BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terdahulu

Penelitian terdahulu yang dilakukan oleh Ayuningsih dan Hasibuan pada tahun 2018 dengan judul Sistem Pakar Mendiagnosa Kerusakan Pada Mesin Penggilingan Padi Menggunakan Metode *Naive Bayes*, penelitian Ayuningsih dan Hasibuan menggunakan objek penelitian yaitu mesin penggiling padi, gejala kerusakan dan solusi perbaikan, pembuatan aplikasi menggunakan pemrograman *android* dan basis data SQLite.

Penelitian terdahulu yang dilakukan oleh Kustini, dkk pada tahun 2021 dengan judul Sistem Pakar *Naive Bayes* Pada Pendeteksi Kerusakan Perangkat *Electrocardiograph* (ECG), penelitian Kustini, dkk menggunakan objek penelitian yaitu perangkat *Electrocardiograph* (ECG), gejala kerusakan dan solusi perbaikan, pembuatan aplikasi menggunakan pemrograman Matlab.

Penelitian terdahulu yang dilakukan oleh Suleman, dkk pada tahun 2018 dengan judul Sistem Pakar Untuk Mendeteksi Kerusakan Komputer Dengan Metode *Naïve Bayes*, penelitian Suleman, dkk menggunakan objek penelitian yaitu perangkat komputer, gejala kerusakan dan solusi perbaikan, pembuatan aplikasi menggunakan pemrograman *android*dan basis data SQLite.

Penelitian terdahulu yang dilakukan oleh Nasa'i dan Dewi pada tahun 2020 dengan judul Sistem Pakar Mendeteksi Kerusakan Sepeda Motor Menggunakan Metode *Naïve Bayes* di Bengkel Motor SMK Islam Tanjung, penelitian Nasa'i

dan Dewi menggunakan objek penelitian yaitu kerusakan sepeda motor, gejala kerusakan dan solusi perbaikan, pembuatan aplikasi menggunakan pemrograman *web* dan basis data MySQL.

Penelitian terdahulu yang dilakukan oleh Subqi dan Anggraini pada tahun 2021 dengan judul Data Mining untuk Pemeliharaan Prediktif Mesin Produksi Berdasarkan *Database* Kerusakan Mesin Menggunakan *Naïve Bayes Classifer*, penelitian Nasa'i dan Dewi menggunakan objek penelitian yaitu kerusakan mesin produksi, gejala kerusakan dan solusi perbaikan, pembuatan aplikasi menggunakan pemrograman *rapidminer* dan *microsoft office*.

II.2. Landasan Teori

Landasan teori dikutip dari beberapa jurnal penelitian terdahulu yang berkaitan dengan penelitian ini.

II.2.1. Sistem Pakar

Sistem pakar adalah suatu sistem yang dirancang unutk dapat menirukan keahlian seorang pakar dalam menjawab pertanyaan dan memecahkan suatu masalah. Sistem pakar akan memberikan pemecahan suatu masalah yang di dapat dari dialog dengan pengguna. Dengan bantuan sistem pakar seorang yang bukan ahli dapat menjawab pertanyaan, menyelesaikan serta mengambil keputusan yang biasanya di ambil oleh seorang pakar. (Nasa'i dan Dewi, 2020: 9).

Sistem pakar adalah salah satu cabang AI (*Artificial Intelegence*) yang membuat penggunaan secara luas *knowledge* yang khusus untuk penyelesaian masalah tingkat manusia oleh seorang pakar dan dirancang untuk dapat menirukan keahlian seorang pakar dalam menjawab pertanyaan dan menyelesaikan permasalahan disemua bidang. Tterdapat empat buah komponen dasar dari sebuah sistem pakar yaitu:

1. Basis Pengetahuan (*Knowledge Base*),
2. Mesin Inferensi (*Inference Engine*),
3. Antarmuka Pemakai (*User Interface*),
4. Mesin Pengembangan (*Development Engine*). (Kustini, dkk, 2021: 2).

II.2.1.1. Kelebihan Sistem Pakar

Sistem pakar memiliki beberapa fitur menarik yang merupakan kelebihannya, seperti:

1. Meningkatnya Ketersediaan (*increased availability*).

   Kepakaran atau keahlian menjadi tersedia dalam sistem komputer. Dapat dikatakan bahwa sistem pakar merupakan produksi kepakaran secara masal (*massproduction*).
2. Mengurangi Biaya (*reduced cost*).

