***LAPORAN PENELITIAN***

**SISTEM PAKAR CERTAINTY FACTOR DALAM MENDIAGNOSIS INDIKASI PENYAKIT KATARAK PADA ANAK**

**Program Studi : Teknik Informatika**

**Jenjang Pendidikan : Strata 1**

****

Diajukan Oleh :

**Randy Permana / 1012128701**

**Rini Sovia / 1005047601**

**Muhammad Reza / 1018119001**

**Hanippa Prima Putra**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS PUTRA INDONESIA “YPTK”**

**PADANG**

# 

# ABSTRAK

Sistem pakar merupakan salah satu bagian dari cabang ilmu kecerdasan buatan yang mengombinasikan pengalaman dan mesin inferensi untuk mendapatkan suatu pengetahuan. Dengan membangun sistem pakar, pengguna bisa mendapatkan pengetahuan yang bersumber dari sistem sama seperti bertemu dengan seorang pakar yang sebenarnya. Terdapat banyak kasus yang dapat dipecahkan oleh sistem pakar antara lain : kesehatan, pertanian, peternakan, otomotif, eletronik dan masih banyak lagi yang lainnya. Pada penelitian ini sistem pakar menerapkan metode certainty factor atau yang disebut dengan faktor kepastian, merupakan metode berbentuk metrik untuk membuktikan apakah suatu fakta yang diberikan pengguna pasti atau tidak dengan yang terdapat di dalam sistem pakar. Kasus yang akan dipecahkan oleh sistem pakar ini adalah pendeteksian secara dini terhadap penyakit katarak yang rentan menyerang bayi maupun anak di bawah umur. Penyakit katarak pada anak umumnya disebabkan oleh beberapa hal seperti kelahiran bayi secara prematur, Ibu yang mengandung mengidap rubella atau toksoplasma, dan juga termasuk makanan yang dikonsumsi oleh ibu yang sedang mengandung. Hasil dari Sistem pakar ini akan menyimpulkan apakah seorang anak menderita penyakit katarak Congenital, Juvenile, Traumatic ataupun tidak sama sekali berdasarkan gejala yang dipilih oleh seorang pengguna berdasarkan nilai keyakinan yang dimiliki oleh seorang pengguna.

**Kata Kunci : Certainty Factor, Kecerdasan Buatan, Katarak, Sistem Pakar**

DAFTAR ISI

[BAB I PENDAHULUAN 5](#_Toc44279053)

[1.1. Latar Belakang Masalah 5](#_Toc44279054)

[1.2. Perumusan Masalah 8](#_Toc44279055)

[1.3. Hipotesa 8](#_Toc44279056)

[1.4. Batasan Masalah 9](#_Toc44279057)

[1.5. Tujuan Penelitian 9](#_Toc44279058)

[1.6. Manfaat Penelitian 9](#_Toc44279059)

[BAB II LANDASAN TEORI 8](#_Toc44279064)

[2.1. Rekayasa Perangkat Lunak (RPL) 1](#_Toc44279065)1

[2.1.1 Pengertian Perangkat Lunak 1](#_Toc44279066)1

[2.1.2 Karakter Perangkat Lunak. 1](#_Toc44279067)2

[2.1.3 Aplikasi Perangkat Lunak 1](#_Toc44279068)2

[2.1.4 Proses Rekayasa Perangkat Lunak 1](#_Toc44279069)4

[2.2 Software Development Life Cycle 1](#_Toc44279071)5

[2.2.2 Tahapan Softawre Life Cycle (SDLC) 15](#_Toc44279073)

[2.2.3 Model Softawre Life Cycle (SDLC) 15](#_Toc44279073)

*[2.3.](#_Toc44279074)* [Kecerdasasn Buatan 2](#_Toc44279074)5

[2.4. Definisi Kecerdasan Buatan 26](#_Toc44279081)

[2.5. Sistem Pakar 3](#_Toc44279084)1

*[2.6.](#_Toc44279085)**[Certainty Factor](#_Toc44279085)* [2](#_Toc44279085)6

[2.7.](#_Toc44279086) *[Website](#_Toc44279086)* [2](#_Toc44279086)7

[2.8. Database 49](#_Toc44279087)

[2.9. Katarak Pada Anak 5](#_Toc44279088)2

[BAB III METODOLOGI PENELITIAN 5](#_Toc44279090)6

[3.1 Kerangka Penelitian 5](#_Toc44279091)6

[3.2 Tahapan Penelitian 5](#_Toc44279092)7

[BAB IV ANALISA DAN PERANCANGAN 6](#_Toc44279090)1

[BAB IV KESIMPULAN...........................................................................................1](#_Toc44279090)04

**[DAFTAR PUSTAKA 1](#_Toc44279128)05**

# BAB I

**PENDAHULUAN**

* 1. **Latar Belakang**

Mata adalah [organ](https://id.wikipedia.org/wiki/Organ_%28anatomi%29" \o "Organ (anatomi)) [penglihatan](https://id.wikipedia.org/wiki/Penglihatan" \o "Penglihatan). Mata mendeteksi [cahaya](https://id.wikipedia.org/wiki/Cahaya" \o "Cahaya) dan mengubahnya menjadi impuls elektrokimia pada [sel saraf](https://id.wikipedia.org/wiki/Sel_saraf" \o "Sel saraf). Pada organisme yang lebih tinggi, mata adalah sistem [optik](https://id.wikipedia.org/wiki/Optika" \o "Optika) kompleks yang mengumpulkan cahaya dari lingkungan sekitarnya, mengatur intensitasnya melalui [diafragma](https://id.wikipedia.org/wiki/Selaput_pelangi" \o "Selaput pelangi), [memfokuskan](https://id.wikipedia.org/wiki/Titik_api" \o "Titik api) melalui penyesuaikan lensa untuk membentuk sebuah gambar, mengkonversi gambar tersebut menjadi satu himpunan sinyal listrik, dan mentransmisikan sinyal-sinyal ke [otak](https://id.wikipedia.org/wiki/Otak" \o "Otak) melalui jalur saraf kompleks yang menghubungkan mata melalui [saraf optik](https://id.wikipedia.org/wiki/Saraf_optik" \o "Saraf optik) menuju [korteks visual](https://id.wikipedia.org/wiki/Sistem_penglihatan" \o "Sistem penglihatan) dan area lain dari otak.

Mata merupakan bagian dari ke lima indera tubuh kita yang sangat kompleks, dimana yang sangat membutuhkan perawatan yang baik supaya penglihatan tetap sehat, jernih, dan setajam mungkin. Walaupun tubuh kita sudah memiliki sistem sendiri untuk menjaga mata dari kerusakan, namun kita juga perlu untuk menyuplai bahan yang tepat yang dibutuhkannya. Dengan demikian, apapun yang kita makan adalah sangat penting bagi kesehatan mata kita. Jika kita kurang asupan nutrisi yang dibutuhkan, maka tubuh kita bisa tak mampu lagi untuk menangkal dan mencegah keausan akibat dari penggunaannya sehari-hari. Hal ini bisa *dimanifestasikan* menjadi masalah penglihatan umum yang mungkin bisa saja terjadi sewaktu-waktu, atau bahkan hingga kebutaan.

Katarak merupakan penyakit pada mata manusia, biasanya penyakit ini datang pada manusia usia lanjut (Manula). Penyebab katarak belum diketahui secara pasti. Seiring bertambahnya usia, protein yang membentuk lensa mata akan berubah, termasuk kandungan airnya. Inilah yang memungkinkan lensa mata yang tadinya bening, berubah menjadi keruh. Salah satu penyebab penyakit ini adalah kurangnya mengkonsumsi vitamin A dan E serta Omega3, dan sebagainya.

Katarak pada anak mempunyai beberapa jenis seperti : katarak *kongenital*, katarak *juvenil*, katarak *traumatik*. Katarak pada bayi disebut juga dengan *Katarak Kongenital* adalah merupakan masalah mata menjadi keruh yang terjadi sejak lahir, lensa mata berperan untuk menfokuskan cahaya yang masuk kedalam mata menuju retina, sehingga dengan begitu mata dapat menangkap gambar dengan jelas. Namun pada seseorang yang menderita katarak, sinar yang masuk ke matanya akan tersebar pada saat melewati lensa mata yang keruh dan menyebabkan gambar yang diterima mata tidak jelas atau kabur dan terdistorsi. Katarak *kongenital* dapat terlihat ketika lahir bagian mata hitam bayi ditutupi dengan noda berwarna keabu-abuan. Umur katarak *kongenital* dapat didiagnosa pada bayi yang baru lahir, katarak *kongenital* umumnya menyertai pada retardasi mental, tuli, penyakit ginjal, penyakit jantung dan gejala sistemik. Katarak *Juvenil* merupakan katarak lembek yang terdapat pada orang usia muda dan mulai terbentuknya pada usia tiga bulan sampai sembilan tahun. Katarak *Traumatik* merupakan katarak yang muncul sebagai akibat cedera pada mata yang dapat diakibatkan benturan benda tumpul ataupun trauma perforasi yang terlihat sesudah beberapa hari ataupun beberapa tahun.Katarak *traumatik* ini dapat muncul akut, subakut, atau pun gejala sisa dari trauma mata.

Seiring berkembangannya zaman, dibidang kedokteran telah memanfaatkan teknologi dalam upaya peningkatan pelayanan yang lebih baik dalam pendiagnosaan penyakit mata, salah satunya penyakit katarak. Katarak adalah penyebab utama kebutaan di dunia, katarak juga dapat terjadi pada bayi dan anak-anak meski kemungkinannya sangat kecil. Penyakit ini umumnya ditemukan pada orang-orang lanjut usia dan dikenal sebagai katarak manula.

Katarak *kongenital* bertanggung jawab sekitar 10% dari seluruh kehilangan penglihatan pada anak, dan diperkirakan 1 dari 250 bayi lahir memiliki beberapa bentuk katarak. Di Inggris, setiap tahunnya terdapat 200-300 kasus bayi lahir dengan katarak *kongenital*. Prevalensi katarak *kongenital* dan infantile pada negara berkembang sekitar 2-4 tiap 10.000 kelahiran hidup. Sedangkan di Indonesia sendiri, belum ada data yang signifikan tentang angka kejadian katarak *kongenital*. Teknologi di bidang kedokteran sekarang ini telah memanfaatkan sistem pakar sebagai salah satu metode dalam mendiagnosa penyakit dan sebagai sistem dalam pengambilan keputusan.

Sistem pakar adalah salah satu cabang dari kecerdasan buatan (*Artificial Intelligence*), yang merupakan suatu aplikasi komputerisasi yang berusaha menirukan proses penalaran dari seorang ahli dalam memecahkan masalah spesifik dan membuat suatu keputusan atau kesimpulan karena pengetahuannya disimpan di dalam basis pengetahuan untuk diproses pemecahan masalah. Dasar dari sistem pakar adalah bagaimana memindahkan pengetahuan yang dimiliki oleh seorang pakar ke komputer, dan bagaimana membuat keputusan serta mengambil kesimpulan berdasarkan pengetahuan itu. Sehingga para pakar nantinya bisa tahu penyebab katarak pada bayi dan membuat sebuah solusi bagi penderitanya.

Dengan permasalahan tersebut maka penulis berkeinginan untuk menuangkan permasalahan tersebut kedalam sebuah karya tulis ilmiah dalam bentuk skripsi yang berjudul **“PENERAPAN SISTEM PAKAR DIAGNOSA KATARAK PADA ANAK MENGGUNAKAN METODE CERTAINTY FACTOR”**

**1.2 Perumusan Masalah.**

Berdasarkan latar belakang yang telah diuraikan di atas, maka dapat dirumuskan beberapa rumusan masalah yaitu sebagai berikut:

1. Bagaimana mengetahui gejala awal terjadinya katarak pada anak?
2. Bagaimana cara penanggulangi katarak pada anak dan dampak buruk terhadap organ lainnya ?
3. Bagaimana cara pencarian solusi terhadap penderita katarak dan masyarakat?

**1.3 Hipotesa**

Berdasarkan perumusan masalah di atas dapat ditarik beberapa hipotesa yaitu:

1. Dengan mengetahui gejala awal penyakit katarak pada anak, diharapkan masyarakat lebih peduli terhadap kesehatan mata, terutama pada orang tua anak terhadap kesehatan mata anaknya.
2. Dengan mengetahui gejala awal tersebut, diharapkan penderita katarak *Congenital, Juvenile* dan *Traumatik* mendapatkan penanganan serius serta penanggulanginya.
3. Dengan adanya pencarian solusi diharapkan mampu memberikan dampak positif bagi penderita katarak dan masyarakat banyak.

**1.4 Batasan Masalah**

Dalam penelitian ini penulis perlu memberikan batasan masalah yang membahas tentang katarak pada anak yaitu katarak *Kongenital, Juvenil* dan *Traumatik* dengan metode *certainty factor*, mulai dari mengetahui gejala awal, penanganannya, pengobatannya, serta pembuatan *Website* dengan menggunakan Bahasa Pemprograman PHP dan *Database MySql* yang nantinya dapat dimanfaat oleh masyarakat luas.

**1.5 Tujuan Penelitian**

Berdasarkan masalah yang telah dirumuskan sebelumnya, maka tujuan penyusunan dari laporan penelitian ini adalah:

1. Untuk mengetahui penyebab dan memberikan solusi bagi penderita katarak dan masyarakat.
2. Sebagai salah satu penerapan sistem pakar pada bidang teknologi dan kedokteran.
3. Sebagai sarana pengetahuan dan informasi tentang masalah penyakit katarak pada anak.

**1.6 Manfaat Penelitian**

Penelitian ini dilakukan dengan harapan dapat memberikan manfaat, diantaranya:

1. Memudahkan para dokter mendeteksi atau mengetahui penyebab atau gejala awal penyakit katarak pada anak dan manula.
2. Agar dapat diterapkan dibidang kedokteran dalam pencarian solusi .
3. Sebagai motivasi bagi para penderita katarak bahwa penyakit tersebut dapat ditangani dan bisa disembuhkan.

# BAB II

**TINJAUAN PUSTAKA**

* 1. **Rekayasa Prangkat Lunak (Software Engineering)**

Rekayasa perangkat lunak (*software engineering*) merupakan pembangunan dengan menggunakan prinsip atau konsep rekayasa dengan tujuan menghasilkan perangkat lunak yang bernilai ekonomi yang dipercaya dan bekerja secara efisien menggunakan mesin. Perangkat lunak banyak di buat dan pada akhirnya sering tidak digunakan karena tidak memenuhi kebutuhan pelanggan atau bahkan karena masalah non-teknis seperti keenggan pemakai perangkat lunak (*user*)untuk mengubah cara kerja dari manual ke otomatis, atau ketidakmampuan *user* menggunakan komputer. Oleh karena itu, rekayasa perangkat lunak dibutuhkan agar perangkat lunak yang dibuat tidak hanya menjadi perangkat lunak yang tidak terpakai [5].

* + 1. **Pengertian Perangkat Lunak (Software)**

Perangkat lunak (*software*) adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain, dan cara penggunaan (*user manual*). Sebuah program komputer tanpa terasosiasi dengan dokumentasinya maka belum dapat disebut perangkat lunak (*software*). Sebuah perangkat lunak juga sering disebut dengan sistem perangkat lunak. Sistem berarti kumpulan kompenen yang saling terkait dan mempunyai satu tujuan yang ingin di capai.

Sistem perangkat lunak berarti sebuah sistem yang memiliki komponen berupa pernagkat lunak yang memiliki hubungan satu sama lain untuk memenuhi kebutuhan pelanggan (*customer*). Pelanggan (*customer*) adalah orang atau organisasi yang memesan atau memebli perangkat lunak (*software*) dari pengembang perangkat lunak atau bisa dianggap bahwa pelanggan (*customer*) adalah orang atau organisasi yang dengan sukarel mengeluarkan uang untuk memesan atau membeli perangkat lunak. *User*  atau pemakai perangkat lunak adalah orang yang memiliki kepentingan untuk memakai atau menggunakan perangkat lunak untuk memudahkan pekerjaannya [5].

* + 1. **Karakter Perangkat Lunak**

Karakter perangkat lunak adalah sebagai berikut [5] :

1. Perangkat lunak dibangun dengan rekayasa (*software engineering*) bukan diproduksi secara manufaktur atau pabrikan.
2. Perangkat lunak tidak pernah usang (*“wear out”*) karena kecacatan dalam perangkat lunak dapat diperbaiki.
3. Barang produksi pabrikan biasanya komponen barunya akan terus diproduksi, sedangkan perangkat lunak biasanya terus diperbaiki seirinh kebutuhan program lainnya.
   * 1. **Aplikasi Perangkat Lunak**

Aplikasi perangkat lunak terdiri dari [5] :

1. Perangkat lunak sistem (*system software*)

Adalah kumpulan program dalam hal ini program yang satu ditulis untuk memenuhi kebutuhan program lainnya.

1. Perangkat lunak waktu nyata (*real-time software*)

Merupakan perangkat lunak yang memonitor, menganalisis, mengontrol sesuatu secara waktu nyata (*real-time*). Reaksi yang dibutuhkan pada perangkat lunak harus langsung menghasilkan respon yang diinginkan.

1. Perangkat lunak bisnis (*business software*)

Merupakan perangkat lunak pengelola informasi bisnis (seperti akuntansi, penjualan, pembayaran, penyimpanan (*inventory*)).

1. Perangkat lunak untuk keperluan rekayasa dan keilmuan (*engineering and scientific software)*

Merupakan perangkat lunak yang mengimplementasikan algoritma yang terkait dengan keilmuan ataupun perangkat lunak yang membantu keilmuan, misalkan perangkat lunak di bidang astronomi, di bidan matematikka dan lain sebagainya.

1. Perangkat lunak tambahan untuk membantu mengerjakan suatu fungsi dari perangkat lunak yang lainnya (*embedded software*)

Misalnya perangkat lunak untuk mencetak dokumen ditambahkan agar perangkat lunak yang memerlukan dapat mencetak laporan, maka perangkat lunak untuk mencetak dokumen ini disebut *embedded software.*

1. Perangkat lunak komputer personal (*personal computer software*)

Merupakan perangkat lunak untuk PC misalnya perangkat lunak pemroses teks, pemroses grafik dan lain sebagainya

1. Perangkat lunak berbasis web (*web based software*)

Merupakan perangkat lunak yang dapat diakses dengan menggunakan browser.

1. Perangkat lunak berintelejensia buatan (*artificial intelligence softare*)

Merupakan perangkat lunak yang menggunakan algoritma tertentu untuk mengelola data sehingga seakan-akan memiliki intelejensia seiring bertambahnya data yang diproses.

**2.1.4 Proses Rekayasa Perangkat Lunak**

Proses perangkat lunak (*software process*) adalah sekumpulan aktifitas yang memiliki tujuan untuk mengembangkan atau mengubah perangkat lunak. Secara umum proses perangkat lunak terdiri dari [5] :

1. Pengumpulan Spesifikasi (*Specification*)

Mengetahui apa saja yang harus dapat dikerjakan sistem perangkat lunak dan batasan pengembangan perangkat lunak.

1. Pengembangan (*Development*)

Pengembangan perangkat lunak untuk menghasilkan sistem perangkat lunak.

1. Validasi (*Validation*)

Memeriksa apakah perangkat lunak sudah memenuhi keinginan pelanggan (*customer*).

1. Evolusi (*Evolution*)

Mengubah perangkat lunak untuk memenuhi perubahan kebutuhan pelanggan (*customer*).

