

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Ada beberapa penelitian terkait dengan Sistem Pakar Mendeteksi Kerusakan Mesin Pada Sepeda Motor Menggunakan Metode Teorema Bayes di antaranya:

Penelitian yang dilakukan oleh Destiansyah Putra Tarigan, Puji Sari Rahmadan dan Suardi Yakub, 2022, dengan judul “Penerapan *Teorema Bayes* Untuk Mendeteksi Kerusakan Mesin Sepeda Motor”, dengan tujuan untuk membantu pengguna mengetahui kerusakan yang terjadi pada Yamaha Vixion. Sebab sistem pakar merupakan sebuah sistem yang mampu mengidentifikasi sebuah sistem dengan menggunakan algoritma tertentu. Metode penelitian yang peneliti gunakan dapat memberikan gambaran rancangan penelitian meliputi: prosedur dan langkah-langkah yang harus ditempuh, waktu penelitian, sumber data dan juga dengan langkah apa data tersebut diperoleh kemudian data tersebut diolah dan dianalisis.[1]

Kemudian penelitian selanjutnya dilakukan oleh Sheryl Febry Irianti Simanjuntak, 2021, dengan judul Skripsi “Analisa *Data Mining* Menggunakan Metode *Bayes* Untuk Mengukur Tingkat Kerusakan Mesin Motor” yang bertujuan untuk menentukan oleh pihak bengkel tentang mesin motor yang rusak ringan maupun rusak berat. [2]

Penelitian yang dilakukan oleh Alfian Nazala Putra, Nurul Hidayat dan Suprpto, 2018, dengan sebuah judul “Sistem Deteksi Kerusakan Mesin Pada

Sepeda Motor Menggunakan *Naïve Bayes-Certainty Factor*”, yang bertujuan untuk mengetahui jenis kerusakan apa yang muncul dengan membutuhkan perangkat lunak yang mampu untuk membantu pengendara sepeda motor. Metode *Bayes* merupakan suatu metode klasifikasi probabilitas sederhana yang berdasarkan pada *teorema bayes* namun inferensinya menggunakan asumsi independensi fitur yang kuat.[3]

Penelitian selanjutnya yang dilakukan oleh Surya Darma Pranata, Milfa Yetri, M.Saifuddin, 2020, dengan judul penelitian “Sistem Pakar Untuk Mengidentifikasi Kerusakan Sepeda Motor PCX 150 Dengan Menggunakan Metode *Teorema Bayes*”, dengan tujuan penelitian untuk mendeteksi kerusakan-kerusakan pada sepeda motor PCX 150 matic injeksi berdasarkan gejala-gejala yang ada secara cepat dan juga akurat dengan menggunakan bahasa pemrograman visual basic 2008 dan ms. *Acces* sebagai database-nya.[4]

Kemudian penelitian yang dilakukan Saiful Nurarif, Iskandar Zulkarnain, Hendra Jaya dan Sri Kusnasari, 2021, dengan sebuah judul “Sistem Pakar Mendiagnosa Kerusakan Pada Mesin Motor Suzuki Satria FU Menggunakan *Metode Bayes*”, maka adapun tujuan dilakukannya penelitian ini adalah dengan terciptanya sebuah aplikasi sistem pakar dengan metode *teorema bayes* yang dapat membantu pemilik satria FU dalam mengetahui kerusakan sistem pakar ini yang sudah banyak digunakan untuk kepentingan komersial karena sistem pakar dapat dilihat sebagai cara penyimpanan pengetahuan pakar dalam bidang tertentu dalam program komputer yang sudah dirancang sedemikian rupa dengan sangat cerdas.[5]

Selanjutnya adalah penelitian yang dilakukan oleh Foni Agus Setiawan, Dewi Primasari dan Wahyudin, 2020, dengan judul “Sistem Pakar Identifikasi Kerusakan Sepeda Motor Suzuki Satria F150 Dengan *Teorema Bayes*”, yang bertujuan untuk membangun sebuah sistem pakar diagnosis atau identifikasi kerusakan pada sepeda motor Suzuki Satria F150 dan juga menawarkan solusi perbaikannya. Di mana sistem pakar dibangun dalam bentuk sebuah aplikasi berbasis web.[6]

