

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Penelitian terkait merupakan salah satu acuan bagi peneliti dalam melakukan penelitian dan dapat menjadi referensi bagi penelitian yang dilakukan saat ini untuk mengefisienkan dan mengefektifkan pengembangan ilmu pengetahuan. Adapun beberapa penelitian terkait dengan sistem pakar diagnosa penyakit mata diantaranya adalah:

Penelitian yang dilakukan oleh Dwi dewi prasetio dan kawan-kawan pada tahun 2021 dengan Judul penelitian “Diagnosa dini penyakit mata menerapkan metode *case based Reasoning*” yang didalam penelitiannya membahas tentang menganalisis dan merancang sistem pakar diagnosa penyakit dini pada mata yang mencakup informasi penyakit, baik gejala maupun solusinya[6].

Penelitian yang dilakukan Shinta Sindri Suntari dan kawan-kawan pada tahun 2022 dengan judul penelitian “rancang bangun aplikasi sistem pakar diagnosa penyakit mata berbasis web dengan metode *forward cahining* dan *case based reasoning (CBR)*”, tujuan dari penelitian ini untuk memeberikan informasi kepada para penderita penyakit mata yang masih awam tentang gejala dan penyakitnya guna menekan dan menurunkan resiko volume prevelensi gangguan penglihatan setiap tahunnya[7].

Penelitian yang dilakukan Gusti Ari Prasetyo, wiwien Hadikurnawati pada tahun 2021 dengan judul “sistem pakar menggunakan *metode case based raesoning (CBR)* untuk mendiagnosa penyakit kucing “dengan penelitiannya yang dilakukan yaitu membuat sistem pakar untuk mendiagnosa penyakit yang diderita oleh kucing tentang gejala apa saja yang dialami oleh kucing berguna untuk membantu pemilik kucing dan memahami memelihara kucing[8].

Penelitian yang dilakukan oleh Rizal Rachman, Sera Moritami pada tahun 2020 dengan judul penelitian “refraksi sistem pakar deteksi penyakit refraksi mata dengan menggunakan metode *teorema bayes* berbasis web” membahas tentang kelainan mata adalah gangguan penglihatan pada umumnya dimana cahaya tidak dapat di fokuskan dengan jelas untuk masuk ke mata[9].

Penelitian yang dilakukan oleh Norma Jaya Telaumbanua Nofriadi dan Ari Dermawan pada tahun 2022, dengan Judul “Sistem Pakar Untuk Mendeteksi Penyakit Mata, Menerapkan Metode *case based reasoning*” dalam penelitiannya membahas tentang penelitian dapat menemukan masalah-masalah yang terjadi dalam mendiagnosa gejala dari penyakit yang dialami pasien sehingga dapat membantu pasien dalam mengatasi penanganan penyakit mata[10].

2.2 Landasan Teori

2.2.1 Sistem pakar

Sistem pakar merupakan suatu sistem informasi yang didalamnya terdapat beberapa pengetahuan para pakar yang terkait suatu bidang dan cenderung spesifik. Sistem pakar merupakan kemampuan untuk merekomendasikan suatu rangkaian atau *behaviouruser* yang bisa mengoperasikan sistem koreksi yang tepat dan penglihatan yang sangat penting dalam kehidupan. Sistem pakar tersebut muncul pertama kali adalah *General purpose problem solver(GPS)* kemudian dikembangkan Newel dan Simon pada pertengahan 1960. Sistem tersebut juga menggunakan kapabilitas proses penalaran kemudian mencapai hasil kesimpulan yang maksimal berdasarkan data dan fakta yang ada [11].

2.2.2 Penyakit mata

Adapun jenis-jenis penyakit mata yaitu, katarak, glaukoma, konjungtivitis, pterigium, retinopati diabetik, degenerasi makula, dan Strabismus. Penyakit mata merupakan salah satu gangguan kesehatan yang sering muncul dalam kehidupan masyarakat. Efek dari itu bisa menyebabkan mata merah, gatal, perih, gangguan penglihatan sampai akhirnya buta. Mata merupakan organ pada manusia yang mampu mendeteksi cahaya didalam pupil dan diteruskan ke retina di belakang mata, kemudian retina mengubahnya menjadi implus saraf lalu di bawa ke otak melalui saraf.[11]

