BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Dalam penyusun penelitan ini, penulis mendapatkan referensi dari peneliti sebelumnya yang berkaitan dengan latar belakang masalah pada penelitian ini berikut: Penelitian yang dilakukan oleh Octaviani Tanjung [2] dengan judul sistem pakar tanaman herbal untuk pengobatan penyakit mengunakan *Metode Case Base Reasoning* permasalah dari penelitian ini bagaimana sistem pakar ini dapat membantu user untuk mengetahui penyakit dan herbal yang cocok untuk setiap penyakit.

Penelitian yang dilakukan oleh Eni Mariani [3] dengan judul "sistem pakar *monitoring* pertumbuhan balita berbasis web mengunakan metode *Case Based Reasoning* dimana masalah dari penelitian ini adalah bagaimana nantinya sistem pakar ini dapat digunakan untuk *memonitoring* pertumbuhan anak melalui *web* bukan lagi mengunakan (Kartu Menuju Sehat).

Penelitian yang dilakukan oleh, Muhammad Fathoni Hervi Hermawan dkk [4], dengan judul "deteksi kerusakan *handpone* Samsung melalui sistem pakar mengunakan kombinasi algolitma K-*Nearest Neighbour* dengan *Case Base Reasoning"*. masalah dari penelitian ini adalah kurangan pengetahuan orang awam, mengenai krusakan tipe *handphone* tertentu dengan adanya sistem pakar ini dapat membantu kaum orang mendiagnosa kerusakan dengan

mengunakan sistem tersebut, sebelum mendatangi tempat *service handphone.*

Penelitian yang dilakukan oleh Suzuki Syofianimmi Kridiagung yang berjudul "implementasi *Case Based Reasoning* (*CBR)* dan K-Nearst Neighbour (K-NN) untuk mendiangnosa gejala kerusakan kendaraan roda 2". masalah dari penelitian ini adalah kebanyakan penguna roda 2 hanya mengunakan kendaraan tanpa mengetahui bila ada kerusakan yang tiba-tiba terjadi [5].

Pada penelitian yang dilakukan oleh Sana Mega Wika Sirait yang berjudul sistem pakar "terapi anak autism pada pusat layanan *autis* (PLA) provinsi riau mengunakan *Case Base Reasoning*," masalah pada penelitian ini yaitu pada pusat pelanyana PLA provinsi Riau masih menentukan terapi anak Autis dengan *from* kertas yang manual sehingga membutuhkan waktu yang lama dan menghabiskan biaya arsipyang tidak sedikit [6].

2.2 Landasan Teori

2.2.1 Kecerdasan Buatan

Kecerdasan adalah ide-ide untuk membuat suatu perangkat lunak komputer yang memilki kecerdasan buatan sehingga perangkat lunak komputer tersebut dapat melakukan suatu pekerjaan yang dilakukan oleh manusia.adapun pekerajaan itu berupa konsultasi yang dapat memberikan suatu informasi berupa saran-saran yang akan sangat berguna.

Kecerdasan buatan memungkinkan komputer berpikir dengan cara menyederhanakan program. kecerdasan buatan juga dapat meniru proses pembelajaran manusia sehingga informasi baru dapat diserap dan digunakan

sebagai acuan di masa mendatang. Kecerdasan buatan atau kepandaian diperoleh dengan dari pengetahuan dan pengalaman, agar perangkat lunak dapat mempunyai kecerdasan untuk itu perangkat lunak harus diberi suatu pengetahuan dan kemapuan untuk menalar dari pengetahuan yang telah didapat untuk menjadi solusi atau kesimpulan seperti seorang pakar dalam bidang tertentu Yang bersifat spesifik [7].

Dalam kecerdasan buatan menawarkan media dan uji teori kecerdasan. Kemudian dapat dipaparkan dalam bahasa program komputer dan dibuktikan melaluieksekusi pada komputer nyata.

2.2.2 Sistem Pakar

Sistem pakar adalah pengembangan dari kecerdasan buatan yang mengabungkan pengetahuan ahli dan penelusuran data untuk memecahkan masalah. tujuan dari sistempakar sebenarnya bukan untuk mengantikan peran manusia tetapi untuk menerapakan ilmu pengetahuan ahli kedalam bentuk sistem, [8] sehingga dapat digunakan oleh para teknisi. Sistem pakar yang pertama kali muncul adalah *General Purpose Problem Solver* (GPS) yang dikembangkan oleh Newel dan Simon. Sampai saat ini masih banyak sistem pakar yang dibuat seprti Mycin, Denrai, Xcon&Xsel, Sophie, Prospector, Folio Dan Delta [9].