2.2 Landasan Teori

2.2.1 Sistem

Sistem merupakan kumpulan orang yang saling bekerja sama dengan berbagai macam ketentuan serta aturan yang sistematis dan secara terstruktur agar terarah untuk membentuk satu kesatuan yang melakukan suatu fungsi untuk mencapai tujuan tersebut. Sistem ini memiliki beberapa sifat atau karakteristik yang terdiri dari komponen sistem, lingkungan luar sistem, masukan sistem, batasan sistem, keluaran sistem, pengolahan sistem dan juga sasaran sistem.[7]

2.2.2 Pakar

Pakar merupakan seorang individu yang memiliki pengetahuan khusus, keahlian, pengalaman, pemahaman dan juga metode-metode yang digunakan untuk memecahkan setiap permasalahan pada bidang tertentu. Pakar juga memiliki kemampuan untuk mengaplikasikan pengetahuannya dan juga memberikan saran-saran serta memecahkan masalah pada domain tertentu. Biasanya pakar memiliki beberapa konsep umum seperti: 1) harus mampu

memecahkan setiap persoalan yang ada dan mencapai tingkat performa yang secara signifikan jauh lebih baik dari kebanyakan orang, 2) pakar adalah relatif.[8]

2.2.3 Sistem Pakar

Sistem pakar (*expert system*) aplikasi berbasis komputer yang banyak dipergunakan dalam penyelesaian permasalahan yang berkaitan dengan pemikiran ataupun keahlian seorang pakar disebut dengan Sistem pakar. Sistem ini mencoba membantu dalam memecahkan sebuah masalah yang tidak dapat diselesaikan orang awam dan hanya bisa diselesaikan oleh seorang pakar di bidangnya. Sistem pakar dikatakan berhasil jika sistem ini mampu menghasilkan sebuah keputusan yang sama seperti yang dilakukan oleh pakar aslinya baik pada saat proses pengambilan keputusannya begitu juga dengan hasil keputusannya.

Sistem pakar hadir menjadi pembantu atau asisten yang akan menuntun seseorang menyelesaikan permasalahan dengan dukungan data kepakaran yang disimpan dalam komputer. Melalui bantuan kepakaran, informasi dirangkum dalam database sebagai sumber penanganan diagnosis kerusakan sampai solusi yang akan dilakukan sebagai langkah penyelesaian permasalahan. *Knowledge Based System* adalah suatu sistem yang menggunakan pengetahuan (*knowledge*) yang diubah kedalam bahasa mesin atau dikodekan untuk dapat melakukan suatu tugas dan menyimpulkannya. *Knowledge Based System* atau Sistem Berbasis Pengetahuan digunakan agar dapat membantu manusia dalam menyelesaikan suatu masalah yang sedang dihadapinya dengan berdasarkan pada pengetahuan yang telah diprogramkan ke dalam sistem. Oleh karena itu digunakan *Knowledge*

Based System dalam memecahkan suatu masalah yang berhubungan dengan *Expert System*. [9]

A. Kelebihan sistem pakar

Adapun beberapa keunggulan yang dimiliki oleh sistem pakar sehingga membuatnya berguna untuk memecahkan masalah yang ada di dalam domain tertentu, seperti berikut:

- a. Dapat membantu para ahli dan juga non para ahli dalam bidang tertentu untuk memberikan solusi layaknya seorang ahli
- b. Memberikan kemudahan untuk mengakses pengetahuan seorang pakar
- c. Sistem pakar diyakini bekerja lebih cepat dibandingkan manusia dengan demikian memberikan layanan yang produktivitas
- d. Dapat dioperasikan pada situasi atau lingkungan yang berbahaya
- e. Memberikan kualitas dengan menawarkan penanganan yang terbarukan dan konsisten
- f. Integrasi sistem pakar dan computer membuatnya lebih efektif dan mencakup lebih banyak aplikasi
- g. Sistem pakar dapat dipercaya karena tidak akan pernah lelah, sakit atau bosan membantu orang awam memahami tentang keahlian di bidang tertentu
- h. Memiliki kemampuan untuk menyelesaikan masalah karena sistem pakar menangkap atau menampung sumber pengetahuan dari banyak pakar

- i. Sistem pakar dapat menjadi media pelengkap dalam pelatihan yang tentu dapat memberikan kepada pakar pada bidang itu sendiri dan dapat menampung pengalaman dari pakar sehingga fasilitas sistem pakar ini dapat pula berfungsi sebagai guru.
- j. Sistem pakar ini dapat menangani informasi yang tidak lengkap atau tidak pasti, dengan terus memberikan jawaban dengan mempertimbangkan jawaban yang diberikan oleh pengguna
- k. Memiliki kemampuan untuk menggabungkan dan menyesuaikan keahlian seseorang.