   Biaya yang diperlukan untuk menyediakan keahlian per satu orang *user* menjadi berkurang.
3. Mengurangi Bahaya (*reduced danger*).

   Sistem pakar dapat digunakan di lingkungan yang mungkin berbahaya bagi manusia.

4. Permanen (*permanence*).

   Sistem pakar dan pengetahuan yang terdapat di dalamnya bersifat lebih permanen dibandingkan manusia yang dapat merasa lelah, bosan, dan pengetahuannya hilang saat sang pakar meninggal dunia.

5. Keahlian *Multiple* (*multiple expertise*).

   Pengetahuan dari beberapa pakar dapat dimuat ke dalam sistem dan bekerja secara simultan dan kontinyu menyelesaikan suatu masalah setiap saat. Tingkat keahlian atau pengetahuan yang digabungkan dari beberapa pakar dapat melebihi pengetahuan yang digabungkan dari beberapa pakar dapat melebihi pengetahuan satu orang pakar.

6. Meningkatkan Kehandalan (*increased reliability*).

   Sistem pakar meningkatkan kepercayaan dengan memberikan hasil yang benar sebagai alternatif pendapat dari seorang pakar atau sebagai penengah jika terjadi konflik antara beberapa pakar. Namun hal tersebut tidak berlaku jika sistem dibuat oleh salah seorang pakar, sehingga akan selalu sama dengan pendapat pakar tersebut kecuali jika sang pakar melakukan kesalahan yang mungkin terjadi pada saat tertekan atau stres.

7. Penjelasan (*explanation*).

   Sistem pakar dapat menjelaskan detail proses penalaran (*reasoning*) yang dilakukan hingga mencapai suatu kesimpulan. Seorang pakar mungkin saja terlalu lelah, tidak bersedia atau tidak mampu melakukannya setiap waktu. Hal ini akan meningkatkan tingkat kepercayaan bahwa kesimpulan yang dihasilkan adalah benar.

8. Respon Yang Cepat (*fast response*).

   Respon yang cepat atau *real time* diperlukan pada beberapa aplikasi. Meskipun bergantung pada *hardware* dan *software* yang digunakan, namun sistem pakar relatif memberikan respon yang lebih cepat dibandingkan seorang pakar.

9. Stabil

   Yang lengkap setiap saat (*steady, unemotional, and complete response at all times*). Karakteristik ini diperlukan pada situasi *real-time* dan keadaan darurat (*emergency*) ketika seorang pakar mungkin tidak berada pada kondisi puncak disebabkan oleh stress atau kelelahan.

10. Pembimbing Pintar (*intelligent tutor*)

    Sistem pakar dapat berperan sebagai *intelligent tutor* dengan memberikan kesempatan pada *user* untuk menjalankan contoh program dan menjelaskan proses *reasoning* yang dilakukan.

11. Basis Data Cerdas (*intelligent database*)

    Sistem pakar dapat digunakan untuk mengakses basis data secara cerdas (Rika Rosnelly, 2012: 6).

II.2.1.2. Konsep Umum Sistem Pakar

Pengetahuan yang dimiliki sistem pakar direpresentasikan dalam beberapa cara. Salah satu metode yang paling umum digunakan adalah tipe *rule* menggunakan format *IF THEN*. Banyak sistem pakar yang dibangun dengan mengekspresikan pengetahuan dalam bentuk *rules*. Bahkan, pendekatan berbasis pengetahuan (*knowledgebased approach*) untuk membangun sistem pakar telah mematahkan pendekatan awal yang digunakan pada sekitar tahun 1950-an dan

1960-an yang menggunakan teknik penalaran (*reasoning*) yang tidak mengandalkan pengetahuan (Rika Rosnelly, 2012: 7).

II.2.1.3. Elemen Manusia Pada Sistem Pakar

Sistem pakar tidak lepas dari elemen manusia yang terkait di dalamnya. Personil yang terkait dengan sistem pakar ada 4 yaitu:

1. Pakar (*expert*)
2. Pembangun pengetahuan (*knowledge engineer*)
3. Pembangunan sistem (*system engineer*)
4. Pemakai (*user*)

Paling tidak terdapat dua komponen orang atau lebih yang berpartisipasi dalam pembangunan dan penggunaan sistem pakar, yakni sedikitnya seorang pembangun pengetahuan dan seorang pakar (Rika Rosnelly, 2012: 10).

