

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terdahulu

Berikut adalah beberapa jurnal terdahulu dengan beberapa judul yang menggunakan penerapan metode *Certainty Factor* dapat dilihat dibawah ini :

1. Nelly Astuti Hasibuan (2017) dengan judul “Sistem Pakar Mendiagnosa Penyakit Kaki Gajah Menggunakan Metode Certainty Factor”. Dimana didalam penelitian ini penyakit kaki gajah atau filariasis adalah penyakit menular yang disebabkan oleh cacing filarial yang ditularkan melalui jenis nyamuk. Penyakit ini bersifat kronis dan bila tidak mendapatkan pengobatan akan mengakibatkan cacat menetap berupa pembesaran kaki, alat kelamin baik perempuan maupun laki-laki. Selain itu berbagai metode pemeriksaan juga dilakukan untuk mendiagnosa penyakit kaki gajah diantaranya ialah dengan yang dikenal sebagai penjarangan membrane, metode konsentrasi knott dan teknik pengendapan. diberikan pertanyaan-pertanyaan untuk mendapatkan nilai kemungkinan dan kemudian nilai yang didapat akan diselesaikan dengan rumusan.
2. Aryu Hanifah Aji (2018) dengan judul “Sistem Pakar Diagnosa Penyakit Ibu Hamil Menggunakan Metode *Certainty Factor* (CF)”. Aplikasi yang dibangun merancang sistem pakar diagnosa ibu hamil yang dapat membantu mengenali penyakit selama kehamilan berlangsung, berdasarkan gejala-gejala yang disarankan ibu hamil. Berdasarkan hal tersebut, diusulkan solusi berupa sistem pakar diagnosis penyakit ibu hamil menggunakan metode *certanty*

factor yang dapat membantu mengenali penyakit selama kehamilan berlangsung berdasarkan gejala-gejala yang dirasakan ibu hamil serta tempat rujukan yang harus dituju oleh pasien. Metode CF memiliki kinerja sistem yang mampu berjalan sesuai kebutuhan fungsional dan hasil persentase akurasi tinggi. Selain itu metode CF dapat menggambarkan tingkat keyakinan seorang pakar terhadap masalah yang sedang dihadapi. Berdasarkan hasil pengujian, diperoleh hasil 100% fungsionalitas sistem pakar diagnosa penyakit ibu hamil berjalan sesuai dengan daftar kebutuhan sistem dan sistem mempunyai tingkat akurasi sebesar 100%.

3. Mohammad Arifin (2017) dengan judul “Penerapan Metode *Certainty Factor* Untuk Sistem Pakar Diagnosis Hama Dan Penyakit Pada Tanaman Tembakau”. Dimana aplikasi yang dirancang untuk membantu para petani mendiagnosis hama dan penyakit tanaman tembakau, dengan membedakan antara hama dan penyakit. Dalam penelitian ini, sistem pakar diagnosis hama dan penyakit pada tanaman tembakau dibangun untuk membantu mendiagnosis jenis hama atau penyakit yang menyerang tanaman tembakau serta memberikan berbagai solusi untuk hama atau penyakit. Metode yang digunakan pada sistem pakar ini adalah metode *certainty factor*. Metode *certainty factor* dipilih karena metode ini cocok dalam proses penentuan identifikasi hama dan penyakit, dan hasil dari penerapan metode ini adalah presentase. Presentase sistem disini merupakan tingkat akurasi penentuan penyakit atau hama yang menjangkit tanaman tembakau. Penentuan presentase dipengaruhi oleh nilai MB yang dapat dari sistem dan nilai MD yang dapat dari penilaian seorang pakar. Berdasarkan penelitian yang

dilakukan, penentuan hama atau penyakit yang menyerang tanaman tembakau dipengaruhi oleh pemilihan gejala. Persentase pada konsultasi sistem pakar diambil dari hasil tertinggi pertama dan kedua, sebagai alternatif hama lain atau penyakit yang menyerang tanaman tembakau.

