

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Penyusunan pada tugas akhir ini, melakukan pengumpulan data dari sumber yang berupa karya skripsi maupun jurnal yang memiliki relevansi terhadap topik yang diteliti oleh penulis, berikut karya ilmiah terdahulu yang berkaitan

Penelitian oleh Liana & Lubis ditahun 2018 dengan mengangkat judul penelitian "Sistem Pakar Diagnosis Kerusakan Komputer dan Internet Dengan Certainty Factor Berbasis Web" dalam penelitian ini menggunakan dua utama komponen masalah yang saya atasi adalah perangkat keras dan perangkat lunak. Ada dua fitur utama dari ahli ini sistem, yang terdiri dari lingkungan pengembangan dan lingkungan konsultasi. lingkungan pengembangan dibuat dengan tujuan untuk mengelola diagnosis, gejala, dan hubungan dari masalah yang mungkin didekati pengguna. Seperti di lingkungan konsultasi, pengguna dapat menggunakannya dengan memasukkan input ke dalam tabel konsultasi yaitu disediakan dalam program, maka sistem akan menghitung hasil berdasarkan metode faktor kepastian dengan forward sistem rantai [5].

Penelitian oleh FA Sianturi ditahun 2019 dengan mengangkat judul penelitian "Implementasi Metode Certainty Factor untuk Diagnosa Kerusakan Komputer" dalam penelitian ini menggunakan Pakar sistem yang memecahkan masalah dengan meniru cara yang dilakukan oleh pakar ahli dengan permasalahan di bidangnya salah satunya dalam bidang reparasi komputer, yaitu masalah kerusakan komputer menjadi masalah yang cukup rumit, masalah ini umumnya dialami oleh individu dan institusi. Salah satunya di sekolah lembaga yang memiliki laboratorium computer [2]. untuk mendiagnosis kerusakan komputer dapat gunakan metode faktor kepastian yang membantu mengidentifikasi kerusakan pada komputer dan mencari penyebab kerusakan komputer berdasarkan gejala yang terjadi dan solusi untuk memperbaikinya.

Penelitian oleh Yusra Fadhilah di tahun 2020 dengan mengangkat judul penelitian “Sistem Pakar menggunakan Forward Chaining dan Certainty Factor Untuk Diagnosa Kerusakan Smartphone” dalam penelitian ini menggunakan metode forward chaining dan Certainty Factor, diawali dengan informasi berupa nyata mereka digabungkan menjadi aturan untuk menghasilkan kesimpulan. Hasil dari apa yang disimpulkan di depan. Faktor rantai dan kepastian dapat mencapai tingkat kesesuaian sebesar 73,33 % dibandingkan dengan hasil oleh ahli terkenal. Jumlah tersebut sudah menunjukkan bahwa metode ini cocok untuk membantu orang [4].

Penelitian oleh Yeyi & Putra ditahun 2020 dengan mengangkat judul penelitian ” Sistem Pakar Menggunakan Forward Chaining dan Certainty Factor Untuk Diagnosa Kerusakan Smartphone” dalam penelitian ini menggunakan metode forward chaining dan certainty factor, dimulai dengan informasi yang ada berupa data real lalu digabungkan menjadi sebuah rule untuk menghasilkan suatu kesimpulan[6].

Penelitian oleh Saputra ditahun 2022 dengan mengangkat judul penelitian “Sistem Pakar Diagnosa Kerusakan Hardware Komputer Menggunakan Metode Forward Chaining dan Certainty Factor Berbasis Website” dalam penelitian ini aplikasi sistem pakar ini, peneliti menggunakan penggabungan dua metode, yaitu metode Forward Chaining dan Metode Certainty Factor. Tools yang digunakan dalam pembangunan sistem pakar ini menggunakan aplikasi Sublime Text 3 sedangkan basis datanya menggunakan MySQL dan menggunakan bahasa pemrograman PHP [10]. Tujuan dikembangkannya aplikasi sistem pakar ini untuk membantu pengguna komputer dalam melakukan diagnosis awal terhadap suatu kerusakan pada Hardware komputer yang dialami beserta penyebab dan solusi untuk mengatasi kerusakan tersebut.

