

## BAB II: TINJAUAN PUSTAKA

### 2.1 Penelitian Terdahulu

Salah satu acuan penulis dalam melakukan penelitian untuk memperbanyak referensi dan teori dalam mengkaji penelitian adalah dengan mengacu pada penelitian terdahulu. Banyak penelitian tentang sistem pakar untuk diagnosis penyakit tanaman *hortikultura* namun tidak secara spesifik. Penulis hanya mengangkat penelitian dengan kasus yang berbeda namun metode yang digunakan hampir sama ataupun sebaliknya sebagai referensi dalam memperkaya bahan kajian. Berikut adalah penelitian terdahulu berupa jurnal atau skripsi terkait.

Salah satu jurnal terkait pada tahun 2019, yaitu berjudul “Sistem Pakar Untuk Diagnosa Hama dan Penyakit Pada Bunga Krisan Menggunakan *Forward Chaining*” Dengan penulis Hanip Afandi dan Danang Arbian Sulistyono[8]. Pada penelitian tersebut menggunakan metode yang sama, yaitu metode *forward chaining* untuk mendiagnosa hama dan penyakit pada bunga krisan yang menghasilkan kesimpulan bahwa Sistem dapat mendiagnosa hama dan penyakit pada bunga krisan melalui gejala-gejala yang dia alami dan diinputkan oleh para petani. Dan dalam melakukan diagnosa hama dan penyakit bunga krisan sistem akan memberikan pertanyaan gejala-gejala yang terdapat yang mungkin terjadi pada bunga krisan untuk diproses dengan runut maju (*forward chaining*) sehingga sistem akan memberikan informasi berupa hasil diagnosa[8].

Penelitian lainnya pada tahun 2019, yaitu berjudul “Penerapan Sistem Pakar untuk Diagnosa Penyakit dan Hama Tanaman Cabai dengan Metode *Forward Chaining*” Dengan penulis Maulida Ayu Fitriani dan Dany Candra Febrianto[5]. Pada penelitian tersebut menggunakan metode yang sama, yaitu metode *forward chaining* untuk mendiagnosa Penyakit pada cabai yang menghasilkan kesimpulan yaitu penerapan metode *Forward Chaining* untuk mendiagnosa Penyakit pada cabai sudah menggunakan sistem pakar untuk memudahkan memberikan informasi berupa kesimpulan tentang penyakit tanaman cabai sehingga hal ini bisa menjadi acuan[5].

Penelitian lainnya pada tahun 2019, yaitu berjudul “Penerapan Metode *Forward Chaining* Dalam Sistem pakar Untuk Diagnosa Hama dan Penyakit Tanaman Tomat” Dengan penulis Muhammad Arif Yulianto dan Hartatik[12]. Pada penelitian tersebut menggunakan metode yang sama, yaitu metode *forward chaining* untuk mendiagnosa hama dan penyakit pada tanaman tomat yang menghasilkan kesimpulan Secara umum metode *forward chaining* berhasil di implementasikan di dalam sistem pakar untuk diagnosa hama dan penyakit tomat[12].

Penelitian lainnya pada tahun 2019, yaitu berjudul “Implementasi Metode *Forward Chaining* Untuk Mendiagnosis Organisme Pengganggu Tanaman (Opt) Kopi” Dengan penulis Andy Victor Pakpahan dan Doni[9]. Pada penelitian tersebut menggunakan metode yang sama, yaitu metode *forward chaining* untuk mendiagnosis organisme pengganggu tanaman kopi yang menghasilkan kesimpulan implementasi metode *forward chaining* dapat membantu para petani kopi dilapangan untuk bisa melakukan diagnosis secara mandiri tanpa harus berkonsultasi langsung dengan pakar tanaman kopi[9].

Penelitian lainnya pada tahun 2020, yaitu berjudul “Sistem Pakar Penentuan Gaya Belajar Siswa Dengan Metode *Forward Chaining* Berbasis *Web*” Dengan penulis Laelia Puti Aditasari , Mega Novita dan Rahmat Robi Waliyansyah[10]. Pada penelitian tersebut menggunakan metode yang sama, yaitu metode *forward chaining* untuk membantu dan memberikan kemudahan guru dalam mengetahui gaya belajar visual, *auditori*, *kinestetik*, *verbal*, *logis*, *interpersonal*, dan *intrapersonal* dari tiap-tiap siswa supaya proses belajar lebih baik dalam hal pengerjaan soal-soal tes dan pembelajaran yang dilakukan oleh siswa serta pengajaran yang diberikan oleh guru menjadi lebih baik[10].