4. Rudini Putra Tanjung (2017) dengan judul “Sistem Pakar Mendeteksi Kerusakan Mesin Las Inverter Dengan Metode Certainty Factor”. Mesin las inverter adalah pengembangan dari generasi mesin las konvensional dimana pada mesin las konvensional menggunakan transformator/koil dengan frekuensi kerja yang rendah 50-60Hz, sehingga untuk mendapatkan arus besar maka perlu memperbesar ukuran transformator, sehingga semakin besar ukuran transformator maka membutuhkan daya listrik yang besar pula. Pada mesin las inverter daya dihasilkan melalui proses switching transformator inti ferit dimana frekuensi kerja *switching* yang sangat tinggi antara 18khz sampai 100khz sehingga ukuran transformator sangat kecil dan daya yang dihasilkan besar. sebuah sistem pakar yang dapat menganalisa gejala-gejala kerusakan menjadi sebuah keputusan nama kerusakan dengan menggunakan metode *Forward Chaining* sebagai metode penelusuran dan metode *Certainty Factor* sebagai metode untuk menghitung nilai kepercayaan atas gejala yang diberikan oleh *welder*.
5. Adi Sucipto (2018) dengan judul “Penerapan Metode Certainty Factor Pada Diagnosa Penyakit Saraf Tulang Belakang”. *Certainty Factor* merupakan metode yang mendefinisikan ukuran kepastian terhadap fakta atau aturan untuk menggambarkan keyakinan seorang pakar terhadap masalah yang sedang dihadapi. Untuk membantu dan mempermudah masyarakat dalam

mendiagnosa penyakit saraf tulang belakang dapat menerapkan metode certainty factor pada sistem pakar. Saraf bagian tulang belakang merupakan organ yang penting bagi manusia. Terbatas dan tidak meratanya dokter spesialis saraf tulang belakang di Indonesia mengakibatkan masyarakat kesulitan dalam mendiagnosa penyakit tersebut. Sistem pakar dapat membantu untuk diagnosa penyakit, dimana sistem ini untuk merekonstruksi keahlian dan penalaran kemampuan seorang pakar. Pada penelitian ini, certainty factor diimplementasikan pada aplikasi diagnosa penyakit saraf tulang belakang. Sistem diujicobakan pada sejumlah masukan, hasil pengujian didapatkan memberikan hasil sesuai dengan perhitungan manual. Hasil pengujian dengan uji coba pada sejumlah masukan yang dilakukan didapatkan bahwa pengujian memberikan hasil sesuai dengan perhitungan manual. Sedangkan pengujian akurasi kesesuaian dari data testing yang didapatkan oleh pakar dengan output sistem pakar didapatkan hasil output yang sesuai sebanyak sebesar 90%.

II.2. Landasan Teori

Untuk mendukung keberhasilan penelitian ini, penyusun melakukan pendekatan teoritis melalui beberapa literatur yang berhubungan dengan penelitian yang dilakukan. Beberapa landasan teori pada penelitian ini yaitu :

1. Oli

Oli atau pelumas (*lubricant* atau sering disebut *lube*) adalah suatu bahan (biasanya berbentuk cairan) yang berfungsi untuk mereduksi kehausan antara dua permukaan benda bergerak yang saling bergesekan. Suatu bahan cairan dapat

dikategorikan sebagai pelumas jika mengandung bahan dasar (bisa berupa *oil based* atau *water/glycol based*) dan paket aditif. Pelumas mempunyai tugas pokok untuk mencegah atau mengurangi keausan sebagai akibat dari kontak langsung antara dua permukaan logam yang saling bergesekan sehingga keausan dapat dikurangi, besar tenaga yang diperlukan akibat gesekan dapat dikurangi dan panas yang ditimbulkan oleh gesekan pun akan berkurang. Pelumas yang baik harus bisa membuat kinerja mesin lebih ringan dan bertugas sebagai pelindung komponen metal didalam mesin dari friksi akibat gesekan antar logam, (Mustajib Furqon Haqiqi, 2019).