Tabel 2. 1 Penelitian Terdahulu

Peneliti	Judul	Hasil
Liana dan Lubis pada tahun 2018	Sistem Pakar Diagnosis Kerusakan Komputer dan Internet Dengan Certainty Factor Berbasis Web	Hasil pengujian antara lingkungan pengembangan dan lingkungan konsultasi ini memiliki beberapa perbedaan kasus dalam mendiagnosa gejala yang dialami oleh user. (dalam bentuk angka dalam bentuk persentasi). Aplikasi sistem pakar ini tidak sepenuhnya akurat untuk dijadikan pedoman dalam memberikan solusi, dikarenakan ada beberapa perbedaan opini antara pakar yang ahli di bidangnya dibandingkan dengan data kasus yang saya kumpulkan dari forum di internet.
Sianturi FA pada tahun 2019	Implementasi Metode Certainty Factor untuk Diagnosa Kerusakan Komputer	Penelitian ini menghasilkan sebuah Sistem pakar mendiagnosa kerusakan komputer dengan metode certainty factor dapat di implementasikan dengan baik dan efisien serta dapat memberikan solusi cara mengatasinya. Mendiagnosa kerusakan komputer dengan metode certainty factor, dilakukan dengan cara mendiagnosa berdasarkan gejala yang muncul dan menghasilkan nilai presentase keyakinan terhadap kerusakan

Yusra Fadhilah pada tahun 2020	Sistem Pakar menggunakan Forward Chaining dan Certainty Factor Untuk Diagnosa Kerusakan Smartphone	Penggunaan aplikasi sistem pakar dalam mendiagnosa kerusakan Smartphone ini bisa mendapatkan hasil yang cukup baik dengan nilai keakuratan mencapai 73,33% jika dibandingkan dengan representasi yang diberikan oleh seorang tenaga ahli service langsung atau pakar.
Yeyi dan Putra pada tahun 2020	Sistem Pakar Menggunakan Forward Chaining dan Certainty Factor Untuk Diagnosa Kerusakan Smartphone	Hasil yang cukup baik dengan nilai keakuratan mencapai 73,33% jika dibandingkan dengan representasi yang diberikan oleh seorang tenaga ahli service langsung atau pakar.
Saputra pada tahun 2022	Sistem Pakar Diagnosa Kerusakan Hardware Komputer Menggunakan Metode Forward Chaining dan Certainty Factor Berbasis Website	Penelitian ini menghasilkan satu aplikasi berbasis website yang ditujukan untuk mendiagnosa kerusakan Hardware pada komputer dengan menggunakan bahasa pemrograman PHP dengan metode Gabungan Forward Chaining dengan Certainty Factor

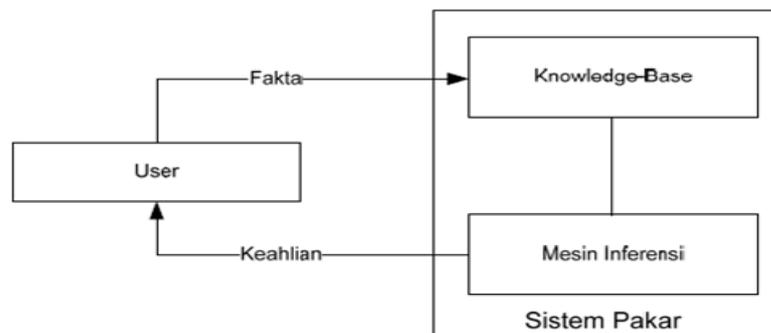
Berdasarkan Tabel 2.1 penelitian yang diuraikan maka terdapat perbedaan dengan penelitian ini. Pada penelitian ini menggunakan studi kasus untuk objek penelitian, studi kasus dilakukan pada salah satu toko komputer yang menyediakan layanan perbaikan komputer yaitu Legenda komputer Wonosobo dan metode pengembangan *system web* pakar pada penelitian ini menggunakan metode *Extreme Programming*. Pada penelitian sebelumnya memiliki kesamaan yaitu menerapkan metode Sistem Pakar *Forward chaining* dan *Certainty factor*.