1. Katarak

Adalah gangguan penglihatan dimana lensa mata menjadikeruh dan seperti berawan sehingga pandangan menjadi kabur. Orang yang mengalami katarak akan merasa seperti selalu melihat kabut atau berasap. Katarak banyak dijumpai pada manusia yang sudah berumur 40 tahun keatas tetapi juga ada kasus katarak yang terjadi pada usia yang lebih mudah bahkan juga terdapat pada bayi yang baru lahir. Penuaan ini membuat protein pada mata mengumpal, katarak ini juga di sebabkan karena adanya penyakit diabetes, luka pada mata, pancaran sinar ultraviolet, sering merokok, juga efek samping dari obat-obatan misalkan kortikosteroid dan radiasi terapi.[12]

2. konjungtivitas

Konjungtivitis adalah mata merah akibat peradangan pada selaput yang melapisi permukaan bola mata dan kelopak mata bagian dalam mata. Selain mata merah, konjungtivitis dapat disertai rasa gatal pada mata dan mata berair[13]

3. Presbiopi

Menurunnya kemampuan mata melihat objek yang dekat secara bertahap. Kondisi ini terjadi secara alami sebagai bagian dari proses penuaan. Umumnya, seseorang baru menyadari bahwa dirinya menderita presbiopi ketika harus menjauhkan handphone atau buku agar bisa membacanya.[14]

4. Gleukoma

gangguan penglihatan yang disebabkan oleh kerusakan saraf mata. Biasanya, kerusakan saraf mata tersebut terjadi karena adanya tekanan tinggi pada bola mata.

Namun, ada beberapa kasus glaukoma yang terjadi walau tekanan pada bola matanya masih dalam batas normal sekalipun.[15]

2.2.3 Metode Case-Based-Reasoning (CBR)

Metode *Case Based Reasoning* merupakan metode untuk merancang sistem untuk mengambil keputusan berdasarkan kasus-kasus sebelumnya yang digunakan sebagai solusi untuk kasus baru. Beberapa masalah yang dihadapi dapat diselesaikan dengan menerapkan problem solving dari pengalaman terdahulu. *Case Based Reasoning* merupakan metode yang pemecahan masalahnya dengan menitik beratkan pada pengetahuan dari kasus-kasus sebelumnya. Apabila terdapat kasus baru maka kasus tersebut akan disimpan pada sistem, sehingga kedepan sistem akan melakukan learning dan knowledge agar menjadi pemecahan masalah berikutnya[16].

Tahapan-tahapan CBR :

1. Retrieve

Memperoleh kembali kasus yang paling g menyerupai/relevan (similar) dengan kasus yang baru. Bagian ini mengacu pada segi identifikasi, kemiripan awal, pencarian dan pemulihan serta eksekusi.

2. Reuse

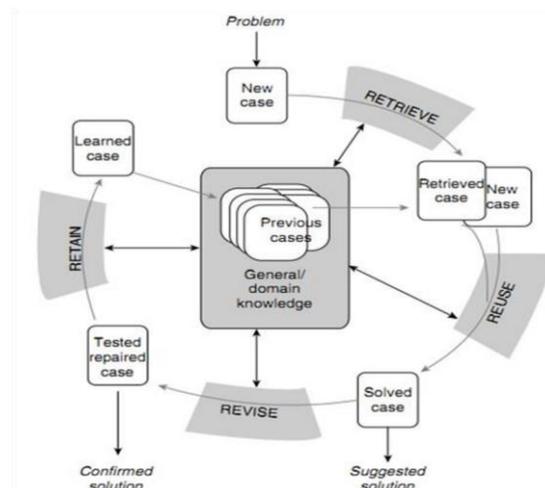
Menggunakan informasi dan pengetahuan dari kasus tersebut untuk memecahkan permasalahan. Proses reuse dari solusi kasus yang telah diperoleh dalam konteks baru difokuskan pada dua aspek yaitu perbedaan antara kasus yang sebelumnya dan yang sekarang, bagian apa dari kasus yang telah diperoleh yang dapat ditransfer menjadi kasus baru.

3. *Revise*

Meninjau kembali solusi yang diusulkan kemudian menegetesnya pada kasus nyata (simulasi dan jika di perlukan memperbaiki solusi agar cocok dengan kasus yang baru.