* 1. ***Software Development Life Cycle* (SDLC)**

**2.2.1 Pengertian *Software Development Life Cycle* (SDLC)**

SDLC atau *Software Development Life Cycle* atau sering disebut juga dengan *System Development Life Cycle* adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya (berdasarkan *best practice* atau cara-cara yang sudah teruji baik) [5].

* + 1. **Tahapan *Software Development Life Cycle* (SDLC)**

Tahapan-tahapan yang ada pada SDLC secara global adalah sebagai berikut [5] :

1. Inisiasi (*Initiation*)

Tahap ini biasanya ditandai dengan pembuatan proposal proyek perangkat lunak.

1. Pengembangan Konsep Sistem (*System Concept Development*)

Mendefinisikan lingkup konsep termasuk dokumen lingkup sistem, analisis manfaat biaya, manajemen rencana, dan pembelajaran kemudahan sistem.

1. Perencanaan (*Planning*)

Mengembangkan rencana manajemen proyek dan dokumen perencanaan lainnya.Menyediakan dasar untuk mendapatkan sumber daya (*resources*) yang dibutuhkan untuk memperoleh solusi.

1. Analisis Kebutuhan (*Requirements Analysis*)

Menganalisis kebutuhan pemakai sistem perangkat lunak (*user*) dan mengembangkan kebutuhan *user*.Membuat dokumen kebutuhan fungsional.

1. Desain (*Design*)

Mentransformasikan kebutuhan detail menjadi kebutuhan yang sudah lengkap, dokumen desain sistem fokus pada bagaimana dapat memenuhi fungsi-fungsi yang dibutuhkan.

1. Pengembangan (*Development*)

Mengkonversi desain ke sistem informasi yang lengkap termasuk bagaimana memperoleh dan melakukan instalasi lingkungan sistem yang dibutuhkan, membuat basis data dan mempersiapkan prosedur kasus pengujian, mempersiapkan berkas dan *file* pengujian, pengodean, pengompilasian, memperbaiki dan membersihkan program, peninjauan-peninjauan.

1. Integrasi dan Pengujian (*Integration and Test*)

Mendemonstrasikan sistem perangkat lunak bahwa telah memenuhi kebutuhan yang dispesifikasikan pada dokumen kebutuhan fungsional.Dengan diarahkan oleh staf penjamin kualitas (*quality assurance*) dan *user*.Menghasilkan laporan analisis pengujian.

1. Implementasi (*Implementation*)

Termasuk pada persiapan implementasi, implementasi perangkat lunak pada lingkungan produksi (lingkungan pada *user*) dan menjalankan resolusi dari permasalahan yang teridentifikasi dari fase integrase dan pengujian.

1. Operasi dan Pemeliharaan (*Operations and Maintenance*)

Mendeskripsikan pekerjaan untuk mengoperasikan dan memelihara sistem informasi pada lingkungan produksi (lingkungan pada *user*), termasuk implementasi akhir pada proses peninjauan.

1. Disposisi (*Disposition*)

Mendeskripsikan aktifitas akhir dari pengembangan sistem dan membangun data yang sebenarnya sesuai dengan aktifitas *user*.

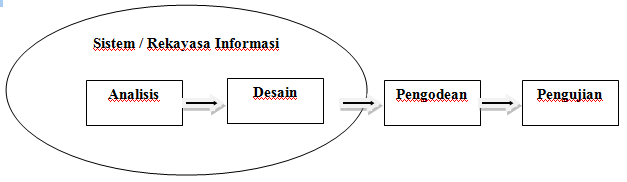
* + 1. **Model *Software Development Life Cycle* (SDLC)**

SDLC memiliki beberapa model dalam penerapan tahapan prosesnya. Beberapa model SDLC adalah [5] :

1. Model *Waterfall*

Model SDLC air terjun *(waterfall)* sering juga disebut model sekuensial linier *(sequential linear)* atau alur hidup klasik (*classic life cycle).* Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara *sekuensial* atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support).*

Berikut adalah gambar model air terjun *(waterfall)* pada Gambar 2.1 :



**Sumber :** *S. Rosa A dan M. Shalahudin,2014 : 29*

**Gambar 2.1 Model *Waterfall***

Keterangan dari gambar 2.2 dapat dijelaskan sebagai berikut :

1. Analisis kebutuhan perangkat lunak

Proses pengumpulan kebutuhan dilakukan secara *intensif* untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user.*

1. Desain

Desain perangkat lunak adalah proses *multi* langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, *representasi* antarmuka, dan prosedur pengkodean.

1. Pembuatan kode program

Desain harus ditranslasikan ke dalam program perangkat lunak.Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

1. Pengujian

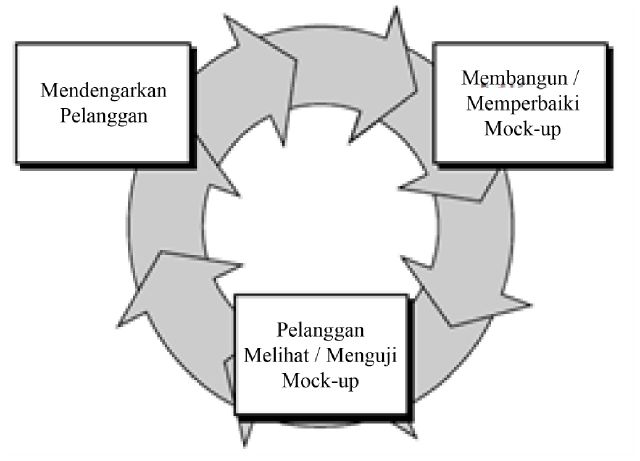
Pengujian fokus pada perangkat lunak secara dari segi lojik dan fungsional dan memastikan bahwa semua bagian sudah diuji.Hal ini dilakukan untuk meminimalisir kesalahan *(error)* dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

1. Pendukung *(support)* atau pemeliharaan *(maintenance)*

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user.*Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru.

1. Model *Prototipe*

Model *prototipe* dimulai dari mengumpulkan kebutuhan pelanggan terhadap perangkat lunak yang akan dibuat.lalu dibuatlah program *prototipe* agar pelanggan lebih terbayang dengan apa yang sebenarnya diinginkan. Program *prototipe* biasanya merupakan program yang belum jadi.Program ini biasanya menyediakan tampilan dengan simulasi alur perangkat lunak sehingga tampak seperti perangkat lunakyang sudah jadi.Program *prototipe* ini dievaluasi oleh pelanggan atau *user* sampai ditemukan *spesifikasi* yang sesuai dengan keinginan pelanggan atau *user*. Untuk lebih jelasnya Model *Prototipe* dapat dilihat pada Gambar 2.2 :



**Sumber :** *S. Rosa A dan M. Shalahudin,2014 : 32*

**Gambar 2.2 Model *Prototipe***

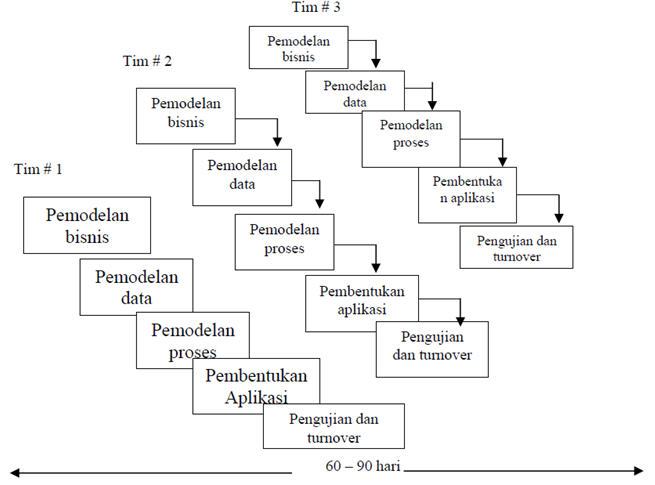
Model *prototipe* juga memiliki kelemahan sebagai berikut :

1. Pelanggan dapat sering mengubah-ubah atau menambah-tambah *spesifikasi* kebutuhan karena menganggap aplikasi sudah dengan cepat dikembangan, karena adanya *iterasi* ini dapat menyebabkan pengembang banyak mengalah dengan pelanggan karena perubahan atau penambahan *spesifikasi* kebutuhan perangkat lunak.
2. Pengembangan lebih sering mengambil kompromi dengan pelanggan untuk mendapatkan *prototipe* dengan waktu yang cepat sehingga pengembang lebih sering melakukan segala cara guna menghasilkan *prototipe* untuk didemontrasikan.

Model *prototipe* cocok digunakan untuk menjabarkan kebutuhan pelanggan secara lebih detail karena pelanggan sering kali kesulitan menyampaikan kebutuhannya secara *detail* tanpa melihat gambaran yang jelas. Untuk mengantisipasi agar proyek dapat berjalan sesuai dengan target waktu dan biaya diawal, maka seabaiknya spesifikasi kebutuhan sistem harus sudah disepakati oleh pengembang dengan pelanggan secara tertulis. Dokumen tersebut akan menjadi patokan agar *spesifikasi* kebutuhan sistem masih dalam ruang lingkup proyek.

1. Model *Rapid Application Development* (RAD)

*Rapid Application Development* (RAD) adalah model proses pengenbangan perangkat lunak yang bersifat incremental terutama untuk waktu pengerjaan yang pendek. Model RAD adalah *adaptasi* dari model air terjun *versi* kecepatan tinggi dengan menggunakan model air terjun untuk pengembangan setiap komponen perangkat lunak. Model RAD membagi tim pengembang menjadi beberapa tim untuk mengerjakan beberapa komponen masing-masing tim pengerjaan dapat dilakukan secara paralel. Untuk lebih jelasnya Model RAD dapat dilihat pada Gambar 2.3 :



**Sumber :** *S. Rosa A dan M. Shalahudin,2014 : 35*

**Gambar 2.3 Model RAD**

Berdasarkan gambar 2.3, dapat dijelaskan sebagai berikut :

1. Pemodelan bisnis

Pemodelan yang dilakukan untuk memodelkan fungsi bisnis untuk mengetahui informasi apa yang terkait proses bisnis, informasi apa saja yang harus dibuat, siapa yang harus membuat informasi itu, bagaimana alur informasi itu, proses apa saja yamg terkait informasi itu.

1. Pemodelan data

Memodelkan data apa saja yang dibutuhkan berdasarkan pemodelan bisnis dan mendefinisikan atribut-atributnya beserta relasinya dengan data-data yang lain.

1. Pemodelan proses

Mengimplementasikan fungsi bisnis yang sudah didefinisikan terkait dengan pendefenisian data.

1. Pembuatan aplikasi

Mengimplementasikan pemodelan proses dan data menjadi program. Model RAD sangat menganjurkan pemakaian komponen yang sudah ada jika dimungkinkan.

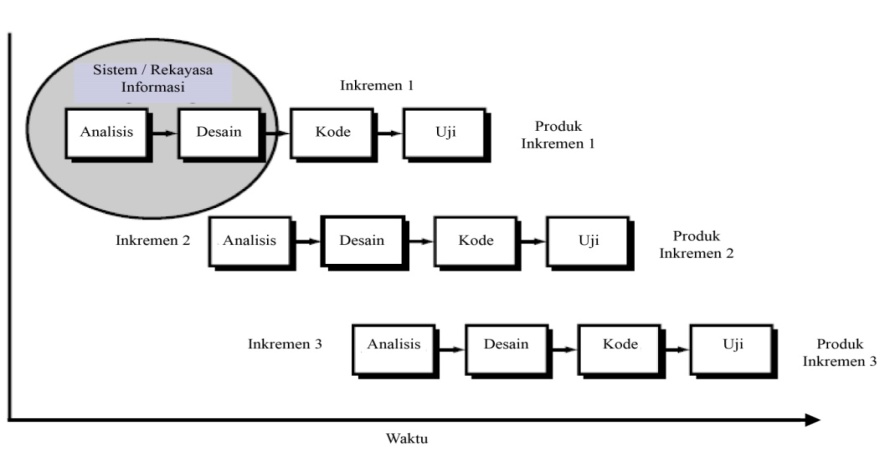
1. Pengujian dan pergantian

Menguji komponen-komponen yang dibuat. Jika sudah teruji maka tim pengembang komponen dapat beranjak untuk mengembangkan komponen berikutnya.

Model RAD memiliki kelemahan sebagai berikut :

1. Pembuatan sistem perangkat lunak dengan sekala besar memerlukan sumber daya manusia yang cukup besar untuk membentuk tim dan mengembangkan komponen.
2. Jika tidak ada persetujuan untuk mengembangkan perangkat lunak secara dengan cepat *(rapid)* maka proyek dengan model ini akan gagal karena hanya akan bingung mendefenisikan kebutuhan pelanggan.
3. Model RAD tidak cocok digunakan untuk sistem perangkat lunak memiliki risiko teknis sangat tinggi.
4. Model Iteratif

Model *iteratif* (*iteratif model)* mengkombinasikan proses-proses pada model air terjun dan *iteratif* pada model *prototipe*. Model *inkrementa*l akan menghasilkan versi-versi perangkat lunak yang sudah mengalami penambahan fungsi untuk setiap pertambahannya (inkremen/*increment)*. Untuk lebih jelasnya Model *Iteratif*  dapat dilihat pada Gambar 2.4:



**Sumber :** *S. Rosa A dan M. Shalahudin,2014 : 38*

**Gambar 2.4 Model *Iteratif***

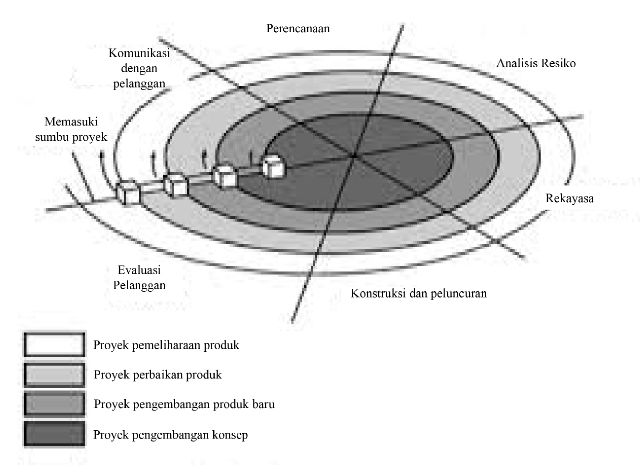
Model *inkremental* dibuat untuk mengatasi kelemahan dari model air terjun yang tidak mengakomodasi *iterasi*, dan mengatasi kelemahan dari metode *prototipe*yang memiliki proses terlalu pendek dan setiap *iteratif* prosesnya tidak selalu menghasilkan produk (bisa jadi hanya *prototipe*). Model *inkremental* menghasilkan produk/aplikasi untuk setiap tahapan *inkremental*.

Model *inkremental* sangat cocok digunakan jika staf yang dimiliki memiliki pergantian (*turnover)* yang tinggi sehingga staf tidak dapat terus ikut dalam pengembangan perangkat lunak.Mekanisme tahapan inkremental perlu direncanakan terlebih dahulu agar hasil produk dan pengerjaan setiap tahapan *inkremen* menjadi lebih baik.

1. Model *Spiral*

Model *spiral(spiral model)* memasangkan iteratif pada model *prototipe* dengan control dan aspek *sistematik* yang diambil dari model air terjun. Model *spiral* menyediakan pengembangkan dengan cara cepat dengan perangkat lunak yang memiliki *versi* yang terus bertambah fungsinya *(increment).*Model *spiral* dibagi menjadi beberapa kerangka aktifitas atau disebut juga kerja *(task region)*. Banyaknya wilayah kerja biasanya diantara tiga sampai enam wilayah sebagai berikut :

1. Komunikasi dengan pelanggan *(customer communication)*, aktifitas ini diperlukan untuk membangun komunikasi yang *efektif* antara pengembang *(developer)* dan pelanggan *(customer)*.
2. Perencanaan *(planning)*, aktifitas ini diperlukan untuk mendefenisikan sumber daya, waktu, dan informasi yang terkait dengan proyek.
3. Analisis risiko *(risk analysis)*, aktifitas ini diperlukan untuk memperkirakan risiko dari segi teknis maupun manajemen.
4. Rekayasa *(engineering)*, aktifitas ini diperlukan untuk membangun satu atau lebih *representasi* dari aplikasi perangkat lunak (dapat juga berupa *prototipe*)
5. Kontruksi dan peluncur *(construction and release)*, aktifitas ini dibutuhkan untuk mengkontraksi, menguji, melakukan *instalasi*, dan menyediakan dukungan terhadap *user* (misalnya dari segi dokumentasi dan pelatihan).
6. Evaluasi pelanggan *(customer evaluation),* aktifitas ini dibutuhkan untuk mendapatkan umpan balik berdasarkan *evaluasi representasi* perangkat lunak yang dihasilkan dari proses rekayasa dan diimplementasikan pada tahap *instalasi*. Untuk lebih jelasnya Model Spiral dapat dilihat pada Gambar 2.5 :



**Sumber :** *S. Rosa A dan M. Shalahudin,2014 : 40*

**Gambar 2.5 Model *Spiral***

**2.3 Kecerdasan Buatan (*Artificial Intelligence*)**

**2.3.1 Sejarah Kecerdasan Buatan**

Istilah Kecerdasan Buatan pertama kali ditemukan pada tahun 1956 di konferensi Darthmouth. Sejak saat itu, Kecerdasan Buatan terus dikembangkan sebab berbagai penelitian mengenai teori-teori dan prinsip-prinsipnya juga terus berkembang. Meskipun istilah Kecerdasan Buatan baru muncul tahun 1956, tetapi teori-teori yang mengarah ke Kecerdasan Buatan sudah muncul sejak tahun 1941 [3].

Tahapan-tahapan sejarah perkembangan Kecerdasan Buatan[RUS95] [3]:

1. Era komputer elektronik (1941)

Pada tahun 1941 telah ditemukan alat penyimpanan dan pemrosesan informasi. Penemuan tersebut dinamakan komputer elektronik yang dikembangkan di USA dan Jerman. Komputer pertama ini memerlukan ruangan yang luas dan ruang AC yang terpisah. Saat itu komputer melibatkan konfigurasi ribuan kabel untuk menjalankan suatu program. Hal ini sangat merepotkan para *programmer.* Pada tahun 1949,berhasil dibuat komputer yang mampu menyimpan program sehingga memnuat pekerjaan untuk memasukan program menjadi lebih mudah. Penemuan ini menjadi dasar pengembangan program yang mengarah ke Kecerdasan Buatan.

1. Masa Persiapan Kecerdasan Buatan (1943 – 1956)

Pada tahun 1943, Warren McCulloch dan Walter Pitts mengemukakan tiga hal : pengetahuan fisiologi dasar dan fungsi sel saraf dalam otak, analasis formal tentang logika proposisi(*propositional logic*), dan teori komputasi Turing. Pada tahun 1950, Nobert Wiener membuat penelitian mengenai prinsip-prinsip teori *feedback.* Penemuan ini juga merupakan awal dari perkembangan Kecerdasan Buatan. Pada tahun 1956, Jhon McCarthy yang meyakinkan Minsky, Claude Shannon dan Nathaniel Rochester untuk membantunya melakukan penelitian dalam bidang *Automata,* Jaringan Syaraf dan pembelajaran intelejensia. Hal ini menjadikan McCarthy disebut sebagai *Father of* *Kecerdasan Buatan* (Bapak Kecerdasan Buatan).