B. Kekurangan Sistem Pakar

Selain mempunyai kelebihan, sistem pakar ini pun memiliki beberapa kekurangan diantaranya:

- a. Terdapat kemungkinan sistem pakar ini tidak dapat memberikan sebuah keputusan
- b. Sistem pakar tidak memberikan hasil 100% bernilai benar
- c. Sulit untuk dikembangkan sebab terdapat keterbatasan pembiayaan yang cukup besar untuk membuat dan pemeliharannya, dalam hal keahlian dan ketersediaan seorang pakar. [10]

2.2.4 Sepeda Motor

A. Sepeda Motor

Sepeda motor adalah alat transportasi yang minimalis karena sangat memudahkan seseorang yang ingin bepergian sebab memiliki fungsi yg lebih handal dan bebas hambatan daripada kendaraan roda empat, yang akhir-akhir ini

lebih diminati oleh pengguna jalan raya khususnya di kota-kota besar yang intensitas kemacetannya tinggi, sehingga suka tidak suka bagi para teknisi kendaraan roda dua ini dituntut untuk bekerja lebih cepat serta akurat.



Gambar 2. 1 Motor Honda Matic[11]

2.2.5 Metode Teorema Bayes

Metode *Teorema Bayes* adalah cara untuk mengetahui probabilitas bersyarat dari suatu peristiwa yang terjadi, mengingat bahwa itu memiliki beberapa hubungan dengan satu atau lebih peristiwa lainnya. Probabilitas *bayes* merupakan salah satu cara dalam mengatasi suatu ketidakpastian data dengan menggunakan formula *Bayes*.

Metode *bayes* juga memandang sebuah tolak ukur sebagai variabel yang menggambarkan sebuah pengetahuan awal tentang tolak ukur sebelum pengamatan dilakukan dan dinyatakan dalam suatu nilai yang disebut dengan distribusi prior. Kemudian setelah pengamatan dilakukan, informasi dalam distribusi prior kembali digabungkan dengan data sampel melalui *Teorema Bayes*.

Adapun algoritma dari penyelesaian dari metode *Teorema Bayes* yaitu sebagai berikut. [9]

Rumus

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)}$$

Persamaan 2.1 Rumus Metode *Teorema Bayes*

Dimana:

X : Data dengan class yang belum diketahui

H : Hipotesis data X

$P(H|X)$: Probabilitas hipotesis H berdasar kondisi X

$P(H)$: Probabilitas hipotesis H

$P(X|H)$: Probabilitas X berdasarkan kondisi pada hipotesis H

$P(X)$: Probabilitas dari X

A. Kelebihan *teorema bayes* yaitu:

- a) Hanya memerlukan pengkodean yang sederhana
- b) Sangat mudah untuk dipahami
- c) Lebih cepat dalam perhitungan

B. Kekurangan *teorema bayes* yaitu:

- a) Pada teori ini, satu probabilitas saja tidak bisa untuk mengukur seberapa dalam tingkat keakuratannya (kurangnya bukti untuk membuktikan kebenaran yang dihasilkan dari teori tersebut).

Metode *bayes* yang digunakan yaitu *Teorema bayes* yang merupakan salah satu metode yang digunakan untuk menghitung ketidakpastian data menjadi data yang pasti dengan membandingkan antara data “ya” atau “tidak”. [12]

Dalam membuat *web* sistem pakar deteksi kerusakan mesin sepeda motor, penulis menentukan beberapa tabel, di antaranya:

2.2.6 Pengembangan Sistem

A. Metode *Waterfall*

Metode *waterfall* adalah model pengembangan sistem informasi yang sistematis dan juga sekuensial. Metode *waterfall* merupakan model pengembangan sistem informasi yang sistematis dan sekuensial metode *waterfall* ada 5, diantaranya:

1. *Requirements*

Pada tahap ini, yang pertama dilakukan yakni mempersiapkan dan menganalisa kebutuhan dari perangkat lunak (*software*) yang akan dikerjakan. Informasinya dapat diperoleh melalui survei, wawancara, studi literatur, observasi hingga diskusi.

2. *Design*

Selanjutnya adalah tahap pembuatan desain dari aplikasi yang akan dikerjakan sebelum proses *coding*. Tahapan ini bertujuan agar memberikan gambaran yang jelas mengenai struktur data ataupun arsitektur *software*, fungsi internal dan eksternal dari algoritma yang diperlukan hingga tampilan dari *software*.