4. *Retain* (menyimpan)

Menyimpan menyimpan kasus baru yang telah berhasil mendapatkan solusi agar dapat digunakan oleh kasus-kasus selanjutnya yang mirip dengan kasus tersebut[17]



Gambar 2. 1 Proses Metode CBR

Berikut adalah rumus dari perhitungan bobot kemiripan (*similarity*):

$$\text{Similarity}(\text{problem}, \text{case}) = \frac{S_1 * W_1 + S_2 * W_2 + \dots + S_n * W_n}{W_1 + W_2 + \dots + W_n}$$

Keterangan :

$S = \textit{similarity}$ (nilai kemiripan) pada $\textit{similarity}$ jika terdapat kemiripan kasus akan bernilai 1, sedangkan tidak mirip maka akan bernilai 0.

$W = \textit{weight}$ (bobot yang diberikan)

Pembobotan ditentukan berdasarkan hasil pembelajaran atau pengamatan pada kasus. Semakin berpengaruh suatu gejala terhadap kasus, maka bobotnya semakin tinggi begitu pula sebaliknya.

Case Based Reasoning bekerja dengan melalui kasus baru berdasarkan kasus lama yang pernah terjadi dan memberikan solusi pada kasus baru berdasarkan pada kasus lama yang memiliki nilai kemiripan tertinggi.

Kekurangan dan kelebihan pada metode CBR

Kelebihan :

- a. Memecahkan masalah dengan mudah
- b. Semakin banyak pengalaman, sistem akan menjadi semakin pintar sehingga dapat memecahkan masalah dengan mudah.

Kekurangan:

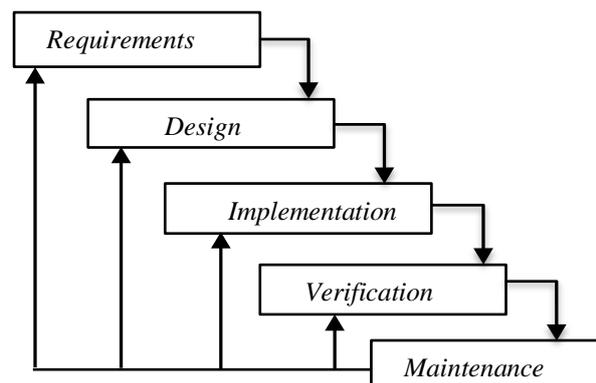
- a. Tidak menjamin solusi terbaik atau optimum karna penalaran ini berdasarkan kasus lampau, apabila solusi lampau salah maka tahapan revise sangat diperlukan untuk mengurangi tingkat kesalahannya.

- b. Semakin banyak pengalaman, proses pencarian semakin lama karna harud membandingkan dengan kasus yang paling mirip.

2.2.4 Pengembangan Sistem

A. Metode Waterfall

Metode *waterfall* adalah metode pengembangan perangkat lunak yang dilakukan secara sistematis dan terurut. Analoginya yakni seperti air terjun, jadi setiap proses perlu dikerjakan secara berurutan (dari atas sampai kebawah) tahapan dalam melakukan pengembangan perangkat lunak dengan menggunakan metode *waterfall* ada 5 yaitu[21] :



Gambar 2. 2 Alur Metode Waterfall[22]

1. Requirements

Pada tahap ini, dilakukan persiapan kemudian menganalisa kebutuhan pada perangkat lunak (*software*) yang dikerjakan. Informasinya dapat diambil melalui diskusi, observasi, survey, studi literature, dan wawancara.

2. Design

Selanjutnya adalah tahap pembuatan desain dari aplikasi yang akan dikerjakan sebelum proses *coding*. Tahapan ini bertujuan agar memberikan gambaran yang

jelas mengenai struktur data ataupun arsitektur *software*, fungsi internal dan eksternal dari algoritma yang diperlukan hingga tampilan dari *software*.

3. *Implementation*

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

4. *Verification*

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

5. *Maintenance*

Tahapan yang terakhir adalah *maintenance*, pada tahapan ini dilakukan pengoperasian dan perbaikan dari *software* atau aplikasi. Setelah pengujian sistem telah selesai, maka akan masuk ke tahap *software* tersebut dicoba oleh user. Proses pemeliharaan dari *software*, pengembang biasanya meminta *feedback* atau laporan dari user apabila user mendapat *error* atau *bug* dari aplikasi yang telah dibuat[18].

