistem dan interaksi pengguna dapat digambarkan dengan lebih jelas dan mudah dipahami.

a. *Use Case Diagram*

Use case diagram merupakan salah satu diagram dalam UML yang digunakan untuk menggambarkan hubungan antara pengguna (aktor) dengan sistem. Menurut Aryani et al., (2025:1035) menjelaskan bahwa *use case diagram* bertujuan untuk mengilustrasikan bagaimana individu akan menggunakan dan memanfaatkan sistem, sementara aktor merupakan entitas yang berinteraksi dengan sistem. Penelitian ini, memanfaatkan *use case diagram* untuk menggambarkan interaksi antara pengguna dengan fitur-fitur yang tersedia pada sistem pakar rekomendasi resep diet berbasis website.

Tabel 2.2 Simbol-Simbol *Use Case Diagram*

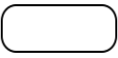
Gambar	Nama	Keterangan
	<i>Actor</i>	Orang atau sistem lain yang berinteraksi dengan sistem saat ini.
	<i>Use Case</i>	Berisi nama dari sistem yang diletakkan di dalam atau di bagian atas <i>boundary</i> .
	<i>System Boundary</i>	Menghubungkan aktor dengan <i>use case</i> .
	<i>Association Relationship</i>	Memasukkan satu <i>use case</i> dalam <i>use case</i> lainnya.
	<i>Include Relationship</i>	Memperluas <i>use case</i> untuk memasukkan perilaku opsional.
	<i>Extend Relationship</i>	Memperluas <i>use case</i> untuk memasukkan perilaku opsional.

Sumber: (Rasiban et al., 2024:284)

b. Activity Diagram

Activity diagram merupakan diagram UML yang digunakan untuk memvisualisasikan alur kerja (*workflow*) atau aktivitas dalam sebuah sistem. Menurut Nauli et al., (2022:1804) menyatakan bahwa *activity diagram* merupakan sebuah diagram yang memvisualisasikan alur kerja atau aktivitas dalam sebuah sistem, proses bisnis, maupun menu yang terdapat di dalam perangkat lunak. Penelitian ini memanfaatkan *activity diagram* digunakan untuk menggambarkan alur proses yang terjadi pada setiap fitur sistem, mulai dari proses input data pengguna, perhitungan kalori, hingga penampilan rekomendasi resep diet. Adapun simbol-simbol *Activity Diagram* disajikan dalam tabel 2.3.

Tabel 2.3 Simbol-Simbol *Activity Diagram*

Gambar	Nama	Keterangan
	<i>Initial State</i>	Digunakan untuk memulai aktifitas yang dilakukan dalam <i>activity diagram</i>
	<i>Final State</i>	Digunakan untuk mengakhiri aktifitas yang dilakukan dalam <i>activity diagram</i>
	<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
	<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktifitas yang terjadi.
	<i>Control Flow</i>	Digunakan untuk menghubungkan antara satu aktifitas dengan aktifitas lainnya.

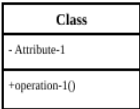
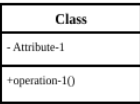
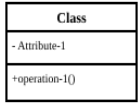
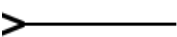
Sumber: (Rasiban et al., 2024:284)

c. Class Diagram

Class diagram merupakan diagram struktural statis dalam UML yang digunakan untuk menggambarkan struktur sistem mencakup kelas-

kelas sistem, atribut, metode, dan hubungan antar objek. Menurut Nauli et al., (2022:1804) menjelaskan bahwa class diagram adalah suatu jenis diagram struktural statis dalam UML serta umum digunakan untuk menggambarkan bagaimana struktur sistem mencakup kelas-kelas sistem, atribut, metode, dan hubungan antar objek. Penelitian ini memanfaatkan *class diagram* untuk mendefinisikan struktur basis data dan hubungan antar kelas yang menjadi dasar dari sistem yang dikembangkan. Adapun simbol-simbol *Class Diagram* disajikan dalam tabel 2.4.