Ada beberapa pemecahan masalah yang dilakukan dengan adalnya sistem pakar yaitu:

1. Pembuatan keputusan (*decision making*)
2. Pemanduan pengetahuan (*knowledge fusing*)
3. Pembuatan desain (*desingning*)

4. Perencanaan (*planning*)
5. Prakiraan (f*orecasting*)
6. Pengaturan (*regulating)*
7. Pengendalian (*controlling)*
8. Diagnosis (*diagnosing*)
9. Perumusan (*prescribing*)
10. Penjelasan (*explaining*)
11. Pemberian nasihat (*advising*)
12. Pelatihan (*tutoring*)

Secara umum sistem pakar merupakan sistem yang mampu pengadopsi pengetahuan manusia ke komputer yang dirancang untuk memodelkan kemampuan menyelesaikan masalah seperti layaknya seorang pakar. Sistem pakar juga dapat membantu aktivitas para pakar sebagai asisten yang berpengalaman dan mempunyai pengetahuan yang dibutuhkan dan para pakar tidak perlu melakukan pengulan langkah-langkah yang sama.

Pada penyusunan sistem pakar mengkombinasikan cara-cara penarikan kesimpulan (*inference rules*) dengan basis pengetahuan tertentu yang diberikan oleh para pakar tertentu. kombinasi dalam dari kedua hal tersebut disimpan dalam komputer, yang selanjutnya digunakan dalam pemgambilan keputusan untuk menyelesaikan masalah yang ada.

Ciri-ciri sistem pakar yang baik menurut Kusrini [10].

a. Terbatas pada bidang yang spesifik
b. Dapat memberi penalaran untuk data-data yang tidak lengkap atau tidak

pasti.

c. Dapat mengemukakan rangkaian alasan yang diberikan dengan cara yang dapatdipahami.

d. Berdasarkan *rule* atau kaidah tertentu

e. Dirancang untuk dapat dikembangkan secara bertahap.

f. Outputnya bersifat nasihat dan anjuran.

g. Outputnya tergantung dari dialog yang dilakuan dengan *user*.

h. *Knowledge bas*e dan *interface engine* terpisah

i. Dapat digunakan dalam berbagai jenis komputer.

a) Keuntungan sistem pakar

Secara garis besar banyak manfaat yang dapat diambil dengan adanya sistempakar antara lain.

a. Bisa melakukan pekerjaan berulang secara otomatis

b. Dapat menyimpan pengetahuan dan keahlian para pakar

c. Meningkatkan produktivitas

d. Mengingatkan kualitas

e. Mampu beroperasi dalam lingkungan berbahaya

f. Memiliki reabilitas serta menghemat waktu dalam pengambilan keputusan

g. Meningkatkan kapabilitas sistem komputer

h. Sebagai media pelengkap dalam pelatihan

b) Kelemahan sistem pakar

a. Masalah dalam mendapatkan pengetahuan yang tidak selalu bisa didapatkan dengan mudah karena kadangkala pakar dari masalah

yang dibuat tidak ada dan jika ada kadang pendekatan yang dimiliki para pakar berbeda-beda.

b. Untuk membuat sistem pakar benar-benar berkualitas tinggi sangat sulit
c. Kadang sistem tidak dapat membuat keputusan
d. Sistem pakar tidak 100%menguntungkan

Alasan pengembangan sistem pakar

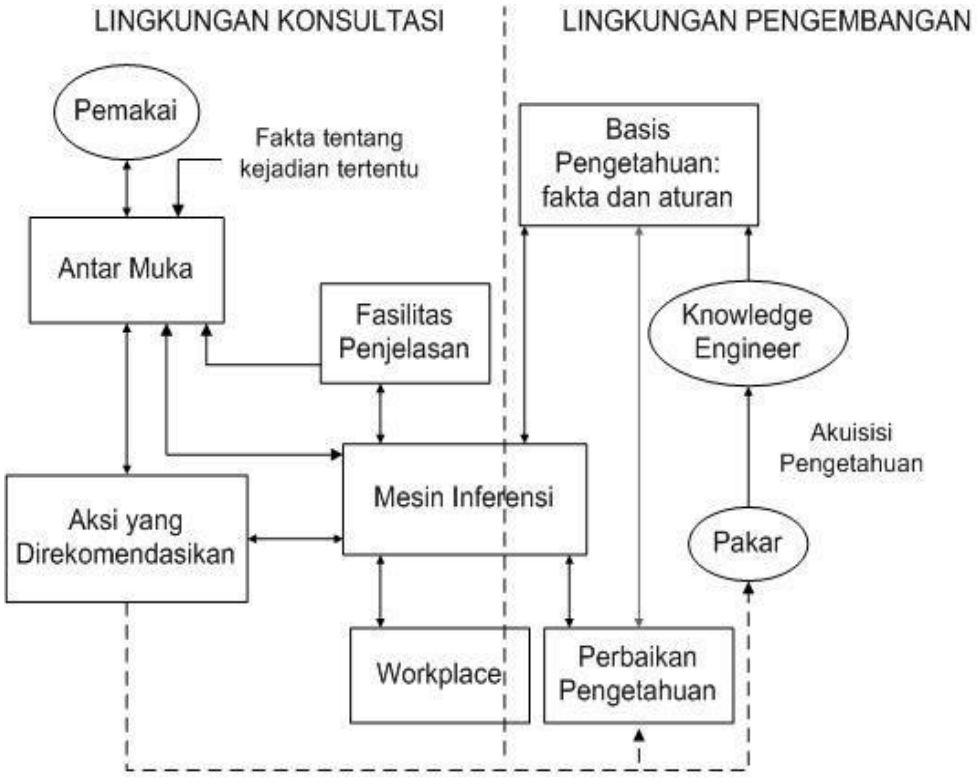
1. Menyediakan kepakaran setiap waktu di berbagai lokasi
2. Dapat bekerja secara otomatis pada tugas rutin yang mebutuhkan sistem pakar
3. Seorang pakar akan pensiun atau pergi