Tabel 2.1 Penelitian Sebelumnya

No	Penulis	Tahun	Judul	Metode	Persamaan	Perbedaan	Kesimpulan
1.	Hanip afandi, Danang Arbian Sulistyo	2019	Sistem Pakar Untuk Diagnosa Hama dan Penyakit Pada Bunga Krisan Menggunakan <i>Forward Chaining</i>	Metode <i>forward chaining</i>	Metode yang digunakan	Objek penelitian	Berdasarkan hasil perancangan dan pembahasan pada perancangan sistem pakar diagnosa hama dan penyakit pada bunga krisan maka dapat disimpulkan bahwa Sistem dapat mendiagnosa hama dan penyakit pada bunga krisan melalui gejala-gejala yang dia alami dan diinputkan oleh para petani. Dan dalam melakukan diagnosa hama dan penyakit bunga krisan sistem akan memberikan pertanyaan gejala-gejala yang terdapat yang mungkin terjadi pada bunga krisan untuk diproses dengan runut maju ( <i>forward chaining</i> ) sehingga sistem akan

							memberikan informasi berupa hasil diagnosa [8].
2	Maulida Ayu Fitriani, Dany Candra Febrianto	2019	Penerapan Sistem Pakar untuk Diagnosa Penyakit dan Hama Tanaman Cabai dengan Metode <i>Forward Chaining</i>	Metode pengembangan sistem pakar ( <i>Expert System Development Life Cycle</i> )	Metode <i>Forward Chaining</i> untuk mendiagnosa	Objek penelitian	Dari hasil penelitian yang telah dilakukan dapat diambil kesimpulan yaitu penerapan metode <i>Forward Chaining</i> untuk mendiagnosa Penyakit pada cabai sudah menggunakan sistem pakar untuk memudahkan memberikan informasi berupa kesimpulan tentang penyakit tanamann cabai sehingga hal ini bisa menjadi acuan [5].
3.	Muhammad Arif Yulianto, Hartatik	2019	Penerapan Metode <i>Forward Chaining</i> Dalam Sistem pakar Untuk Diagnosa Hama dan Penyakit Tanaman Tomat	Metode <i>forward chaining</i>	Metode yang digunakan	Objek penelitian	Secara umum metode <i>forward chaining</i> berhasil di implementasikan di dalam sistem pakar untuk diagnosa hama dan penyakit tomat. Hasil pengujian

							diagnosa sistem dengan diagnosa pakar menyimpulkan bahwa sistem layak digunakan dengan tingkat akurasi 100%. Sistem dapat memberikan hasil diagnosa hama dan penyakit berdasarkan masukan gejala <i>user</i> [12].
4.	Andy Victor Pakpahan, Doni	2019	Implementasi Metode <i>Forward Chaining</i> Untuk Mendiagnosis Organisme Pengganggu Tanaman (Opt) Kopi	Metode pengembangan sistem <i>Software Development Life Cycle</i> (SDLC)	Metode yang digunakan	Objek penelitian	Dari hasil pengujian yang dilakukan terhadap aplikasi web yang dibangun, implementasi metode <i>forward chaining</i> dapat membantu para petani kopi dilapangan untuk bisa melakukan diagnosis secara mandiri tanpa harus berkonsultasi langsung dengan pakar tanaman kopi. Selain itu, sistem pakar yang dibangun dapat diandalkan sebagai media edukasi

							budidaya serta pengendalian serangan OPT kopi [9].
5.	Laelia Puti Aditasari , Mega Novita , Rahmat Robi Waliyansyah	2020	Sistem Pakar Penentuan Gaya Belajar Siswa Dengan Metode <i>Forward Chaining</i> Berbasis <i>Web</i>	Metode <i>Forward Chaining</i>	Metode yang digunakan	Objek penelitian	Aplikasi analisis perancangan sistem pakar penentuan gaya belajar siswa dirancang untuk membantu dan memberikan kemudahan guru dalam mengetahui gaya belajar visual, auditori, kinestetik, verbal, logis, interpersonal, dan intrapersonal dari tiap-tiap siswa supaya proses belajar lebih baik dalam hal pengerjaan soal-soal tes dan pembelajaran yang dilakukan oleh siswa serta pengajaran yang diberikan oleh guru menjadi lebih baik. Berdasarkan pengujian <i>Black Box</i> , aplikasi analisis perancangan sistem pakar penentuan gaya belajar siswa ini

							layak digunakan dengan persentase 100% dan dapat berfungsi dengan baik dan berdasarkan pengujian Validasi Ahli yang dilakukan ke tiga responden penguji. Aplikasi analisis perancangan sistem pakar penentuan gaya belajar siswa ini memiliki rata-rata 76% maka <i>software</i> sudah berjalan dengan baik [10].
--	--	--	--	--	--	--	---

Berdasarkan penelitian terdahulu yang meneliti menggunakan metode *forward chaining* menjelaskan bahwa metode *forward chaining* dapat bekerja maksimal dalam menentukan kesimpulan ketika permasalahan dimulai dari pengumpulan data atau fakta. Oleh karena itu pada penelitian ini menerapkan metode *forward chaining* karena untuk menentukan kesimpulan dari hama dan penyakit tanaman kangkung dengan pengumpulan gejala dari tanaman kangkung.