Adapun kelebihan dan kekurangan dari metode *waterfall* yaitu :

- a. Kelebihan pada metode *waterfall*
 - 1) Menghemat biaya
 - 2) Dokumentasi yang sangat bagus
 - 3) *Workflow* (aliran kerja) yang jelas

- 4) Digunakan untuk pengembangan *software* berskala besar.
- b. Kekurangan pada metode *waterfall*
 - 1) Kurangnya fleksibilitas
 - 2) Memerlukan tim yang *solid*
 - 3) Memerlukan waktu yang lama.
 - 4) Tidak bisa melihat gambaran sistem yang jelas dan benar

2.2.5 Desain sistem

A. Unified Modelling Language (UML)

UML Singkatan dari Unified Modelling Language yang berarti bahasa pemodelan standar. Chonoles mengatakan sebagai bahasa, berarti UML memiliki sintaks dan semantic. Disini penulis hanya akan menggunakan 2 macam diagram, yaitu[19] :

1. Use Case Diagram

Use case diagram adalah jenis pemodelan yang menggambarkan suatu hubungan interaksi antara satu aktor atau lebih dengan sistem aplikasi yang akan dibuat. Aktor disimbolkan menyerupai seseorang sedangkan sistem disimbolkan menyerupai elips. Seorang actor adalah yang melakukan interaksi dengan sistem informasi dan berada diluar sistem informasi yang akan dibuat itu sendiri, walaupun simbol dari aktor menyerupai orang, belum tentu aktor itu sendiri merupakan orang. *Use case* adalah fungsi yang dipersiapkan oleh sistem sebagai unit-unit yang saling berinteraksi atau bertukar pesan antar unit atau aktor. Berikut ini merupakan simbol-simbol yang sering digunakan dalam membuat *use case* diagram[23]

Tabel 2. 1 Simbol Use Case Dagram

No	Nama	Simbol	Keterangan
1.	<i>Actor</i>		Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari <i>actor</i> belum tentu merupakan orang: biasanya dinyatakan menggunakan kata benda diawal <i>frase</i> nama <i>actor</i> .
i2.	<i>Use Case</i>		Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau <i>actor</i> ; biasanya dinyatakann dengan menggunakan kata kerja di awal frase nama <i>Use Case</i> .
3.	<i>Asosiation</i>		Komunikasi antara actor dan Use Case yang berpartisipasi pada Use Case atau Use Case memiliki interaksi dengan actor.
4.	<i>Generalization</i>		Menspesifikasikan bahwa <i>Use Case</i> target memperluas perilaku dari <i>Use Case</i> sumber pada suatu titik yang diberikan.
5.	<i>Include</i>		Relasi <i>Use Case</i> tambahan ke sebuah <i>Use Case</i> dimana <i>Use Case</i> yang ditambahkan memerlukan <i>Use Case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.

6.	<i>Extend</i>		Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>Use Case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari hasilnya.
----	---------------	---	---

1. Activity diagram

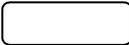
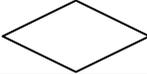
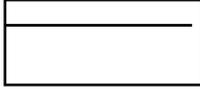
Activity diagram merupakan pemodelan yang menggambarkan *workflow* atau aliran kerja dari sebuah sistem atau aplikasi yang dibuat. Pada diagram ini hanya aktivitas dari sistem saja yang digambarkan, jadi aktivitas dari faktor tidak dapat dilihat. *Activity* diagram sering digunakan untuk hal-hal berikut,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

- a. Merancang menu yang ditampilkan pada *software*.
- b. Pengelompokkan dari sistem atau *user interface*, dimana setiap aktivitas dianggap memiliki suatu antarmuka tampilan.
- c. Merancang proses bisnis yang digambarkan merupakan proses bisnis dari sistem yang didefinisikan.
- d. Merancang pengujian, pada setiap aktivitas dianggap membutuhkan sebuah pengujian yang perlu didefinisikan kasus ujinya.

Berikut merupakan simbol yang selalu dipergunakan dalam pemodelan *activity* diagram :

Tabel 2. 2 *Activity Diagram*

No	Nama	Simbol	Keterangan
1.	Status Awal		Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah awal.