Penyusunan sistem pakar menurut Stuggard dalam jurnal suatu sistem pakar tersusun atas 3 modul utama [11]:

1. Modul penerimaan pengetahuan (*knowledge Acuisition mode*) pada modul ini, dimana proses penerimaan dan pengumpulan pengetahuan dari pakar, yang digunakan dalam pengembangan sistem yang dilakukan dengan bantuan *knowledge engineer*. Berperan utuk menghubungkan atar suatu sistem pakar dengan pakarnya.
2. Modul konstultasi (*consultation mode*) pada modul ini sistem pakar berada pada pemberikan jawaban atas permasalahan yang dilakukan oleh *user*, pada bagian ini user berinteraksi dengan sistem dengan menjawab pertanyaan yang diajukanoleh sistem.
3. Modul penjelasan (*explanation mode*) pada modul ini memaparkan proses pengambilan keputusan oleh sistem.

a. Struktur sistem pakar

Pada dasarnya sistem pakar terdiri dari dua bagian inti yaitu: lingkungan pengembangan (*development environment*) dan lingkungan konsultasi (*consultation environment*).lingkungan pengembangan digunakan sebagai pembangunan sistem maupun basis pengetahuan.lingkungan konsultasi digunakan oleh seseorang yang bukan ahli untuk berkonsultasi dengan masalahyang dialami.



Gambar 2.1 Alur Konsultasi

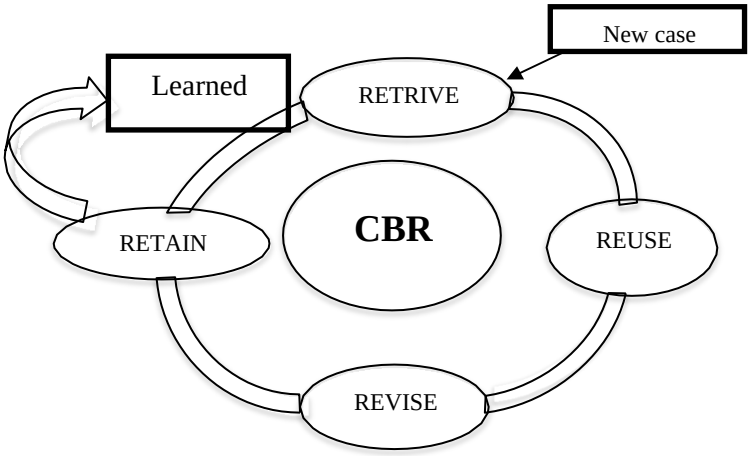
2.2.3 Jantung

Jantung adalah salah satu modifikasi pembuluh darah yang berperan untuk memompa darah keseluruh tubuh, dengan membentuk pembuluh darah menjadi 2 sirkulasi, yaitu sikulasi *purmonal (*sirkulasi kecil) menyalurkan darah dari dan ke paru-paru, sedangkan *sirkulasi sistemik* (sirkulasi besar)membawah darah dari dan ke seluru tubuh. Dari jantung darah dialirkan ke pembuluh e*fferent* jantung, dimulai dari pembuluh terbesar yang disebut *Aorta* yang melalui cabang-cabang pembuluh yang lebih kecil. *Arteri* berguna membawah darah bersama *nutrien* dan oksigen jaringan. Darah akhirnya mencapai *kapiler y*ang *beranastomos*e (*capillary bed*) di mana terjadi pertukaran zat [12] antara darah dan jaringan. Dari kapiler darah kembali ke jantung melewati sistem pembuluh darah lebih besar yakni pembuluh *vena* yang merupakan pembuluh *afferen* jantung yang berfungsi membawa *metabolit* dan CO2 (Santa, 2002). dalam jurnal [12]. Ada beberapa jenis penyakit jantung yakni, jantung *coroner*, serangan jantung, *aritmia*, *kardiomioopat*i, gagal jantung, penyakit jantung

bawaan, *katup* jantung, *endokarditis*, tumor jantung.