## 2.2 Dasar Teori

Dalam laporan tugas akhir ini digunakan beberapa teori yang diperlukan untuk mendukung kegiatan yang dilakukan. Beberapa landasan teori yang dikemukakan tersebut meliputi konsep dasar dan definisi yang berkaitan dengan perangkat yang digunakan sebagai faktor-faktor pendukung dalam melaksanakan tugas akhir ini.

### 2.2.1 Jenis-Jenis Kangkung

Berdasarkan jenis tempat tumbuhnya, kangkung dibedakan menjadi dua jenis, yaitu kangkung darat atau disebut kangkung Cina dan kangkung air yang tumbuh secara alami di perairan, seperti sawah, rawa atau parit [2].

#### 1) Kangkung Darat (*Ipomoea reptans*)

Kangkung darat umumnya tumbuh di dataran tinggi atau di tempat yang kering. Kangkung darat memiliki ciri-ciri yang membedakan dengan kangkung air, yaitu sebagai berikut :

- a) Warna bunga putih bersih.
- b) Batangnya lebih kecil dan bentuk daunnya lebih tipis serta lebih lunak.
- c) Batangnya bewarna putih kehijauan.
- d) Kangkung darat memproduksi biji lebih banyak daripada kangkung air sehingga proses budi dayanya umumnya menggunakan penyebaran biji [2].

#### 2) Kangkung Air (*Ipomoea aquatic*)

Kangkung air tumbuh di daerah yang basah dan berair, seperti sawah, rawa, sungai, parit, dan sebagainya. Kangkung air memiliki ciri-ciri sebagai berikut.

- a) Warna bunga putih kemerahan.
- b) Batangnya lebih besar dan daunnya lebih lebar dan sedikit agak keras.
- c) Kangkung air sedikit memproduksi biji sehingga cara menanamnya dengan stek pucuk batangnya [2].



### 2.2.2 Hama dan Penyakit Kangkung

Dalam budidaya sayuran kangkung sering kali terkendala dengan POPT (Petugas Pengendali *Organisme* Pengganggu Tumbuhan) yang menjadi kendalanya adalah hama dan penyakit yang dapat mempengaruhi pertumbuhan dan perkembangan tanaman kangkung.

Sebenarnya tanaman kangkung ini merupakan salah satu jenis tanaman yang tahan terhadap serangan hama dan penyakit. Akan tetapi, yang menyebabkan tanaman kangkung rentan terhadap hama dan penyakit adalah faktor iklim dan sanitasi lingkungan [13].

Ada beberapa jenis hama dan penyakit yang menyerang tanaman kangkung. Penulis mengemukakan 5 hama dan 2 jenis penyakit dalam sistem pakar ini sebagai berikut :

1) Hama Bekicot (*Achatina fulica*)

Hama yang satu ini sering menggerogoti dan merusak daun dan batang tanaman kangkung, sehingga menyebabkan batang dan daun tanaman menjadi busuk. Bekas gigitan bekicot akan menyebabkan tanaman menjadi layu, kemudian busuk. Pemberantasannya adalah dengan. Penyebab timbulnya hama bekicot adalah kebersihan di sekitar tanaman kangkung tidak terjaga, sehingga menimbulkan kondisi yang lembap dan menjadi area kesukaan bekicot [13].

2) Hama Ulat Grayak (*Spodoptera*)

Gejala yang timbul akibat serangan hama ini adalah daun menjadi berlubang dan di bagian pinggir daun menjadi bergerigi karena bekas gigitan. Pengendalian hama ini bisa dilakukan apabila terjadi over populasi, bisa menyemprotkan dengan *Insektisida Diazinon 60 EC*, dengan dosis sebesar 2 cc/liter air. Pada waktu membasmi hama, sebaiknya lahan dikeringkan terlebih dahulu selama 4-5 hari, kemudian diberi air kembali [13].