2.2 Dasar Teori

2.2.1 Sistem pakar

Sistem pakar adalah jenis kecerdasan buatan (AI) yang menggunakan pengetahuan khusus untuk memecahkan masalah tingkat manusia yang ahli. “Pakar adalah orang yang memiliki pengetahuan atau kemampuan luar biasa dalam suatu bidang tertentu yang tidak diketahui atau mampu oleh orang lain [11].

Secara umum, sistem pakar adalah sistem yang berusaha mengadopsi pengetahuan manusia ke komputer yang dirancang untuk memodelkan kemampuan menyelesaikan masalah seperti layaknya seorang pakar. Dari pernyataan tersebut menunjukkan bahwa sistem pakar adalah suatu sistem komputer yang dirancang dengan memasukkan keahlian seorang pakar untuk menyelesaikan masalah layaknya pakar itu sendiri. Bagi para ahli, sistem pakar ini juga akan membantu aktivitasnya sebagai asisten yang sangat berpengalaman.



Gambar 2. 1 Konsep dasar sistem pakar [12]

Sistem pakar memiliki sejumlah komponen utama sistem dan antar muka dengan individu dalam berbagai peran yang diilustrasikan pada Gambar 2.1 [12]. Komponen utamanya adalah:

1. *Knowledge base*, representasi deklaratif keahlian dan biasanya dalam aturan *IF THEN*.

2. Mesin inferensi, kode pada inti sistem yang memperoleh rekomendasi dari basis pengetahuan dan data spesifik masalah.
3. *User interface*, kode yang mengontrol komunikasi antara pengguna dengan sistem.

Beberapa keunggulan dan kekurangan yang dimiliki oleh sistem pakar [12].

Keunggulan sistem pakar :

1. Sistem pakar dapat menyimpan pengetahuan dan keahlian dari pakar.
2. Memungkinkan seseorang yang bukan seorang pakar, dapat mengerjakan tugas dari seorang pakar.
3. Dapat melakukan proses yang sama secara berulang.
4. Mampu bekerja dalam ketidakpastian.
5. Mampu beroperasi dalam lingkungan yang berbahaya.

Kelemahan sistem pakar :

1. Biaya yang diperlukan untuk membuat, memelihara, dan mengembangkan sistem pakar yang tidak murah.
2. Sulit dikembangkan, karena keterbatasan pakar dan sulitnya mengekstrak keahlian dari pakar ke komputer.
3. Sistem pakar tidak selamanya benar.
4. Pendekatan oleh setiap pakar untuk suatu masalah terkadang berbeda- beda.

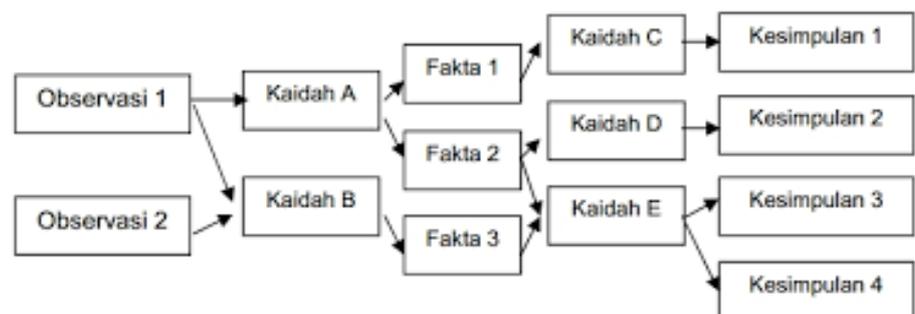
2.2.2 Kerusakan Hardware Komputer

Kerusakan pada perangkat keras komputer adalah masalah umum di antara pengguna komputer/pc; kerusakan terjadi karena kurangnya pemahaman pengguna tentang cara melakukan perawatan dan cara mengatasi kerusakan perangkat keras, sehingga banyak pengguna langsung pergi ke bengkel tanpa mengetahui apa yang terjadi pada komputer mereka.[13]