2.2.4 Metode *Case Base Reasoning* (CBR)

Menurut Aamodt dan Plaza (1994) dalam jurnal *Case-Base Reasoning* adalah suatu pendekatan untuk menyelesaikan suatu permasalahan (*problem solving*) berdasarkan solusi dari permasalahan sebelumnya.



Gambar 2. 1 Tahap Proses Metode CBR

a. *Retrieve* untuk mendapatkan kasus-kasus yang mirip dibandingkan dengan kumpulan kasus di masa lalu, dimulai dengan tahapan mengenali masalah dan berakir ketika kasus yang ingin dicari solusinya telah ditemukan serupa dengan kasus lama. Tahapanyang ada pada *retrieve* ini antara lain:
   1. Identifikasi masalah
   2. Memulai pencocokan
   3. Menyeleksi
b. *Reuse* : mengunakan kembali kasus-kasus yang ada dan dicoba untuk menyelesaikan kasus baru(sekarang).*Reuse* suatu kasus dalam konteks kasus baru terfokus pada aspek,diantaranya: perbedaan antara kasus yang ada dengan kasus yang baru dan bagian dimana dari *retrieve case* yang dapat digunakan pada kasus yang baru. Ada dua cara yang digunakan untuk me-*reuse* kasus yang telah ada (*transformatial reuse*) atau *reuse* metode kasus yang ada untuk membuat solusi *(derivational reuse*).

c. *Revise:* dimana terjadi proses merubah dan mengadopsi solusi yang

ditawarkan jika perlu.terdapat dua tugas utama dari tahapan ini, yaitu: evaluasi solusi dan memperbaikikesalahan.

d. *Retain* : dimana tetap memakai solusi yang terakhir sebagai bagian dari kasus baru.pada tahap ini juga terjadi suatu proses penggabungan dari solusi kasus yang baru yang benar ke *knowledge* yang telah ada. kemuadian terdapat tiga tahapan antaranya: *extract, index* dan *integrate*.Berikut adala rumus untuk mencari nilai kemiripan (*similiarity)* [8].

Kemiripan (*similarity*) adalah langkah yang digunakan untuk mengenali kesamaan atau kemiripan antara kasus-kasus yang tersimpan dalam kasus yang baru. Kasus dengan nilai *similarity* yang paling besar dianggap sebagai kasus yang paling mirip. Nilai *similarity* berkisar antara 0-1.ketentuan bobot: [13] Gejala biasa = 1 sedangkan gejala sedang =3.

2.2.5 Algoritma Nearest Neighbour (K-NN)

Algoritma *nearest neigbour* merupakan suatu algoritma yang betujuan untuk melakukan klasifikasi teradap objek berdasarkan data pembelajaran dimana mengambil jarak yang paling dekat dengan suatu objek kasus. Algoritma K-NN ini mengelompokkan data baru yang belum di ketahui *class* yang cocok, dengan memilih data seumlah K akan dipilih menjadi class yang diperkirakan untuk data baru tersebut.pada umunnya nilai k mengunakan jumlah ganjil agar tidak terdapat jarak yang sama.

dalam proses klasifikasi.jauh atau dekatnya tetangga dihitung mengunakan jarak *Euclidean*.

Kasus khusus dimana klasifikasi di prediksi berdasarkan data

pemebelajaran yang paling dekat (atau k=1) disebut *algoritma K- Nearest Neighbour*. rumus *similarity* dengan *K Nearest Neigbour* di turunkan.

$$similiarity\ (T,S) = \frac{\sum_{I=1}^{N} f(T_i, S_i) \times W_i}{W_i} \quad (2.1)$$

Keterangan:

T : Kasus baru

S : Kasus yang ada dalam penyimpanan

N : Jumlah *atribut* dalam setiap kasus

I : *Atribut* individu antara 1 s.d. *n*

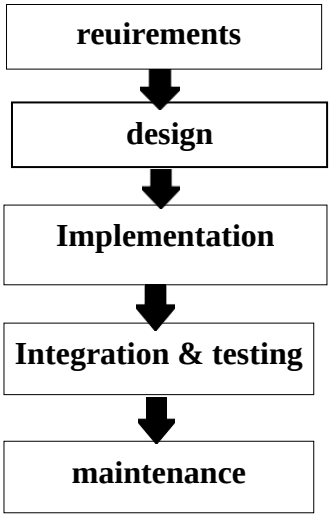
F : Fungsi *similarity* atribut *i* antara kasus *T* dan kasus *S*

W : Bobot yang diberikan antara *atribut* ke-*i*

2.2.6 Pengembangan Sistem

A. **Metode *Waterfall***

Metode *waterfall* merupakan pengembangan perangkat lunak yang dilakukan secara sistematis dan terurut. Analoginya, yakni seperti air terjun. Jadi setiap perlu dikerjakan secara berurutan. Tahap dalam melakukan pengembangan perangkat lunak dengan mengunakan metode *waterfall* ada lima yaitu:



Gambar 2. 2. Metode *Waterfall*.

1) Requirements

Pada tahap ini yang pertama kali dilakukan adalah mempersiapkan dan menganalisis kebutuhan dari *software* yang akan dikerjakan. Informasinya dapat diperoleh mealalui *survey*, wawancara, studi *literature*, o*bservasi* hingga diskusi.

2) Design

Tahap selanjutnya adalah tahap pembuatan desain dari apalikasi yang akan dikerjakan sebelum proses *coding*. tujuan dari proses ini bertujuan agar memberikan gambaran yang jelas pada struktur data ataupun arsitektur *software* fungsi eksternal dan internal dari algoritma sampai tampilan dari *software.*

3) Implementation

Pada tahap ini desain dari *software* yang diinginkan kemudian di implementasikan kedalam kode program dengan mengunakan berbagai *tools* dan bahasa pemrograman yang di inginkan.

4) Integration& Testing

Tahap ini adalah tahap dimana dilakukan proses integrasi dan pengujian sistem yang telah dibuat. Yang biasanya bertanggungjawab melakukan testing biasanya adalah QA (*Quality Assurance*) dan QC *(Quality Control*). Mereka akan melakukanpengecekan apaka *software* sudah sesuai desain serta apakah terdapat error atau *bug*.

5) Maintenance

Tahap terakhir ini yang dilakukan adalah pengoperasian dan perbaikan dari s*oftware* atau aplikasi.setelah pengujian sistem selesai, maka akan masuk tahap peguji coba oleh *user* (pengguna). Kemudian untuk pemeliharaan dari *software p*engembang biasa meminta *feetback,* atau laporan dari user jika mendapatkan *error* atau *bug* dari sistem yang dibuat.

Ada beberapa kelebihan dan kekurangan metode *waterfall*:

a. Kelebihan metode *warerfall*

1. *Workflow* (aliran kerja) yang jelas
2. Hasil dokumentasi yang baik
3. Dapat menghemat biaya
4. Digunakan untuk pengembangan *software* berskala besar.

b. Kekurangan metode *waterfall*

1. Membutuhkan tim yang solid
2. Tidak dapat melihat gambaran sistem yang jelas
3. Masih kurang *flesibilitas*
4. Membutuhkan waktu yang lama [14].

2.2.7 Desain Sistem

A. *Unified Modelling Language*(UML)

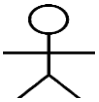
Unified modelling language (UML) merupakan metode suatu rancangan sistem berorientasi objek secara visual.dalam UML terdapat 14 macam diagram seperti *class* diagram, *object* diagram, *package* diagram, *sequence* diagram, *communication* diagram, *timing* diagram, dan *interaction overview*.

1. *Use case* diagram

Use Case Diagram adalah jenis diagram pemodelan yang mengambarkan hubungan interaksi antara satu atau lebih *actor* dengan sistem atau aplikasi yang di buat.aktor disimbolkan menyerupai seseorang sedangkan sistem di simbolkan menyerupai *elips*. *aktor* yang dibuat, berada di luar sistem informasi yang akan dibuat sendiri, walaupun simbol dari actor menyerupai orang belum tentu *actor* itu adalah seseorang.

Use case adalah fungsi yang disediakan sistem bagi unit yang saling berinteraksidan bertukar pesan antar unit atau *actor*.

Tabel 2. 1 *Use Case Diagram*

No	Nama	Simbol	Keterangan
1.	*Actor*		Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari *actor* belum tentu merupakan orang: biasanya dinyatakan menggunakan kata benda diawal *frase* nama *actor*.