II.2.1.4. Pakar

Pakar adalah seorang individu yang memiliki pengetahuan khusus, pemahaman, pengalaman, dan metode-metode yang digunakan untuk memecahkan persoalan dalam bidang tertentu (Rika Rosnelly, 2012: 10).

II.2.1.5. Pembangun/ Pembuat Pengetahuan

Pembuat pengetahuan memiliki tugas utama menerjemahkan dan merepresentasikan pengetahuan yang diperoleh dari pakar, baik berupa pengalaman pakar dalam menyelesaikan masalah maupun sumber terdokumentasi lainnya ke dalam bentuk yang bisa diterima oleh sistem pakar. Dalam hal ini pembangunan pengetahuan (*knowledge engineer*) menginterpretasikan dan merepresentasikan

pengetahuan yang diperoleh dalam bentuk jawaban-jawaban atas pertanyaan-pertanyaan yang diajukan pada pakar atau pemahaman, penggambaran analogis, sistematis, konseptual yang diperoleh dari membaca beberapa dokumen cetak seperti *text book*, jurnal, makalah, dan sebagainya. Kurangnya pengalaman *knowledge engineer* merupakan kesulitan utama dalam mengkonstruksi sistem pakar. Untuk mengatasi hal tersebut, perancang sistem pakar menggunakan *tools* komersial (Seperti pada editor-editor khusus maupun *logic debuggers*) dan usahanya akan dipusatkan pada pembangunan mesin inferensi (Rika Rosnelly, 2012: 11)

II.2.1.6. Pembangun/ Pembuat Sistem

Pembangunan sistem adalah orang yang bertugas untuk merancang antarmuka pemakai sistem pakar, merancang pengetahuan yang sudah diterjemahkan oleh pembangun pengetahuan ke dalam bentuk yang sesuai dan dapat diterima oleh sistem pakar dan mengimplementasikannya ke dalam mesin inferensi. Selain hal tersebut, pembangun sistem juga bertanggung jawab apabila sistem pakar akan diintegrasikan dengan sistem komputerisasi lain. Alat pembangun (*tool builder*) dapat dipakai untuk menyajikan atau membangun *tool* yang spesifik. Penjual (*vendor*) dapat memberikan *tool* dan saran, staf pendukung dapat memberikan saran dan bantuan secara teknis dalam proses pembangunan sistem pakar (Rika Rosnelly, 2012: 12).

II.2.1.7. Pengguna Sistem

Banyak sistem berbasis komputer mempunyai susunan pengguna tunggal. Hal ini berbeda jauh dengan sistem pakar yang memungkinkan mempunyai beberapa kelas pengguna. Pengguna mungkin tidak terbiasa dengan komputer dan mungkin pada domain masalah. Bagaimanapun juga, banyak solusi permasalahan menjadi lebih baik dan kemungkinan lebih murah dan keputusan yang cepat bila menggunakan sistem pakar. Pakar dan pembangun sistem harus mengantisipasi kebutuhan-kebutuhan pengguna dan membuat batasan-batasan ketika mendesain sistem pakar (Rika Rosnelly, 2012: 12).

II.2.1.8. Struktur Sistem Pakar

Komponen yang terdapat dalam struktur sistem pakar ini adalah *knowledge base* (*rules*), *inference engine*, *working memory*, *explanation facility*, *knowledge acquisition facility, user interface* (Rika Rosnelly, 2012)

1. *Knowledge Base* (Basis Pengetahuan)

Basis pengetahuan mengandung pengetahuan untuk pemahaman, formulasi, dan penyelesaian masalah. Komponen sistem pakar disusun atas dua elemen dasar, yaitu fakta dan aturan. Fakta merupakan informasi tentang objek dalam area permasalahan tertentu, sedangkan aturan merupakan informasi tentang cara bagaimana memperoleh fakta baru dari fakta yang telah diketahui.

2. *Inference Engine* (Mesin Inferensi)

Mesin inferensi merupakan otak dari sebuah sistem pakar dan dikenal juga dengan sebutan *control structure* (struktur kontrol) atau *rule interpreter* (dalam

sistem pakar berbasis kaidah). Komponen ini mengandung mekanisme pola pikir dan penalaran yang digunakan oleh pakar dalam menyelesaikan suatu masalah. Mesin inferensi disini adalah *processor* pada sistem pakar yang mencocokkan bagian kondisi dari *rule* yang tersimpan di dalam *knowledge base* dengan fakta yang tersimpan di *working memory*.