3) Hama Kutu Daun (*Aphidoidea*)

Gejala yang ditimbulkan akibat serangan kutu daun ini adalah tanaman menjadi kerdil dan daun melengkung, karena kutu daun senang menghisap cairan tanaman [13].

4) Hama Ulat Keket (*Agrius convolvuli*)

Ciri-ciri hama ini adalah warnanya hijau muda dengan garis menyilang kuning. Bagian daun yang terserang akan rusak dan berlubang [13].

5) Penyakit Karat Putih :

Penyakit ini disebabkan oleh jamur *Albugo Ipomoe panduratae*. Gejala yang terlihat adalah pada bagian permukaan daun yang lama-kelamaan akan muncul bercak putih, bersamaan dengan rusaknya kondisi daun [13].

6) Penyakit Bercak Daun :

Penyakit ini disebabkan oleh jamur *Cercospora bataciola* dan *Fusarium sp.* Gejala yang timbul adalah munculnya bercak kecokelatan hingga kehitaman pada daun, jika tanaman kangkung sudah terkena penyakit ini, kondisi daun kangkung akan menjadi rusak [13].

7) Penyakit yang disebabkan oleh Bakteri :

Bakteri adalah salah satu organisme yang dapat membusukkan daun, batang, dan akar tumbuhan. Bagian tumbuh-tumbuhan yang terkena serangan bakteri akan mengeluarkan lendir keruh, baunya sangat busuk, dan lengket jika disentuh. Tumbuhan yang membusuk lama-kelamaan akan mati [2].

8) Penyakit yang disebabkan oleh Virus :

Virus dapat menyerang tanaman dalam kondisi yang sehat. Penyakit yang disebabkan oleh virus cukup berbahaya bagi tanaman. Hal ini karena tanaman yang terkena virus dapat menular dan menyebar ke seluruh tumbuhan dengan cepat. Tanaman yang sudah terlanjur diserang akan sulit disembuhkan [2].

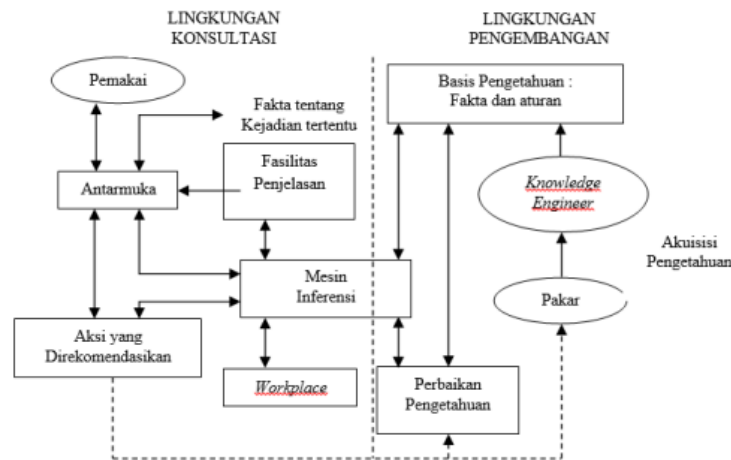
9) Penyakit yang disebabkan oleh Alga :

Keberadaan alga juga perlu diwaspadai karena dapat menyebabkan bercak karat merah pada daun. Bagian yang diserang oleh alga umumnya adalah daun. Tanaman yang terserang alga dapat dikenali dengan adanya bercak berwarna kelabu kehijauan pada daun. Kemudian, pada permukaannya tumbuh rambut berwarna coklat kemerahan. Meskipun alga berukuran kecil, apabila bercak yang ditimbulkan sangat banyak maka dapat merugikan tanaman [2].

### 2.2.3 Sistem Pakar

Sistem berbasis komputer yang dibangun berdasarkan fakta, pengetahuan, maupun penalaran yang dapat membantu menyelesaikan suatu permasalahan disebut sistem pakar [6]. Sistem pakar telah banyak digunakan memecahkan masalah di berbagai bidang, seperti bidang pertanian, perbankan dan pendidikan [5]. Tujuan Sistem Pakar adalah mentransfer kepakaran seorang pakar ke komputer, kemudian melanjutkannya dari komputer ke orang lain (yang bukan pakar), sedangkan Seorang pakar/ahli (*human expert*) adalah seorang individu yang memiliki kemampuan pemahaman yang superior dari suatu masalah [7].