2.	Aktivitas		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
3.	Percabangan atau <i>Decision</i>		Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.	Penggabungan atau <i>join</i>		Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.	Status Akhir		Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6	<i>Swimlane</i>		Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

2.2.6 Desain Database

A. Entity Relationship Diagram (ERD)

Entity relationship diagram (*ERD*) adalah model data yang dikembangkan menurut objek *ERD* juga berfungsi sebagai penghubung antara data dan basis data kepada pengguna secara logis. *ERD* terdiri atas objek-objek yang berdasarkan pada suatu persepsi. User *ERD* lebih mudah dimengerti, terlebih bagi user pada umumnya. Bagi penganalisa sistem atau perancang *ERD* berfungsi bagi pemodelan sistem basis data untuk dikembangkan nantinya. Perancang ini membantu melakukan analisis dan juga pada perancangan *database*, model ini juga bisa memberikan berbagai macam data yang diperlukan dan kerelasiannya dalam data [20]. Terdapat 3 komponen dalam membentuk suatu *ERD* diantaranya :

1. Entitas, adalah sebuah objek untuk membedakan dari yang lain yang akan diwujudkan dalam basis data nantinya.

2. Hubungan atau relasi, adalah hubungan antara 2 jenis entitas yang digambarkan melalui garis lurus.
3. Atribut, memberikan detail informasi dari entitas, jenis atribut ada 5 tergantung dari tipe data entitas, diantaranya :

Atribut *Primary Key*, ialah atribut yang unik karena tidak memiliki nilai yang sama pada baris data yang lain.

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

Dalam perancangan ERD seringkali dijumpai derajat relasi, yang bertujuan untuk memberikan total uraian yang maksimal antara 1 entitas dan entitas lain

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

Setiap anggota A hanya bisa berinteraksi dengan 1 anggota B, begitu pula sebaliknya.

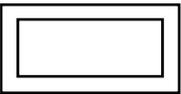
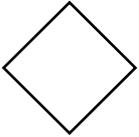
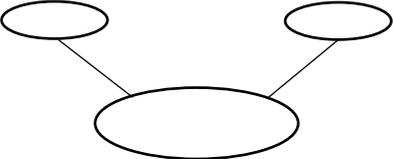
2. *One to Many* (1:M)

Setiap anggota A bisa berinteraksi dengan semua anggota dalam entitas B, namun tidak sebaliknya.

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

Setiap anggota A bisa berinteraksi dengan semua anggota dalam entitas B, begitu pula sebaliknya. Berikut adalah simbol-simbol yang sering digunakan dalam pembuatan erd, diantaranya :

Tabel 2. 3 *Simbol ERD*

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

9		Atribut Derivatif
---	---	-------------------

2.2.7 Implementasi

A. Hypertext Preprocessor (PHP)

Merupakan salah satu bahasa pemrograman *web* yang bersifat *server side* dan bertujuan untuk mewujudkan skrip kemudian di-*generate* dalam kode *html* yang merupakan bahasa standar *web*. Menurut Abhar, *php* merupakan bahasa pemrograman *web server-side* yang bersifat *open source*, dan *script* yang terintegrasi dengan *HTML* yang berada pada server yang digunakan untuk membuat halaman *website* yang sangat dinamis untuk membuat halaman tampilan yang akan ditampilkan pada saat dibuat halaman yang diminta oleh *client*. Secara resmi *PHP* merupakan bahasa *script server-side* yang disisipkan pada *HTML*[21].

Menurut Andi, *PHP* adalah bahasa pemrograman yang berjalan dalam sebuah *web server* yang berfungsi sebagai pengolah data pada server, data yang dikirim oleh *user client* akan diolah dan disimpan pada *database web server* kemudian ditampilkan kembali pada saat di akses diakses untuk menjalankan kode-kode program *PHP* harus diupload kedalam server, upload merupakan proses mentransfer data atau file dari komputer client kedalam *web server*. *PHP* dapat berinteraksi dalam dokumen *HTML (Hypertext Markup Language)* yang menghasilkan halaman yang sesuai dengan permintaan *user*, adanya *PHP* ini kita bisa merubah situs menjadi sebuah aplikasi berbasis web dan tidak sekedar sekelompok halaman statik yang jarang diperbaharui.