3. *Working Memory*

Berguna untuk menyimpan fakta yang dihasilkan oleh *inference engine* dengan penambahan parameter berupa derajat kepercayaan atau dapat juga dikatakan sebagai global *database* dari fakta yang digunakan oleh *rule-rule* yang ada.

4. *Explanation Facility*

Menyediakan kebenaran dari solusi yang dihasilkan kepada *user* (*reasoning chain*).

5. *Knowledge Acquistion Facility*

Meliputi proses pengumpulan, pemindahan dan perubahan dari kemampuan pemecahan masalah seorang pakar atau sumber pengetahuan terdokumentasi ke program *computer*, yang bertujuan untuk memperbaiki atau mengembangkan basis pengetahuan.

6. *User Interface*

Mekanisme untuk memberi kesempatan kepada *user* dan sistem pakar untuk berkomunikasi. Antar muka menerima informasi dari pemakai dan mengubahnya ke dalam bentuk yang dapat diterima oleh sistem. Selain itu antarmuka menerima informasi dari sistem dan menyajikannya ke dalam bentuk yang dapat dimengerti oleh pemakai (Rika Rosnelly, 2012: 15).

II.2.1.9. Karakteristik Sistem Pakar

Sistem pakar umumnya dirancang untuk memenuhi beberapa karakteristik umum berikut ini:

1. Kinerja sangat baik (*high performance*). Sistem harus mampu memberikan respon berupa saran (*advice*) dengan tingkat kualitas yang sama dengan seorang pakar atau melebihinya.
2. Waktu respon yang baik (*adequate respon time*). Sistem juga harus mampu bekerja dalam waktu yang sama baiknya (*reasonable*) atau lebih cepat dibandingkan dengan seorang pakar dalam menghasilkan keputusan. Hal ini sangat penting terutama pada sistem waktu nyata (*real-time*).
3. Dapat diandalkan (*good reliability*). Sistem harus dapat diandalkan dan tidak mudah rusak/ *crash*.
4. Dapat dipahami (*understandable*). Sistem harus mampu menjelaskan langkah-langkah penalaran yang dilakukannya seperti seorang pakar.

Fleksibel (*flexibility*). Sistem menyediakan mekanisme untuk menambah, mengubah, dan menghapus pengetahuan (Rika Rosnelly, 2012: 21).

II.2.2. Diagnosa

Diagnosa merupakan tahapan dan hasil dari diagnosis suatu penyakit yang diderita oleh penderita. Menurut Kamus Besar Bahasa Indonesia (KBBI), Diagnosis adalah penentuan jenis penyakit dengan cara memeriksa atau meneliti gejala-gejalanya. Secara umum diagnosa adalah upaya untuk mengetahui atau

mengindentifikasi suatu penyakit atau masalah kesehatan yang diderita atau dialami seseorang pasien atau penderita. Diagnosis adalah menentukan sebab malfungsi dalam situasi kompleks yang didasarkan pada gejalagejala yang teramati, diantaranya medis, elektronis, mekanis, dan diagnosis perangkat lunak. (Ayudia, 2020: 383).

II.2.3. Kerusakan

Kerusakan merupakan produk yang tidak sesuai dengan standar mutu yang telah direncanakan perusahaan diawal sebelum proses produksi berlangsung yang secara ekonomis tidak dapat diperbaiki menjadi produk selesai.Produk rusak yang timbul dari proses produksi harus diperhitungkan dengan benar, karena produk tersebut dapat mempengaruhi tujuan utama perusahaan yaitu memperoleh laba serta juga dapat berdampak kerugian pada perusahaan. Oleh karena itu, produk rusak dapat diperhitungkan dengan cara mengetahui karakteristik produk rusak terlebih dahulu. (Pratiwi, 2018: 3).

II.2.4. Mobil

Mobil merupakan salah satu alat transportasi yang sangat diminati dan banyak digunakan oleh masyarakat, karena dapat berpergian ke berbagai tempat dengan nyaman dan terlindungi dari cuaca buruk. (Anggraini, dkk, 2020: 1).

Kendaraan mobil merupakan salah satu alat transportasi yang banyak dibutuhkan oleh masyarakat karena mobil dapat menampung banyak orang dan juga cocok digunakan untuk bepergian jauh karena akan terhindar dari panas sinar

mataharidan hujan. Mobil merupakan salah satu kebutuhan primer, ini berlaku untuk masyarakat ekonomi kelas atas. Dimana mobil sekarang bukanlah barang langka, dan bagi kaum yang memiliki uang berlebih merupakan sebuah kebutuhan yang harus dipenuhi karena fungsi dari mobil sangatlah bermanfaat. (Setiadi, 2019: 247).