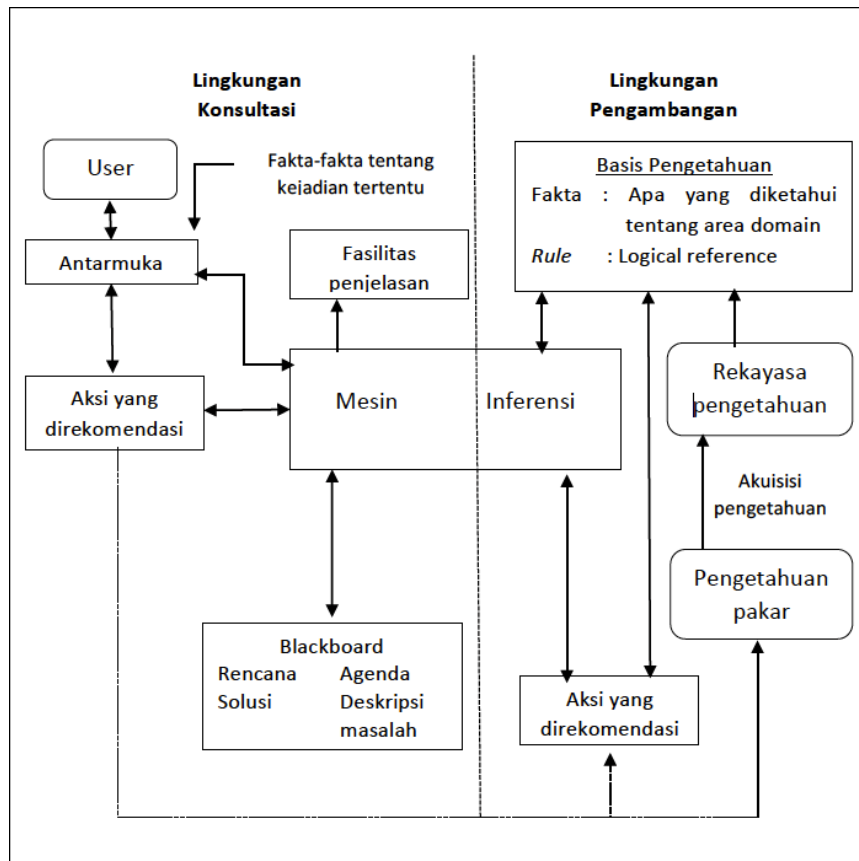
2. Sepeda Motor Automatic

Transmisi otomatis atau dikenal dengan sebutan *Continuous Variable Tranmision* (CVT) telah lama digunakan di dunia otomotif termasuk sepeda motor. Penggunaan komponen CVT asalnya dari paten DAF Belanda yang saat ini dasar teknologinya telah melewati masa perlindungan paten. Di tanah air, penggunaan transmisi otomatis pada sepeda motor telah dimulai pada tahun 1980-an. Pada tahun 1984, pabrikan asal Taiwan memasukan skutik Adly. Skutik asal Taiwan tersebut merupakan sepeda motor pertama bertransmisi CVT dengan mesin 2-tak berkapasitas 50 cc. Namun, masyarakat saat itu belum mengerti teknologi CVT. Produk ini dianggap terlalu maju di Era itu, terlebih lagi jaringan penjualan dan kesediaan suku cadangnya tidak merata. Terbukti sambutan masyarakat terhadap kehadiran sepeda motor Adly tidak begitu sukses. Tahun 1992 Piaggio coba menawarkan kembali skuter bertransmisi CVT dengan merek Corsa. Penjualan Corsa ternyata tidak menggembirakan hingga akhirnya

dihentikan produksinya pada tahun 1998. Selain modelnya yang dianggap ketinggalan dibandingkan dengan sepeda motor Jepang, sepeda motor ini tidak diminati karena masyarakat belum siap menerima produk sepeda motor bertransmisi CVT. Kultur masyarakat Indonesia yang cenderung konservatif dalam menerapkan teknologi baru menjadikan penggemar skuter matic lebih sedikit dibandingkan dengan sepeda motor tipe bebek. Selain itu, desain dan model yang kurang mencerminkan kebutuhan masyarakat dan kurangnya promosi menjadi penyebab kurang lakunya sepeda motor tipe matic dipasaran, (Douglas Alfrido, 2017).

3. Sistem Pakar

Istilah sistem pakar dari istilah *knowledge- based expert system*. Sistem pakar menggunakan pengetahuan seorang pakar yang dimasukkan ke dalam komputer. Seorang yang bukan pakar/ ahli menggunakan sistem pakar untuk meningkatkan kemampuan pemecahan masalah, sedangkan seorang pakar menggunakan sistem pakar untuk *knowledge assistant*. Ada dua bagian penting dalam sistem pakar, yaitu lingkungan pengembangan dan lingkungan konsultasi. Lingkungan pengembangan digunakan oleh pembuatan sistem pakar untuk membangun komponen-komponennya dan memperkenalkan pengetahuan ke dalam *knowledge base* (basis pengetahuan). Lingkungan konsultasi digunakan oleh pengguna untuk berkonsultasi sehingga pengguna mendapat pengetahuan dari sistem pakar seperti berkonsultasi dengan seorang pakar. Pada Gambar dibawa ini ditunjukkan komponen pada sistem pakar (Aryu Hanifa Aji; 2018).