B. *Javascript*

Merupakan bahasa *script* yang populer di *internet* dan berinteraksi di sebagian besar *browser* populer misalnya *Internet explorer (IE)*, *Mozillafirefox*, *Netscape* dan *opera*. *Javascript* bisa disisipkan dalam halaman *web* dengan *tag script*.

Berikut adalah hal-hal tentang *javascript* yaitu [28]:

1. *Javascript* mendesain dan menambah interaktif pada *web*
2. *Javascript* adalah bahasa *scripting*
3. *Scripting* adalah bahasa pemrograman yang mudah
4. *Javascript* berisi baris kode yang dijalankan di komputer (*web browser*)
5. *Javascript* disisipkan (*embedded*) dalam halaman *HTML*
6. *Javascript* adalah bahasa interpreter (yang berarti skrip dieksekusi tanpa proses kompilasi)
7. Setiap orang bisa menggunakan *Javascript* tidak memungut biaya lisensi.

C. MySQL

MySQL adalah bahasa pemrograman yang bisa menangani data dalam berkapasitas besar tetapi tidak menuntut *resource* yang besar. *MySQL* merupakan *database* yang paling populer daripada *database* lain.

MySQL merupakan *Foss license exception* dan tetapi juga memiliki versi komersial. *Tag MySQL* merupakan “*The World's most popular open source database*”. *MySQL* menyediakan beberapa platform, di antaranya pada versi windows dan linux. *MySQL* memudahkan dalam melakukan administrasi dengan

menggunakan software tertentu, di antaranya adalah *php my admin* dan *mysql yog*. Kesempatan kali ini, kita akan menggunakan *php my admin*, yang terdapat dalam *bundle xampp*[22].

D. Bootstrap

Bootstrap merupakan *framework front-end* dan *powerful* yang mengembangkan suatu aplikasi *web* yang lebih mudah dan menggunakan *HTML*, *CSS* dan *Javascript*. *Bootstrap* dikembangkan oleh Mark Otto dan Jacob Thornton dari *Twitter*. *Framework* di jalan sebagai produk *open source* pada Agustus 2011 di *GitHub*, *bootstrap* juga mempunyai fitur-fitur komponen *interface* yang istimewa seperti *typography*, *forms*, *buttons*, *tables*, *navigations*, *dropdowns*, *alerts*, *modals*, *tabs*, *accordion*, *carousel* dan lain-lain.

D. XAMPP

XAMPP merupakan salah satu *software* yang menjalankan *website* berbasis *PHP* dan memanfaatkan pengolah data *MySQL* pada komputer lokal, *Xampp* juga berperan sebagai *server web* pada komputer. *Xampp* merupakan salah satu kumpulan paket software dan terdiri dari *apache*, *mysql*, *php my admin*, *php*, *Perl*, *Freetype2*, dan lain-lain. *Xampp* berguna untuk memudahkan instalasi lingkungan *php* dan biasanya lingkungan pengembangan web memerlukan *php*, *apache*, *mysql* dan *php my admin* serta *software-software* kemudian terkait pada pengembangan *web*. Dalam menggunakan *xampp*, kita tidak perlu menginstall aplikasi tersebut satu persatu.

2.2.8 Pengujian

a. *Black Box Testing*

Blackbox Testing yaitu pengujian program dari segi spesifikasi fungsional tanpa menguji *source code* untuk mengetahui fungsi dari kesalahan interface yang ada dalam struktur data pengujian ini hanya memeriksa nilai keluaran berdasarkan nilai masukan masing-masing. Metode pengujian *black box* ini fokus pada bagian hasil eksekusi dari fungsi pada sistem. Setiap fungsi pada sistem yang diuji kesesuaian dari hasil output-nya[23].

Black Box Testing cenderung untuk menemukan hal-hal berikut:

1. Fungsi yang tidak benar atau tidak ada.
2. Kesalahan antarmuka (*interface errors*).
3. Kesalahan pada struktur data dan akses basis data.
4. Kesalahan performansi (*performance errors*).
5. Kesalahan inisialisasi dan terminasi.

b. *User Acceptance Testing (UAT)*

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

1. UAT mengukur bagaimana sistem sudah sesuai dengan kebutuhan pengguna[26].
2. UAT mengekspos fungsionalitas/*logic* bisnis yang belum ditemukan, karena *unit testing* dan *system testing* tidak berfokus pada fungsionalitas/*logic* bisnis.