3. *Implementation*

Pada tahap ini desain dari *software* yang diinginkan diimplementasikan ke dalam kode program dengan menggunakan berbagai *tools* dan bahasa pemrograman yang diinginkan.

4. *Integration & Testing*

Kemudian tahapan selanjutnya adalah *integration & testing*. Pada tahap ini dilakukan proses integrasi dan pengujian dari sistem yang telah dibuat. Yang bertanggung jawab untuk melakukan proses *testing* biasanya adalah QA (*Quality Assurance*) dan QC (*Quality Control*). Mereka mengecek apakah *software* sudah sesuai dengan desain tadi serta apakah terdapat *error* atau *bug*.

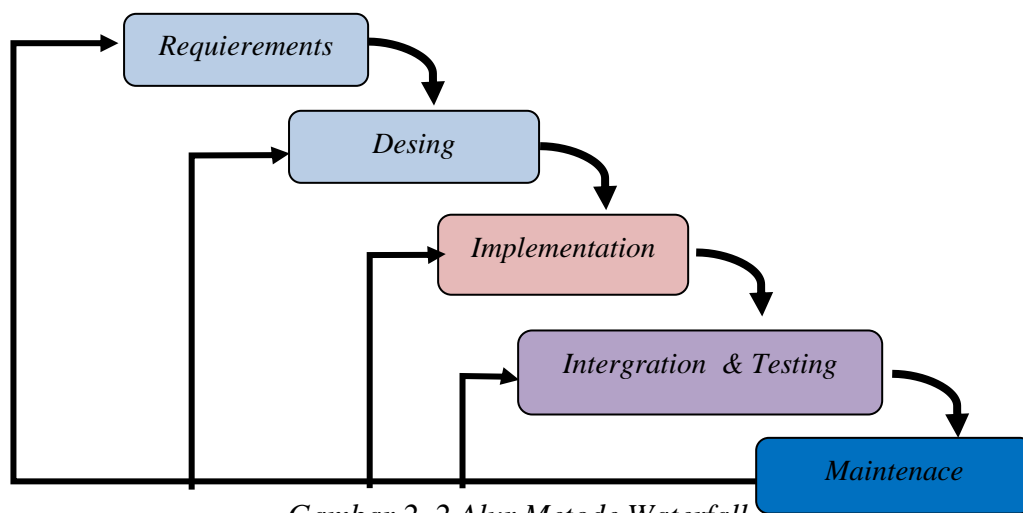
5. *Maintenance*

Tahapan yang terakhir adalah *maintenance*. Pada tahapan ini dilakukan pengoperasian dan perbaikan dari *software* atau aplikasi. Setelah pengujian sistem telah selesai, maka akan masuk ke tahap *software* tersebut dicoba oleh *User* (pengguna). Untuk proses pemeliharaan dari *software*, pengembang biasanya meminta *feedback* atau laporan dari *User* apabila *User* mendapat *error* atau *bug* dari aplikasi yang telah dibuat.

Adapun kelebihan dan kekurangan dari metode *waterfall*, yakni:

- a. Kelebihan metode *waterfall*
 - a) *Workflow* (aliran kerja) yang jelas
 - b) Hasil dokumentasi yang baik
 - c) Dapat menghemat biaya
 - d) Digunakan untuk pengembangan *software* berskala besar.
- b. Kekurangan metode *waterfall*

- a) Membutuhkan tim yang *solid*
- b) Masih kurangnya fleksibilitas
- c) Tidak dapat melihat gambaran sistem yang jelas
- d) Membutuhkan waktu yang lama. [13]



Gambar 2. 2 Alur Metode Waterfall

2.2.7 Desain Sistem

A. UML (*Unified Modelling Language*)

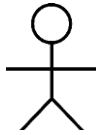

Menurut Miles dalam [14] *Unified Modelling Language* adalah sebuah metode untuk memodelkan suatu perancangan sistem berorientasi objek secara visual. Dalam UML terdapat 14 macam diagram seperti *Class diagram*, *Object diagram*, *Component diagram*, *Composite diagram*, *Compoite Sructure diagram*, *Package diagram*, *Deployment diagram*, *Use Case diagram*, *Actifity diagram*, *State Machine diagram*, *Sequence diagram*, *Communication diagram*, *Timing diagram*, dan *Interaction Overview*[15]. Disini penulis hanya akan menggunakan 2 macam diagram yaitu:





1. Use Case Diagram

Use Case diagram adalah jenis diagram pemodelan yang menggambarkan hubungan interaksi antara 1 atau lebih aktor dengan sistem atau aplikasi yang akan dibuat. Aktor disimbolkan menyerupai seseorang sedangkan sistem disimbolkan menyerupai elips. Aktor merupakan orang, proses atau sistem lain yang melakukan interaksi dengan sistem informasi yang akan dibuat itu sendiri. Jadi walaupun simbol dari aktor menyerupai orang, belum tentu aktor itu sendiri merupakan orang. *Use Case* adalah fungsi yang disediakan sistem sebagai unit-unit yang saling berinteraksi atau bertukar pesan antara unit atau aktor.

Berikut adalah simbol-simbol yang sering digunakan dalam membuat *Use Case* diagram:

Tabel 2. 1 Simbol Use Case Diagram

No	Nama	Simbol	Keterangan
1.	<i>Aktor</i>		Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari <i>actor</i> belum tentu merupakan orang: biasanya dinyatakan menggunakan kata benda diawal <i>frase</i> nama <i>actor</i> .
2.	<i>Use Case</i>		Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau <i>actor</i> ; biasanya dinyatakan dengan menggunakan kata kerja diawal frasa nama <i>Use Case</i> .

3.	<i>Association</i>		Komunikasi antara <i>actor</i> dan <i>Use Case</i> yang berpartisipasi pada <i>Use Case</i> atau <i>Use Case</i> memiliki interaksi dengan <i>actor</i> .
4.	<i>Generalization</i>		Menspesifikasikan bahwa <i>Use Case</i> target memperluas perilaku dari <i>Use Case</i> sumber pada suatu titik yang diberikan.
5.	<i>Include</i>		Relasi <i>Use Case</i> tambahan ke sebuah <i>Use Case</i> dan <i>Use Case</i> yang ditambahkan memerlukan <i>Use Case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.
6.	<i>Extend</i>		Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>Use Case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari hasilnya.

[16]

2. Activity Diagram


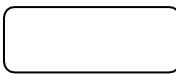
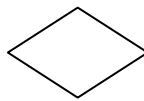


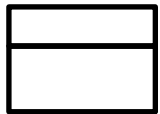
Activity diagram adalah jenis diagram pemodelan yang menggambarkan *workflow* atau aliran kerja dari sebuah sistem atau aplikasi yang dibuat. Pada diagram ini hanya aktivitas dari sistem saja yang digambarkan, jadi aktivitas dari aktor tidak dapat dilihat. *Activity* diagram sering digunakan untuk hal-hal berikut, yaitu [17]:

- a. Rancang menu yang ditampilkan pada *software*.
- b. Urutan atau pengelompokkan tampilan dari sistem atau *User interface*, dimana setiap aktivitas dianggap memiliki suatu antarmuka tampilan.

- c. Rancangan proses bisnis, dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis dari sistem yang didefinisikan.
- d. Rancangan pengujian, dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.

Berikut adalah simbol-simbol yang sering digunakan dalam pemodelan *activity diagram* :

Tabel 2. 2 Simbol Activity Diagram

No	Nama	Simbol	Keterangan
1.	Status Awal		Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah awal.
2.	Aktivitas		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.	Percabangan atau <i>Decision</i>		Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.	Penggabungan atau <i>join</i>		Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.	Status Akhir		Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6.	Swimlane		Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

Sumber: Aprianti dan Maliha [18]

2.2.8 Desain Database

A. Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) merupakan suatu model data yang dikembangkan berdasarkan objek.” *Entity Relationship Diagram* (ERD) digunakan untuk menjelaskan hubungan antar data dalam basis data kepada pengguna secara logis. *Entity Relationship Diagram* (ERD) didasarkan pada suatu persepsi bahwa *real world* terdiri atas objek-objek dasar tersebut. Penggunaan *Entity Relationship Diagram* (ERD) relatif mudah dipahami, bahkan oleh para pengguna yang awam. Bagi perancang atau analis sistem, *Entity Relationship Diagram* (ERD) berguna untuk memodelkan sistem yang nantinya, basis data akan di kembangkan. Model ini juga membantu perancang atau analis sistem pada saat melakukan analis dan perancangan basis data karena model ini dapat menunjukkan macam data yang dibutuhkan dan kerelasian antar data di dalamnya. [19]

Terdapat 3 komponen dalam membentuk suatu ERD, di antaranya :

1. Entitas, adalah sebuah objek untuk membedakan dari yang lain yang akan diwujudkan dalam basis data nantinya.
2. Hubungan / Relasi, adalah hubungan antara 2 jenis entitas yang digambarkan melalui garis lurus.
3. Atribut, memberikan detail informasi dari entitas. Jenis atribut ada 5 tergantung dari tipe data entitas, di antaranya :
 - a. Atribut *Primary Key*, adalah atribut yang unik karena tidak memiliki nilai yang sama pada baris data yang lain.