Tabel 2.4 Simbol-Simbol Class Diagram

Gambar	Nama	Keterangan
	<i>Class</i>	Mewakili orang, tempat, atau hal-hal yang dibutuhkan sistem untuk menangkap dan menyimpan informasi
	<i>Attribute</i>	Merupakan properti yang menggambarkan keadaan suatu objek
	<i>Operation</i>	Mewakili tindakan atau fungsi yang dapat dilakukan oleh kelas
	<i>Generalization</i>	Mewakili hubungan sejenis di antara beberapa kelas
	<i>Association</i>	Mewakili hubungan antara beberapa kelas atau kelas dan dirinya sendiri

Sumber: (Rasiban et al., 2024:284)

d. Sequence Diagram

Sequence diagram merupakan diagram dalam UML yang digunakan untuk menggambarkan urutan interaksi antar objek dalam sistem secara kronologis berdasarkan waktu. Wayahdi & Ruziq,

(2023:1516) menjelaskan bahwa pemodelan UML menggunakan sequence diagram dapat membantu menggambarkan urutan dalam membuat aplikasi sehingga alur komunikasi antar komponen sistem menjadi lebih terstruktur dan mudah dipahami. Penelitian ini memanfaatkan *sequence diagram* untuk menggambarkan urutan proses komunikasi antara pengguna, antarmuka sistem, dan basis data pada setiap fungsionalitas utama sistem. Adapun simbol-simbol *Sequence Diagram* disajikan dalam tabel 2.5.

Tabel 2.5 Simbol-Simbol *Sequence Diagram*

Gambar	Nama	Keterangan
	<i>Actor</i>	Orang atau sistem yang memperoleh manfaat dari dan berada di luar sistem
	<i>Boundary Class</i>	Menggambarkan tampilan program
	<i>Control Class</i>	Menggambarkan <i>controller</i>
	<i>Entity Class</i>	Menggambarkan class
	<i>Lifeline</i>	Menunjukkan kehidupan suatu objek selama suatu urutan
	<i>Execution Occurrence</i>	Merupakan persegi panjang sempit panjang yang ditempatkan di atas lifeline
	<i>Message</i>	Menyampaikan informasi dari satu objek ke objek lainnya
	<i>Create Message</i>	Pesan yang digunakan untuk membuat objek baru dalam sistem (panah putus-putus).
	<i>Return Message</i>	Pesan balasan yang dikirimkan kembali ke objek pengirim (panah putus-putus).

	<i>Self Message</i>	Pesan yang dikirimkan oleh suatu objek kepada dirinya sendiri.
	<i>Guard Condition</i>	Merupakan tes yang harus dipenuhi untuk pesan yang akan dikirim
	<i>Frame</i>	Menunjukkan konteks sequence diagram

Sumber: (Rasiban et al., 2024:284)

10. Metode Pengembangan Sistem

a. Model ADDIE (*Analysis, Design, Development, Implementation, dan Evaluation*)

Model ADDIE merupakan salah satu model pengembangan yang digunakan sebagai pedoman dalam merancang dan mengembangkan suatu produk atau sistem secara sistematis. Menurut Astuti et al., (2025:1921) menjelaskan bahwa model ADDIE terdiri atas lima tahapan utama, yaitu *Analysis, Design, Development, Implementation, dan Evaluation*, yang saling berkaitan dalam membentuk suatu proses pengembangan yang terarah. Selain itu, Ariel & Yusuf, (2025:2646) menyatakan bahwa ADDIE bukan sekadar urutan tahapan, melainkan kerangka konseptual yang memberikan pedoman bagi pengembang dalam menghasilkan produk yang sesuai dengan tujuan pengembangan.