2.2.3 *Forward Chaining*

Chain (rantai) merupakan suatu perkalian inferensi yang menghubungkan suatu permasalahan dengan solusinya. Suatu rantai yang dicari atau dilewati/dilintasi dari suatu permasalahan untuk memperoleh solusinya disebut dengan *Forward Chaining*. Cara lain untuk menggambarkan *Forward Chaining* yaitu dengan penalaran dari fakta menuju konklusi yang terdapat dari fakta [14]. Metode inferensi *Forward Chaining* disebut juga *data driven*, dimulai dengan fakta-fakta dan menelusuri aturan-aturan yang sesuai sampai diperoleh kesimpulan [15].

Forward Chaining merupakan teknik pencarian yang dimulai dengan fakta yang diketahui, kemudian mencocokkan fakta-fakta tersebut dengan bagian *IF* (informasi masukan) dari *rule IF-THEN* [16]. Bila ada fakta yang cocok dengan bagian *IF*, maka *rule* tersebut akan dieksekusi. Bila sebuah *rule* dieksekusi, maka sebuah fakta baru (bagian *THEN*) ditambahkan ke dalam basis data. Setiap kali pencocokan, dimulai dari *rule* teratas. Setiap *rule* hanya boleh dieksekusi sekali saja. Proses pencocokan berhenti bila tidak ada lagi *rule* yang dieksekusi.



Gambar 2.2 Diagram *Forward Chaining*

Gambar 2.2 merupakan diagram *Forward Chaining* secara umum dalam menyelesaikan suatu masalah. Hal ini dapat dijelaskan ketika fakta dipilih maka sistem akan menuju pada *rule* yang telah dibangun. Sistem akan mencocokkan fakta yang ada dengan *rule* yang telah dibangun, jika

sistem menemukan *rule* yang sesuai maka sistem akan menampilkan kesimpulan berdasarkan *rule* yang ada [17].

2.2.4 *Certainty Factor*

Metode *certainty factor* merupakan metode yang mendefinisikan ukuran kapasitas terhadap suatu fakta atau aturan, dalam mengekspresikan tingkat keyakinan seorang pakar terhadap suatu masalah yang sedang dihadapi. Biasanya dalam suatu masalah sering ditemukan jawaban yang tidak memiliki kepastian. Ketidakpastian ini dapat berupa kemungkinan atau probabilitas yang tergantung pada hasil kejadian. Hasil yang tidak pasti ini dapat disebabkan oleh aturan yang tidak pasti atau jawaban pengguna yang tidak pasti. Menurut Anik dalam bukunya mengatakan sistem pakar dituntut untuk mampu bekerja dalam ketidakpastian, sehingga dapat mengatasi masalah ini dan dapat membuat keputusan yang benar. Nilai tertinggi dalam *certainty factor* adalah +1,0 (pasti benar), dan terendah adalah -1,0 (pasti salah).

Tabel 2.2 istilah dan besaran *certainty factor* [18]

Istilah	<i>Certainty Factor</i>
Pasti Tidak	-1,0
Hampir Pasti Tidak	-0,8
Kemungkinan Besar Tidak	-0,6
Mungkin Tidak	-0,4
Tidak Tahu	-0,2 sampai 0,2
Mungkin Benar	0,4
Kemungkinan Besar Benar	0,6
Hampir Pasti Benar	0,8
Pasti Benar	1,0

Berikut merupakan rumus dari *certainty factor* :

$$CF[h, e] = MB[h, e] - MD[h, e] \dots\dots (1)$$

Jika ditemukan lebih dua *evidence*, maka kombinasikan CF dari setiap *evidence* menggunakan persamaan seperti di bawah:

$$CF = CF[h, e]_1 + CF[h, e]_2 \times (1 - CF[h, e]_1) \dots\dots (2)$$

Jika terdapat lebih dari dua *evidence*, maka kombinasikan nilai CF sebelumnya (CF_{old}) dengan *evidence* baru menggunakan persamaan seperti dibawah:

$$CF = CF_{old} + [h, e]_3 \times (1 - CF_{old}) \dots\dots (3)$$

Keterangan:

$[h, e]$: Faktor kepastian

$[h, e]$: *Measurement Believe* (MB) merupakan tingkat keyakinan terhadap *hipotesa* “h”, jika dipengaruhi *evidence* “e”.