3. UAT membatasi bagaimana sistem telah “selesai” dibuat[24].

c. *Matrix Confusion*

Confusion matrix merupakan salah satu metode yang dapat digunakan untuk mengukur kinerja suatu metode klasifikasi[27].

Tabel 2. 4*Multiclas Confusion Matrix*

	Prediksi	
Aktual	TRUE	FALSE
TRUE	TP	FP
FALSE	FN	TN

Dimana:

True Positive (TP) Jumlah dari satu kelas True yang diprediksi benar pada kelas True
True Negative (TN) Jumlah dari satu kelas

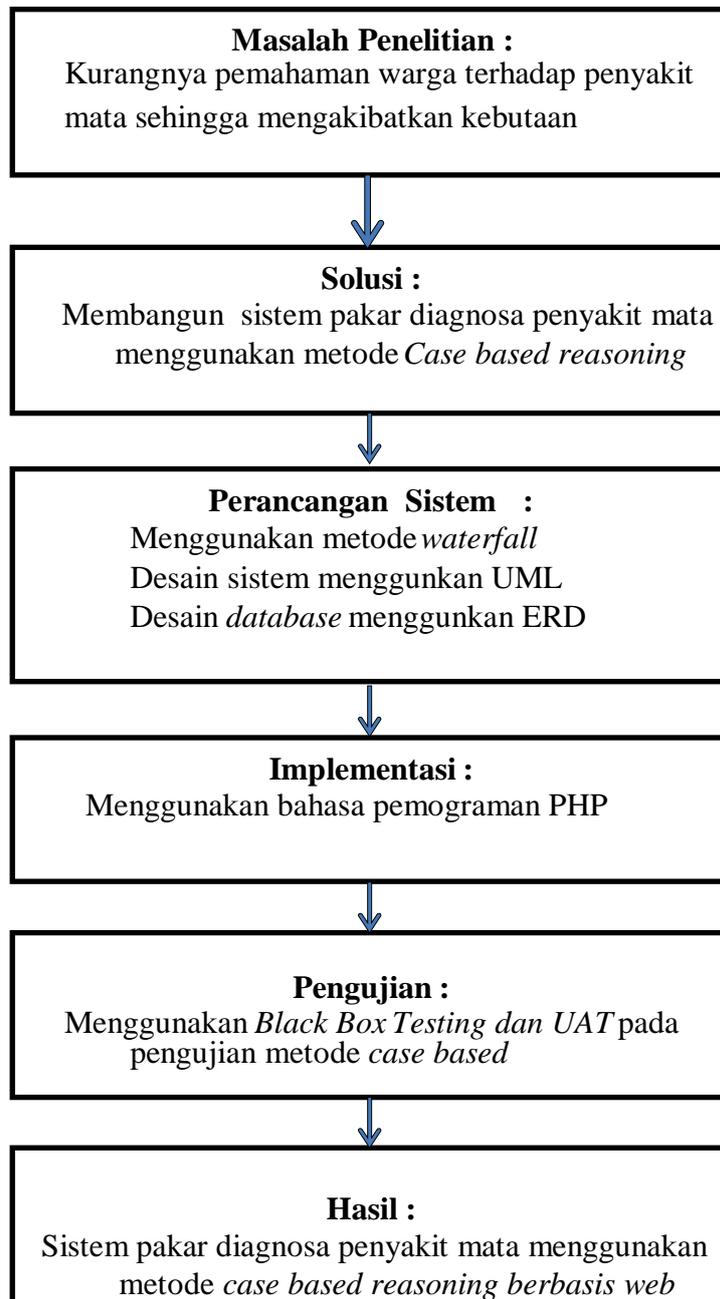
True yang diprediksi False yang diprediksi benar pada kelas False, *False Positive* (FP) Kondisi dimana kelas True diprediksi salah pada kelas False

False Negative (FN) Kondisi dimana Kelas False diprediksi salah pada kelas True

$$\text{Rumus Akurasi} = \frac{TPos+TNeg}{\text{Jumlah Set Data}} \times 100\%$$

2.2 Kerangka pikir

Kerangka pikir yang dilaksanakan dapat dilihat pada Gambar 2.2 :



Gambar 2. 3 Kerangka Pikir