- b. Atribut *Simple*, adalah atribut yang bernilai *atomic* atau tidak dapat dipecah maupun dipilah lagi.
- c. Atribut *Multivalued*, adalah atribut yang memiliki lebih dari 1 nilai dari atribut yang bersangkutan.
- d. Atribut *Composite*, adalah atribut yang terdiri dari beberapa atribut yang lebih kecil yang berarti masih dapat dipecah lagi atau memiliki sub atribut.
- e. Atribut Derivatif, adalah atribut yang tidak harus disimpan dalam basis data.

Dalam perancangan ERD sering kali dijumpai derajat relasi, yang bertujuan untuk menjelaskan jumlah maksimum antara 1 entitas dengan entitas yang lain.

1. *One to One* (1 : 1)

Setiap anggota entitas A hanya dapat berhubungan dengan 1 anggota dalam entitas B, begitu pula sebaliknya.

2. *One to Many* (1 : M)



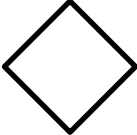




Setiap anggota entitas A dapat berhubungan dengan banyak anggota dalam entitas B, namun tidak sebaliknya.

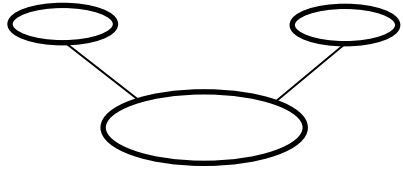

3. *Many to Many* (M : M)

Setiap anggota entitas A dapat berhubungan dengan banyak anggota dalam entitas B, begitu pula sebaliknya.

Berikut adalah simbol-simbol yang sering digunakan dalam pembuatan ERD, di antaranya :

Tabel 2. 3 Simbol ERD

No	Notasi	Arti
1		<i>Entity</i>
2		<i>Weak Entity</i>
3		<i>Relationship</i>
4		<i>Identifying Relationship</i>
5		Atribut
6		<i>Atribut Primary Key</i>
7		<i>Atribut Multivalued</i>

8		Atribut <i>Composite</i>
9		Atribut Derivatif

Sumber: Riski dan Ropianto[20]

2.2.9 Implementasi

A. Hypertext Preprocessor (PHP)

PHP adalah bahasa pemrograman web server-side yang bersifat open source. PHP merupakan singkatan dari PHP: Hypertext Preprocessor. PHP adalah script yang berintegrasi dengan HTML dan berada pada server (*server side HTML embedded scripting*). PHP berfungsi untuk membuat halaman *website* yang dinamis. Dinamis artinya halaman yang akan ditampilkan dan dibuat pada saat halaman itu diminta oleh *client*. Dengan adanya mekanisme ini menyebabkan informasi yang diterima oleh *client* selalu yang terbaru/*scrip up to date*. Semua script PHP dieksekusi pada server di mana script tersebut dijalankan.[21]

B. Javascript

JavaScript merupakan bahasa pemrograman web yang pemrosesan dilakukan di sisi client, karena berjalan disisi *client javascript* dapat dijalankan hanya dengan menggunakan browser. *javascript* dapat dijalankan tanpa memerlukan *refresh* pada *browser*, hal ini berbeda dengan PHP yang bekerja pada sisi server. *Java Script* biasanya dijalankan ketika ada event tertentu yang terjadi pada halaman web, baik itu event yang dilakukan oleh *User* maupun even yang

terjadi karena adanya perubahan pada halaman *website*. Sejak munculnya *query* sebagai salah satu library yang sangat memudahkan penggunaan *JavaScript* maka dari situlah *JavaScript* mulai berkembang secara populer. Beberapa hal tentang *Javascript* :