$[h, e]$: *Measurement Disbelieve*(MD) merupakan tingkat ketidakyakinan terhadap hipotesis “h”, jika dipengaruhi Evidence “e” .

h : (*Hipotesa*) atau dugaan sementara (jenis penyakit).

e : (*Evidence*) atau gejala-gejala yang muncul.

Berikut merupakan contoh perhitungan dengan contoh kasus:

Si Tono memiliki komputer dengan kerusakan layar tidak tampil. Teknisi memperkirakan komputer Tono terkena blue screen dengan kepercayaan $MB[\text{blue screen, layar tidak tampil}] = 0,7$ dan $MD[\text{blue screen, layar tidak tampil}] = 0,2$. Maka nilai dari *certainty factor* :

$$CF[\text{blue screen, layar tidak tampil}] = 0,7 - 0,2 = 0,5$$

Rumus CF dengan kombinasi 2 evidence:

$$CF_{\text{combine}} = CF[h, e]_1 + CF[h, e]_2 \times (1 - CF[h, e]_1)$$

Jika ada observasi baru dan hasilnya komputer Tono terdeteksi bios rusak, dengan kepercayaan $MB[\text{blue screen, bios rusak}] = 0,6$ dan $MD[\text{blue screen, bios rusak}] = 0,2$.

$$CF[\text{blue screen, bios rusak}] = 0,6 - 0,2 = 0,4 \text{ Maka}$$

:

$$CF_{\text{combine}}[\text{blue screen, layar tidak tampil} \wedge \text{bios rusak}] = 0,5 + 0,4 * (1 - 0,5) = 0,7$$

Jika dilihat dari contoh di atas, gejala yang dialami komputer Tono yaitu layar tidak tampil 0,5 menyebabkan orang lain beranggapan bahwa komputer tonu blue screen. Setelah ditemukannya gejala baru bios rusak faktor keyakinan bahwa komputer Tono blue screen turun menjadi 0,7.

2.2.5 *Hypertext Preprocessor (PHP)*

HyperText PreProcessor (PHP) merupakan bahasa pemrograman yang di proses di server, Fungsi utama PHP dalam membangun website adalah untuk melakukan pengelolaan data dalam database (Rohi Abdulloh, 2016:2) PHP ialah suatu bahasa pemrograman yang digunakan untuk menerjemahkan baris kode program menjadi kode mesin yang dapat di mengerti oleh komputer yang bersifat serverside yang dapat di tambahkan ke dalam HTML (Supono, 2018:3).

PHP dikembangkan khususnya untuk mengakses dan memanipulasi data yang ada di database server open source seperti MySQL (Teguh Wahyono, 2005, h.5) [19].

2.2.6 *Extreme Programming*

Extreme Programming (XP) adalah metodologi pengembangan perangkat lunak yang adaptif, berfokus pada kecepatan, dan menekankan pada kualitas perangkat lunak yang tinggi. XP melibatkan serangkaian tahapan yang berfokus pada komunikasi intensif, kolaborasi tim, dan pengujian terus-menerus.



Gambar 2.3 Extreme Programming

1. Tahap Eksplorasi

Tahapan eksplorasi adalah tahapan dimana klien menuliskan kebutuhan-kebutuhan dari sistem yang paling mendasar. Setiap kebutuhan yang dituliskan oleh klien akan dibuat dalam bentuk modul yang sederhana atau disebut juga dengan *User Stories*. hasil dari tahapan eksplorasi adalah mengetahui dokumentasi atas visi dan ruang lingkup pekerjaan. Berikut contoh tahap eksplorasi dengan menggunakan user stories.