2.	*Use Case*		Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesanantar unit atau *actor*; biasanya dinyatakann dengan menggunakan kata kerja di awal *frase* nama
3.	*Asosiation*		Komunikasi antara *actor* dan *Use Case* yang berpartisipasi pada *Use Case* atau *Use Case* memiliki interaksi dengan *actor.*
4.	*Generalization*		Menspesifikasikan bahwa *Use Case* target memperluas perilaku dari *Use Case* sumber pada suatu titik yang diberikan.
5.	*Include*		Relasi *Use Case* tambahan ke sebuah *Use Case* dimana *Use Case* yang ditambahkan memerlukan *Use Case* ini untuk menjalankan fungsinya atau sebagai syarat dijalankan *user case ini.*
6.	*Extend*		Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah *Use Case* dimana fungsi yang satu adalah fungsi yang lebih umum dari hasilnya.

2. *Activity Diagram*

Activity diagram adalah jenis diagram 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 aktor tidak dapat dilihat. *Activity* diagram sering digunakan untuk hal-hal berikut, yaitu : [15].

a. Rancang menu yang ditampilkan pada *software*.

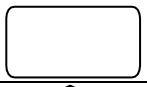
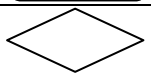
b. Urutan atau pengelompokkan tampilan dari sistem atau *user interface*, dimanasetiap aktivitas dianggap memiliki suatu antarmuka tampilan.

c. Rancangan proses bisnis, dimana setiap urutan aktivitas yang digambarkanmerupakan proses bisnis dari sistem yang didefinisikan.

d. Rancangan pengujian, dimana setiap aktivitas dianggap memerlukan sebuahpengujian yang perlu didefinisikan kasus ujinya.

Berikut adalah simbol-simbol yang sering digunakan dalam pemodelan *activity* diagram :

Tabel 2. 2 Simbol Activity Diagram

No	Nama	Simbol	Keterangan
1.	Status Awal		Status awal aktivitas sistem, sebuah aktivitas memiliki sebuah awal.
2.	Aktivitas		Aktivitas yang dilakukan sistem, akti
3.	Percabangan atau *Decision*		Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.	Penggabungan atau *join*		Asosiasi penggabungan dimana lebih dari satuaktivitas digabungkan menjadi satu.
5.	Status Akhir		Status akhir yang dilakukan sistem, sebuahdiagram aktivitas memiliki sebuah status akhir.

2.2.8 Desain Data Base

a. Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) merupakan suatu model data yang dikembangkan berdasarkan objek." *Entity Relationship Diagram* (ERD) digunakan untuk menjelaskan hubungan antar data dalam basis data kepada pengguna secara logis. *Entity Relationship Diagram* (ERD) didasarkan pada suatu persepsi bahwa *real world* terdiri atas obyek-obyek dasar tersebut. Penggunaan *Entity Relationship Diagram* (ERD) relatif mudah dipahami, bahkan oleh para pengguna yang awam. Bagi perancang atau analis sistem,

Entity Relationship Diagram (ERD) berguna untuk memodelkan sistem yang nantinya, basis data akan di kembangkan. Model ini juga membantu perancang atau analis sistem pada saat melakukan analis dan perancangan basis data karena model ini dapat menunjukkan macam data yang dibutuhkan dan kerelasian antardata didalamnya [3].

Terdapat 3 komponen dalam membentuk suatu ERD di antaranya :

1. *Entitas,* adalah sebuah objek untuk membedakan dari yang lain yang akandiwujudkan dalam basis data nantinya.
2. Hubungan / Relasi, adalah hubungan antara 2 jenis *entitas* yang digambarkanmelalui garis lurus.
3. Atribut, memberikan detail informasi dari *entitas.* Jenis atribut ada 5 tergantungdari tipe data *entitas*, diantaranya :

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

b. Atribut *Simple*, ialah atribut yang bernilai *atomic* atau tidak dapat dipecahmaupun dipilah lagi.

c. Atribut *Multivalue*, ialah atribut yang memiliki lebih dari 1 nilai dari atribut yangbersangkutan.

d. Atribut *Composite,* ialah atribut yang terdiri dari beberapa atribut yang lebih kecil yang berarti masih dapat dipecah lagi atau memiliki sub atribut.

e. Atribut *Derivatif*, ialah atribut yang tidak harus disimpan dalam basis data.

Dalam perancangan ERD seringkali dijumpai derajat relasi, yang bertujuan untuk menjelaskan jumlah maksimum antara 1 entitas dengan entitas yang lain.

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

   Setiap anggota *entitas* A hanya dapat berhubungan dengan 1 anggota dalam*entitas* B, begitu pula sebaliknya.

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

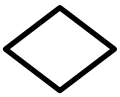
   Setiap anggota *entitas* A dapat berhubungan dengan banyak anggota dalam *entitas* B, namun tidak sebaliknya.