Secara garis besar sistem pakar ini dirancang untuk melakukan diagnosa secara mandiri agar lebih awal mengetahui hama dan penyakit pada tanaman kangkung. Cara kerja sistem pakar ini dirancang untuk memasukkan informasi *user*, memilih ciri-ciri yang muncul pada tanaman kangkung. Kemudian data-data yang dimasukkan tadi akan diproses oleh mesin inferensi berdasarkan basis pengetahuan dari artikel, jurnal hama dan penyakit pada tanaman kangkung dan juga diperoleh dari metode wawancara narasumber yaitu bapak Irman Rudianto. Nantinya, hasil atau *output* yang diberikan sistem kepada *user* berisi diagnosis hama dan penyakit pada tanaman kangkung. Sistem pakar diagnosis ini bekerja menggunakan perhitungan mesin inferensi metode *forward chaining* berdasarkan ciri-ciri yang terlihat oleh *user* di tanaman kangkung. Arsitektur sistem pakar dapat dilihat pada Gambar 2.1.



**Gambar 2.1 Arsitektur Sistem Pakar[14]**

Di dalam membangun dan merancang sistem pakar diagnosis hama dan penyakit pada tanaman kangkung ini terdapat beberapa komponen wajib untuk menunjang fungsi kerja sistem pakar ini [15], yaitu :

### 1. Tampilan Antarmuka

Tampilan antarmuka adalah bagian visual dari sistem pakar. Tampilan antarmuka atau biasa disebut UI (*User Interface*) menampilkan semua informasi sistem pakar pada layar serta bertujuan untuk meningkatkan *usability* dan UX (*User Experience*).

### 2. Mesin Inferensi

Komponen ini biasa disebut sebagai otaknya sistem pakar, karena mengandung penalaran metodologi yang biasa dipakai oleh seorang pakar untuk menyelesaikan masalah. Penalaran yang dimaksud adalah informasi yang ada di dalam basis pengetahuan dan memori kerja untuk memformulasikan kesimpulan. Metodologi dalam mesin inferensi yang biasa digunakan ada dua macam yaitu metode maju (*forward Chaining*) dan metode mundur (*backward chaining*).

### 3. Basis Pengetahuan

Berisi pengetahuan, formulasi dan penyelesaian masalah. Komponen basis pengetahuan disusun oleh dua elemen yaitu fakta dan aturan. Fakta adalah informasi objek dalam permasalahan, sedangkan

aturan adalah bagaimana cara memperoleh fakta baru dari fakta yang telah diketahui.

#### 4. Memori kerja

Sebagai tempat menyimpan fakta-fakta hasil dari proses konsultasi untuk kemudian diproses oleh mesin inferensi berdasarkan basis pengetahuan untuk menentukan suatu keputusan.

##### **2.2.2.1 Kelebihan dan Kelemahan Sistem Pakar**

Sebagai sebuah sistem yang bekerja dengan cara mengadopsi kemampuan pengetahuan manusia ke dalam sebuah perangkat komputer dengan tujuan untuk menyelesaikan permasalahan layaknya seorang pakar pastinya mempunyai beberapa kelebihan dan kelemahan dalam proses pengambilan suatu keputusannya. Berikut adalah uraian kelebihan dan kelemahan dari suatu sistem pakar. Kelebihan sistem pakar, yaitu :

1. Membantu orang awam untuk menyelesaikan masalah 'tanpa' bantuan para pakar.
2. Meningkatkan kualitas dan produktivitas.
3. Mampu beroperasi dalam lingkungan yang berbahaya.
4. Memiliki kemampuan untuk mengakses pengetahuan dan keahlian para ahli baik yang biasa maupun yang langka.
5. Sebagai asisten para ahli sehingga meringankan pekerjaan para ahli.
6. Memiliki reliabilitas.
7. Dapat menghemat waktu dalam pengambilan keputusan [16].

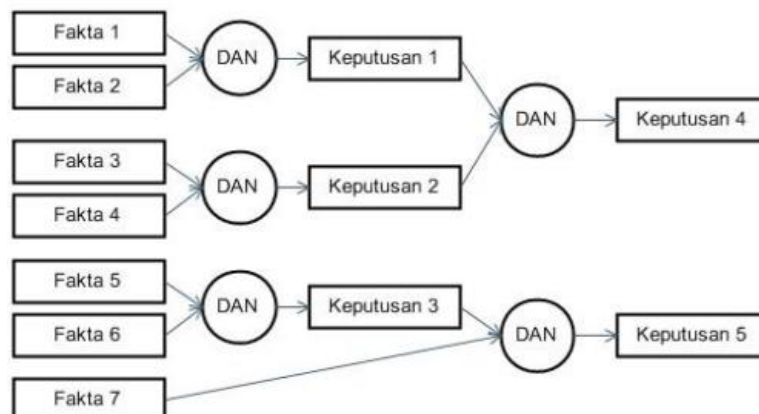
Adapun kekurangan dari sistem pakar adalah :

1. Tidak ada jaminan bahwa sistem pakar memuat 100% kepakaran yang diperlukan.

2. Pengembangan sistem pakar tergantung ada tidaknya pakar di bidangnya sehingga pengembangannya dapat terkendala.
3. Biaya untuk mendesain, mengimplementasikan dan memeliharanya dapat sangat mahal tergantung seberapa lengkap dan kemampuannya[16].