Gamabar II.1. Komponen-Komponen Yang Penting Dalam Sistem Pakar
(Sumber : Aryu Hanifa Aji; 2018)

4. Ciri-Ciri Sistem Pakar

Adapun beberapa ciri-ciri sistem pakar dapat dijelaskan sebagai berikut :

- Terbatas pada bidang yang spesifik.
- Dapat memberikan penalaran untuk data-data yang tidak lengkap atau tidak pasti.
- Dapat mengemukakan rangkaian alasan yang diberikannya dengan cara yang dapat dipahami.
- Berdasarkan pada rule atau kaidah tertentu.
- Dirancang untuk dapat dikembangkan secara bertahap.
- Outputnya* bersifat nasihat atau anjuran.

- g. *Output* tergantung dari dialog dengan user.
- h. *Knowledge base* dan *inference engine* terpisah, (Ricky Hamidi, 2017).

5. Keuntungan Sistem Pakar

Adapun beberapa keuntungan sistem pakar dapat dijelaskan sebagai berikut :

- a. Membuat seorang awam dapat bekerja seperti layaknya seorang pakar.
- b. Dapat bekerja dengan informasi yang tidak lengkap atau tidak pasti.
- c. Meningkatkan output dan produktivitas, karena sistem pakar dapat bekerja lebih cepat dari manusia.
- d. Meningkatkan kualitas.
- e. Menyediakan nasihat yang konsisten dan dapat mengurangi tingkat kesalahan.
- f. Membuat peralatan yang kompleks lebih mudah dioperasikan karena sistem pakar dapat melatih pekerja yang tidak berpengalaman.
- g. Handal (reliability).
- h. Sistem pakar tidak dapat lelah atau bosan. Juga konsisten dalam memberi jawaban dan selalu memberikan perhatian penuh.
- i. Memungkinkan pemindahan pengetahuan ke lokasi yang jauh serta memperluas jangkauan seorang pakar, dapat diperoleh dan dipakai dimana saja, (Ricky Hamidi, 2017).

6. Metode *Certainty Factor* (CF)

Faktor kepastian (*certainty factor*) diusulkan oleh *Shortliffe* dan *Buchanan* pada 1975 untuk mengakomadasikan ketidakpastian pemikiran (*inexact*

reasoning) seorang pakar. Seorang pakar, (misalnya dokter) sering kali menganalisis informasi yang ada dengan ungkapan seperti misalnya : mungkin, kemungkinan, besar, hampir pasti. Untuk mengakomodasi hal ini dengan menggunakan *certainty factor* (CF) guna menggambarkan tingkat keyakinan pakar terhadap masalah yang dihadapi. Secara umum, rule direpresentasikan dalam bentuk sebagai berikut :

IF E1 [AND/OR] E2 [AND/OR] ... Then H (CF=CFi) [1]

Dimana :

E1. . .En : fakta-fakta (evidence) yang ada

H : hipotesa atau konklusi yang dihasilkan

CF : tingkat keyakinan (certainty factor)

Penjelasan pada rumus persamaan satu merupakan pencocokan fakta atau pernyataan dimulai dari bagian kiri (IF) atau dengan kata lain penalaran dimulai dari fakta terlebih dahulu untuk menguji kebenaran. Metode ini sering disebut *data-driven* karena mesin inferensi menggunakan informasi yang ditentukan oleh pemakai untuk memindahkan ke seluruh jaringan dari logika „AND“ dan „OR“ sampai sebuah terminal ditentukan sebagai objek. Bila mesin inferensi tidak dapat menentukan objek maka akan meminta informasi lain. Aturan (*Rule*) dimana menentukan objek, membentuk lintasan (*path*) yang mengarah ke objek. Oleh karena itu, hanya satu cara untuk mencapai suatu objek adalah dengan memenuhi semua aturan.

$CF(H,E) = MB(H,E) - MD(H,E)$ [2]

Dimana :

CF (H, E) : Certainty Factor dari hipotesis H yang dipengaruhi oleh gejala (evidence) E. Besarnya CF berkisar antara -1 sampai 1. Nilai -1 menunjukkan ketidakpercayaan mutlak sedangkan nilai 1

menunjukkan kepercayaan mutlak.