- a. *Javascript* didesain untuk menambah interaktif suatu *web*
- b. *Javascript* merupakan bahasa *scripting*
- c. Bahasa *scripting* merupakan bahasa pemrograman yang ringan
 - a) *Javascript* berisi baris kode yang dijalankan di komputer (*web browser*)
 - d. *Javascript* biasanya disisipkan (*embedded*) dalam halaman HTML
 - a) *Javascript* adalah bahasa interpreter (yang berarti skrip dieksekusi tanpa proses kompilasi)
 - b) Setiap orang dapat menggunakan *Javascript* tanpa membayar lisensi.[22]

C. MySQL

MySQL adalah singkatan dari *My Structure Query Language* yang merupakan salah satu *DataBase Management System (DBMS)* dari sekian banyaknya *DBMS* seperti *Oracle*, *MS SQL*, *Postgre SQL* dan lainnya. *MySQL* berfungsi untuk mengolah database menggunakan bahasa *SQL*. *MySQL* bersifat *open source* sehingga kita bisa menggunakannya secara gratis. Pemrograman *PHP* juga sangat mendukung/support dengan database *MySQL*.[21]

D. XAMPP

XAMPP merupakan perangkat lunak (*free software*) bebas, yang mampu mendukung banyak sistem operasi dan juga merupakan kompilasi dari beberapa program. *XAMPP* berfungsi sebagai server yang mampu untuk berdiri sendiri (*localhost*), di dalamnya terdiri dari beberapa program seperti: *Apache HTTP Server*, *Perl*, penerjemahan bahasa yang ditulis dengan bahasa pemrograman, dan *MySQL*. Nama *XAMPP* sendiri merupakan singkatan dari X (empat sistem operasi apa pun), *Apache*, *MySQL*, *PHP* dan juga *Perl*. Program ini tersedia di dalam *GNU General Public License* dan bebas, merupakan *web server* yang begitu mudah untuk digunakan serta dapat menampilkan halaman web yang dinamis. Untuk mendapatkannya *XAMPP* dapat mendownload langsung dari web resminya yaitu di situs <http://www.apachefriends.org>. [23]

2.2.10 Pengujian

A. Black Box Testing

Black Box Testing berfokus pada spesifikasi fungsional dari perangkat lunak. Tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program[24].

Black Box Testing cenderung untuk menemukan hal-hal berikut:

- a. Fungsi yang tidak benar atau tidak ada.
- b. Kesalahan antarmuka (*interface errors*).
- c. Kesalahan pada struktur data dan akses basis data.
- d. Kesalahan performansi (*performance errors*).
- e. Kesalahan inisialisasi dan terminasi.

Pengujian didesain untuk menjawab pertanyaan-pertanyaan berikut:

- a. Bagaimana fungsi-fungsi diuji agar dapat dinyatakan valid?
- b. Input seperti apa yang dapat menjadi bahan kasus uji yang baik?
- c. Apakah sistem sensitif pada input-input tertentu?
- d. Bagaimana sekumpulan data dapat diisolasi?
- e. Berapa banyak rata-rata data dan jumlah data yang dapat ditangani sistem?
- f. Efek apa yang dapat membuat kombinasi data ditangani spesifik pada operasi sistem?

B. User Acceptance Testing (UAT)

User Acceptance Testing (UAT) merupakan salah satu metodologi yang sangat inovatif untuk mencegah kegagalan proyek IT. Dalam pengembangan perangkat lunak, terdapat tiga hal yang dilakukan dalam proses UAT yaitu :