#### 2.2.2.2 *Forward Chaining*

Teknik yang digunakan dalam proses rancang bangun sistem pakar ini menggunakan metode *Forward Chaining*. *Forward Chaining* merupakan proses yang dimotori data (data driven) [17]. Dalam proses ini pelacakan dimulai dari informasi masukan dan selanjutnya mencoba menggambarkan kesimpulan. Pelacakan ke depan mencari fakta yang sesuai dengan bagian *IF* dari aturan *IF-THEN* [18]. “Dalam *forward chaining*, kaidah interpreter mencocokkan fakta dalam pangkalan data dalam situasi yang dinyatakan dalam bagian sebelah kiri atau kaidah *IF*. Bila fakta yang ada dalam pangkalan data itu sudah sesuai dengan kaidah *IF*, maka kaidah di eksitasi [16].



**Gambar 2.2 *Forward Chaining* [18].**

Pada saat konsultasi, dilakukan pengujian fakta-fakta yang dimasukkan pengguna adalah mesin inferensi *forward*

*chaining* [19][20], fakta-fakta tersebut adalah gejala dari serangan penyakit dan hama tanaman kangkung. Dalam sistem pakar ini mengimplementasikan pengetahuan atau keahlian seorang pakar untuk menjadikan sebuah kesimpulan [21][22].

## 2.3 **Tools Pembangun Sistem Pakar**

### 2.3.1 **XAMPP**

*XAMPP* adalah perangkat lunak yang penamaannya diambil dari akronim kata *Apache*, *MySQL/MariaDB*, *PHP*, dan *Perl*. Imbuhan huruf “X” berasal dari istilah *cross platform* sebagai simbol bahwa aplikasi ini bisa dijalankan di empat sistem operasi berbeda, seperti *OS Linux*, *OS Windows*, *Mac OS*, dan juga *Solaris* [23].

Keuntungan penggunaan *XAMPP* adalah pengganti peran *web hosting* dimana file-file pembangun *web* disimpan didalam *hosting* lokal (*localhost*) agar bisa ditampilkan di *browser*. Bagian-bagian penting di dalam *XAMPP* yaitu :

1. *Htdocs* adalah folder atau direktori untuk menyimpan *files* pembangun *web* seperti *HTML*, *PHP*, dan file lainnya.
2. *PhpMyAdmin* adalah suatu *server* di komputer untuk manajemen basis data *MySQL*.
3. *Control Panel* adalah panel untuk mengontrol *start* atau *stop* layanan *service XAMPP*.

### 2.3.2 **HTML**

*HTML (Hyper Text Markup Language)* adalah bahasa *mark up*. *HTML* digunakan untuk *mark up* (penanda) terhadap suatu dokumen teks. Simbol *mark up* yang digunakan oleh *html* ditandai dengan tanda lebih kecil (<) dan tanda lebih besar (>). Kedua tanda ini disebut tag. Tag yang digunakan sebagai tanda penutup diberi karakter garis miring (</.>) [23].

### 2.3.3 **PHP**

*PHP (Hypertext Preprocessor)* merupakan bahasa *scripting* yang tergabung menjadi satu dengan *HTML* dan dijalankan pada *server side* atau semua perintah yang diberikan akan secara penuh dijalankan pada *server*.

Sedangkan yang dikirim ke klien (*browser*) hanya berupa hasilnya saja. Sintak dan semantik PHP memiliki kesamaan dengan bahasa pemrograman C, JAVA, PERL, dengan menambahkan tambahan tag `<?php` sebagai pembuka dan diakhiri dengan `?>`, pasangan kedua kode inilah yang berfungsi sebagai tag kode PHP. Berdasarkan tag inilah pihak server dapat memahami kode php dan kemudian memprosesnya, hasilnya dikirim ke browser [23].

#### 2.3.4 CSS

CSS (*Cascading Style Sheet*) adalah *stylesheet language* yang digunakan untuk mendeskripsikan penyajian dari dokumen yang dibuat dalam *mark up language*. CSS merupakan sebuah dokumen yang berguna untuk melakukan pengaturan pada komponen halaman web, inti dari dokumen ini adalah memformat halaman web standar menjadi bentuk web yang memiliki kualitas yang lebih indah dan menarik [23].