Tabel 2. 3 contoh user stories [9]

User	Kebutuhan Sistem
<i>Admin</i>	<i>Admin</i> adalah orang yang mengisikan dan menganalisa data barang, menganalisa data penjualan, menganalisa <i>guest</i> yang mendaftar sebagai <i>member</i> , membalas pesan dari <i>member</i> , dan <i>memberikan</i> peringatan kepada <i>member</i> yang belum membayar barang yang sudah dipesan.
<i>Member</i>	<i>Member</i> adalah user yang berbelanja pada aplikasi <i>e-Commerce</i> ini. <i>Member</i> dapat melihat barang, melakukan pemesanan barang, mengubah profil yang terdiri dari nama <i>member</i> , alamat rumah <i>member</i> , dan kontak <i>member</i> . Kemudian <i>member</i> juga dapat melakukan pengiriman pesan kepada <i>admin</i> untuk <i>memberikan</i> pertanyaan, kritik ataupun saran.

2. Tahap Perencanaan

Tahap perencanaan berorientasi kepada tahapan eksplorasi. Tahapan ini akan memperkirakan kebutuhan bisnis, kebutuhan user, dan kebutuhan sistem. Tahapan ini juga akan menghasilkan penjadwalan yang menggambarkan perencanaan waktu pelaksanaan pembangunan system. Pada tahap perencanaan ini, akan dihasilkan kebutuhan bisnis, dan kebutuhan sistem, dengan mengacu kepada tahapan sebelumnya, yakni tahap eksplorasi. Kebutuhan system pada tahap ini berisi tentang analisis yang dilakukan terhadap fungsionalitas system serta perancangan data [9].

3. Iterasi Pengembangan Sistem

Pada tahap ini, akan terjadi beberapa kali iterasi, setiap iterasi terdiri dari 3 tahapan yaitu analisis sistem, desain sistem, dan pembuatan dan pengujian sistem.

a) Analisis Sistem

Tahapan ini menganalisa kebutuhan sistem untuk user admin. Tabel dibawah merupakan contoh untuk kebutuhan sistem.

Tabel 2. 4 contoh rincian kebutuhan system [9]

Halaman Admin	Fungsi-fungsi yang Disediakan
Halaman Home	<ul style="list-style-type: none"> • Label nama admin (tanda bahwa admin sedang login) • Menu tab : Item, Sold Item, Member, Inbox • Form input barang baru • Menampilkan tabel barang
Halaman <i>Sold Item</i>	<ul style="list-style-type: none"> • Label nama admin (tanda bahwa admin sedang login) • Menu tab : Item, Sold Item, Member, Inbox • Menampilkan tabel data penjualan
Halaman <i>Member</i>	<ul style="list-style-type: none"> • Label nama admin (tanda bahwa admin sedang login) • Menu tab : Item, Sold Item, Member, Inbox • Menampilkan tabel data member
Halaman <i>Inbox</i>	<ul style="list-style-type: none"> • Label nama admin (tanda bahwa admin sedang login) • Menu tab : Item, Sold Item, Member, Inbox • Menampilkan seluruh pesan dari member • Tombol balas pesan

b) Pembuatan dan pengujian

Tahapan ini berisi tentang klien memberikan *feedback* kepada developer untuk merubah dan menambah fungsi maupun desain. Tabel dibawah merupakan contoh hasil dari pembuatan dan pengujian.