II.2.5. Metode *Naive Bayes*

Metode *Naive bayes* adalah pengklasifikasian dengan metode probabilitas dan statistik yang di kemukakan oleh ilmuwan Inggris Thomas Bayes, yaitu memprediksi peluang di masa depan berdasarkan pengalamandi masa sebelumnya sehingga dikenal sebagai *Teorema Bayes*. *Naive Bayes* memiliki beberapa kelebihan dan kekurangan. Persamaan teorema Bayes adalah:

$$P(ai|vj) = \frac{nc+m.p}{n+m} \quad \text{.....(1)}$$

Keterangan:

n : Jumlah term pada data latih dimana v = vj

nc : Jumlah term dimana v = vj dan a = aj

p : Probabilitas setiap kelas dalam data latih

m : Jumlah term pada data uji. (Ayuningsih dan Hasibuan, 2018: 372).

II.2.6. *Web*

Situs *web* merupakan salah satu media yang dapat digunakan untuk memberikan layanan informasi. *Web Browser* adalah aplikasi perangkat lunak yang digunakan untuk mengambil dan menyajikan sumber informasi *web*. Sumber informasi *web* diidentifikasikan dengan *Uniform Resource Identufuer* (URL) yang dapat terdiri dari halaman *web*, *video*, gambar, ataupun konten lainnya. (Choiri, dkk, 2021: 198).

II.2.6.1. Sejarah *Web*

Penemu situs *web* yaitu Sir Timothy John ¨Tim¨ Berners-Lee, sedangkan situs *web* yang tersambung dengan jaringan pertama kali muncul pada tahun 1991. Tujuan dari Tim ketika merancang situs *web* adalah untuk memudahkan tukar menukar dan memperbarui informasi pada sesama peneliti di tempat ia bekerja. Pada tanggal 30 April 1993, CERN yaitu suatu tempat dimana tim bekerja mengumumkan bahwa WWW dapat digunakan secara gratis oleh publik. Sejarah *web service* berada pada perkembangan *web*. Dalam perkembangannya *web* dibagi menjadi 3 yaitu, web 1.0, web 2.0 dan *web* 3.0. Pada masa web 1.0 dimana *web* pertama kali berawal dan masi statis karena pembuat *web* dan pengguna *web* hanya terjadi komunikasi satu arah. (Nurhayati, dkk, 2020: 3).

II.2.6.2. Perkembangan *Web*

Berkembangnya *web* lebih baik dimana *user* tidak hanya berinteraksi satu arah. Dimana pengguna dapat berkomunikasi dengan pengguna lainnya pada situs

seperti *facebook*, *twitter* dan lainlain, bahkan kita dapat mengupload video dan *sharing* di *youtube* pada situs *web* yang menjadi *platform*nya itu terjadi dikarenkan ada pelayanan pada sistem web. Pada *web* 3.0 perkembangan *web* lebih berkembang setelah *web* 2.0. konsep *web* 3.0 pertama kali diperkenalkan pada tahun 2001, yaitu saat penemu *world wide web* Berners-lee menulis sebuah artikel ilmiah yang menggambarkan *web* 3.0 dimana sebagai sarana bagi mesin untuk membaca halaman- halaman *web*. Hal ini mesin akan memiliki kemampuan untuk membaca halaman *web* yang manusia lakukan seperti saat ini. Dan ciri dari *web* 3.0 adalah adanya *trend* teknologi salah satunya API (*Application Programming Interface*) merupakan sebuah *interface* yang diimplementasikan dengan menggunakan *software* sehingga *softwer* dapat saling berinteraksi dengan *software* lain. Ketika kita memperhatikan di masa *web* 3.0 kita dapat melihat bahawa *web service* mulai ada dan menjadi trend teknologi. (Nurhayati, dkk, 2020: 3).