1. Awal perkembangan Kecerdasan Buatan (1952 – 1969)

Diawali dengan kesuksesan Newell dan Simon dengan sebuah dengan program yang disebut *General Problem Solver.* Program ini dirancang untuk memulai penyelesaian masalah secara manusiawi. Pada tahun 1958, McCarthy di MIT AI Lab Memo No.1 mendefinisikan bahasa pemograman tingkat tinggi yaitu LISP, yang sekarang mendominasi pembuatan program-program Kecerdasan Buatan. Pada tahun 1959, Nathaniel Rochester dari IBM dan mahasiswa-mahasiswanya mengeluarkan pogram Kecerdasan Buatan yaitu *Geometri Theorm Prover.* Program ini dapat membuktikan suatu teorema menggunakan axioma-axioma yang ada. Pada tahun 1963, program yang dibuat James Slagle mampu menyelesaikan masalah integral tertutup untuk matakuliah kalkulus. Pada tahun 1968, program analogi buatan Tom Evan menyeesaikan masalah analogi geometris yang ada pas tes IQ.

1. Perkembangan Kecerdasan Buatan melambat

Pada 10 tahun kemudian, perkembangan Kecerdasan buatan melambat. Hal ini disebabkan adanya 3 kesulitan utama yang dihadapi Kecerdasan Buatan, yaitu :

1. Masalah pertama : program-program Kecerdasan Buatan bermunculan hanya mengandung sedikit atau bahkan tidak mengandung sama sekali pengetahuan (*knowledge*) pada subjeknya.
2. Masalah kedua : banyak masalah yang harus diselesaikan oleh Kecerdasan Buatan. Karena terlalu banyak masalah yang berkaitan, maka tidak jarang terjadi kegagalan pada pembuatan program Kecerdasan Buatan
3. Masalah ketiga : ada beberapa batasan pada struktur dasar yang digunakan untuk menghasilkan perilaku intelejensia.
4. Sistem berbasis pengetahuan (1969 – 1979)

Pengetahuan adalah kekuatan pendukung Kecerdasan Buatan. Hal ini dibuktikan dengan program yang dibuat oleh Ed Feigenbaum, Bruce Buchanan dan Joshua Lederberg yang membuat program untuk memecahkan masalah struktur molekul dari informasi yang iddapat dari *spectometer* massa. Program ini dinamakan *Dendral Progrms* yang berfokus pada segi pengetahuan kimia.

1. Kecerdasan Buatan menjadi sebuah industri (1980 – 1988)

Industrialisasi Kecerdasan Buatan diawali dengan ditemukannya *expert system* (sistem pakar) yang dinamakan R1 yang mampu mengkonfigurasi sistem-sistem komputer baru. Program tersebut mulai dioperasikan di *Digital Equipment Corporaton (DEC)* pada tahun 1982. Pada tahun 1988, kelompok Kecerdasan Buatan di DEC menjalankan 40 sistem pakar.

1. Kembalinya Jaringan Syaraf Tiruan (1986 – sekarang)

Para ahli fisika seperti Hopfield (1982) menggunakan teknik-teknik mekanika statistika untuk menganalisa sifat-sifat penyimpanan dan optimasi pada jaringan syaraf. Para ahli psikolog, David Rumelhart dan Geoff Hinton, melanjutkan penelitian mengenai model jaringan syaraf pada memori. Pada tahun 1985-an sedikitnya empat kelompok riset menemukan kembali algoritma belajar propagasi bali (*Back-Propagation learning*). Algoritma ini berhasil diimplementasikan kedalam bidang ilmu komputer dan psikolog.

* + 1. **Definisi Kecerdasan Buatan**

Kecerdasan buatan berasal dari baha inggris “*Artificial Intelligence*” atau disingkat AI, yaitu *intelligence* adalah kata sifat yang berarti cerdas, sedangkan *artificial* artinya buatan. Kecerdasan buatan yang dimaksud disini merujuk pada mesin yang mampu berpikir, menimbang tindakan yang akan diambil, dan mampu mengambil keputusan seperti yang dilakukan oleh manusia [6].

Para ahli mendefinisikan Kecerdasan Buatan secara berbeda-beda tergantung pada sudut pandang mereka masing-masing. Ada yang fokus pada logika berpikir manusia saja, tetapi ada juga yang mendefinisikan Kecerdasan Buatan secara lebih luas pada tingkah laku manusia. Pada [RUS95], Stuart Russel dan Peter Norvig mengelompokan definisi kecerdasan buatan, yang diperoleh dari beberapa *textbook* yang berbeda, kedalam empat kategori, yaitu [3] :

1. ***Thinking humanly****: the cognitive meodling approach*

Pendekatan ini dilakukan dengan dua cara :

1. Melalui intropeksi : mencoba untuk menangkap pemikiran-pemikiran kita sendiri pada saat kita berpikir.
2. Melalui eksperimen-eksperimen psikologi.
3. ***Acting humanly*** *: the Turing test approach*

Pada tahun 1950, Alan Turing merancang suatu ujian bagi komputer berintelijensia untuk menguji apakah komputer tersebut mampu mengelabuhi seirang manusia yang menginterogasinya melalui *teletype* (komunika berbasis teks jarak ajuh). Jika interrogator tidak dapat membedakan yang diinterogasi adalah manusia atau komputer, maka komputer berintelijensia tersebut lolos dari *Turring tes.* Komputer tersebut perlu emilikikemampuan : *Natural Languange Processing, Knowledge Representation, Automated Reasoning, Machine Learning, Computer Vision, Robotics. Turing test* sengaja menghindari interaksi fisik antara *interrogator* dan komputerkarena simulasi fisik manusia tidak memerlukan intelijensia.

1. ***Tinking rationally*** *: the laws of thought approach*

Terdapat dua masalah dalam pendekatan ini :

1. Tidak mudah untuk membuat pengetahuan informal dan memnyatakan pengetahuan tersebut ke dalam *formal term* yang diperlukan oleh notasi logika, khususnya ketika pengetahuan tersebut memiliki kepastion kurang dari 100%.
2. Terdapat oerbedaan besar antara dapat memecahkan masalah “dalam prinsip” dan memecahkannya “ dalam dunia nyata”.
3. ***Acting rationally*** : *the rational agent approach*

Membuat inferensi yang logis merukan bagian dari suatu *rational agent*. Hal ini disebabkan satu-satunya cara untuk melakukan aksi secara rasional adalah dengan menalar secara logis. Degan menalar secara logis, maka bisa didapatkan kesimpulan bahwa aksi yang diberikan akn mencapai tujuan atau tidak. Jika mencapai tujuan, maka *agent* dapat melakukan aksi berdasarkan kesimpulan tersebut.

Definisi kecerdasan buatan yang paling tepat untuk saat ini adalah *acting rationally* dengan pendekatan *rational agent*. Hal ini berdasarkan pemikiran bahwa komputer bisa melakukan penalaran secaa login dan juga bisa melakukan aksi secara rasional berdasarkan hasil dari penalaran tersebut [3].

**2.4.3 Kecerdasan Alami dan Kecerdasan Buatan**

Jika dibandngkan dengan kecerdasan alami (kecerdasan yang dimiliki oleh manusia), kecerdasan buatan memiliki keuntungan komersial, antara lain [6]:

1. Kecerdasan buatan lebih bersifat permanen. Kecerdasan alami akan cepat mengalami perubahan.
2. Kecerdasan buatan lebih mudah diduplikasi dan disebarkan.
3. Kecerdasan buatan lebih murah dibandingkan kecerdasan alami.
4. Kecerdasan buatan bersifat konsisten.
5. Kecerdasan buatan dapat didokumentasikan.
6. Kecerdasan buatan dapat mengerjakan pekerjaan lebih cepat dibandingkan dengan kecerdasan alami.
7. Kecerdasan buatan dapat mengerjakan pekerjaan lebih baik dibandingkan dengan kecerdasan alami.

Kecerdasan alami memberikan keuntungan sebagai berikut [6]:

1. Kreatif. Pengetahuan seorang manuia selalu bertambah seiring dengan perkembangan waktu.
2. Kecerdasan alami memungkin orang menggunakan pengalaman secara langsung.
3. Pemikiran manusia dapat digunakan secara luas, sedangkan kecerdasan buatan sangat terbatas.

**2.5 Sistem Pakar (*Expert System*)**

**2.5.1 Definisi Sistem Pakar**

Menurut Stephani Halim dan Seng Hansum, Sistem pakar atau *expert system* adalah sistem yang mengambil pengetahuan manusia dan memanfaatkannya ke komputer, supaya komputer dapat menyelesaikan masalah layaknya manusia atau yang dilakukan oleh pakar pada umumnya, sehingga sistem pakar dapat menyelesaikan suatu masalah, bahkan meniru kerja dari pakar.

Sistem pakar adalah suatu sistem yang dirancang untuk menirukan keahlian seorang pakar dalam menjawab pertanyaan dan memecahkan suatu masalah. Sistem pakar akan memberikan pemecahan suatu masalah yang didapat dari dialog dengan pengguna. Dengan bantuan sistem pakar seseorang yang bukan pakar/ahli dapat menjawab pertanyaan, menyelesaikan masalah serta mengambil keputusan yang biasanya dilakukan oleh seorang pakar [6].

Menurut Imam Gunawan, Sistem pakar adalah sistem yang menggunakan pengetahuan manusia yang dimasukkan ke dalam komputer untuk memecahkan masalah-masalah yang biasanya diselesaikan oleh pakar.

Sistem pakar adalah aplikasi berbasis komputer yang digunakan untuk menyelesaikan masalah sebagaimana yang dipikirkan oleh pakar. Pakar yang dimaksud disini adalah orang yang mempunyai keahlian khusus yang dapat menyelesaikan masalah yang tidak dapat diselesaikan oleh orang awam. Sistem pakar yang mencoba memecahkan masalah yang biasanya hanya bisa dipecahkan oleh pakar, dipandang berhasil ketika mampu mengambil keputusan seperti yang dilakukan oleh pakar aslinya baik dari sisi proses pengambilan keputusan maupun hasil keputusan yang diperoleh. [2].

Sistem pakar adalah salah satu bidang ilmu komputer yang mendayangunakan komputer sehingga dapat berprilaku cerdas seperti manusia. Ilmu komputer tersebut mengembangkan perangkat lunak dan perangkat keras untuk menirukan tindakan manusia. Aktifitas yang ditirukan seperti penalaran, penglihatan, pembelajaran, pemecahan masalah, pemahaman bahasa alami dan sebagainya [1].

Secara umum, sistem pakar adalah sistem yang berusaha mengadopsi pengetahuan manusia ke komputer yang dirancang untuk memodelkan kemampuan untuk menyelesaikan masalah seperti layaknya seorang pakar. Dengan sistem pakar ini, orang awam pun dapar menyelesaikan masalahnya atau hanya sekedar mencari suatu informasi berkualitas yang sebenarnya hanya dapat diperoleh dengan bantuan para ahli di bidangnya.

Pengetahuan sistem pakar dibentuk dari kaidah atau pengalaman tentang perilaku elemen dari domain bidang pengetahuan tertentu. Pengetahuan pada sistem pakar diperoleh dari orang yang mempunyai pengetahuan pada suatu bidang(pakar bidang tertentu), buku-buku, jurnal ilmiah, majalah, maupun dokumentasi yang tercetak lainnya. Pengetahuan-pengetahuan tersebut direpresentasikan dalam format tertentu, dan dihimpun dalam suatu basis pengetahuan [1].

Beberapa pengertian sistem pakar menurut para ahli [6]:

1. Turban (2001, p402)

Sistem pakar adalah sebuah sistem yang menggunakan pengetahuan manusia dimana pengetahuan tersebut dimasukkan ke dalam sebuah komputer dan kemudian digunakan untuk menyelesaikan masalah-masalah yang biasanya membutuhkan kepakaran atau keahlian manusia.

1. Jackson (1999, p3)

Sistem pakar adalah program komputer yang mempresentasikan dan melakukan penalaran dengan pengetahuan beberapa pakar untuk memecahkan masalah atau memberikan saran.

1. Luger dan Stubblefield (1993, p308)

Sistem pakar adalah program yang berbasis pengetahuan yang menyediakan solusi ‘kualitas pakar’ kepada masalah-masalah dalam bidang (domain) yang spesifik.

**2.5.2 Manfaat Sistem Pakar**

Sistem pakar menjadi sangat populer karena banyak kemampuan dan manfaat yang diberikan, diantaranya [6]:

1. Menigkatkan produktivitas, karena sistem pakar dapat bekerja lebih cepat dari pada manusia.
2. Membuat seorang yang awam bekerja seperti layaknya seorang pakar.
3. Menigkatkan kualitas, dengan memberikan nasehat yang konsisten dan mengurangi kesalahan.
4. Mampu menangkap pengetahuan dan kepakaran seseorang.
5. Dapat beroperasi dilingkungan yang berbahaya.
6. Memudahkan akses pengetahuan seorang pakar.
7. Andal. Sistem pakar tidak pernah menjadi bodan dan kelelahan atau sakit.
8. Meningkatkan kapabilitas sistem komputer.
9. Mampu bekerja dengan informasi yang tidak lengkap atau tidak pasti.
10. Bisa digunakan sebagai media pelengkap dalam pelatihan.
11. Meningkatkan kemampuan untuk menyelesaikan masalah karena sistem pakar mengambil sumber pengetahuan dari banyak pakar.

**2.5.3 Ciri-ciri Sistem pakar**

Ciri-ciri sistem pakar adalah sebagai berikut [6]:

1. Terbatas pada domain keahlian tertentu.
2. Dapat memberikan penalaran untuk data-data yang tidak lengkap atau tidak pasti.
3. Dapat menjelaskan alasan-alasan dengan cara yang dapat dipahami.
4. Bekerja berdasarkan kaidah */ rule* tertentu.
5. Mudah dimodifikasi.
6. Basis pengetahuan dan mekanisme inferensi terpisah.
7. Keluarannya bersifat anjuran.
8. Sistem dapat mengaktifkan kaidah secara searah yang sesuai, dituntun oleh dialog dengan pengguna.

**2.5.4 Konsep Dasar Sistem Pakar**

Konsep dasar sistem pakar meliputi enam hal berikut ini [6]:

**2.5.4.1 Kepakaran (*Expertise*)**

Kepakaran merupakan suatu pengetahuan yang diperoleh dari pelatihan, membaca, dan pengalaman. Kepakaran inilah yang memungkin para ahli dapat mengambil keputusan lebih cepat dan lebih baik dari pada seseorang yang bukan pakar [6].

Kepakaran itu sendiri meliputi pengetahuan tentang :

1. Fakta-fakta tentang bidang permasalahan tertentu.
2. Teori-teori tentang bidang permasalahan tertentu.
3. Aturan-aturan dan prosedur-prosedur menurut bidang permasalahan umumnya.
4. Aturan *heuristic* yang harus dikerjakan dalam suatu situasi tertentu.
5. Strategi global untuk memecahkan permasalahan.
6. Pengetahuan tetang pengetahuan (*meta knowledge*).

**2.5.4.2 Pakar (*Expert*)**

Pakar adalah seseorang yang mempunyai pengetahuan, pengalaman, dan metode khusus, serta mampu menerapkan untuk memecahkannya untuk memecahkan masalah atau memeberikan nasehat. Seorang pakar harus mampu menjelaskan dan mempelajari hal-hal yang baru yang berkaitan dengan topik permasalahan, jika perlu harus mampu menyusun kembali pengetahuan-pengetahuan yang didapatkan, dan dapat memecahkan aturan-aturan serta menentukan relevansi kepakaran [6].

Jadi seorang pakar harus mampu melakukan kegiatan-kegiatan berikut [6]:

1. Mengenali dan memformulasikan permasalahan
2. Memecahkan permasalahan secara cepat dan tepat.
3. Menerangkan pemecahannya.
4. Belajar dari pengalaman.
5. Merestrukturisasi pengetahuan.
6. Memecahkan aturan-aturan.
7. Menentukan relevansi.

**2.5.4.3 Pemindahan Kepakaran (*Transferring Expertise)***

Tujuan dari sistem pakar adalah memindahkan kepakaran dari seorang pakar ke dalam komputer, kemudian ditransferkan kepada orang lain bukan pakar. Proses ini melibatkan empat kegiatan, yaitu [6]:

1. Akuisisi pengetahuan (dari pakar atau sumber lain).
2. Representasi pengetahuan (pada komputer).
3. Inferensi pengetahuan.
4. Pemindahan pengetahuan ke pengguna.

**2.5.4.4 Inferensi (*Inferencing)***

Inferensi adalah sebuah prosedur (program) yang mempunyai kemampuan dalam melakukan penalaran. Inferensi ditampilkan pada suata komponen yang disebut mesin inferensi yang mencakup prosedur-prosedur mengenai pemecahan masalah. Semua pengetahuan yang dimiliki oleh seorang pakar disimpan pada basis pengetahuan oleh sistem pakar. Tugas mesin inferensi adalah mengambil kesimpulan berdasarkan basis pengetahuan yang dimilikinya [6].

**2.5.4.5 Aturan-aturan (*Rule*)**

Kebanyakan software sistem pakar komersial adalah sistem yang berbasis *rule* (*rule-based syste*), yaitu pengetahuan disimpan terutama dalam bentuk *rule,* sebagai prosedur-prosedur pemecahan masalah [6].

**2.5.4.6 Kemampuan Menjelaskan (*Explanation Capability*)**

Fasilitas lain dari sistem pakar adalah kemampuannya untuk menjelaskan saran atau rekomendasi yang diberikannya. Penjelasan dilakukan dalam subsistem yang disebut subsistem penjelesan (*explanation*). Bagian dari sistem ini memungkinkan sistem untuk memeriksa penalaran yang dibuatnya sendiri dan menjelaskan operi-operasinya [6].

**2.5.5 Struktur Sistem Pakar**

Ada dua bagian penting dari sistem pakar, yaitu lingkungan pengembangan (*development environment*) dan lingkungan konsultasi (*consultation environment*). Lingkungan pengembangan digunakan oleh pembuat sistem pakar untuk membangun komponen-komponennya dan memperkenalkan pengetahuan ke dalam *knowledge*  *base* (basis pengetahuan). Lingkungan konsultasi digunakan oleh pengguna untuk berkonsultasi sehingga pengguna mendapatkan pengetahuan dan nasihat dari sistem pakar layaknya berkonsultasi dengan seorang pakar [6].

**2.5.6 Komponen Sistem Pakar**

Sistem pakar sebagai sebuah program yang difungsikan untuk menirukan pakar manusia harus bisa melakukan hal-hal yang dapat dikerjakan oleh seorang pakar. Unutk membangun sistem yang seperti itu maka komponen-komponen yang harus dimiliki adalah sebagai berikut [1] :

1. Antar Muka Pengguna (*User Interface*)

Sistem pakar mengantikan seorang pakar dalam suatu situasi tertentu, maka sistem harus menyediakan pendukung yang diperlukan oleh pemakai yang tidak memahami masalah teknis. Sistem pakar juga menyediakan komunikasi antara sistem dan pemakainya, yang disebut sebagai antar muka. Antar muka yang efektif dan ramah pengguna (*user-friendly*) penting sekali terutama bagi pemakai yang tidak ahli dalam bidang yang diterapkan pada sistem pakar.

1. Basis Pengetahuan (*Knowledge Base*)

Basis pengetahuan merupakan kumpulan pengetahuan bidang tertentu pada tingkatan pakar dalam format tertentu. Basis pengetahuan bersifat dinamis, bisa berkembang dari waktu ke waktu. Perkembangan ini disebabkan karena pengetahuan selalu bertambah. Pada sistem pakar basis pengetahuan terpisah dari mesin inferensi. Pemisahaan ini bermanfaat untuk pengembangan sistem pakar secara leluasa disesuaikan dengan perkembangan pegetahuan pada suatu domain. Penambahan dan pengurangan dapat dilakukan pada basis pengetahuan ini tanpa menggangu mesin inferensi.