Tabel 2.5 contoh penambahan requirment [9]

Halaman Admin	Fungsi-fungsi yang diinginkan
Halaman <i>Home</i>	<ul style="list-style-type: none"> • Ganti halaman home menjadi kalimat pembuka “Welcome to TMS Komputer” dan gambar yang menunjukkan pelayanan untuk pelanggan dari pegawai. • Menu tab yang pertama diganti menjadi Home, dan selanjutnya Item, Sold Item, Member dan Inbox.
Halaman <i>Item</i>	<ul style="list-style-type: none"> • Tambahkan halaman Item menggantikan halaman home, sehingga isi dari halaman home yang pertama dipindahkan ke halaman Item.
<i>Desain</i>	<ul style="list-style-type: none"> • Ganti backgorund menjadi warna putih • Ganti menu tab

c) *Desain system*

Tahapan ini berisi tentang pembuatan desain system yang akan digunakan untuk pembuatan aplikasi. Keluaran dari tahapan ini berisi tentang desain wireframe dari aplikasi web yang akan dibuat [9].

4. Tahap produksi akhir

Tahap produksi akhir adalah tahapan dimana sistem sudah siap untuk di *release*. Tahapan ini akan melakukan *testing* terhadap keseluruhan sistem yang telah dibuat kepada klien. Setelah developer selesai membangun seluruh halaman dan sesuai dengan keinginan klien, maka developer kembali bertemu dengan klien untuk menunjukkan kepada klien aplikasi yang sudah siap untuk diimplementasikan. Berikut contoh pengujian keseluruhan system yang telah dibuat kepada klien.

Tabel 2. 6 contoh pengujian system [9]

Fungsi	Hasil Pengujian
Input data barang	Berhasil
Edit data barang	Berhasil
Hapus data barang	Berhasil
Melihat data barang	Berhasil
Melihat data penjualan	Berhasil
Menghapus data penjualan	Berhasil

2.2.7 **PHP Native**

PHP *native* adalah bahasa pemrograman PHP murni hanya berisi *script* PHP biasa tanpa adanya *library-library* tertentu, penggunaan PHP *native* lebih simple dibandingkan *framework*. PHP *native* memiliki instruksi yang dibuat tanpa adanya campur tangan pengembang lain untuk memproses nya, sehingga kode program tersebut hanya diketahui oleh pembuat PHP *Native* [20].

2.2.8 *Hypertext Markup Language (HTML)*

HTML sebuah *markup language* yang digunakan menentukan struktur dari sebuah *file*. Bahasa ini digunakan di sebuah *web page*. HTML dapat membantu format *text* atau gambar tepat pada *website*. Tanpa HTML dalam *browser* tidak dapat menampilkan struktur dan format *website* dengan baik

Perkembangan HTML yang pesat mengalami perkembangan dari waktu ke waktu. HTML pertama dibuat oleh IBM pada tahun 1980, dan membentuk program untuk memformat dokumen secara otomatis dari penempatan elemen *tag* [20].

2.2.9 *Cascading Style Sheet (CSS)*

Cascading Style Sheets sebuah standar W3C dimungkinkan untuk mengontrol bagaimana *file* HTML ditampilkan kepada pengguna. Penting untuk dicatat bahwa sintak CSS dan *JavaScript* berbeda dari HTML. Sederhananya, CSS hanyalah bahasa deklaratif yang digunakan untuk menentukan nilai yang kemudian digunakan untuk menentukan bagaimana elemen HTML ditampilkan di browser [21].

2.2.10 *Unified Modelling Language (UML)*

Unified Modelling Language dapat disingkat UML digunakan untuk melakukan permodalan visualisasi perancangan, dokumentasi, dan spesifikasi system

1. *Activity Diagram*

Activity Diagram merupakan pemodelan dalam alur proses sebuah sistem, alur proses suatu sistem direpresentasikan dalam diagram pemodelan ini sebagai gambar vertikal. Unsur-unsur dari suatu proses dapat direpresentasikan dalam diagram aktivitas sebagai opsi, pengulangan, atau aktivitas dan tindakan [21].