Menurut Istiningsih et al., (2021:913) tahap *Analysis* bertujuan mengidentifikasi kebutuhan, karakteristik pengguna, serta permasalahan yang menjadi dasar pengembangan. Tahap *Design* berfokus pada penyusunan rancangan produk atau sistem, kemudian dilanjutkan dengan tahap *Development* sebagai proses mewujudkan rancangan tersebut

menjadi produk yang siap digunakan. Selanjutnya, tahap *Implementation* dilakukan untuk menerapkan produk kepada pengguna, sedangkan tahap *Evaluation* bertujuan menilai hasil pengembangan guna mengetahui tingkat kesesuaian produk serta menjadi dasar dalam melakukan penyempurnaan apabila masih ditemukan kekurangan.

Berdasarkan uraian tersebut, model ADDIE merupakan model pengembangan yang menekankan proses yang sistematis melalui lima tahapan yang saling berkesinambungan, yaitu *Analysis*, *Design*, *Development*, *Implementation*, dan *Evaluation*. Setiap tahapan memiliki peran yang berbeda dalam mendukung proses pengembangan, sehingga model ADDIE banyak digunakan sebagai landasan teoretis dalam pengembangan berbagai produk maupun sistem yang memerlukan proses perancangan secara terstruktur.

11. Pengujian Sistem

a. Black Box Testing

Pengujian sistem merupakan tahap krusial dalam siklus pengembangan perangkat lunak untuk memastikan sistem berjalan sesuai dengan spesifikasi yang telah ditentukan. Pengujian adalah sebuah proses yang diberikan sebagai siklus hidup perangkat lunak dan merupakan proses standar yang digunakan oleh organisasi perangkat lunak (Nur Syihab et al., 2024:112).

Dalam penelitian ini digunakan metode *black box testing* yang berfokus pada pengujian fungsi sistem dari sudut pandang pengguna.

Black box testing adalah pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak tanpa mengetahui yang terjadi dalam proses detail, melainkan hanya mengetahui input dan output (Pratama et al., 2023:561). Pengujian black box testing dilakukan dengan memberikan berbagai nilai masukan pada setiap fungsi sistem dan kemudian memverifikasi apakah keluaran yang dihasilkan sesuai dengan yang diharapkan.

12. Evaluasi Sistem

a. *System Usability Scale (SUS)*

Evaluasi sistem dilakukan untuk mengetahui tingkat kemudahan penggunaan sistem yang dikembangkan oleh pengguna. Salah satu metode yang banyak digunakan untuk mengukur usability sistem adalah *System Usability Scale (SUS)*. SUS adalah kuesioner sederhana yang digunakan untuk mengukur persepsi pengguna terhadap tingkat usability suatu sistem melalui sepuluh butir pertanyaan. Metode ini banyak digunakan karena mudah diterapkan, cepat dalam proses pengisian, serta mampu memberikan gambaran mengenai tingkat penerimaan pengguna terhadap suatu sistem (Subiksa et al., 2025:677). *System Usability Scale (SUS)* telah banyak digunakan dalam penelitian pengembangan sistem informasi karena dinilai efektif dalam mengukur aspek kemudahan penggunaan.

B. Kajian Empiris

Kajian empiris merupakan kajian terhadap penelitian terdahulu yang relevan dengan penelitian yang dilakukan. Penelitian yang dilakukan oleh Hafidz & Widayat, (2021:45) berjudul "Rancang Bangun Sistem Pakar Diagnosis Stunting Menggunakan Metode *Forward Chaining* Berbasis Website" bertujuan mengembangkan sistem pakar berbasis website untuk membantu proses diagnosis dini stunting pada anak menggunakan metode *Forward Chaining*. Hasil penelitian menunjukkan bahwa sistem yang dikembangkan mampu melakukan proses diagnosis berdasarkan aturan yang terdapat pada basis pengetahuan serta telah dinyatakan layak digunakan berdasarkan hasil validasi pakar dan pengujian *System Usability Scale* (SUS). Persamaan penelitian tersebut dengan penelitian ini terletak pada penggunaan metode *Forward Chaining* sebagai mekanisme inferensi dalam sistem pakar berbasis website. Adapun perbedaannya, penelitian tersebut berfokus pada diagnosis stunting berdasarkan data antropometri anak, sedangkan penelitian ini bertujuan memberikan rekomendasi menu diet berdasarkan hasil perhitungan kebutuhan kalori harian pengguna.