#### 2.3.5 MySQL

Beberapa pengertian MySQL menurut para ahli:

1. Menurut Raharjo (2011:21), “MySQL merupakan RDBMS (atau *server database*) yang mengelola *data base* dengan cepat menampung dalam jumlah sangat besar dan dapat di akses oleh banyak *user*”.
2. Menurut Kadir (2008:2), “MySQL adalah sebuah *software open source* yang digunakan untuk membuat sebuah *database*.”

Berdasarkan pendapat yang dikemukakan di atas dapat ditarik kesimpulan bahwa MySQL adalah suatu *software* atau program yang digunakan untuk membuat sebuah *database* yang bersifat *open source*.

Didalam *MySQL* terdapat suatu bahasa perintah atau *query* yaitu SQL yang merupakan singkatan dari *Structured Query Language*. Terdapat 3 klausa struktur dasar dari ekspresi *SQL*:

- a. *Select* digunakan untuk mendaftar semua atribut yang diinginkan sebagai hasil suatu *query*.
- b. *From* ini mencatat semua relasi yang dipindai dalam evaluasi suatu *query*.



- c. *Where* terdiri dari sebuah predikat yang menyangkut atribut dari relasi yang muncul dalam klausa *from*.  
Ekspresi dasar modifikasi data ada 3, yaitu :
  - a. *Delete* untuk menghapus data.
  - b. *Insert* untuk memasukkan data, dan
  - c. *Update* untuk mengganti data.

### 2.3.6 Visual Studio Code

*Visual Studio Code* adalah kode editor sumber yang dikembangkan oleh Microsoft untuk Windows, Linux dan macOS. Ini termasuk dukungan untuk *debugging*, kontrol git yang tertanam dan GitHub, penyorotan sintaksis, penyelesaian kode cerdas, *snippet*, dan *refactoring* kode. Ini sangat dapat disesuaikan, memungkinkan pengguna untuk mengubah tema, pintasan *keyboard*, preferensi, dan melakukan instalasi ekstensi yang menambah fungsionalitas tambahan.

### 2.3.7 StarUML

Perancangan sistem pakar ini memerlukan suatu metode atau langkah-langkah sistematis dari awal sampai akhir. Metode yang dipakai dapat menjelaskan dan mendokumentasikan langkah-langkah dalam pembangunan sistem pakar ini. StarUML adalah *software* yang digunakan penulis untuk menyelesaikan metode tersebut.

*Unified Modeling Language* (UML) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menjelaskan dan membangun perangkat lunak[24]. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem.

Ada beberapa komponen diagram yang dipakai penulis di dalam *starUML* dalam perancangan sistem pakar ini adalah sebagai berikut:

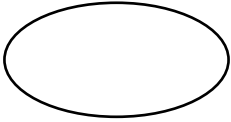



#### 1. *Use Case Diagram*

Sebuah *Use Case Diagram* menyatakan visualisasi interaksi yang terjadi antara pengguna (aktor) dengan sistem. Diagram ini bisa menjadi

gambaran yang bagus untuk menjelaskan konteks dari sebuah sistem sehingga terlihat jelas batasan dari sistem [25].

Di dalam *use case diagram* terdapat beberapa simbol-simbol yaitu :

Tabel 2.2 Simbol *Use Case Diagram*[25]




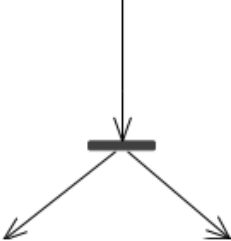
Simbol	Keterangan
	<p><i>Use Case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, yang dinyatakan dengan menggunakan kata kerja</p>
	<p><i>Actor</i> atau Aktor adalah <i>Abstraction</i> dari orang atau sistem yang lain yang mengaktifkan 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 kontrol terhadap <i>use case</i></p>
	<p>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 data.</p>
	<p>Asosiasi antara aktor, dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem</p>

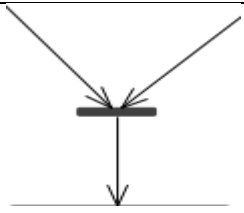
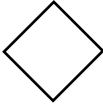
Pada tabel 2.2 Adalah simbol-simbol yang akan dipakai pada saat pembuatan *use case* diagram. Simbol simbol tersebut adalah *use case* menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, yang dinyatakan dengan menggunakan kata kerja, *Actor* atau Aktor adalah *Abstraction* dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem, Asosiasi antara aktor dan *use case*, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan Asosiasi antara aktor, dan *use case* yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.