1. Mekanisme Inferensi (*Inference Machine*)

Mesin inferensi merupakan otak dari sistem pakar, berupa perangkat lunak yang melakukan tugas inferensi penalaran sistem pakar, bisa dikatakan sebagai mesin pemikiran (*Thinking Machine*). Pada prinsipnya mesin inferensi inilah yang akan mencari solusi dari suatu permasalahan. Mesin inferensi sesungguhnya adalah program komputer yang menyediakan metodologi untuk melakukan penalaran tentang informasi pada basis pengetahuan dan pada memori kerja, serta untuk merumuskan kesimpulan-kesimpulan.

1. Memori Kerja (*Working Memory*)

Merupakan bagian dari sistem pakar yang menyimpan fakta-fakta yang diperoleh saat dilakukan proses konsultasi. Fakta-fakta inilah yang nantinya akan diolah oleh mesi inferensi berdasarkan pengetahuan yang disimpan dalam basis pengetahuan untuk menentukan suatu keputusan pemecahan masalah. Konklusinya bisa berupa hasil diagnosa, tindakan, akibat.

Untuk menjadikan sistem pakar menjadi lebih menyerupai seorang pakar yang berinteraksi dengan pemakai, maka dilengkapi dengan fasilitas berikut [1] :

1. Fasilitas Penjelasan (*Explanation Facility*)

Proses menentukan keputusan yang dilakukan oleh mesin inferensi selama sesi konsultasi mencerminkan proses penalaran seorang pakar. Bentuk penjelasannya dapat berupa keterangan yang diberikan setelah suatu pernyataan diajukan, yaitu penjelasan atas pertanyaan mengapa, atau penjelasan atas pertanyaan bagaimana sistem mencapai konklusi. Tujuan adanya fasilitas penjelasan dalam sistem pakar antara lain membuat sistem menjadi lebih cerdas, menunjukan adanya proses analisa dan yang tidak kalah pentingnya adalah memuaskan psikologi pemakai. Beberapa sistem pakar saat ini mempunyai sistem penjelasan yang berupa daftar kaidah yang digunakan salama eksekusi.

1. Fasilitas Akuisisi Pengetahuan (*Knowledge Acquisition Facility*)

Pengetahuan pada sistem pakar dapat ditambahkan kapan saja pengetahuan baru diperoleh atau saat pengetahuan yang sudah ada tidak berlaku lagi. Akusisi pengetahuan adalah proses pengumpulan, perpindahan, dan tranformasi dari keahlian/kepakaran pemecahan masalah yang beasak dari beberapa sumber pengetahuan ke dalam bentuk yang dimengerti oleh komputer. Dengan adanya fasilitas ini pada sistem, maka seorang pakar akan dengan mudah menambahkan pengetahuan ataupun kaidah baru pada sistem pakar. Untuk menjamin bahwa pengetahuan pada sistem pakar ini *up to date* dan valid, maka fasilitas akuisisi pengetahuan hanya bisa diakses oleh pakar. Pengguna awam tidak berhak memakai fasilitas akuisisi pengetahuan.

**2.5.7 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] :

1. Pakar (*expert*)

Pakar adalah seorang individu yang memiliki pengetahuan khusus, pemahaman, pengalaman, dan metode-metode yang digunakan untuk memecahkan persoalan dalam bidang tertentu. Selain itu, pakar juga memiliki kemampuan untuk mengaplikasikan pengetahuannya dan memberikan saran serta pemecahan masalah pada domain tertentu. Ini merupakan pekerjaan pakar, memberikan pengetahuan tentang bagaimana seseorang melaksanakan tugas utuk menyelsaikan masalah. Penyelesaian masalah ini didukung atau bahkan secara ekstrim akan dilaksakan oleh sistem basis pengetahuan/sistem pakar.

1. Pembangun pengetahuan (*knowledge engineer*)

Pembangunan pengetahuan memiliki tugas utama menterjemahkan 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 pakar. Dalam hal ini pembangunan pengetahuan (*knowledge engineer*) menginterprestasikan, dan merepresentasikan pengetahuan yang diperoleh dalam bentuk jawaban-jawaban atas pertanyaan-pertanyaan yang diajukan pada pakar, atau pemahaman, pengambaran analogis, sistematis, konseptual yang diperoleh dari membaca beberapa dokumen cetak seperti *text book*, jurnal, makalah, dan sebagainya.

1. Pembangun sistem (*system engineer*)

Pembangunan sistem adalah orang yang bertugas untuk merancang antar muka 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. Alat pembangun (*tool builder*) dapat dipakai untuk meyajikan dan 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.

1. Pengguna (*user*)

Banyaknya sistem berbasis komputer mempunyai susunan pengguna tunggal. Hal ini berbeda jauh dengan sistem pakar yang memungkinkan mempunyai beberapa kelas pengguna. Pengguna memungkinkan tidak terbiasa dengan komputer dan mungkin pada domain masalah. Bagaimanaun juga, banyak solusi permasalahan menjadi lebih baik dan kemungkinan lebih murah dan keputusan yang cepat bila menggunakan sistem pakar. Pakar dan pembangunan sistem harus mengantisipasi kebutuhan-kebutuhan pengguna dan membuat batasan-batasan ketika mendesain sistem pakar.

**2.5.8 Kategori Problem dan Aplikasi Sistem Pakar**

Banyak permasalahan yang dapat diangkat menjadi aplikasi sistem pakar. Pembuatan sistem pakar diawali sejak tahun 1965 dengan mulai dibuatnya DENDRAL. DENDRAL merupakan apliaksi sistem pakar di bidang kimia yang digunakan untuk menganalisis dengan mengambil domain pada berbagai bidang,misalnya : medis, geologi, elektonika,teknik, dan computer [1].

Saat ini, aplikasi sistem pakar masih banyak dikembangkan dalam berbagai bidang, misalnya bidang medis, agronomi, pertanian, perpajakan, perekonomian.pada dasarnya di setiap bidang terdapat pengetahuan, pengetahuan ini dikelola dan disimpan ke dalam komputer untuk dipakai oleh komputer melakukan penalaran dalam menyelesaikan permasalahan [1].

**2.6 *Certainty Factor* (Faktor Kepastian)**

*Certainty Factor* (Faktor Kepastian) diperkenalkan oleh Shortliffe Buchanan dalam pembuatan MYCIN (Wesley, 1984). *Certainty factor* (CF) merupakan nilai parameter klinis yang diberikan MYCIN untuk menunjukkan besarnya kepercayaan [2].

*Certainty factor* adalah suatu metode untuk membuktikan apakah suatu fakta itu pasti ataukah tidak pasti yang berbentuk *metric* yang biasanya digunakan dalam sistem pakar. Metode ini sangat cocok untuk sistem pakar yang mendiagnosis sesuatu yang belum pasti. (Joseph Giarratano, 2004) [17].

*Certainty Factor* (*Theory*) ini diusulkan oleh Shortliffe dan Buchanan pada tahun 1975 untuk mengakomadasi ketidakpastian pemikiran (*inexact reasoning*) seorang pakar. Teori ini berkembang bersamaan dengan pembuatan sistem pakar MYCIN. Tim pengembang MYCIN mencatat bahwa dokter sering kali menganalisa informasi yang ada dengan ungkapan seperti misalnya: mungkin, kemungkinan besar, hampir pasti, dan sebagainya. Untuk mengakomodasi hal ini tim MYCIN menggunakan *certainty factor* (CF) guna menggambarkan tingkat keyakinan pakar terhadap masalah yang sedang dihadapi [18].

Teori *Certainty Factor* (CF) diusulkan oleh Shortliffe dan Buchanan pada tahun 1975 untuk mengkombinasi ketidakpastian pemikiran (*inexact reasoning*) seorang pakar. Seorang pakar, (misalnya dokter) sering kali menganalisis informasi yang ada dengan ungkapan seperti “mungkin”, “kemungkinan besar”, “hampir pasti”. Untuk mengakomodasi hal ini kita menggunakan *certainty factor* (CF) guna menggambarkan tingkat keyakinan pakar terhadap masalah yang sedang dihadapi [6].

Formulasi *Certainty Factor* :

CF (Rule) = MB(H, E) – MD(H, E)

MB(H, E) = { (max[P(H | E), P(H)] – P(H)) / (max[1, 0] – P(H))

MD(H, E) = { (min[P(H | E), P(H)] – P(H)) / (min[1, 0] – P(H))

Dimana :

CF(Rule) = faktor kepastian

MB(H, E) = *measure of belife* (ukuran kepercayaan) terhadap hipotesis H, jika diberikan *evidence* E (antara o dan 1)

MD(H, E) = *measure of disbelife* (ukuran ketidakpastian) terhadap *evidence* H, jika diberikan *evidence* E (antara 0 dan 1)

P(H) = probabilitas kebenaran hipotesis H

P(H | E) = probabilitas bahwa H benar karena fakta E

**Interpretasi *Certainty Factor***

Nilai CF (Rule) didapat dari interprestasi “term” dari pakar, yang diubah menjadi nilai CF tertentu sesuai tabel berikut :

**Tabel 2.8 Nilai Interprestasi**

|  |  |  |
| --- | --- | --- |
| **No** | ***Uncertain Term*** | **CF** |
| 1.  2.  3.  4.  5.  6.  7.  8.  9. | *Definitely not* (pasti tidak)  *Maybe not* (mungkin tidak)  *Unknow* (tidak tahu)  *Probably* (kemungkinan besar tidak)  *Maybe* (mungkin)  *Maybe yes* (mungkin iya)  *Most likely yes* (kemungkinan besar iya)  *Almost certainty* (hampir pasti)  *Defiitely* (pasti) | 0  0.2  0.3  0.4  0.5  0.6  0.7  0.8  1.0 |

**Sumber :** *(Sutojo. T. dkk, 2011 : 198)*

**2.6.1 Kelebihan dan Kekurangan Metode *Certainty Factor***

Kelebihan metode *Certainty Factor* adalah [6]:

1. Metode ini cocok dipakai dalam sistem pakar yang mengandung ketidakpastian.
2. Dalam sekali proses perhitungan hanya dapat mengelolah 2 data saja sehingga keakuratan data dapat terjaga.

Kekurangan metode *Certainty Factor* adalah [6]:

1. Pemodelan ketidakpastian yang menggunakan perhitungan metode *certainty factor* biasanya masih diperdebatkan.
2. Untuk data lebih dari 2 buah, harus dilakukan beberapa kali pengolahan data.

**2.7 *Website***

**2.7.1 Pengertian *Website***

*World Wide Web* (WWW) atau lebih sering dikenal sebagai *web* adalah sesuatu layanan sajian informasi yang menggunakan konsep *hyperlink* (tautan), yang memudahkan *surfer* (sebutan para pemakai komputer yang melakukan *browsing* atau penelusuran informasi melalui internet). Keistimewaan inilah yang menjadikan *web* sebagai *service* yang paling cepat pertumbuhannya [8].

Makna *website* adalah sekumpulan halaman informasi yang disediakan melalui jalur internet sehingga bisa di akses diseluruh dunia selama terkoneksi dengan jaringan internet. *Website* merupakan sebuah komponen yang terdiri dari teks, gambar, suara animasi sehingga menjadi media informasi yang menarik untuk dikunjungi oleh orang lain. Bisa dipahami bahwa definisi *website*  secara sederhana adalah informasi apa saja yang bisa diakses dengan menggunakan koneksi jaringan internet [4].

Situs-situs *web* menampilkan informasi apapun secara cepat, mudah dan murah, seperti informasi, teknologi, olahraga, *entertainment* dan lain-lain, sehingga manusia cenderung menggunakan *web* sebagai agen informasi terbesar saat ini. *Web* bukan saja digunakan untuk menampilkan informasi secara tepat, namun fungsi *web* makin meluas. *Web* mulai digunakan untuk aplikasi yang banyak membutuhkan interaksi dari user. Untuk membuat situs *web* yang menarik dan informative diperlukan suatu aplikasi untuk menggantikan *Hypertext Markup Language* *(HTML)* konvensional yang bersifat statis. [19]

Sebuah *website* biasanya ditempatkan setidaknya pada sebuah *server web* yang dapat diakses melalui jaringan seperti internet, ataupun *Local Area Network* [LAN]. Meskipun setidaknya halaman utama dari sebuah *website* umumnya dapat diakses publik secara bebas, pada praktiknya tidak semua situs memberikan kebebasan bagi publik untuk mengaksesnya, beberapa *website*  mewajibkan pengunjung untuk melakukan pendaftaran sebagai anggota [4].

Halaman dari seuah *website* merupakan berkas yang ditulis sebagai berkas teks biasa (*plain text*) yang diatur dan dikombinasi sedemikian rupa dengan instruksi-instruksi berbasis HTML (*Hypertext Markup Language*). Halaman-halaman *website*  tersebut diakses oleh pengguna melalui protokol komunikasi jaringan yang disebut sebagai *Hypertext Tranfer Protocol* (HTTP), sebagai tambahan untuk menigkatkan aspek keamanan dan aspek privasi yang lebih baik [4].

**2.7.2 Jenis-jenis *Website***

*Website* digolongkan menjadi tiga jenis, yaitu [4] :

1. *Website* Statis

*Website* statis adalah *website* yang memiliki halaman yang tidak berubah. Artinya adalah untuk melakukan perubahan pada suatu halaman dilakukan secara manual dengan mengubah kode menjadi struktur dati situs itu.

1. *Website* Dinamis

*Website* dinamis merupakan *website* yang secara spesifik didesain agar isi yang terdapat dalam situs tersebut dapat diperbarui secara berkala dengan mudah. Sesuai dengan namanya, isi yang terkandung didalam *website* ini umumnya akan berubah setelah melewati satu periode tertentu.

1. *Website* Interakti

*Website* interaktif adalah *website* yang saat ini memang sedang banyak digemari. Di *website* ini *user* bisa berinteraksi dan beradu argumen mengenai apa yang akan terjadi pemikiran mereka. Biasanya *website* seperti memiliki moderator untuk mengatur supaya topik yang diperbincangkan tidak melenceng dari topik pembicaraan.

**2.7.3 Aplikasi *Website***

Aplikasi web adalah jenis aplikasi yang diakses melalui *browsing,* misalnya Internet Explorer dan Mozilla Firefox. Kepopuleran internet diseluruh penjuru dunia mendorong apliaksi web semakin diminati. Dengan menggunakan aplikasi web, seseorang hanya perlu menempatkan apliaksi dalam sebuah *server* dan dengan sendirinya aplikasi tersebut dapat diakses dimananpun, sepanjang pemakai dapat mengakses *web* *server*-nya [8].

**2.7.4 HTML**

HTML adalah suatu bahasa untuk menampilkan informasi dengan lebih menarik dibandingkan dengan tulisan teks biasa (*plain text*) [11].

*Hyper Text Markup Language* (HTML) adalah bahasa standar untuk membuat halaman-halaman [8].

**2.7.4.1 Pengenalan Dasar Kode HTML**

Kode HTML memiliki aturan dan struktur penulisan tersendiri yang disebut tag HTML. Tag adalah kode yang digunakan untuk memoles (*mark-up*) teks ASCII (*Ameican Standard Code For Information Interchange*) menjadi file HTML. Setiap tag diapit dengan tanda kurung runcing. Ada tad pembuka yaitu <html> dan ada tag penutup yaitu </html> yang ditadai dengan tanda garis miring (*slash*) didepan awal tulisannya.

Struktur kode HTML :

<html>

<head>

<title></title>

</head>

<body></body>

</html>

* + 1. **Program Web**

**2.7.5.1 Definisi PHP**

PHP adalah akronim dari *Hypertext Preprocessor*, yaitu suatu bahasa pemograman berbasis kode-kode (*scrip*) yang digunakan untuk mengelolah suatu data dan mengirimkannya kembali ke web *browser* menjadi kode HTML [11].

Kode PHP mempunyai ciri-ciri khusus, yaitu :

1. Hanya dapat dijalankan menggunakan *web server*, misalnya : apache
2. Kode PHP diletakan dan dijalankan di *web server*.
3. Kode PHP dapat digunakan untuk mengakses *database*, seperti : MySQL, PostgreSQL, Oracle, dan lain-lain.
4. Merupakan software yang bersifat *open source*.
5. Gratis untuk di-download dan digunakan.
6. Memiliki sifat multiplatform, artinya dapat dijalankan menggunakan sistem operasi apapun, seperti : Linux, Unix, Windows, dan lain-lain.

PHP adalah perangkat lunak yang bersifat *free* (gratis). Namun perlu diketahui, PHP terkadang dikemas dalam bundel perangkat lunak, misalnya pada WAMP5. Hal yang menarik lainnya adalah PHP bersifat multiplatform. Artinya,PHP dapat berjalan pada berbagai sistem, seperti Windows, Linux, dan Unix [8].

**2.8 Database**

**2.8.1 Definisi Database**

*Database* sering didefnisikan sebagai kumpulan data yang terkait. Secara teknis, yang berada dalam sebuah *database* adalah sekumpulan tabel atau objek lain (*index, review,* dan lain-lain). Tujuan utama pembuatan *database* adalah untuk memudahkan dalam mengakses data. Data dapat ditambahkan, dihabpus, atau dibaca dengan relatif mudah dan cepat [8].

Basisdata, menurut Stephens dan Plew (2000), adalah mekanisme yang digunakan untuk menyimpan informasi atau data. Informasi adalah sesuatu yang kita gunakan sehari-hari untuk berbagai alasan. Dengan basisdata, pengguna dapat menyimpan data secara terorganisasi [7].

**2.8.2 Arsitektur Basisdata**

Arsitektur sistem basisdata sangat dipengaruhi oleh sistem komputer dimana ia dijalankan, terutama oleh aspek arsitektur komputer komputer seperti jaringan, paralelisme, dan distribusi [7] :

1. Jaringan komputer memungkinkan beberapa tugas dieksekusi pada sistem *server* dan beberapa lainnya pada sistem *client*. Pembagian pekerjaan memunculkan sistem basisdata *client-server.*
2. Pemrosesan paralel dalam sistem komputer memungkinkanaktifitas sistem basisdata dipercepat, respons lebih cepat pada transaksi, serta lebih banyak transaksi yang dapat direspons per detik. Query dapat diproses dengan mengeksploitasi paralelisme yang ditawarkan oleh sistem komputer. Kebutuhan akan pemrosesan query paralel memunculkan sistem basisdata paralel.
3. Distribusi data keseluruh atau separtemen dalam organissasi memungkinkan data berada ditempat dimana mereka dibuat atau paling dibutuhkan, tetapi masih tetap dapat diakses dari tempat lain dan dari departemen lain.