II.2.7. *Hyper Text Markup Language* (HTML)

HTML adalah "bahasa" untuk membuat dokumen yang bisa diakses lewat *web*. HTML memuat dua bagian yakni isi dan pemformat. Isi adalah materi yang akan disampaikan, sedangkan bagian pemformat berwujud "*tag*" untuk menjadikan isi bisa diterjemahkan sebagai paragraf, ganti baris, *heading*, *unordered list*, *ordered list*, *table*. Dokumen HTML merupakan dokumen teks biasa yang bisa dibuka dengan pengedit teks seperti Notepad. Bila dibuka dengan Notepad akan terlihat bahwa yang tersaji di *web browser* sebenarnya sudah ada di sana. Tetapi di Notepad terlihat ada tambahan di sana-sini berupa "*tag*" sehingga di *browser*

terlihat seperti itu. Anatomi dokumen HTML terdiri dari dua bagian, yakni HEAD dan BODY. Bagian HEAD biasanya memuat informasi tentang atribut dari dokumen itu. Sedangkan bagian BODY memuat informasi utama yang akan dikomunikasikan lewat *web*. Adanya link dalam dokumen HTML menjadikan dokumen ini bisa menunjuk ke dokumen-dokumen HTML lain baik di folder yang sama, di *server* yang sama, maupun di *server* lain. Inilah alasan mengapa dokumen ini disebut hiperteks. Satu dokumen HTML juga bisa "memuat" dan menampilkan *file-file* lain seperti gambar, audio, Word, PDF, dan sebagainya. (Huzaeni, dkk, 2019: 140).

Hypertext Markup Language atau HTML adalah bahasa yang digunakan pada dokumen *web*. Dengan kata lain, HTML merupakan bahasa pemrograman berupa tagtag yang digunakan untuk membuat dan mengatur penampilan dari halaman *website*. (Irmaniawan, 2020: 44).

II.2.7.1. Perkembangan *Hyper Text Markup Language* (HTML)

HTML pertama kali diperkenalkan pada tahun 1990-an. Tim Berners-Lee pada tahun 1989 menciptakan HTML sederhana namun sangat efektif untuk pengkodean dokumen elektronik. *Web browser* pada zaman itu digunakan untuk membuka dokumen-dokumen dengan format HTML. Pada saat tahun 90-an inilah yang menjadi sejarah lahirnya HTML sehingga dinamakan HTML versi 1.0. Sebelum versi HTML yang terbaru keluar, ada proses panjang harus melalui persetujuan dari W3C (*World Wide Web Consortorium*) dengan evaluasi yang ketat. Dengan adanya seperti ini, setiap ada perkembangan versi

terbaru dari HTML bisa dipastikan ada update dan fitur baru dari versi sebelumnya. Sampai saat ini versi HTML yang terbaru sudah sampai HTML veris 5.0. Dalam kegiatan pelatihan ini, tim dosen pengabdi mengingatkan para peserta untuk memastikan terlebih dahulu apakah peseta sudah memiliki text editor pada komputer yang gunakan. Biasanya Windows memiliki text editor bawaan bernama "Notepad". Jika mau menggunakan text editor lain juga bisa tergantung keperluan saja. Selain itu pastikan sudah terinstall Web Browser untuk melihat hasil dari script yang Andatuliskan. (Sinaga, dkk, 2021: 52).

II.2.8. *Hypertext Preprocessor* (PHP)

PHP adalah bahasa standar yang digunakan dalam *website*. PHP adalah bahasa program yang berbentuk skrip yang diletakan di dalam *server web*. (Irawan, 2020: 46).

PHP singkatan dari *Hypertext PreProcessor* merupakan bahasa pemogram yang skript bersifat *Open Source*. Program ini bersifat *server side*, artinya tanpa adanya *server* yang berjalan disisinya script program PHP tidak dapat dijalankan PHP adalah *script* yang ditanamkan dalam HTML untuk membuat halaman *website* dinamis yang berkerja secara otomatis dan berfungsi sebagai pengolahan data pada *server* di mana *script* tersebut di jalankan. (Huzaeni, 2019: 140).