## 2. *Activity Diagram*

*Activity Diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis[25]. Simbol-simbol yang digunakan dalam *activity diagram* yaitu:

Tabel 2.3 Simbol *Activity Diagram*[25]

Simbol	Keterangan
	<i>Start Point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktivitas
	<i>End Point</i> , akhir aktivitas
	<i>Activities</i> , menggambar kan suatu proses/kegiatan bisnis
	<i>Fork</i> atau percabangan, digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu


Simbol	Keterangan
	<i>Join</i> (penggabungan) atau <i>rake</i> , digunakan untuk menunjukkan adanya dekomposisi
	<i>Decision Points</i> , menggambar kan pilihan untuk pengambilan keputusan, <i>true</i> atau <i>false</i>



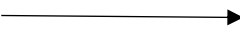
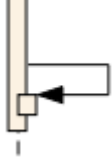


Pada tabel 2.3 Adalah simbol-simbol yang akan dipakai pada saat pembuatan *activity diagram*. Simbol simbol tersebut adalah *Start Point*, diletakkan pada pojok kiri atas dan merupakan awal aktivitas. *End Point*, akhir aktivitas. *Activities*, menggambar kan suatu proses/kegiatan bisnis. *Fork*/percabangan, digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabung kan dua kegiatan paralel menjadi satu. *Join* (penggabungan) atau *rake*, digunakan untuk menunjukkan adanya dekomposisi. *Decision Points*, menggambar kan pilihan untuk pengambilan keputusan, *true* atau *false*.

### 3. *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 2.4 Simbol *Sequence Diagram*[25]

Simbol	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

Simbol	Keterangan
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interfaces</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan <i>form entry</i> dan <i>form cetak</i>
	<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> , mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi pengaktifan sebuah operasi
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i>


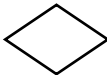
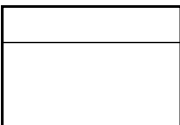
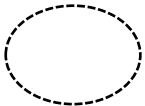
Pada tabel 2.4 Adalah simbol-simbol yang akan dipakai pada saat pembuatan *sequence diagram*. Simbol simbol tersebut adalah *Entity Class*, merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem. *Boundary Class*,

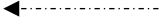
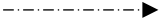
berisi kumpulan kelas yang menjadi *interfaces* atau interaksi antara satu atau lebih aktor dengan sistem. *Control class*, suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas. *Message*, simbol mengirim pesan antar *class*. *Recursive*, menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri. *Activation*, mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi pengaktifan sebuah operasi. Dan *Lifeline*, garis titik-titik yang terhubung dengan objek, sepanjang *lifeline* terdapat *activation*.

#### 4. Class Diagram

*Class Diagram* 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[25].

Tabel 2.5 Simbol *Class Diagram*

Gambar	Nama	Keterangan
	<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).
	<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
	<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
	<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor

Gambar	Nama	Keterangan
	<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (independent) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri

Pada tabel 2.5 Adalah simbol-simbol yang akan dipakai pada saat pembuatan *class diagram*. Simbol simbol tersebut adalah *Generalization*, *Nary Association*, *Class*, *Collaboration*, *Realization*, dan *Dependency*. Simbol-simbol tersebut mempunyai fungsi masing-masing yang sudah dijelaskan di tabel.

### 2.3.8 Confusion Matrix

Untuk mengukur kinerja suatu metode klasifikasi digunakan salah satu metode yaitu *Confusion Matrix* [26][27]. Pada pengukuran kinerja menggunakan *confusion matrix*, terdapat 4 (empat) sebutan sebagai perwakilan hasil proses klasifikasi. Keempat istilah tersebut adalah *True Positive* (TP), *True Negative* (TN), *False Positive* (FP) dan *False Negative* (FN). Nilai *True Negative* (TN) merupakan jumlah data negatif yang terdeteksi dengan benar, sedangkan *False Positive* (FP) merupakan data negatif namun terdeteksi sebagai data positif, pada dasarnya *confusion matrix* mengandung informasi yang membandingkan hasil klasifikasi yang dilakukan oleh sistem dengan hasil klasifikasi yang seharusnya [26][28]. Berikut adalah rumus dari *confusion matrix* (2.1) [29][30].

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \times 100\% \quad (2.1)$$

### **2.3.9 *Black-box Testing***

Setelah dilakukan pengukuran akurasi, langkah selanjutnya adalah membuat sebuah kegiatan untuk mengevaluasi kebenaran yaitu berupa pengujian kepada sistem yaitu *black-box testing*. *black-box testing* adalah pengujian perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program untuk mengetahui apakah fungsi, masukan dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan [31].