**2.8.3 Arsitektur Database**

Sistem pada *web* terdiri dari dua macam yaitu  *web browser* dan *web server.* Antara keduanya membentuk hubungan komunikasi yang saling tergantung. Sebuah *web browser* membuat *reques* (permintaan) kepada *web server* dan sebaliknya *web server* memberikan *respons* (pelayanan kepada *web browse [8].*

Aplikasi web berjalan pada protokol HTTP dan semua protocol yang ada melibatkan server dan *client*. Membangun aplikasi *web* dibutuhkan sebuah server *web*, ketika seseorang mengetik alamat *web browser*  maka *browser* akan mengirim oerintah tersebut ke *web server*. *Web server* berguna sebagai tempat apikasi *web* dan sebagai penerima *request* dan *client.*

**2.8.4 Pengenalan Database MySQL**

Saat ini banyak perangkat lunak yang ditujukan untuk mengelola *database*. Perangkat lunak seperti itu biasa dinamakan DBMS (*Database Management System*). *Acces, MS SQL Server,* dan *MySQL* merupakan segelintir contoh produk pengelola *database*. Beberapa diantaranya berkelas *database server*, yaitu jenis yang secara aktif memantau permintaan akses terhadap data. Dalam hal ini, *databse server* akan segera menanggapi permintaan data. Adapun yang bukan termasuk *database server* adalah *Acces.*

MySQL adalah program apliaksi *database* yang berbasis *open source*. MySQL mampu menangani *database* yang kompleks dan cukup besar. MySQL juga dapat menangani *database* *client server [4].*

Pengaksesan data dalam *database* dapat dilakukan dengan mudah melalui SQL (*Structured Query Language*). Data dalam *databse* bisa diakses melalui aplikasi ­*non-web* (misalnya dengan visual basic) maupun aplikasi *web* (misalnya dengan PHP) [8].

**2.9 Katarak Pada Anak**

**2.9.1 Definisi Mata Dan Katarak**

Mata adalah alat indra penglihatan dibentuk untuk menerima rangsangan, berkas-berkas cahaya pada *retina* dengan perantara mengalihkan rangsangan ini kepusat penglihatan pada otak, bagian mata berfungsi memfokuskan rangsangan cahaya ke *retina* adalah *lensa* (Wijaya dan Putri,2013) [15].

Mata merupakan suatu panca indra yang sangat penting dalam kehidupan manusia untuk melihat. Jika mata mengalami gangguan atau penyakit mata, maka akan berakibat sangat fatal bagi kehidupan manusia. Jadi sudah mestinya mata merupakan anggota tubuh yang perlu dijaga dalam kesehatan sehari-hari [14].

Mata adalah salah satu indera yang penting bagi manusia, melalui mata manusia menyerap informasi visual yang digunakan untuk melaksanakan berbagai kegiatan. Namun, gangguan terhadap penglihatan banyak terjadi, mulai dari gangguan ringan hingga gangguan yang berat yang dapat mengakibatkan kebutaan (Kemenkes, 2014) [13].

Meskipun sangat penting, seringkali kita lupa untuk merawatnya secara baik yang dikarenakan kurangnya pengetahuan masyarakat untuk mencegah penyakit itu. Selain itu, terbatasnya sarana pelayanan kesehatan mata di puskesmas dan rumah sakit, serta kurangnya tenaga dokter spesialis mata yang bisa memeriksa dan melakukan operasi mata, membuat gangguan mata tak tertangani sejak dini [14].

Katarak merupakan kekeruhan yang terjadi pada lensa mata, dimana pada keadaan normal jernih. Katarak pada anak dapat terjadi sejak lahir atau terlihat segera setelah lahir yang disebut dengan katarak kongenital, katarak yang terjadi pada tahun pertama disebut katarak infantil, sedangkan yang berkembang selama 12 tahun pertama disebut dengan katarak juvenil.

Insiden katarak pada anak dinegara berkembang lainnya seperti Singapura diperkirakan sekitar 1:5000 sampai 1: 10,000 pada setiap kelahiran. insiden ini kemungkinan dapat meningkat menjadi 1: 1000 kelahiran [16].

Katarak ditandai dengan adanya *lensa* mata yang berangsur-angsur menjadi buram yang pada akhirnya dapat menyebabkan kebutaan total. Penyakit katarak terutama disebabkan oleh proses *degenerasi* yang berkaitan dengan usia. Katarak kini masih menjadi penyakit paling dominan pada mata dan merupakan penyebab utama dari kebutaan di seluruh dunia. Paling sedikit 50% dari semua kebutaan disebabkan oleh katarak, dan 90% diantaranya terdapat di negara berkembang tidak terkecuali di Indonesia (Tana.L, 2007) [15].

Katarak yang merupakan penyebab utama berkurangnya penglihatan di dunia diperkirakan jumlah penderita kebutaan katarak di dunia saat ini sebesar 17 juta orang dan akan meningkat menjadi 40 juta pada tahun 2020. Katarak terjadi 10% orang Amerika Serikat dan prevalensi ini meningkat sampai sekitar 50% untuk mereka yang berusia antara 65 dan 74 tahun. Dan sampai sekitar 70% untuk mereka yang berusia lebih dari 75 tahun (Soehardjo,2004) [15].

Katarak merupakan suatu keadaan di mana lensa mata yang biasanya jernih dan bening menjadi keruh. Faktor penyebab dan proses terjadinya katarak: [14]

1. Katarak juvenile: katarak yang terlihat pada usia di atas 1 tahun dan di bawah 50 tahun. Merupakan lanjutan dari katarak kongenital yang makin nyata.
2. Katarak kongenital: katarak yang terlihat pada usia di bawah 1 tahun**/** Kekeruhan lensa yang didapatkan sejak lahir.
3. Katarak Traumatik: katarak yang disebabkan oleh akibat cedera pada mata yang dapat merupakan trauma perforasi ataupun tumpul yang terlihat sesudah beberapa hariataupun beberapa tahun.

Lensa mata merupakan struktur globular yang transparan, terletak di belakang iris, di depan badan kaca. Bagian depan ditutupi kapsul anterior dan belakang oleh kapsul posterior. Di bagian dalam kapsul terdapat korteks dan nukleus. Posisi lensa tergantung pada zonula zinn yang melekat pada prosesus siliaris. Fungsi dari lensa ada dua yaitu refraksi dan akomodasi. Fungsi refraksi adalah sebagai bagian optik bola mata untuk memfokuskan sinar ke bintik kuning, lensa menyumbang + 18.0 – Dioptri. Fungsi akomodasi adalah dengan kontraksi otot-otot siliaris ketegangan zonula zinn berkurang sehingga lensa lebih cembung untuk melihat obyek dekat (Ilyas dkk, 2010) [13].

Keadaan patologik dari lensa dapat dijumpai dalam beberapa bentuk seperti katarak dan dislokasi lensa. Katarak dibagi menjadi beberapa macam yaitu katarak perkembangan/pertumbuhan misalnya kongenital atau juvenile, katarak degenerative misalnya katarak senil, katarak komplikata, katarak trauma. Sedangkan dislokasi lensa merupakan kelainan lensa akibat fi ksasi lensa pada zonula zinn tidak normal. Bila hanya sebagian zonula zinn yang putus maka disebut subluksasi lensa (Ilyas dkk,2010) [13].

Katarak kini masih menjadi penyakit paling dominan pada mata dan penyebab paling utama kebutaan. Paling sedikit 50% dari semua kebutaan disebabkan oleh katarak, dan 90% diantaranya terdapat di negara berkembang. Tidak terkecuali Indonesia, dimana berdasarkan hasil survei kesehatan indera penglihatan dan pendengaran tahun 1993–1996, prevalensi kebutaan mencapai 1,5% dan lebih dari separuhnya disebabkan oleh katarak yang belum dioperasi (Depkes RI, 1998) [12].

Katarak adalah setiap keadaan kekeruhan pada lensa yang dapat terjadi akibat hidrasi (penambahan cairan) lensa, denaturasi protein lensa atau akibat kedua-duanya. Katarak umumnya merupakan penyakit pada usia lanjut, akan tetapi dapat juga akibat kelainan kongenital atau penyulit penyakit mata lokal menahun (Ilyas, 2002;Vaughan, 2000) [12].

# BAB III

**METODOLOGI PENELITIAN**

**3.1 Kerangka penelitian**

Agar langkah-langkah yang diambil penulis dalam penelitian ini tidak melenceng dari pokok pembahasan dan lebih mudah dipahami, maka urutan langka-langkah akan dibuat secara sistematis sehingga dapat dijadikan pedoman yang jelas dan mudah untuk menyelesaikan permasalahan yang ada. Urutan langkah-langkah yang akan dibuat pada penelitian ini dapat dilihat pada gambar 3.1 berikut :

Penelitian Pendahuluan

Pengumpulan Data

Analisis

Perancangan

Implementasi

Pengujian

**Gambar 3.1 Kerangka Kerja Penelitian**

**3.2 Tahapan Penelitian**

Tahapan penelitian ini menjelaskan langkah-langkah dalam melakukan pencatatan data serta mengumpulkan beberapa laporan yang diperlukan untuk dapat dijadikan pedoman dalam pembuatan penelitian ini:

**3.2.1 Penelitian Pendahuluan**

Penelitian pendahuluan ini merupakan langkah pertama melakukan suatu penelitian. Tujuan dari penelitian ini adalah untuk mengetahui permasalahan penyakit katarak pada anak, yang umumnya banyak para orang tua yang belum mengetahui gejalah awal dan pencegahannya, serta kurangnya perhatian orang tua terhadap permasalahan penyakit yang diderita. Tujuan dari pembuatan aplikasi ini agar para orang tua khususnya penderita katarak bisa mengetahui informasi dan gejalah penyakit secara dini. Tujuan dari menggunakan metode *certainty factor* adalah menentukan atau membuktikan seorang anak penderita katarak tersebut mengalami penyakit yang serius atau bisa diatasi dengan pengobatan biasa saja.

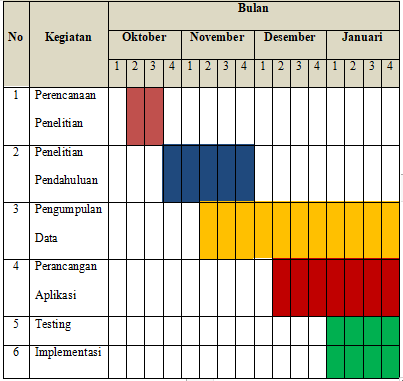
**3.2.2 Pengumpulan Data**

Dalam penelitian ini, dilakukan pengumpulan data dengan cara mewawancarai dokter / pakar. Untuk mendapatkan data tentang penyakit katarak pada anak berdasarkan gejalah, pengobatan dan pencegahan, maka metode yang digunakan dalam proses pengumpulan data dilakukan dengan tahapan-tahapan sebagai berikut :

1. Waktu Penelitian

Waktu penelitian dimulai sejak bulan September 2017. Perancangan sistem dan pengumpulan data dilakukan pada November 2017 sampai selesai.

**Tabel 3.1 Waktu Penelitian**



Keterangan :

|  |
| --- |
|  |
|  |
|  |
|  |
|  |

9 Oktober – 20 Oktober :

23 Oktober – 23 November :

7 November – 25 Januari :

4 Desember – 25 Januari :

2 Januari – 25 Januari :

1. Tempat Penelitian

Tempat dilakukannya penelitian ini adalah di Rumah Sakit Umum Citra Bunda Medical Center (BMC) jln. Proklamasi No.37, Alang Laweh, Padang Barat, Kota Padang, Sumatera Barat. Dengan dokter yang bersangkutan yaitu dr. Naima Lessie, SpM.

1. Metode Penelitian

Untuk melakukan penelitian ini, metode pengumpulan data yang digunakan adalah dengan cara :

1. Wawancara (*Interview*)

Dalam hal ini penulis mengajukan pertanyaan-pertanyaan dan menanyakan kebenaran data dan informasi yang sudah ada kepada dokter / pakar.

1. Penelitian Pustaka (*Library Research*)

Pengambilan data tentang katarak pada anak dilakukan dengan cara *browsing internet* atau buku / jurnal yang ada. Sehingga data yang diperoleh dapat digunakan sebagai landasan pada tahap penelitian selanjutnya.

**3.2.3 Analisa**

Dalam tahap analisa ini dapat dilakukan dengan dua tahap antara lain sebagai berikut :

1. Analisa Data

Analisa dilakukan untuk mendapatkan suatu fakta berdasarkan metode penelitian yang dilakukan. Data yang diperoleh dari kasus katarak pada anak dan metode penelitian akan dianalisis sehingga memperoleh seperangkat aturan (*rule base*).

1. Analisis Proses

Pada tahap analisa proses ini, penulis mnggunakan metode *Certainty Factor*. Dimana metode ini dapat membantu pakar dalam menyelesaikan hal ketidakpastian. Hasil yang tidak pasti disebabkan oleh dua faktor, yaitu aturan yang tidak pasti dan jawaban pengguna yang tidak pasti atau suatu pertanyaan yang diajukan oleh sistem pakar.

1. Analisa sistem

Pada tahap analisa sistem ini dilakukan untuk merancang sistem yang akan dibuat dengan menggunakan metode *certainty factor*. Perancangan tersebut meliputi, merancang tampilan *user*, merancang basis data untuk sistem tersebut agar manajemen *file* yang ada lebih teratur, dan merancang koding program dari suatu informasi. Dimana program yang akan dibuat menggunakan bahasa pemograman PHP dan MySQL.

**3.2.4 Perancangan**

**3.2.4.1 Perancangan Model**

Adapun UML(*Unified Modelling Language*) yag akan digunakan adalah sebagai berkut :

1. ***Use Case Diagram***

Pada diagram *use case* akan dirancang sebuah interaksi antara *actor* dengan sistem pakar. *Actor* disini terdiri dari *admin*, *user*, dan pengunjung. Interaksi antara *actor* dan *system* dalam bentuk naratif, yang terdiri dari input *user* dan *respon* sistem pakar.

1. ***Class Diagram***

Diagram ini menjelaskan bagaimana hubungan antara nama *class,* admin, gejala, analisa hasil, konsultasi, solusi *user* dan berita pada aplikasi sistem pakar.

1. ***Sequence Diagram***

*Sequence diagram* ini menjelaskan urutan-urutan kejadian yang akan terjadi seiring dengan waktu pada saat *user* menggunakan aplikasi ini mulai dari *login* sampai *logout*, serta mengambarkan menu lihat, tambah, hapus, *edit,* *update* pada *sequence*.

1. ***Collaboration Diagram***

*Collaboration diagram* ini dibagi menjadi *collaboration diagram user*, *collaboration diagram* penyakit, dan *collaboration diagram* analisa hasil.

1. ***Statechart* *Diagram***

Diagram ini menjelaskan untuk memodelkan perubahan keadaan yang dipengaruhi oleh suatu kejadian waktu, UML ini dibagi menjadi diagram *statechart user* untuk melakukan konsultasi, diagram *statechart* admin untuk menambah gejala, dan diagram *statechart* admin untuk menambah data.

1. ***Activity Diagram***

Diagram aktivitas ini dibagi menjadi diagram admin, *activity diagram* *user* dan *activity diagram* pengunjung.

1. ***Deployment Diagram***

Disini perangkat lunak *browser* akan berhubungan dengan *web server* untuk meminta data yang diinginkan *user* dan *web server* akan berhubungan dengan *database browser*.

**3.2.5 Pengujian**

Pengujian sistem merupakan tahap melakukan *testing* untuk mengetahui kesalahan dalam sistem. Salah satu hal penting dalam sistem pakar yaitu dapat melakukan konsultasi secara komputerisasi, dan melihat jenis kesalahan-kesalahan yang terjadi dalam sistem tersebut.

1. Pengujian *Black Box*

Metode berpusat pada fungsional perangkat lunak yang digunakan. Tujuannya adalah menemukan kesalahan-kesalahan seperti:

1. Fungsi -fungsi yang tidak sesuai, atau hilang.
2. Kesalahan atau kekeliruan interface.
3. Kesalahan performa sistem.
4. Kesalahan inisialiasi (proses mulai) atau terminasi (proses selesai atau akhir).

**BAB IV**

**ANALISA DAN PERANCANGAN**

**4.1 Analisa Data**

Tahap analisa data merupakan tahap-tahap yang paling penting dalam pengembangan sebuah sistem, karena pada tahap inilah nantinya dilakukan evaluasi, identifikasi terhadap masalah yang ada, rancangan sistem dan langkah-langkah yang dibutuhkan untuk perancangan yang diinginkan sampai pada analisa yang diharapkan. Perancangan pada analisa data ini terdiri dari analisa data penyakit katarak pada anak, dan data gejala. Hal tersebut dapat dijelaskan pada uraian dibawah ini antara lain:

**4.1.1 Data Jenis - Jenis Penyakit Katarak Pada Anak**

Masalah yang dibahas dalam penelitian ini adalah mengenai penyakit katarak. Setelah melakukan wawancara dengan pakar, penulis mendapat kejelasan tentang pembagian dan jenis penyakit tersebut. Beberapa jenis penyakit katarak yang dibahas dalam penelitian ini dapat dilihat pada tabel 4.1 berikut ini :

**Tabel 4.1 Jenis-jenis Penyakit Katarak Pada Anak**

|  |  |  |
| --- | --- | --- |
| **KODE PENYAKIT** | **NAMA PENYAKIT** | **KETERANGAN** |
| P001 | Kongenital | Merupakan masalah mata menjadi keruh yang terjadi sejak lahir, lensa mata berperan untuk menfokuskan cahaya yang masuk kedalam mata menuju retina, sehingga dengan begitu mata dapat menangkap gambar dengan jelas. Namun pada seseorang yang menderita katarak, sinar yang masuk ke matanya akan tersebar pada saat melewati lensa mata yang keruh dan menyebabkan gambar yang diterima mata tidak jelas atau kabur dan terdistorsi. |
| P002 | Juvenil | Merupakan katarak lembek yang terdapat pada manusia, pada usia muda dan mulai terbentuknya pada usia tiga bulan sampai sembilan tahun. |
| P003 | Traumatik | Merupakan katarak yang muncul sebagai akibat cedera pada mata yang dapat diakibatkan benturan benda tumpul ataupun trauma perforasi yang terlihat sesudah beberapa hari ataupun beberapa tahun. Katarak ini dapat muncul akut, subakut, atau pun gejala sisa dari trauma mata. |

**4.1.2 Data Gejala Penyakit Katarak Pada Anak**

Berikut ini merupakan data nama gejala dapat diuraikan pada tabel 4.2 sebagai berikut :

**Tabel 4.2 Data Gejala-Gejala Penyakit Katarak**

|  |  |  |
| --- | --- | --- |
| **Kode gejala** | **Nama Gejala** | **Jenis Penyakit** |
| G001 | Keturunan atau Genetik | *Kongenital* |
| G002 | Infeksi selama kehamilan (rubella, simplex virus herpes, cytomegalovirus, cacar, dan toksoplasmosis) |
| G003 | Reaksi obat |
| G004 | Trauma mata |
| G005 | Diabetes |
| G006 | Toxocariasis (infeksi parasit yang menginfeksi mata) |
| G007 | Galatosemia (kekurangan enzim) |
| G008 | Berawan dilensa |
| G009 | Gerakan mata yang tidak biasa (Nytagmus) |
| G010 | Bola mata bergoyang-goyang atau juling |
| G011 | Pandangan kabur | *Juvenil* |
| G012 | Silau |
| G013 | Perubahan daya lihat warna |
| G014 | Penurunan ketajaman penglihatan |
| G015 | Diplopia monocular (penglihatan ganda pada satu mata) |
| G016 | Luka memar area mata (benda tumpul) | *Traumatik* |
| G017 | Luka perforasi (benda tajam) |
| G018 | Radiasi sinar |
| G019 | Zat kimia |
| G014 | Penurunan ketajaman penglihatan |
| G012 | Silau |
| G020 | Sensivitas kontras |

**4.1.3 Data Penanganan Penyakit Katarak**

Bila penyakit katarak sudah menyerang pada mata manusia maka cara yang dapat dilakukan hanya mengendalikan bagaimana penyakit tersebut dapat diatasi dan tidak bertambah parah. Adapun bentuk-bentuk dari penanganan penyakit katarak dapat kita lihat pada Tabel 4.3 berikut ini :

**Tabel 4.3 Penanganan Penyakit Katarak**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Kode Penanganan** | **Jenis Penyakit** | | | **Penanganan** |
| **P1** | **P2** | **P3** |
| 1 | PN01 | X |  |  | Katarak kongenital dapat diobati dengan cara operasi katarak sedini/secepat mungkin. |
| 2 | PN02 | X |  |  | Mengganti lensa anak dengan lensa buatan, atau memakaikan anak dengan lensa kontak atau kacamata. |
| 3 | PN03 |  | X |  | Pembedahan dilakukan jika penderita tidak dapat melihat dengan baik dengan bantuan kaca mata untuk melakukan kegiatan sehari-hari |
| 4 | PN04 |  | X |  | Melakukan operasi jika pasien mengeluh adanya gangguan penglihatan dalam melakukan rutinitas pekerjaan |
| 5 | PN05 |  | X |  | Mengkonsumsi sayur-sayuran seperti : bayam, wortel, dan brokoli. |
| 6 | PN06 |  |  | X | Tumpul : Dilakukan pemeriksaan yang khusus, serta perlu penanganan yang serius oleh dokter |
| 7 | PN07 |  |  | X | Perforasi/Benda tajam : Dilakukan operasi secepat mungkin sehingga tidak terjadi infeksi pada mata |
| 8 | PN08 |  |  | X | Sinar dan Zat kimia : Dilakukan pemeriksaan khusus dari dokter mata. |

**4.2 Analisa Proses**

Dalam mengembangkan sistem pakar ini pengetahuan dan informasi diperoleh dari beberapa sumber, yaitu dari dokter serta buku tentang penyakit mata dan katarak.pengetahuan ini akan direpresentasikan dalam bentuk *rule* yang berguna untuk menemukan kesimpulan terhadap penyakit katarak dan solusinya. Struktur rule secara logika menghubungkan satu atau lebih kondisi ada bagian IF (yang akan menguji kebenaran dari serangkaian data) dengan satau atau lebih kesimpulan (*Colclusion*) yang terdapat pada bagian *THEN*.