Penelitian lain dilakukan oleh Sufi et al., (2023:8) dengan judul "Sistem Pakar Rekomendasi Menu Makanan Sehat Penderita Penyakit dengan Metode *Forward Chaining*." Penelitian tersebut menghasilkan sistem yang mampu memberikan rekomendasi menu makanan berdasarkan kondisi kesehatan pengguna melalui penerapan aturan-aturan dalam basis pengetahuan. Hasil penelitian menunjukkan bahwa metode *Forward Chaining* dapat digunakan

untuk menghasilkan rekomendasi makanan yang sesuai dengan kebutuhan pengguna. Persamaan penelitian tersebut dengan penelitian ini adalah sama-sama mengembangkan sistem pakar berbasis web yang memberikan rekomendasi makanan, sedangkan perbedaannya terletak pada fokus penelitian yang dilakukan, yaitu rekomendasi menu diet berdasarkan kebutuhan kalori harian pengguna.

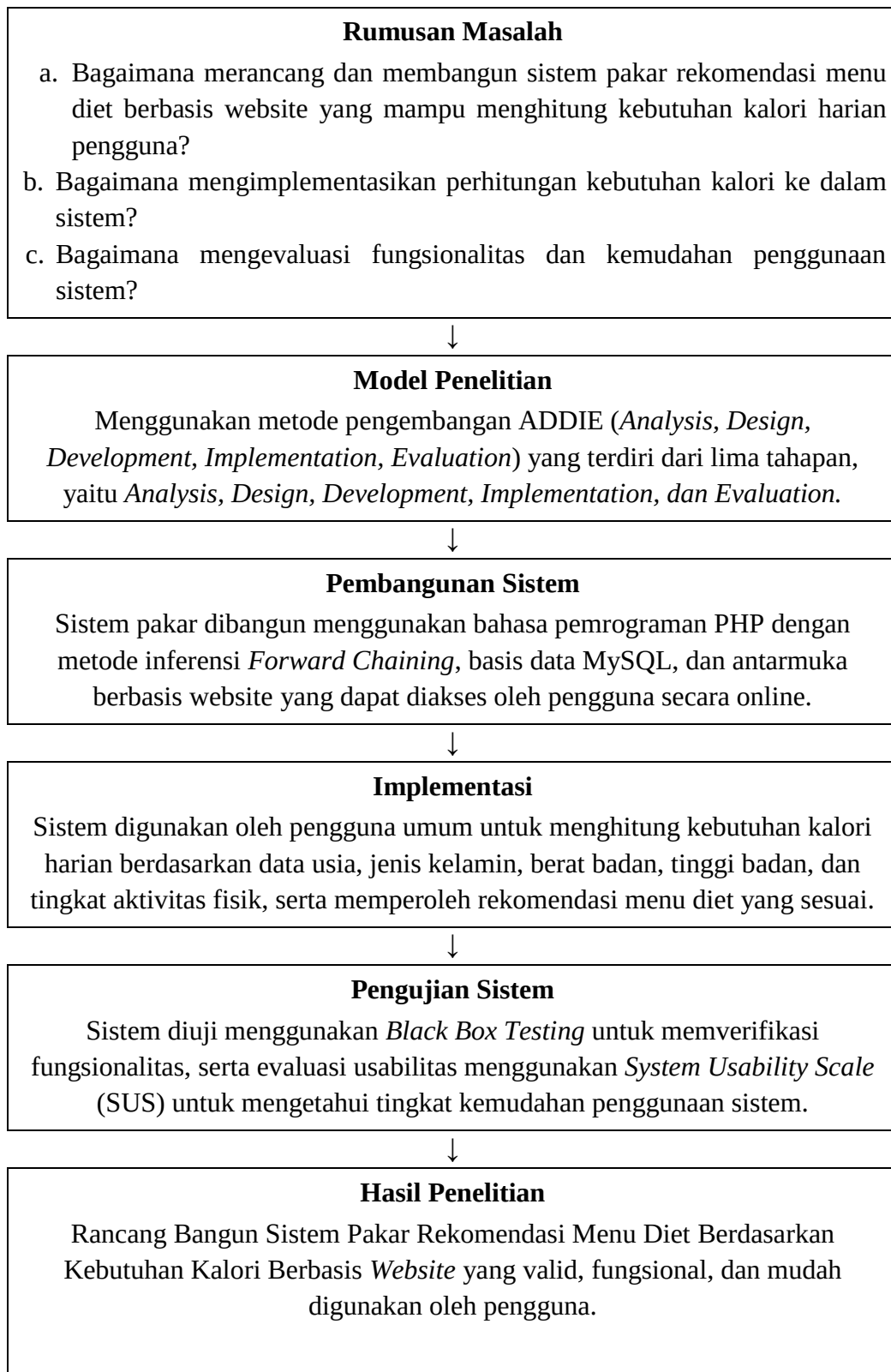
Penelitian yang dilakukan oleh Hasugian et al., (2025:29) berjudul "Sistem Pakar Menentukan Menu Makanan Sehat Berdasarkan Penyakit Menggunakan Metode *Certainty Factor* dan *Forward Chaining* Berbasis Website" mengembangkan sistem pakar berbasis website yang bertujuan membantu pengguna memperoleh rekomendasi menu makanan sehat sesuai dengan kondisi penyakit yang diderita. Sistem tersebut menerapkan metode *Forward Chaining* sebagai mekanisme inferensi untuk mencocokkan fakta dengan aturan pada basis pengetahuan, sedangkan metode *Certainty Factor* digunakan untuk menghitung tingkat keyakinan terhadap hasil rekomendasi yang diberikan. Hasil penelitian menunjukkan bahwa sistem mampu menghasilkan rekomendasi menu makanan yang sesuai dengan kondisi kesehatan pengguna dan dapat menjadi media pendukung dalam menentukan pola makan yang lebih sehat. Persamaan penelitian tersebut dengan penelitian ini terletak pada penggunaan metode *Forward Chaining* dalam sistem pakar berbasis website untuk menghasilkan rekomendasi berdasarkan aturan yang telah ditetapkan. Adapun perbedaannya, penelitian tersebut memberikan rekomendasi menu makanan berdasarkan jenis penyakit yang dialami pengguna,