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

Setiap anggota entitas A dapat berhubungan dengan banyak anggota dalam *entitas* B, begitu pula sebaliknya.

Berikut adalah simbol-simbol yang sering digunakan dalam pembuatan ERD,diantaranya :

Tabel 2. 3 Simbol Atribut ERD

No	Notasi	Arti
1		*Entity*
2		*Weak Entity*
3		*Relationship*
4		*Identifying Relationship*
5		Atribut
6		Atribut *Primary Key*
7		Atribut *Multivalue*
8		Atribut *Composite*
9		Atribut *Derivatif*

2.2.9 Implementasi

A. *Web*

Web merupakan salah satu aplikasi yang berisikan dokumen-dokumen multi media(teks, gamar, suara, animasi, video. yang didalamnya mengunakan *protocol* HTTP (*Hypertext Transfer Protocol*) dan untuk mengaksesnya mengunakan perangkat lunak yang disebut *browse*r [16].

B. *Hypertext Preprocessor*(PHP)

PHP merupakan bahasa pemrograman bersifat *server side* yang bertujuan, untuk mengasiskan *skrip* yang akan di-*generate* dalam kode HTML yang merupakan bahasastandar pembuatan web.

C. *Java Script*

Javascript merupakan bahasa *scripting* yang popular di internet dan dapat bekerja di sebagian besar browser popular seperti internet *explorer, Mozilla Firefox, Netscape,* Dan *Opera. kode JavaCript dapat* disisipkan pada halaman web yang mengunakan *tag script*. dibawah ini beberapa hal tetang *javacript*:

a. *Javascript* merupakan bahasa *scripting*

b. Bahasa *scriping* merupaka bahasa pemrograman yang ringan

c. *Javascript* didesain untuk menambah interaktif suatu web

d. *Javascript* berisi barisan kode yang dijalankan di computer (*web browser*)

e. J*avascript* merupakan bahasa *interpreter* (*script* dieksekusi tanpa proseskompilasi)

f. J*avascript* bersifat *open source*

D. *MySQL*

MySQL merupakan *database* server adalah RDBMS (*relational database management system),* yang dapat mengatasi data yang bervolume besar. namun tidak menuntut *resource* yang besar. MySQL adalah *datase* yang paling popular diantara *database –database* yang lain

E. Xampp

XAMPP adalah *software* yang berfungsi untuk menjalankan *wesite* berbasis PHP dan mengunakan pengelolah data *MySQL*. Pada komputer lokal XAMPP berperan sebagai server web pada komputer.

F. DBMS

DBMS merupakan sebua aplikasi yang menjembatani user di dalam *database*.dengan mengunakan DBMS user mampu mengelolah data-data di dalam secara muda dan cepat.

G. *Flowchart*

Flowcart digunakan untuk mengambarkan aliran kegiatan yang akan terjadi dari program yang dimaksud kedalam suatu bagan. Dari bagan alir ini, dapat diamanti dan ditentukan aliran kedali program. berikut simbol pada *flowchart*:

2.2.10. Pengujian

A. *Back box testing*

Black box testing befokus pada spesifikasi fungsional dari perangkat lunak.tester dapat mengartikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasifungsioal program.

Hal yang sering ditemukan pada *black box testing*:

1. Fungsi yang tidak ada atau fungsi yang salah
2. Kesalahan *interface errors*
3. Kesalahan informasi
4. Kesalahan pada struktur data dan akses basis data
5. Kesalaan inisialisasi dan terminasi

Pengujan desain untuk menjawab pertanyaan:

1. Bagaimana fungsi-fungsi diuji agar dapat menyatakan *valid*?
2. Input seperti apa yang dapat menjadi bahan kasus ui yang baik?
3. Apakah sistem sensitif pada input tertentu?
4. Bagaimana sekumpulan data dapat disosialisasi

B. *User Acceptance Testing* (UAT)

User Acceptance Testing (UAT) merupakan salah satu metodologi yang sangat inovatif untuk mencegah kegagalan proyek IT [17]. Dalam pengembangan perangkatlunak, terdapat tiga hal yang dilakukan dalam proses UAT yaitu:

a. UAT mengukur bagaimana sistem sudah sesuai dengan kebutuhan pengguna.

b. UAT *mengekspos fungsionalitas*/*logic* bisinis yang belum ditemukan, karena

   unit testing dan *system testing* tidak berfokus pada fungsionalitas/*logic* bisnis.

c. UAT membatasi bagaimana sistem telah "selesai" dibuat.