Selain rule, pada sistem pakar juga dibutuhkan database yang berisikan fakta tentang penyakit katarak. Dengan adanya *rule* dan database ini belum cukup untuk menyelesaikan masalah penyakit katarak, untuk menelusuri masalah yang dibutuhkan sebuah metode inferensi. Metode infernsi yang digunakan dalam penelusuran masalah pada sistem pakar mendiagnosa penyakit katarak ini adalah *Certainty Faktor.*

*Certainty faktor* merupakan metode infernsi yang digunakan untuk menghadapi permasalahan dengan jawaban yang tidak memiliki kepastian penuh. Adapun ketentuan interprestasi CF dapat kita lihat pada tabel 4.4 berikut ini :

**Tabel 4.4 Ketentuan Nilai Pada Metode *Certainty Factor***

|  |  |  |
| --- | --- | --- |
| **No** | ***Uncertain Tern*** | **CF** |
| 1 | Tidak | 0 |
| 2 | Mungkin Tidak | 0.4 |
| 3 | Tidak Tahu | 0.6 |
| 4 | Mungkin Iya | 0.8 |
| 5 | Iya | 1.0 |

**4.2.1 Rule (Aturan)**

*Rule* sebuah teknik representasi pengetahuan sintak *rule* IF E (*Evidence* yang ada) *Then* H (Hipotesa atau kesimpulan yang dihasilkan).

R1 = **IF** Keturunan atau genetik **AND** Infeksi selama kehamilan **AND** Reaksi obat **AND** Trauma mata **THEN** Penyakit = Katarak Kongenital ( **CF= 0.80** )

R2 = **IF** Galatosemia **AND** Gerakan mata yang tidak biasa **AND** Bola mata bergoyang-goyang atau juling **AND** Toxocariasis **THEN** Penyakit = Katarak Kongenital (**CF= 0.75** )

R3 = **IF** Keturunan atau genetik **AND** Diabetes **AND** Reaksi obat **AND** Berawan dilensa **THEN** Penyakit = Katarak Kongenital **( CF= 0.80 )**

R4 = **IF** Pandangan kabur **AND** Silau **AND** Perubahan daya lihat warna **THEN** Penyakit = Katarak Juvenil **( CF= 0.80 )**

R5 = **IF** Penurunan ketajaman penglihatan **AND** Diplopia monokular **AND** Silau **THEN** Penyakit = Katarak Juvenil **( CF= 0.65 )**

R6 = **IF** Luka memar area mata (benda tumpul) **AND** Luka perforasi (benda tajam) **AND** Radiasi sinar **AND** Zat kimia **THEN** Penyakit = Katark Traumatik **( CF= 0.80 )**

R7 = **IF** Luka perforasi (benda tajam) **AND** Sensivitas kontras **AND** Penurunan ketajaman penglihatan **AND** Silau **THEN** Penyakit = Katarak Traumatik **( CF= 0.65 )**

R8 = **IF** Radiasi sinar **AND** Luka memar area mata (benda tumpul) **AND** Zat kimia **AND** Silau **THEN** penyakit = Katarak Traumatik **( CF= 0.70 )**

**4.2.2 Dialog User Dengan Sistem Pakar**

Merupakan dialog antara user dengan sistem pakar untuk mengetahui keluhan yang sedang dialami oleh penderita atau orang tua anak.

**SP :**”Apakah keturunan anda ada mengalami penyakit yang sama (katarak) (Nilai kepastian [0-1])”?

**User :”**Ya, CF = (0.4)  **“**

**SP :**”Apakah anda mengalami infeksi selama kehamilan (Nilai kepastian [0-1]”?

**User :”**Ya, CF = (0.8)**“**

**SP :**”Apakah anak anda pernah mengalami reaksi obat (Nilai kepastian [0-1]) “?

**User :”**Ya, CF = (0.4) **“**

**SP :**”Apakah anak anda pernah mengalami trauma pada mata (Nilai kepastian [0-1]) “?

**User :”**Tidak **“**

**SP :**”Apakah anda mengalami penyakit diabetes (Nilai kepastian [0-1]) “?

**User :”**Ya, CF = (0.8) **“**

**SP :**”Apakah anak anda mengalami Toxocariasis (infeksi parasit mata) (Nilai kepastian [0-1]) “?

**User :”**YA, CF **=** (0.4) **“**

**SP :**”Apakah anak anda mengalami Galatosemia (kekurangan enzim) (Nilai kepastian [0-1]) “?

**User :”**Ya, CF = (0.4) **“**

**SP :**”Apakah mata anak anda mengalami berawan dilensa (Nilai kepastian [0-1]) “?

**User :”**Ya, CF = (0.8) **“**

**SP :**”Apakah anak anda mengalami Nygtamus (gerakan mata yang tidak biasa) (Nilai kepastian [0-1]) “?

**User :”**Ya, CF = (0.4) **“**

**SP :**”Apakah bola mata anak anda bergoyang-goyang atau juling bila dibuka (Nilai kepastian [0-1]) “?

**User :”**Ya, CF = (0.4)**“**

**SP :**”Apakah anak anda mengalami pandangan kabur (Nilai kepastian [0-1]) “?

**User :”**Ya, CF = (0.8)**“**

**SP :**”Apakah anak anda mengalami silau (Nilai kepastian [0-1]) “?

**User :”**Ya, CF = (0.8) **“**

**SP :**”Apakah anak anda mengalami perubahan daya lihat jika melihat warna (Nilai kepastian [0-1]) “?

**User :**”Ya, CF = (0.6)“

**SP :**”Apakah anak anda mengalami penurunan ketajaman penglihatan (Nilai kepastian [0-1]) “?

**User :**”Ya, CF = (0.4)“

**SP :**”Apakah anak anda mengalami diplopia monocular (penglihatan ganda pada satu mata) (Nilai kepastian [0-1]) “?

**User :**”Ya, CF = (0.6) “

**SP :**”Apakah anak anda pernah luka memar terkena benda tumpul pada area mata (Nilai kepastian [0-1]) “?

**User :**”Ya, CF = (0.8)“

**SP :**”Apakah anak anda pernah mengalami luka perforasi atau terkena benda tajam pada area mata (Nilai kepastian [0-1]) “?

**User :**”Tidak “

**SP :**”Apakah anak anda pernah terkena radiasi sinar (Nilai kepastian [0-1]) “?

**User :**”Ya CF = (0.4)“

**SP :**”Apakah anak anda pernah terkena zat kimia pada area mata (Nilai kepastian [0-1]) “?

**User :**”Ya, CF = (0.4) “

**SP :**”Apakah anak anda mengalami sensivitas kontras saat menonton televisi atau laptop (Nilai kepastian [0-1])“?

**User :**”Ya, CF = (0.4)“

**4.2.3 Proses Rule Dari Penyakit**

Proses *rule* mengacu pada pohon keputusan yang akan dibuat, berikut terdapat beberapa *rule* untuk pengetahuan mengenai gangguan dan masalah yang terjadi pada penyakit katarak pada anak, proses *rule* tersebut dapat dilihat pada tabel 4.5 berikut ini :

**Tabel 4.5 Proses Rule Dari Penyakit**

|  |  |
| --- | --- |
| **Rule** | **Keterangan** |
| Rule 1 | **IF** G001 (CF= 0.4) **AND** G002 (CF= 0.8) **AND** G003 (CF= 0.4 ) **AND** G004 **THEN** P01 (CF= 0.80) |
| Rule 2 | **IF** G007 (CF= 0.4) **AND** G009 (CF= 0.4) **AND** G010 (CF= 0.4) **AND** G006 (CF= 0.4) **THEN** P01 (CF= 0.75 ) |
| Rule 3 | **IF** G001 (CF= 0.4) **AND** G005 (CF= 0.8) **AND** G003 (CF= 0.4) **AND** G008 (CF= 0.8) **THEN** P01 (CF= 0.80 ) |
| Rule 4 | **IF** G011 (CF= 0.8) **AND** G012 (CF= 0.8) **AND** G013 (CF= 0.6) **THEN** P02 (CF= 0.80 ) |
| Rule 5 | **IF** G014 (CF= 0.4) **AND** G015 (CF=0.6) **AND** G012 (CF= 0.8) **THEN** P02 (CF= 0.65 ) |
| Rule 6 | **IF** G016 (CF= 0.8) **AND** G017 **AND** G018 (CF= 0.4) **AND** G019 (CF= 0.4) **THEN** P03 (CF= 0.80 ) |
| Rule 7 | **IF** G017 **AND** G020 (CF= 0.4) **AND** G014 (CF= 0.8) **AND** G012 (CF= 0.8) **THEN** P03 (CF= 0.65 ) |
| Rule 8 | **IF** G018 (CF= 0.4) **AND** G016 (CF= 0.8) **AND** G019 (CF= 0.4) **AND** G012 (CF= 0.8) **THEN** P03 (CF= 0.70 ) |

Dari rule diatas maka didapatkan fakta baru dari setiap gejala pada rule yang ada :

**Fakta Baru :**

**Tabel 4.6 Fakta Baru**

|  |  |
| --- | --- |
| **Gejala** | **CF** |
| Keturunan atau Genetik | **Evidence CF = 0.4** |
| Infeksi selama kehamilan (rubella, simplex virus herpes, cytomegalovirus, cacar, dan toksoplasmosis) | **Evidence CF = 0.8** |
| Reaksi obat | **Evidence CF = 0.4** |
| Trauma mata |  |
| Diabetes | **Evidence CF = 0.8** |
| Toxocariasis (infeksi parasit yang menginfeksi mata) | **Evidence CF = 0.4** |
| Galatosemia (kekurangan enzim) | **Evidence CF = 0.4** |
| Berawan dilensa | **Evidence CF = 0.8** |
| Gerakan mata yang tidak biasa (Nytagmus) | **Evidence CF = 0.4** |
| Bola mata bergoyang-goyang atau juling | **Evidence CF = 0.4** |
| Pandangan kabur | **Evidence CF = 0.8** |
| Silau | **Evidence CF = 0.8** |
| Perubahan daya lihat warna | **Evidence CF = 0.6** |
| Penurunan ketajaman penglihatan | **Evidence CF = 0.4** |
| Diplopia monocular (penglihatan ganda pada satu mata) | **Evidence CF = 0.6** |
| Luka memar area mata (benda tumpul) | **Evidence CF = 0.8** |
| Luka perforasi (benda tajam) |  |
| Radiasi sinar | **Evidence CF = 0.4** |
| Zat kimia | **Evidence CF = 0.4** |
| Sensivitas kontras | **Evidence CF = 0.4** |

**CF Rule1** = 0,80 **CF Rule5** = 0,65

**CF Rule2** = 0,75 **CF Rule6** = 0,80

**CF Rule3** = 0,80 **CF Rule7** = 0,65

**CF Rule4** = 0,80 **CF Rule 8**= 0,70

Keterangan setiap *rule* dari tabel fakta baru :

**R1** = Tidak dieksekusi karena Evidance TIDAK FAKTA

**R2** = **IF** Galatosemia **AND** Gerakan mata yang tidak biasa **AND** Bola mata bergoyang-goyang atau juling **AND** Toxocariasis **THEN** Penyakit = Katarak **Kongenital** ( CF = 0,75 )

**CF 2 = (** Galatosemia, Gerakan mata yang tidak biasa, Bola mata bergoyang-goyang atau juling, Toxocariasis )

**=** MIN ( 0,4; 0,4; 0,4; 0,4) \* 0,75

= 0,4 \* 0,75

= 0,3

Fakta baru : **Penyakit katarak kongenital Hypotesa = 0,3**

**R3** = **IF** Keturunan atau genetik **AND** Diabetes **AND** Reaksi obat **AND** Berawan dilensa **THEN** Penyakit = Katarak **Kongenital** ( CF = 0,80 )

**CF 3 =(** Keturunan atau genetik**,** Diabetes**,** Reaksi obat**,** Berawan dilensa)

= MIN (0,4; 0,8; 0,4; 0,8) \* 0,80

= 0,4 \* 0,80

= 0,32

Fakta baru : **Penyakit katarak kongenital Hypotesa = 0,32**

**R4** = **IF** Pandangan kabur **AND** Silau **AND** Perubahan daya lihat warna **THEN** Penyakit = Katarak **Juvenil** ( CF = 0,80 )

**CF 4 = (** Pandangan kabur**,** Silau**,** Perubahan daya lihat warna)

**=** MIN (0,8; 0,8; 0,6) \* 0,80

= 0,6 \* 0,80

= 0,48

Fakta baru : **Penyakit katarak Juvenil Hypotesa = 0,48**

**R5** = **IF** Penurunan ketajaman penglihatan **AND** Diplopia monokular **AND** Silau **THEN** Penyakit = Katarak **Juvenil** ( CF = 0,65 )

**CF 5** = ( Penurunan ketajaman penglihatan**,** Diplopia monocular**,** Silau)

**=** MIN ( 0,4; 0,6; 0,8 ) \* 0,65

**=** 0,4 \* 0,65

= 0,26

Fakta baru : **Penyakit katarak Juvenil Hypotesa = 0,26**

**R6** = Tidak dieksekusi karena EvidanceTIDAK FAKTA

**R7** = Tidak dieksekusi karena EvidanceTIDAK FAKTA

**R8** = **IF** Radiasi sinar **AND** Luka memar area mata (benda tumpul) **AND** Zat kimia **AND** Silau **THEN** penyakit = Katarak **Traumatik** ( CF = 0,70 )

**CF 8** = ( Radiasi sinar, Luka memar area mata (benda tumpul)**,** Zat kimia**,** Silau )

= MIN (0,4; 0,8; 0,4; 0,8) \* 0,70

= 0,4 \* 0,70

=0,28

Fakta baru : **Penyakit katarak Traumatik Hypotesa = 0,28**

**R2** = Kongenital **CF1** = 0,3 **R8** = Traumatik **CF5** = 0,28

**R3** = Kongenital **CF2** = 0,32

**R4** = Juvenil **CF3** = 0,48

**R5** = Juvenil **CF4** = 0,26

Karena **R2** dan**R3** memiliki hipotesa yang sama maka **R2** dan **R3** digabung :

CF 2 = 0,3

CF 3 = 0,32

CF gabungan1 (CF2, CF3) = CF2 + CF3 \* (1 – CF2)

= 0,3 + 0,32 \* (1 – 0,3)

= 0,3 + 0,32 \* 0,7

= 0.434

CF gabung R2 dan R3 = 0.434

Karena **R4** dan **R5** memiliki hipotesa yang sama maka **R4** dan **R5** digabung :

CF 4 = 0,48

CF 5 = 0,26

CF gabungan1 (CF4, CF5) = CF4 + CF5 \* (1 – CF4)

= 0,48 + 0,26 \* (1 – 0,48)

= 0,48 + 0,26 \* 0,52

= 0.3848

CF gabung R4 dan R5 = 0.3848

Jadi ***CF*** gabungan = ***Kongenital*** = 0,434 = 43.4 %

***CF*** gabungan = ***Juvenil*** = 0,3848 = 38.48 %

***CF*** = ***Traumatik*** = 0.28 = 28 %

Jadi penyakit katarak yang dimiliki oleh user adalah ***Katarak*** ***Kongenital*** dengan Nilai kepastian (*Certainty Factor*) = 43.4%

**4.3 Perancangan Sistem Menggunakan UML**

Perancangan UML (*Unified Modelling Lenguage*) adalah untuk menentukan cara kerja yang menggunakan *Use Case Diagram, Class Diagram, Sequence Diagram, Collaboratin Diagram, Statechart Diagram, Activity Diagram, dan Deployment Diagram.* Aplikasi ini agar mempermudah memindahkan konsep sistem yang dirancang ke dalam bentuk program dimana perancangannya digambarkan dalam bentuk diagram-diagram sebagai berikut :

**4.3.1 *Use Case Diagram***

*Use case* mengambarkan bagaimana seseorang akan menggunakan atau memanfaatkan sistem, sedangkan aktor adalah seseorang atau sesuatu yang berinteraksi dengan sistem. *Use case diagram* mengambarkan bagaimana proses-proses yang dilakukan oleh aktor terhadap sebuah sistem.