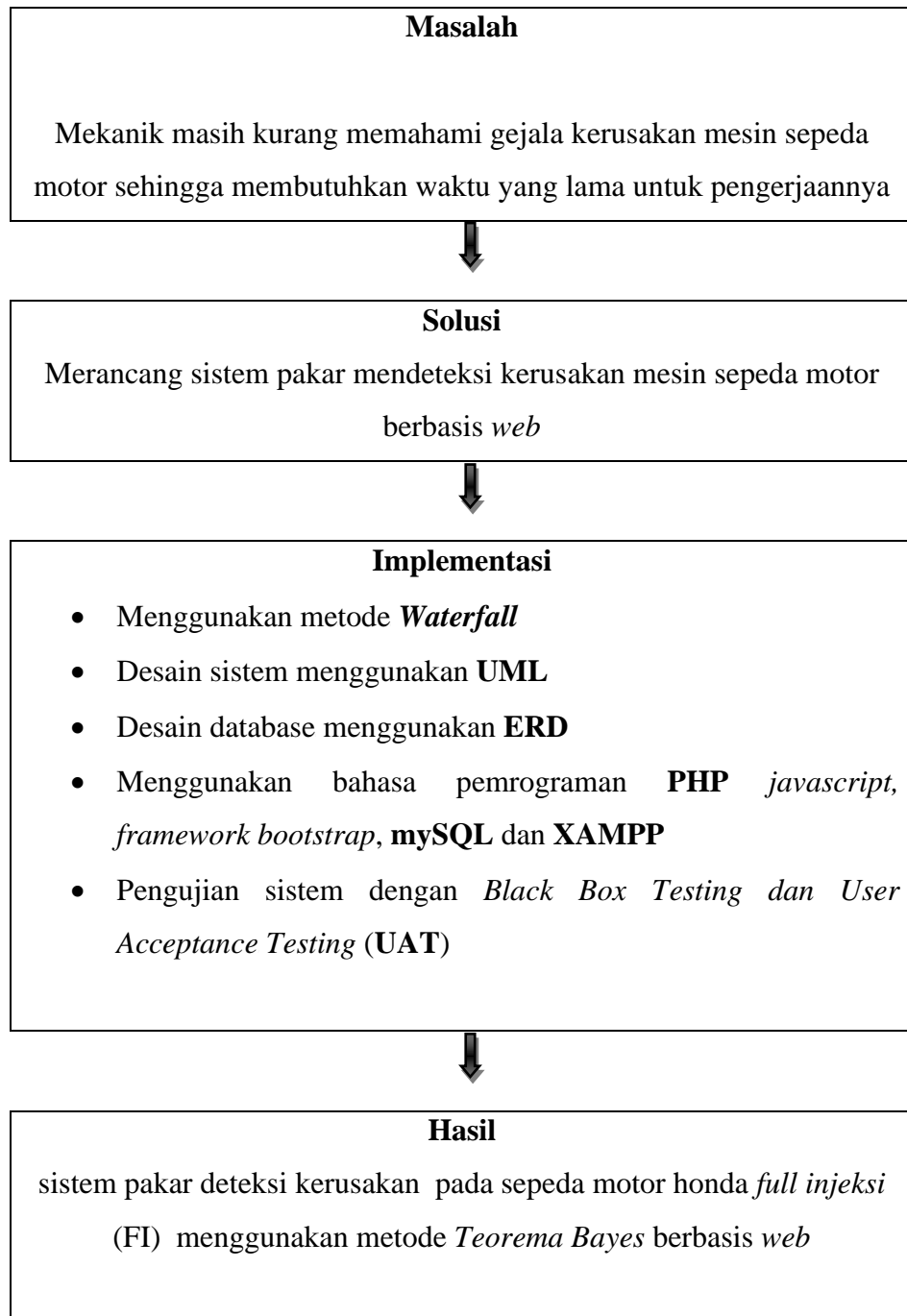
- a. UAT mengukur bagaimana sistem sudah sesuai dengan kebutuhan pengguna.
- b. UAT mengekspos fungsionalitas/*logic* bisnis yang belum ditemukan, karena *unit testing* dan *system testing* tidak berfokus pada fungsionalitas/*logic* bisnis.
- c. UAT membatasi bagaimana sistem telah “selesai” dibuat.

Proses UAT diawali dengan menyediakan dokumentasi persyaratan bisnis, kemudian dilanjutkan dengan proses bisnis (alur kerja) atau skenario dan yang terakhir yaitu pengujian menggunakan data. Efektifitas dalam pengujian sangat dibutuhkan dalam pengembangan sebuah aplikasi ataupun sistem informasi

sehingga produk tersebut dapat sampai kepada pengguna dengan tepat waktu dan sesuai dengan kebutuhan pengguna. Berdasarkan penelitian yang dilakukan oleh *Zhang*, efektifitas terhadap kriteria pengujian yang sudah ada dan yang baru harus dievaluasi untuk membangun teori pengujian yang lebih berguna.

Pengujian UAT menggunakan skala Likert. Skala Likert paling banyak digunakan dalam penelitian yang menggunakan kuisisioner kepada responden untuk menentukan tingkat penilaian terhadap kegunaan aplikasi yang telah dibuat nilai interpretasi persen dari setiap pernyataan yang diajukan kepada 20 responden. Nilai iterpretasi persen tersebut lalu dirata-rata untuk menentukan tingkat penerimaan responden terhadap aplikasi yang telah dibuat.[25]

2.3 Kerangka Pikir



Gambar 2. 3 Kerangka Pikir