MB (H, E) : ukuran kenaikan kepercayaan (measure of increased belief) terhadap hipotesis H yang dipengaruhi oleh gejala E.

MD (H, E) : ukuran kenaikan ketidakpercayaan (measure of increased disbelief) terhadap hipotesis H yang dipengaruhi oleh gejala E. Bentuk dasar rumus certainty factor, adalah sebuah aturan
JIKA E MAKA H.

Dalam memberikan ukuran MB, MD, dan CF, tim MYCIN mempunyai parameter untuk menunjukkan ukuran kepercayaan nilai interpretasi untuk MB dan MD yang diberikan oleh MYCIN. *Certainty Factor* menggunakan ukuran kepercayaan (MB) dan ukuran ketidakpercayaan (MD) dalam mengkombinasikan beberapa *evidence* untuk menentukan nilai CF suatu hipotesis. Konsep ini kemudian di formulasikan ke dalam rumus berikut :

$$CF(H,e) = CF(E,e) * CF(H,E) \dots\dots\dots [3]$$

Dimana :

CF (H, e) : certainty factor hipotesis yang dipengaruhi oleh evidence e.

CF (E, e) : certainty factor evidence E yang dipengaruhi oleh evidence e.

CF (H, E) : certainty factor hipotesis dengan asumsi evidence

$$CF(E,e) = CF(H,E) \dots\dots\dots [4]$$

Dalam aplikasinya, CF(H,E) merupakan nilai kepastian yang diberikan oleh pakar terhadap suatu aturan, sedangkan CF(E,e) merupakan nilai kepercayaan yang diberikan oleh pengguna terhadap gejala yang dialaminya, (Nelly Astuti Hasibuan, 2017).

7. Studi Kasus Metode Certainty Factor

Sistem Pendukung keputusan pemilihan driver berbasis metode Certainty Factor, (Eris Dwi Septiawan Rizal; 2019). Pada penelitian ini diusulkan sistem rekomendasi driver menggunakan metode *Certainty Factor* (CF). Dimana pengambilan data yang hampir mirip dengan aplikasi Sitara untuk dilakukan rekomendasi adalah data pengirim aplikasi Sitara. Dengan memesan aplikasi Sitara dan menentukan tempat penjemputan, posisi – posisi driver dapat terlihat. Saat dilakukan pemesanan salah satu dari driver – driver terdekat akan direkomendasi. Tahapan proses pembuatan sistem rekomendasi driver selanjutnya dapat dilakukan komputasi pengambilan jarak antara Pengirim dan Driver. Data yang kita dapatkan berupa data *latitude* dan *longitude* pada *Google maps*. Konsep ini kemudian diformulasikan dalam rumusan dasar sebagai berikut :

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

$$CF[H,E] = CF[H] * CF[E]$$

$$Cfcombine\ CF[H,E]_{1,2} = CF[H,E]_2 * [1 - CF[H,E]_1]$$

$$Cfcombine\ CF[H,E]_{old,3} = CF[H,E]_{old} + CF[H,E]_3 * [1 - CF[H,E]_{old}]$$

IF Latitude titik awal

AND Longitude titik awal

AND Latitude titik akhir

AND Longitude titik akhir

THEN Jarak Terdekat

Langkah pertama, app sitara menentukan CF untuk masing-masing kaidah sebagai berikut :

$$\text{CFsitara Latitude titik awal} = 0.4$$

$$\text{CFsitara Longitude titik awal} = 0.6$$

$$\text{CFsitara Latitude titik akhir} = 0.6$$

$$\text{CFsitara Longitude titik akhir} = 0.6$$

Misalkan cf driver memiliki nilai sebagai berikut :

$$\text{Latitude titik awal} = 0.2$$

$$\text{Longitude titik awal} = 0.4$$

$$\text{Latitude titik akhir} = 0.4$$

$$\text{Longitude titik akhir} = 0.4$$

Kaidah tersebut kemudian dihitung nilai Cfsitaranya dengan mengalikan Cfdriver dengan CF menjadi :

$$\text{CF[H,E]1} = \text{CF[H]1} * \text{CF[E]1}$$

$$= 0.4 * 0.2$$

$$= 0.08$$

$$\text{CF[H,E]2} = \text{CF[H]2} * \text{CF[E]2}$$

$$= 0.6 * 0.4$$

$$= 0.24$$

$$\text{CF[H,E]3} = \text{CF[H]3} * \text{CF[E]3}$$

$$= 0.6 * 0.4$$

$$= 0.24$$

$$\text{CF[H,E]4} = \text{CF[H]4} * \text{CF[E]4}$$

$$= 0.6 * 0.4$$

$$=0.24$$

Langkah yang terakhir adalah mengkombinasikan nilai CF dari kaidah.