1. **Definisi Aktor**

Aktor adalah aktifitas yang bisa dilakukan oleh para aktor dalam menggunakan sistem dapat kita lihat pada tabel 4.7 berikut ini :

**Tabel 4.7 Defini Aktor**

|  |  |  |
| --- | --- | --- |
| **NO** | **Aktor** | **Deskripsi** |
| 1 | Admin | Aktor yang memonitoring segala isi pada aplikasi sistem pakar. |
| 2 | User | Aktor yang melakukan konsultasi pada aplikasi sistem pakar. |

1. **Definisi *Use Case Diagram***

Definisi dalam *Use case diagram* adalah kegiatan-kegiatan yang akan dilakukan didalam sistem antara para aktor dengan *use case* dapat kita lihat pada tabel 4.8 berikut :

**Tabel 4.8 Definisi *Use Case Diagram***

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Use case** | **Deskripsi** | **Aktor** |
| 1 | Regitrasi | *Register* merupakan proses pendaftaran untuk bisa masuk kedalam sistem | User |
| 2 | Login | *Login* merupakan proses untuk masuk ke dalam sistem pakar | User |
| 3 | Logout | *Logout* merupakan proses keluar dari sistem pakar | User |
| 4 | Home | Merupakan tampilan informasi aplikasi sistem pakar | User |
| 5 | Menambah penyakit | Menambah penyakit merupakan proses untuk menambah data penyakit pada sistem | Admin |
| 6 | Mengedit penyakit | Mengedit penyakit merupakan proses untuk merubah data penyakit pada sistem | Admin |
| 7 | Menghapus penyakit | Menghapus penyakit merupakan proses untuk penghapusan data penyakit pada sistem | Admin |
| 8 | Menambah gejala | Menambah gejala merupakan proses untuk menambah data gejala pada sistem | Admin |
| 9 | Mengedit gejala | Mengedit gejala merupakan proses untuk merubah data gejala pada sistem | Admin |
| 10 | Menghapus gejala | Menghapus gejala merupakan proses untuk penghapusan data gejala pada sistem | Admin |
| 11 | Konsultasi | Tampilan konsultasi merupakan tampilan untuk melakukan konsultasi atau memilih gejalah yang dilakukan oleh seorang user | User |
| 12 | Hasil konsultasi | Tampilan hasil konsultasi merupakan tampilan dari proses yang telah disimpan setelah melakukan konsultasi | User |
| 13 | Cetak hasil | Tampilan hasil konsultasi merupakan tampilan output atau hasil setelah melakukan konsultasi | User |

*Use Case Diagram* yaitu mengambarkan skenario atau interaksi yang dapat dilakukan oleh seorang aktor, aktor disini berupa admin dan user. Prosesnya adalah dilakukan oleh aktor yang dalam sistem ini secara umum. Admin bisa menambah, mengedit dan menghapus data, sedangkan user melakukan konsultasi, registrasi dan melihat home. Adapaun *use case* *diagram* tersebut dapat kita lihat pada gambar 4.1 berikut ini :



**Gambar 4.1 *Use Case Diagram Admin* dan *User***

**4.3.2 Class Diagram**

*Class Diagram* sering digunakan untuk menampilkan kelas tabel, field dan operasi dari program ke *database* yang digunakan.

1. **Definisi *Class***

Spesifikasi yang jika diintansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class diagram* menggambarkan struktur dan deskripsi *class, paskage,* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. Untuk lebih jelasnya dapat kita lihat pada tabel 4.9 berikut ini :

**Tabel 4.9 Definisi *Class***

|  |  |  |
| --- | --- | --- |
| **NO** | ***Class*** | **Deskripsi** |
| 1 | Admin | Berisikan id admin, nama, dan password admin |
| 2 | User | Berisikan id user, nama, email, dan password |
| 3 | Penyakit | Berisikan id penyakit, nama penyakit, dan solusi |
| 4 | Gejala | Berisikan id gejala, dan nama gejala |
| 5 | Deskripsi | Berisikan id deskripsi, nama, deskripsi, dan gambar |
| 6 | Deskripsi penyakit | Berisikan id des penyakit, nama penyakit, dan deskripsi |

Adapun *class diagram* dari sistem yang akan dibuat dapat kita lihat pada gambar 4.2 berikut :



**Gambar 4.2 *Class Diagram***

1. **Rancangan**

Dalam database terdapat *file*, terdiri dari beberapa *record*, *record* terdiriterbagi atas beberapa *field,* setiap *record* akan menampung data untuk menghasilkan informasi. Bentuk *database* dapat dilihat sebagai berikut :

1. Tabel Admin

Tabel yang digunakan sebagai tempat penyimpanan data *username* dan *password* admin. Dimana *username* dan *password* nantinya dapat digunakan sebagai login pada sistem sebagai operator. Adapun bentuk tabel admin yang ada di *database* dapat kita lihat pada tabel 4.10 berikut ini :

**Tabel 4.10 Tabel Admin**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Field** | **Type** | **Keterangan** |
| 1. | id\_admin | Int (15) | Primary Key |
| 2. | Nama | Varchar (25) | Nama Panggilan Admin |
| 3. | Password | Varchar (15) | Merupakan hak akses untuk masuk kedalam layanan |

1. Tabel *User*

Tabel yang digunakan sebagai masuk kedalam sistem dalam aplikasi sistem pakar. Adapun bentuk tabel *user* yang ada di *database* dapat dilihat pada tabel 4.11 sebagai berikut :

**Tabel 4.11 Tabel *User***

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Field** | **Type** | **Keterangan** |
| 1. | Id | Int (11) | Primary key |
| 2. | Nama | Varchar(25) |  |
| 3. | Alamat | Varchar(30) |  |
| 4. | Jenis kelamin | Varchar(10) |  |
| 3. | email | Varchar(25) |  |
| 4. | Password | Varchar(15) |  |

1. Tabel Penyakit

Tabel penyakit berisi tentang penyakit. Adapun bentuk tabel penyakit yang ada di *database* dapat dilihat pada tabel 4.12 sebagai berikut :

**Tabel 4.12 Penyakit**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Field** | **Type** | **Keterangan** |
| 1. | Nama\_penyakit | Varchar(25) |  |
| 2. | Solusi | Text |  |
| 3. | Id\_penyakit | Varchar(10) |  |

1. Tabel Gejala

Tabel gejala berisi tentang gejala. Adapun bentuk tabel gejala yang ada di *database* dapat dilihat pada tabel 4.13 sebagai berikut :

**Tabel 4.13 Gejala**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Field** | **Type** | **Keterangan** |
| 1. | Id\_gejala | Varchar(10) |  |
| 2. | Nama\_gejala | Varchar(100) |  |

1. Tabel Deskripsi

Tabel deskripsi berisi tentang keteranganpenyakit beserta gambar katarak. Adapun bentuk tabel deskripsi yang ada di *database* dapat dilihat pada tabel 4.14 sebagai berikut :

**Tabel 4.14 Deskripsi**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Field** | **Type** | **Keterangan** |
| 1. | Id\_deskripsi | Int(11) | Primery key |
| 2. | Nama | Varchar(25) |  |
| 3. | Deskripsi | Text |  |
| 4. | Gambar | Varchar(25) |  |

1. Tabel Deskripsi Penyakit

Tabel deskripsi penyakit berisi tentang keterangan penyakit serta gejala. Adapun bentuk tabel deskripsi penyakit yang ada di *database* dapat dilihat pada tabel 4.15 sebagai berikut :

**Tabel 4.15 Deskripsi Penyakit**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Field** | **Type** | **Keterangan** |
| 1. | Id\_des\_penyakit | Int(11) | Primery key |
| 2. | Nama\_penyakit | Varchar(30) |  |
| 3. | Deskripsi | Text |  |

**4.3.3 *Sequence Diagram***

*Sequence diagram* mengambarkan interaksi antar objek didalam dan disekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhdap waktu. *Sequence diagram* digunakan untuk mengambarkan perilaku pada sebuah *scenario* yang diterapkan pada sistem untuk modal penerapan pada bahasa pemograman. Berikut gambar-gambar *sequence diagram* yang terdapat pada sistem, diantaranya :

1. *Squence Diagram* Administrator

*Squence Diagram* dibawahini menjelaskan kegiatan aktor admin dalam menjalankan sistem menambah, mengedit, dan menghapus dalam database. Dapat kita lihat pada gambar 4.3 dibawah ini :



**Gambar 4.3 *Squence* *Diagram* Administrator**

1. *Squence Diagram* AdminKelolah Penyakit

*Squence* *diagram* dibawah ini menjelaskan kegiatan yang dilakukan admin dalam melakukan aktivitas edit, tambah dan hapus dalam *database*. Dapat kita lihat pada gambar 4.4 dibawah ini sebagai berikut :



**Gambar 4.4 *Squence* *Diagram* Admin Kelolah Penyakit**

1. *Squence Diagram* Admin Kelolah Gejala

*Squence* *diagram* dibawah ini menjelaskan kegiatan yang dilakukan admin dalam melakukan aktivitas edit gejala, tambah gejala dan hapus gejala dalam *database*. Dapat kita lihat pada gambar 4.5 dibawah ini sebagai berikut :



**Gambar 4.5 *Squence* *Diagram* Admin Kelolah Gejala**

1. *Squence Diagram* Admin Kelolah Pasien

*Squence* *diagram* dibawah ini adalah kegiatan yang dilakukan admin dalam melihat data pasien. Dapat kita lihat pada gambar 4.6 dibawah ini sebagai berikut.



**Gambar 4.6 *Squence* *Diagram* Admin Kelolah Pasien**

1. *Squence Diagram* Registrasi

*Squence* *diagram* registrasi dibawah ini menjelaskan kegiatan yang dilakukan user dalam melakukan aktivitas daftar agar dapat melakukan login. Dapat kita lihat pada gambar 4.7 dibawah ini sebagai berikut :



**Gambar 4.7 *Squence* *Diagram* Registrasi**

1. *Squence Diagram* Konsultasi

*Squence* *diagram* konsultasi dibawah ini menjelaskan kegiatan yang dilakukan user dalam melakukan aktivitas konsultasi dengan melakukan pilih gejalah dan pilih nilai. Dapat kita lihat pada gambar 4.8 dibawah ini sebagai berikut :



**Gambar 4.8 *Squence* *Diagram* Konsultasi**

1. *Squence Diagram* Cetak Hasil Konsultasi

*Squence* *diagram* cetak hasil konsultasi dibawah ini menampilkan hasil konsultasi yang dilakukan dalam aplikasi sistem pakar. Dapat kita lihat pada gambar 4.9 dibawah ini sebagai berikut :



**Gambar 4.9 *Squence* *Diagram* Cetak Hasil Konsultasi**

**4.3.4 *Collaboration Diagram***

*Collaboration diagram* adalah bentuk lain dari *sequence diagram*. Bila *sequence diagram* diorganisir menurut waktu maka *collaboration diagram* diorganisir menurut ruang atau *space*. Berikut gambar-gambar *collaboration diagram* yang terdapat pada dalam sistem :

1. *Collaboration Diagram* Administrator

*Collaboration* *diagram* ini menjelaskan tahapan-tahapan terjadi suatu aktivitas aktor admin dalam kelolah sistem pada administrator dan memperlihatkan bagaimana hubungan antara beberapa objek untuk menambah, edit, dan hapus dalam *database*. Dapat dilihat pada gambar 4.10 Sebagai berikut :



**Gambar 4.10 *Collaboration* *Diagram* Administrator**

1. *Collaboration Diagram* Admin Kelolah Penyakit

*Collaboration* *diagram* ini menjelaskan tahapan-tahapan terjadi suatu aktivitas aktor admin menjalankan sistem berupa tambah, edit, dan hapus dalam database serta memperlihatkan bagaimana hubungan antara beberapa objek. Dapat dilihat pada gambar 4.11 sebagai berikut :



**Gambar 4.11 *Collaboration* *Diagram* Gejala (Admin)**

1. *Collaboration Diagram* Admin Kelolah Gejala

*Collaboration* *diagram* ini menjelaskan tahapan-tahapan terjadi suatu aktivitas aktor admin menjalankan kelolah gejala, serta memperlihatkan bagaimana hubungan antara beberapa objek untuk menambah, edit, dan hapus dalam *database*. Dapat dilihat pada gambar 4.12 sebagai berikut :



**Gambar 4.12 *Collaboration* *Diagram* Admin Kelolah Gejala**

1. *Collaboration Diagram* Admin Kelolah Pasien

*Collaboration* *diagram* ini menjelaskan tahapan-tahapan terjadi suatu aktivitas aktor admin melihat data pasien. Dapat dilihat pada gambar 4.13 sebagai berikut :



**Gambar 4.13 *Collaboration* *Diagram* Admin Kelolah Pasien**

1. *Collaboration Diagram* User untuk Registrasi

*Collaboration* *diagram* ini menjelaskan tahapan-tahapan terjadi suatu aktivitas aktor user dalam registrasi berupa nama, alamat, jenis kelamin, email dan password. Dapat dilihat pada gambar 4.14 sebagai berikut :



**Gambar 4.14 *Collaboration* *Diagram* User untuk Registrasi**

1. *Collaboration Diagram* Konsultasi

*Collaboration* *diagram* konsultasi ini menjelaskan tahapan-tahapan terjadi suatu aktivitas aktor user dalam melakukan konsultasi dengan memilih gejala dan memilih nilai pada cekbox. Dapat dilihat pada gambar 4.15 sebagai berikut :



**Gambar 4.15 *Collaboration* *Diagram* Konsultasi**

1. *Collaboration Diagram* Cetak Hasil Konsultasi

*Collaboration* *diagram* cetak hasil konsultasi ini merupakan hasil dari melakukan aktivitas konsultasi oleh user pada aplikasi sistem pakar. Dapat dilihat pada gambar 4.16sebagai berikut :



**Gambar 4.16 *Collaboration* *Diagram* Deskripsi dan Gambar (User)**

**4.3.5 *Statchart Diagram***

*Statechart diagram* biasanya digunakan untuk memodelkan perilaku dinamis suatu *class* atau objek. *Statechart diagram* memperlihatlkan urutan *state* (aktifitas) yang dilalui sebuah objek, kejadian yang menyebabkan sebuah transisi dari suatu *state* ke *state* yang lain, dan aksi yang menyebabkan perubahan *state* atau aktifitas. Beberapa *Statechart Diagram* pada sistem, anatra lain :

1. *Statchart Diagram* Admin

Diagram ini menjelaskan bagaimana proses penambahan data input penyakit katarak. Diagram ini merupakan gambaran setiap kegiatan yang dilakukan oleh sistem pada halaman tampilan beranda pada admin, terlihat pada gambar 4.17 berikut :



**Gambar 4.17 *Statechart Diagram* Admin**

1. *Statechart Diagram* User

*Diagram* ini menjelaskan bagaimana proses konsultasi pada user. *Diagram* ini merupakan gambaran setiap kegiatan yang dilakukan oleh sistem pada halaman tampilan beranda pada user, terlihat pada gambar 4.18 berikut :



**Gambar 4.18 *Statechart Diagram* User**

1. *Statechart Diagram* Konsultasi

*Diagram* ini menjelaskan bagaimana proses konsultasi pada user. *Diagram* ini merupakan gambaran setiap kegiatan yang dilakukan oleh sistem pada halaman tampilan beranda pada user, seperti memilih gejala setelh itu diproses. *Diagram* ini dapat lihat pada gambar 4.19 berikut :



**Gambar 4.19 *Statechart Diagram* Konsultasi**

**4.3.6 *Activity Diagram***

*Activity diagram* merupakan sebuah diagram dimana dalam diagram tersebut terdapat semua aktifitas dalam sistem yang dirancang.

1. *Activity Diagram* Admin Kelolah Gejala

*Activity diagram* admin kelolah gejala mengambarkan segala aktifitas yang bisa dilakukan admin terhadap sistem dengan melakukan penambahan, edit, dan hapus gejala dalam database. *Activiti diagram*  ini dapat dilihat pada gambar 4.20 berikut:



**Gambar 4.20 *Activity Diagram* Admin Kelolah Gejala**

1. *Activity Diagram* Admin Kelolah Penyakit

*Activity diagram* admin kelolah penyakit mengambarkan segala aktifitas yang bisa dilakukan admin terhadap sistem dengan melakukan penambahan, edit, dan hapus gejala dalam database. *Activiti diagram*  ini dapat dilihat pada gambar 4.21 berikut:



**Gambar 4.21 *Activity Diagram* Admin Kelolah Penyakit**

1. *Activity Diagram* User Konsultasi

*Activity diagram* user konsultasi mengambarkan aktifitas yang bisa dilakukan user terhadap aplikasi dengan memilih gejala, setelah itu mengklik proses pada aplikasi sistem pakar. *Activiti diagram*  untuk user konsultasi dapat dilihat pada gambar 4.22 berikut :

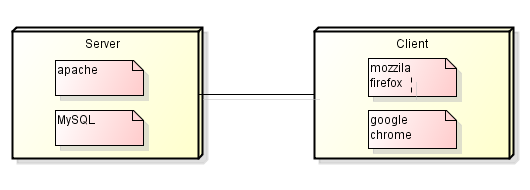


**Gambar 4.22 *Activity Diagram* User**

**4.3.7 *Deployment Diagram***

*Deployment* atau *physical diagram* mengambarkan detail bagamana komponen di-*deplay* dalam infrastruktur sistem, dimana komponen akan terletak (pada mesin, *server* atau piranti keras), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi *server*, dan hal-hal lain bersifat fisikal. Sebuah *node* adalah *server*, *workstation,* atau piranti keras lain yang digunakan untuk men-*deplay* komponen dalam lingkungan sebenarnya. Hubungan antar *node* (misalnya *TCP/IP*) dan *requirement* dapat juga didefinisikan dalam diagram ini.

Pada perancangan sistem ini menggunakan *web server apache* dan *database server* MySQL. Sedangkan *user* dapat melakukan akses dengan menggunakan *web browser* seperti *Mozilla Firefox*, *internet Explorer* dan *Google Crome*.

****

**Gambar 4.23 *Deployment Diagram***

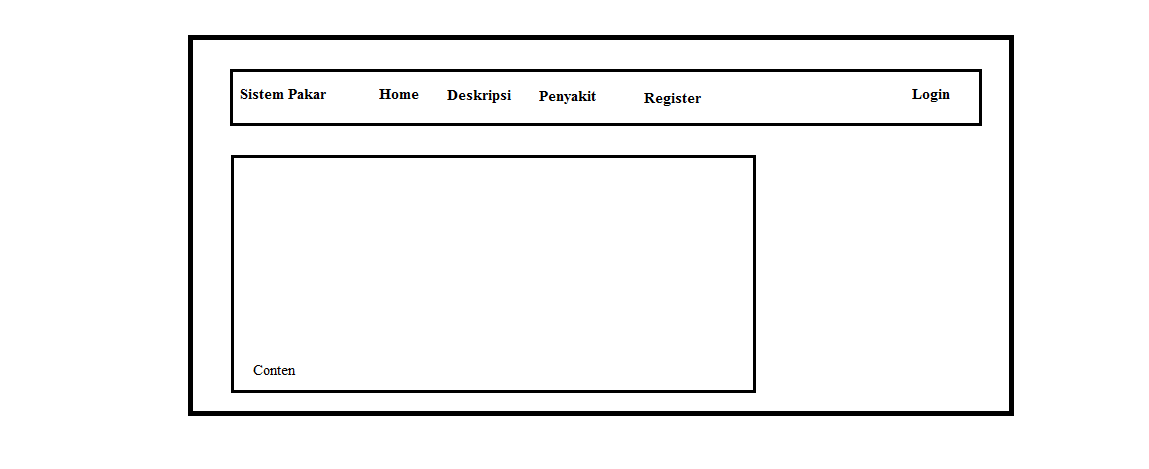
* + 1. **Perancangan *Interface***

Desain antar muka merupakan suatu sistem yang mempertimbangkan keinginan pengguna, yaitu aplikasi yang memenuhi kebutuhan dan kemudahan untuk digunakan. Desain ini digunakan juga untuk mengambarkan tampilan *website* yang akan dirancang. Pada halaman awal terdapat perintah (*link*) yang dapat memanggil atau membuka halaman selanjutnya, dan begitu pula pada halaman-halaman lainnya.