Tabel 2. 7 Simbol-Simbol Activity Diagram

No	Simbol	Nama	Keterangan
1		Status Awal	Sebuah <i>Activity</i> diagram memiliki status awal.
2		Aktivitas	Aktivitas yang dilakukan oleh sistem, biasanya diawali dengan kata kerja.
3		Penggabungan	Menghubungkan penggabungan dari beberapa aktivitas.
4		Percabangan	Percabangan terjadi aktivitas lebih dari satu.
5		Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6		<i>Swimlane</i>	<i>Swimlane</i> memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

2. Use case Diagram

Use case Diagram termasuk dalam *Unified Modelling Language*, *use case* sendiri merupakan diagram yang digunakan sebagai pemodelan urutan aktivitas dalam sebuah sistem, penggambaran yang menghubungkan *user* dengan system [21].

Tabel 2. 8 Simbol-Simbol *Use Case Diagram*

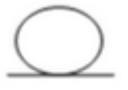
No	Simbol	Nama	Keterangan
1.		<i>Actor</i>	Simbol <i>user</i> atau sistem lain yang berkomunikasi dalam <i>use case</i> .
2.		<i>Use case</i>	Abstraksi dan deskripsi antar <i>actor</i> dan sistem.
3.		<i>Association</i>	Menghubungkan antara objek objek.
4.		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit
5.		<i>Extend</i>	Menunjukkan bahwa fungsi suatu <i>use case</i> merupakan fungsional tambahan dari <i>use case</i> lain.

3. *Sequence Diagram*

Sequence diagram termasuk dalam *Unified Modelling Language* yang menunjukkan bagaimana suatu sistem bekerja dalam urutan tertentu dari waktu ke waktu pemrosesan sistem, dan menggambarkan bagaimana setiap hal dalam suatu sistem berinteraksi satu sama lain. Objek yang terhubungnya berjalan ke arah kanan [21].

Tabel 2. 9 Simbol-Simbol *Sequence Diagram*

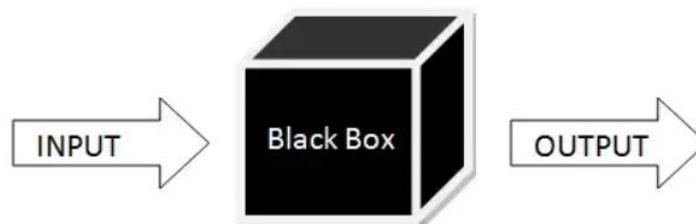
No.	Simbol	Nama	Keterangan
1.		<i>Actor</i>	Menggambarkan orang berinteraksi dengan sistem.
2.		<i>Object</i>	Komponen Utama <i>Sequence Diagram</i> .
3.		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi - informasi aktivitas yang terjadi.

No.	Simbol	Nama	Keterangan
4.		<i>Line Life</i>	Menggambarkan tempat mulai dan berakhirnya dari sebuah <i>message</i> .
5.		<i>Self Message</i>	Spesifikasi dari komunikasi antar <i>life line</i> pada sebuah interaksi.
6.		<i>Control Class</i>	Menghubungkan <i>boundary</i> dengan tabel.
7.		<i>Boundary</i>	Menggambar sebuah <i>form</i> .
8.		<i>Entity Class</i>	Menggambarkan hubungan kegiatan yang dilakukan.

4. Class Diagram

Struktur penggambaran dari sebuah sistem lebih terperinci mendeskripsikan dalam sistem yang saling berkaitan dengan komponen komponennya. Bekerja untuk mengilustrasikan sebuah *system* [21].

2.2.11 *Black Box*



Gambar 2. 4 *Black Box*

Teknik pengujian menggunakan *Black Box* ini digunakan untuk mengevaluasi

fitur dan fungsi situs *web*. Sebelum situs *web* dirilis, langkah ini dapat mengungkapkan masalah atau kesalahan apa pun. Pengujian fungsionalitas *website* menggunakan pengujian ini sangat mudah dan sederhana. Keuntungan menguji menggunakan *Black Box* ini penguji tidak perlu punya pengetahuan tentang bahasa pemrograman, hanya menguji sebatas penggunaan fitur, Pengujian *Black Box* juga berfokus pada *input* dan *output* aplikasi, menentukan apakah aplikasi dapat digunakan saat ini atau tidak dengan melihat data yang dievaluasi [22].