Berikut adalah kombinasi CF[H,E] dengan CF[H,E] :

$$\text{CFcombine CF[H,E]1,2}$$

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

$$= 0.08 + 0.24 (1-0.08)$$

$$= 0.08 + 0.24 (0.92)$$

$$= 0.08 + 0.22$$

$$=0.3\text{old}$$

$$\text{CFcombine CF[H,E] old,3}$$

$$= \text{CF[H,E] old} + \text{CF[H,E]3} * (1 - \text{CF[H,E] old})$$

$$= 0.3 + 0.24 (1-0.3)$$

$$=0.3 + 0.24 (0.7)$$

$$=0.3 + 0.16$$

$$=0.46 \text{ old2}$$

$$\text{CFcombine CF[H,E]old2,4}$$

$$= \text{CF[H,E] old2} + \text{CF[H,E]4} * (1 - \text{CF[H,E] old2})$$

$$= 0.46 + 0.24 (1-0.46)$$

$$= 0.46 + 0.24 (0.54)$$

$$=0.46 + 0.12$$

$$= 0.58\text{old3}$$

$CF[H,E]old4 * 100 = 0.58 * 100 = 58\%$ Dengan demikian dapat dikatakan bahwa perhitungan *Certainty factor* memiliki persentase tingkat keyakinan 58%. Nilai $P(H)$ disini diinisialisasikan 0,00001 dan $P(H|E)$ didapat dari jarak / jarak max, jarak max atau jarak maksimal antara pengirim dan driver adalah 2 km atau 2000 m. Hasil yang akan keluar dari komputasi adalah hasil dengan angka yang benar mendekati 0, untuk menjadikan nilai tersebut menjadi mendekati 1, hasil harus dikurangi 1 kemudian dikalikan -1 sehingga mendapatkan hasil kebenaran dari nilai terbesar. Setelah nilai CF dari semua data didapatkan ambil nilai yang paling mendekati 1. Nilai CF yang memiliki jarak paling mendekati 1 berkemungkinan besar adalah driver yang berkemungkinan besar akan menyetujui aplikasi kemudian menjemput Pengirim (Eris Dwi Septiawan Rizal; 2019).

7. PHP (Hypertext Preprocessor)

PHP (*Hypertext Preprocessor*) adalah suatu bahasa scripting khususnya digunakan untuk web *development*. Karena sifatnya yang *server side scripting*, maka untuk menjalankan PHP harus menggunakan web server. PHP juga dapat diintegrasikan dengan HTML, *Javascript*, *Jquery*, *Ajax*. Namun, pada umumnya PHP lebih banyak digunakan bersamaan *file* bertipe HTML dengan menggunakan PHP anda bisa membuat *website powerful* yang dinamis dengan disertai manajemen *databasenya*. Selain itu juga penggunaan PHP yang sebagian besar dapat jalan di banyak *platform*, menjadi salah satu alasan kenapa anda harus menguasai PHP untuk menjadi *web development* yang hebat.

Dengan demikian kode program yang ditulis dalam PHP tidak akan terlihat oleh user sehingga keamanan halaman web lebih terjamin. PHP dirancang untuk membuat halaman web yang dinamis, yaitu halaman web yang dapat membentuk suatu tampilan berdasarkan permintaan terkini, seperti menampilkan isi basis data ke halaman web. Kelebihan yang dimiliki PHP, sebagai berikut :

- a. PHP berbasis *server side scripting*
- b. *Command line scripting* pada PHP
- c. PHP dapat membuat aplikasi desktop
- d. Digunakan untuk berbagai macam *platform* OS
- e. Mendukung berbagai macam web server
- f. *Object oriented* pemrograman atau procedural
- g. *Output file* PHP pada XHTML, HTML dan XML
- h. Mendukung banyak RDBMS (*database*)
- i. Mendukung banyak komunikasi
- j. Pengolahan teks yang sangat baik, (Anggia Dasa Putri, 2017).