II.2.8.1. Perkembangan *Hypertext Preprocessor* (PHP)

PHP pertama kali dikembangkan oleh Rasmus Lerdorf pada tahun 1994. Pada awalnya PHP merupakan kependekan dari *Personal Home Page* (Situs

personal). Pada waktu itu PHP masih bernama *Form Interpreted* (FI), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dari *web*. Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI. Dengan perilisan kode sumber ini menjadi sumber terbuka, maka banyak pemrogram yang tertarik untuk ikut mengembangkan PHP. Pada November 1997, dirilis PHP/FI 2.0. Pada rilis ini, *interpreter* PHP sudah diimplementasikan dalam program C. Dalam rilis ini disertakan juga modul-modul ekstensi yang meningkatkan kemampuan PHP/FI secara signifikan. Pada tahun 1997, sebuah perusahaan bernama Zend menulis ulang interpreter PHP menjadi lebih bersih, lebih baik, dan lebih cepat. Kemudian pada Juni 1998, perusahaan tersebut merilis *interpreter* baru untuk PHP dan meresmikan rilis tersebut sebagai PHP 3.0 dan singkatan PHP diubah menjadi akronim berulang PHP*: Hypertext Preprocessor*. Pada pertengahan tahun 1999, Zend merilis *interpreter* PHP baru dan rilis tersebut dikenal dengan PHP 4.0. PHP 4.0 adalah versi PHP yang paling banyak dipakai pada awal abad ke-21. Versi ini banyak dipakai disebabkan kemampuannya untuk membangun aplikasi *web* kompleks tetapi tetap memiliki kecepatan dan stabilitas yang tinggi. Pada Juni 2004, Zend merilis PHP 5.0. Dalam versi ini, inti dari interpreter PHP mengalami perubahan besar. Versi ini juga memasukkan model pemrograman berorientasi objek ke dalam PHP untuk menjawab perkembangan bahasa pemrograman ke arah paradigma berorientasi objek. Peladen *web* bawaan ditambahkan pada versi 5.4 untuk mempermudah pengembang menjalankan kode PHP tanpa menginstal peladen perangkat lunak.

Versi terbaru dan stabil dari bahasa pemograman PHP saat ini adalah versi 8.0.(Huzaeni, 2019: 140).

II.2.9. *My Structure Query Language* (MySQL)

MYSQL merupakan salah satu database yang bersifat *open source* yang populer di dunia, dimana saat ini digunakan lebih dari 100 juta penguna diseluruh dunia MYSQL turunan dari SQL (*Structured Query Languaqe*) yang sering digunakan oleh para programmer dalam pembuatan program. Sistem *database* MYSQL selain mudah memahami sintaks dan pengaksesan juga di dukung beberapa fitur seperti *Multithreaded*, *multi-user* dan SQL *database management system*. (Huzaeni, dkk, 2019: 141).

MySQL (*My Structured Query Language*) atau yang biasa dibaca mai-se-kuel adalah sebuah program pembuat dan pengelola *database* atau yang sering disebut dengan DBMS (*Database Management System*), sifat dari DBMS ini adalah *open source*. (Imaniawan, 2020: 45).

II.2.9.1. Perkembangan *My Structure Query Language* (MySQL)

MySQL pada awalnya diciptakan pada tahun 1979, oleh Michael "Monty" Widenius, seorang programmer komputer asal Swedia. Monty mengembangkan sebuah sistem *database* sederhana yang dinamakan UNIREG yang menggunakan koneksi *low-level* ISAM *database engine* dengan *indexing*. Pada saat itu Monty bekerja pada perusahaan bernama TcX di Swedia.TcX pada tahun 1994 mulai mengembangkan aplikasi berbasis *web*, dan berencana menggunakan UNIREG

sebagai sistem *database*. Namun sayangnya, UNIREG dianggagap tidak cocok untuk *database* yang dinamis seperti *web*. TcX kemudian mencoba mencari alternatif sistem *database* lainnya, salah satunya adalah mSQL (miniSQL). Namun mSQL versi 1 ini juga memiliki kekurangan, yaitu tidak mendukung *indexing*, sehingga performanya tidak terlalu bagus. Dengan tujuan memperbaiki performa mSQL, Monty mencoba menghubungi David Hughes (*programmer* yang mengembangkan mSQL) untuk menanyakan apakah ia tertarik mengembangkan sebuah konektor di mSQL yang dapat dihubungkan dengan UNIREG ISAM sehingga mendukung indexing. Namun saat itu Hughes menolak, dengan alasan sedang mengembangkan teknologi indexing yang independen untuk mSQL versi 2. Dikarenakan penolakan tersebut, David Hughes, TcX (dan juga Monty) akhirnya memutuskan untuk merancang dan mengembangkan sendiri konsep sistem *database* baru. Sistem ini merupakan gabungan dari UNIREG dan mSQL (yang *source code*nya dapat bebas digunakan). Sehingga pada May 1995, sebuah RDBMS baru, yang dinamakan MySQL dirilis.David Axmark dari Detron HB, rekanan TcX mengusulkan agar MySQL di 'jual' dengan model bisnis baru. Ia mengusulkan agar MySQL dikembangkan dan dirilis dengan gratis. Pendapatan perusahaan selanjutnya di dapat dari menjual jasa "*support*" untuk perusahaan yang ingin mengimplementasikan MySQL. Konsep bisnis ini sekarang dikenal dengan istilah *Open Source*. Pada tahun 1995 itu juga, TcX berubah nama menjadi MySQL AB, dengan Michael Widenius, David Axmark dan Allan Larsson sebagai pendirinya. Titel "AB" di belakang MySQL, adalah singkatan dari "Aktiebolag", istilah PT (Perseroan Terbatas) bagi perusahaan Swedia.(Huzaeni, dkk, 2019: 14