sedangkan penelitian ini berfokus pada pemberian rekomendasi menu diet berdasarkan kebutuhan kalori harian pengguna.

C. Kerangka Berpikir

Kerangka berpikir merupakan gambaran konseptual yang menjelaskan hubungan antara permasalahan yang diteliti dengan solusi yang ditawarkan berdasarkan kajian teoritis dan kajian empiris. Penyusunan kerangka berpikir bertujuan untuk memberikan arah yang jelas dalam pelaksanaan penelitian sehingga proses penelitian dapat berjalan secara sistematis sesuai dengan tujuan yang telah ditetapkan.

Permasalahan yang melatarbelakangi penelitian ini adalah masih banyak masyarakat yang mengalami kesulitan dalam menentukan jenis dan jumlah makanan yang sesuai dengan kebutuhan kalori harian. Penelitian ini menawarkan solusi berupa pembangunan sistem pakar rekomendasi menu diet berbasis *website*. Pengguna menginputkan data berupa usia, jenis kelamin, berat badan, tinggi badan, dan tingkat aktivitas fisik untuk menghitung kebutuhan kalori harian. Hasil perhitungan digunakan sebagai dasar pemberian rekomendasi menu diet. Sistem dikembangkan menggunakan bahasa pemrograman PHP berbasis *website* dengan metode ADDIE. Selanjutnya, sistem diuji menggunakan *Black Box Testing* dan dievaluasi menggunakan *SUS*.



Gambar 2.1 Kerangka Berfikir