8. Database MySQL

MySQL merupakan salah satu database kelas dunia yang sangat cocok bila dipadukan dengan bahasa pemrograman PHP. MySQL bekerja menggunakan bahasa SQL (*Structure Query Language*) yang merupakan bahasa standar yang digunakan untuk manipulasi *database*. MySQL dikembangkan oleh perusahaan swedia bernama MySQL AB yang pada saat ini bernama Oracle Data Solutions sekitar tahun 1994-1995, namun cikal bakal kodenya sudah ada sejak tahun 1979. Awalnya MySQL merupakan perusahaan pengembang *software* dan konsultan

database, dan saat ini MySQL sudah diambil alih oleh *Oracle Corp.* Pada umumnya perintah yang paling sering digunakan dalam MySQL adalah *Select* (Mengambil), *Insert* (Menambah), *Update* (Mengubah), dan *Delete* (Menghapus). Selain itu, SQL juga menyediakan perintah untuk membuat *database*, *field*, ataupun *index* untuk menambah atau menghapus data, (Anggia Dasa Putri, 2017).

9. XAMPP

Xampp merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai server yang berdiri sendiri (*localhost*) yang terdiri atas program *Apache HTTP Server*, *MySQL database* dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan *Perl*. Nama Xampp merupakan singkatan dari X (empat sistem operasi apapun), *Apache*, *MySQL*, *PHP* dan *Perl*. Program ini tersedia dalam *GNU General Public License* dan bebas. Xampp merupakan web server yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis. Bagian penting XAMPP yang biasa digunakan pada umumnya:

- a. *XAMPP Control Panel Application* berfungsi mengelola layanan (*service*) XAMPP. Seperti mengaktifkan layanan (*start*) dan menghentikan (*stop*) layanan.
- b. *Htdoc* adalah folder tempat meletakkan berkas- berkas yang akan dijalankan. Di Windows, folder ini berada di *C:/xampp*
- c. *PhpMyAdmin* merupakan bagian untuk mengelola *database*, (Anggia Dasa Putri, 2017).

10. UML (*Unified Modelling Language*)

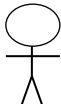
Unified Modeling Language (UML). UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem.

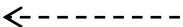
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

1. *Use case* Diagram

Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *Use Case* Diagram yaitu :

Tabel II.1. Simbol *Use Case* Diagram

Gambar	Keterangan
	<i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i> .
	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i> , tetapi tidak memiliki control terhadap <i>use case</i> .
	Asosiasi antara aktor dan <i>use case</i> , digambarkan

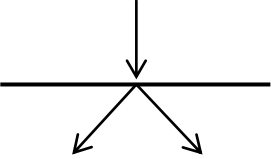
	dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i> , merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

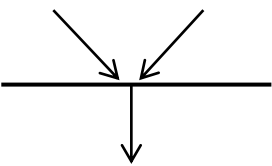
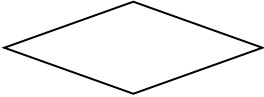
(Sumber : Julianto Simatupang; 2019)

2. Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram*, yaitu :

Tabel II.2. Simbol *Activity Diagram*

Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/ kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan paralel menjadi satu.

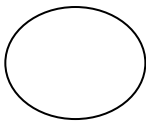
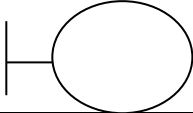
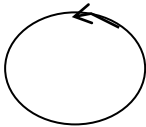
	<i>Join</i> (penggabungan) atau <i>rake</i> , digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
	<i>Swimlane</i> , pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.

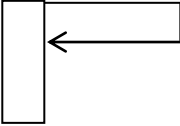
(Sumber : Julianto Simatupang; 2019)

3. Diagram Urutan (*Sequence Diagram*)

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram*, yaitu :

Tabel II.3. Simbol *Sequence Diagram*

Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan <i>formentry</i> dan <i>form</i> cetak
	<i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.

	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> , <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

(Sumber : Julianto Simatupang; 2019)

4. Class Diagram (Diagram Kelas)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/ Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau *cardinality*.

Tabel II.4. Multiplicity Class Diagram

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber : Julianto Simatupang; 2019)