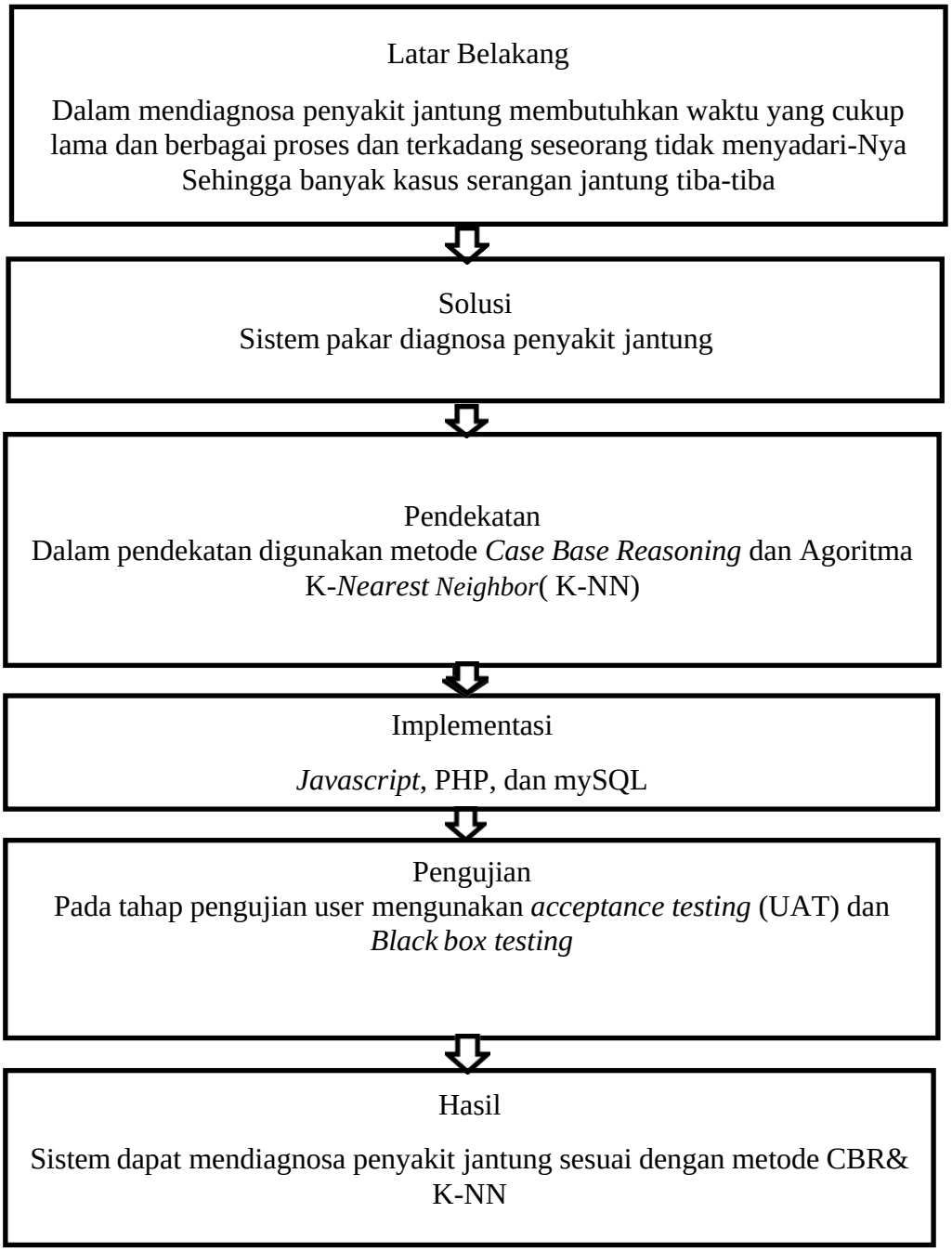
1. Proses UAT diawali dengan menyediakan dokumentasi persyaratan bisnis, kemudian dilanjutkan dengan proses bisnis (alur kerja) atau skenario dan yang terakhir yaitu pengujian menggunakan data. Efektifitas dalam pengujian sangat dibutuhkan dalam pengembangan sebuah aplikasi ataupun sistem informasi sehingga produk tersebut dapat sampai kepada pengguna dengan tepat waktu dan sesuai dengan kebutuhan pengguna. Berdasarkan penelitian yang dilakukan oleh

   Zhang, efektifitas terhadap kriteria pengujian yang sudah ada dan yang

baru harus dievaluasi untuk membangun teori pengujian yang lebih berguna.

2.2.11. Kerangka Pikir

Dibawah ini adalah uraian krangka pikir dari sistem pakar yang akan dibuat;



Gambar 2. 3 Kerangka Pikir