II.2.10. *Unified Modeling Language*(UML)

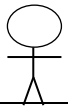
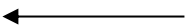
UML yaitu satu kumpulan konvensi permodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem perangkat lunak yang terkait dengan objek. UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan system yang besar dan kompleks. UML tidak hanya digunakan dalam proses pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan. (Andikos, 2019: 39).

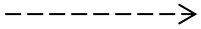
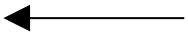
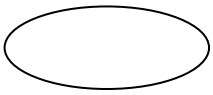
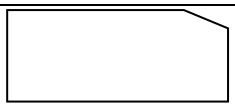
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasiskan UML adalah sebagai berikut:

1. *Use Case Diagram*

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah "apa" yang diperbuat sistem, dan bukan "bagaimana". Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case diagram* dapat digambarkan dengan sumber-sumber pada Tabel II.1.

Tabel II.1. Simbol *Use Case*

Gambar	Nama	Keterangan
	*Actor*	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan *use case*.
	*Depedency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (*independent*) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*independent*).
	*Generalization*	Hubungan dimana objek anak (*descendent*) berbagi perilaku dan

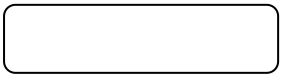
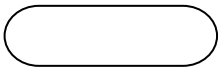
		struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
	*Include*	Menspesifikasikan bahwa *use case* sumber secara eksplisit.
	*Extend*	Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan.
	Association	Apa yang mnghubungkan antara objek satu dengan objek lainnya.
	*System*	Menspesifikan paket yang menampilkan sistem secara terbatas.
	*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu *actor*.
	*Collaboration*	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
	*Note*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

(Sumber:Andikos, 2019: 39)

2. Diagram Aktivitas (*Activity Diagram*)

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* dapat digambarkan dengan simbol-simbol seperti pada tabel II.2.

Tabel II.2. Simbol *Activity Diagram*

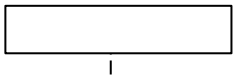
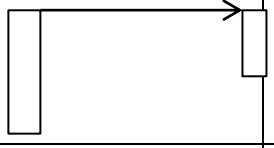
Gambar	Nama	Keterangan
	*Activity*	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
	*Action*	State dari sistem yang mencerminkan eksekusi dari suatu aksi.
	*Initial Node*	Bagaimana objek dibentuk atau diawali.
	*Activity Final*	Bagaimana objek dibentuk dan dihancurkan
	*Fork Node*	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

(Sumber:Andikos, 2019: 39)

3. Diagram Urutan (*Sequence Diagram*)

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa message yang digambarkan terhadap waktu. *Sequence Diagram* dapat digambarkan dengan simbol-simbol seperti pada Tabel II.3.

Tabel II.3. Simbol *Sequence Diagram*

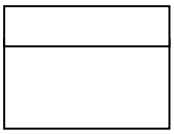
Gambar	Nama	Keterangan
	*Lifeline*	Objek entity, antarmuka yang saling berinteraksi.
	*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.
	*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.

(Sumber:Andikos, 2019: 39)

4. *Class Diagram* (Diagram Kelas)

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain. *Class diagram* dapat digambarkan dengan simbol-simbol seperti pada Tabel II.4.

Tabel II.4. *Class Diagram*

Gambar	Nama	Keterangan
	Generalization	Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
	*Nary Association*	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
	*Class*	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
	*Collaboration*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
	Realization	Operasi yang benar-benar dilakukan oleh suatu objek.
	Depedency	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (*independent*) akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri
	Assocation	Apa yang menghubungkan antara objek satu dengan objek lainnya

(Sumber:Andikos, 2019: 39)