Beberapa desain yang akan dibuat pada sistem sebagai gambaran kasarnya dapat kita lihat sebagai berikut :

1. Halaman Awal pada Aplikasi Sistem Pakar

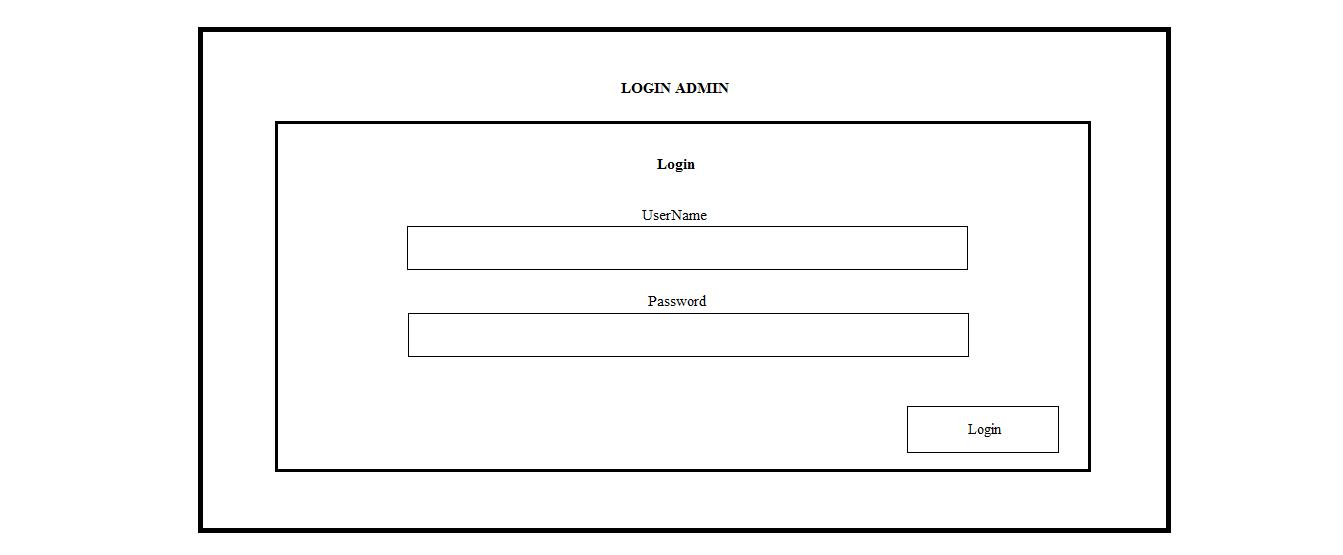
Halaman ini merupakan tampilan awal yang muncul setelah masuk kedalam website, desain tampilan awal aplikasi sistem pakar ini dapat dilihat pada gambar 4.24 :

****

**Gambar 4.24 Halaman Awal Aplikasi Sistem Pakar**

1. Halaman Login

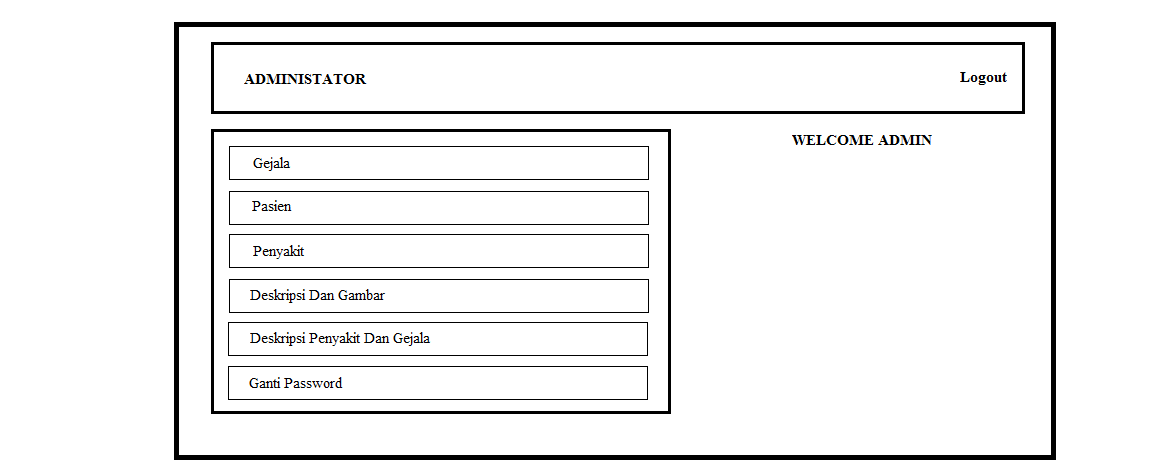
Halaman ini merupakan tampilan login pada administrator, desain tampilan login administrator dapat dilihat pada gambar 4.25 :



**Gambar 4.25 Halaman Login Admin**

1. Halaman Administator

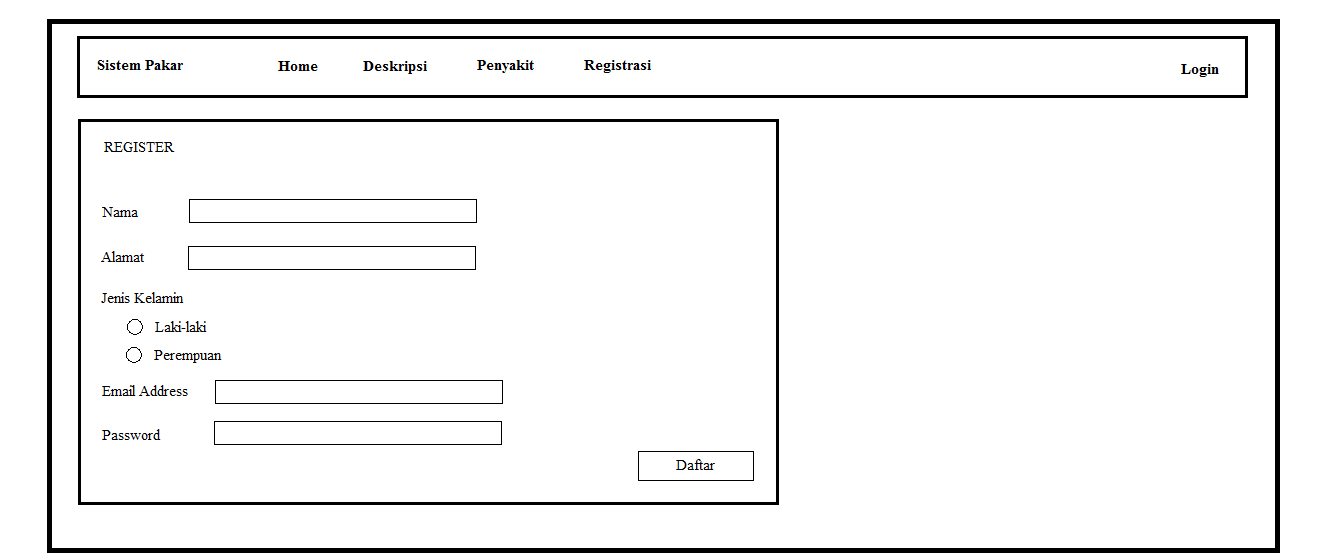
Halaman ini merupakan tampilan pada administrator, desain tampilan administrator dapat dilihat pada gambar 4.26 :



**Gambar 4.26 Halaman Administator**

1. Halaman Registrasi

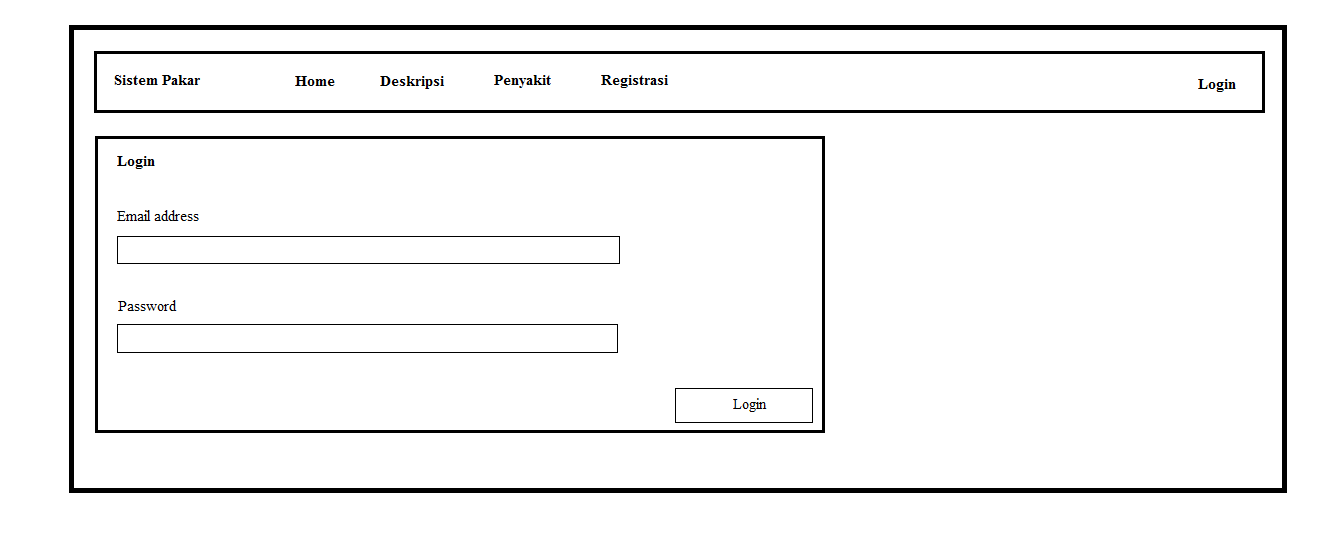
Halaman ini merupakan tampilan biodata user, desain tampilan registrasi pada user dapat dilihat pada gambar 4.27 :



**Gambar 4.27 Halaman Registrasi**

1. Halaman Login User

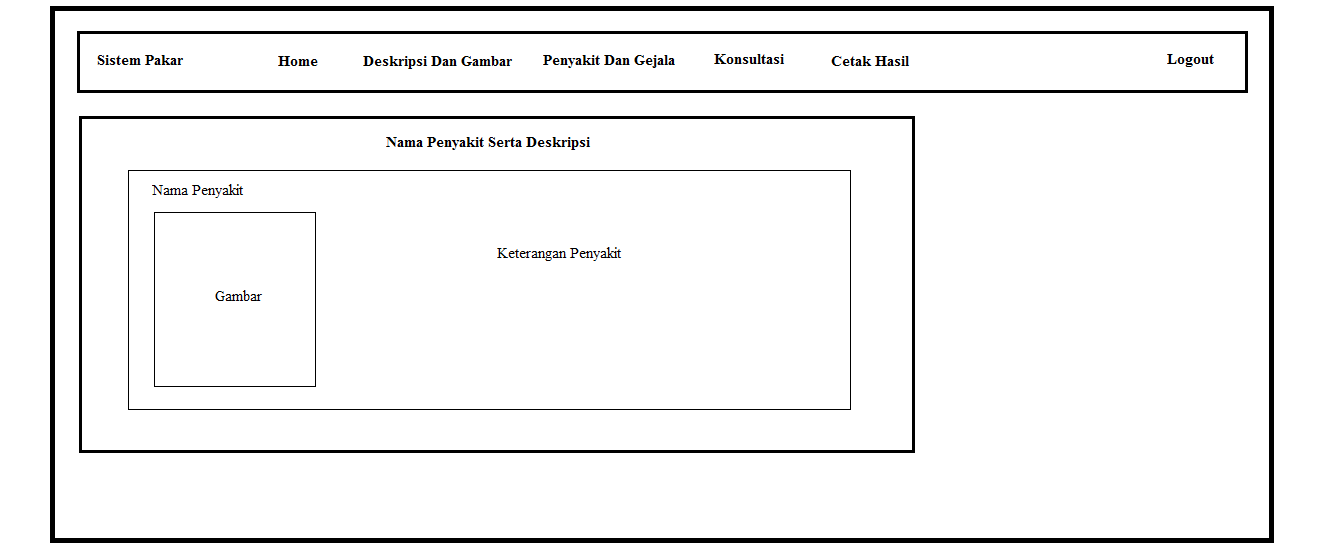
Halaman ini merupakan tampilan awal pada user, desain tampilan awal pada user dapat dilihat pada gambar 4.28 :



**Gambar 4.28 Halaman Login User**

1. Halaman Deskripsi dan Gambar

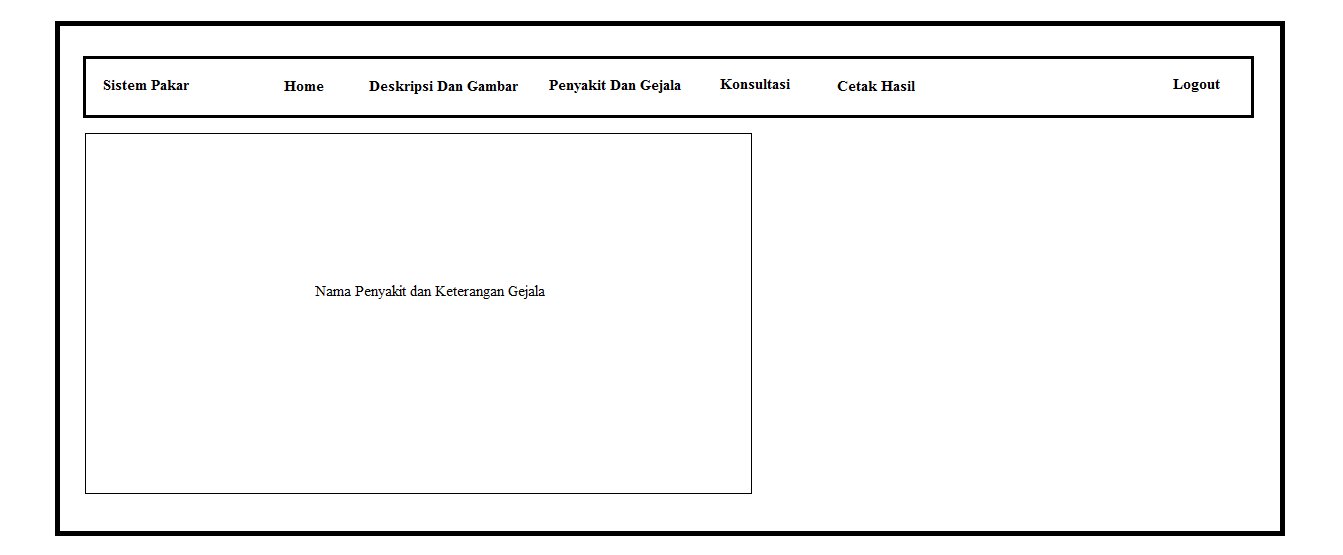
Halaman ini merupakan tampilan keterangan penyakit beserta gambar, desain tampilan halaman ini dapat dilihat pada gambar 4.29 :



**Gambar 4.29 Halaman Deskripsi dan Gambar**

1. Halaman Penyakit dan Gejala

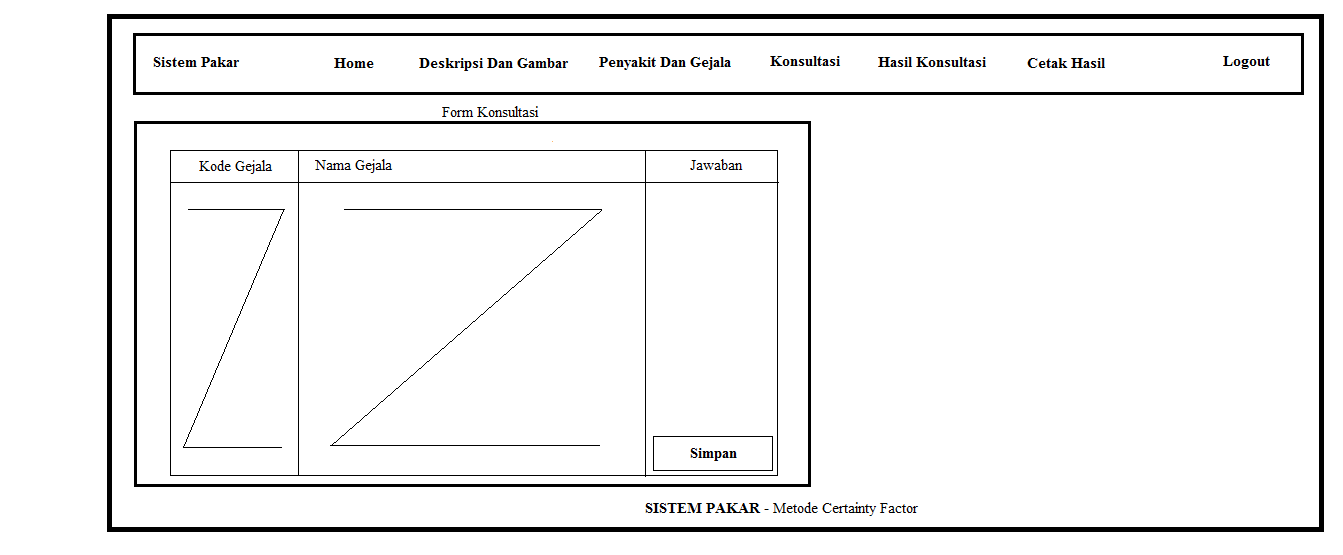
Halaman ini merupakan tampilan keterangan informasi penyakit beserta gejala, desain tampilan halaman ini dapat dilihat pada gambar 4.30 :



**Gambar 4.30 Halaman Penyakit dan Gejala**

1. Halaman Konsultasi

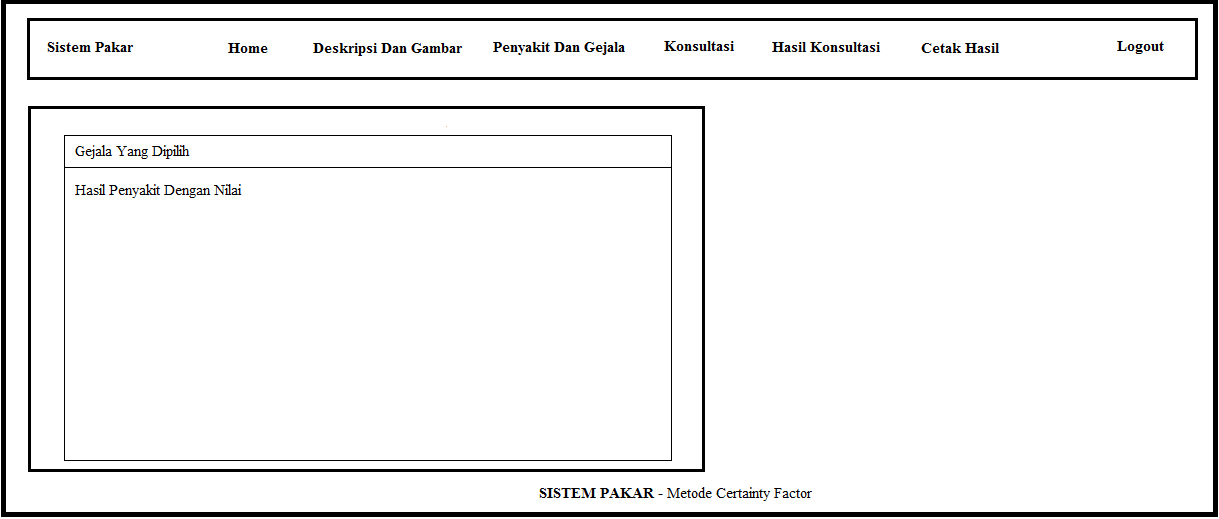
Halaman ini merupakan tampilan situasi konsultasi pada aplikasi sistem pakar, desain tampilan halaman ini dapat dilihat pada gambar 4.31 :



**Gambar 4.31 Halaman Konsultasi**

1. Halaman Hasil Konsultasi

Halaman ini merupakan tampilan akhir situasi konsultasi pada aplikasi sistem pakar, desain tampilan halaman ini dapat dilihat pada gambar 4.32 :



**Gambar 4.32 Halaman Hasil Konsultasi**

**BAB V**

**KESIMPULAN**

**5.1 Kesimpulan**

Sistem Pakar dapat digunakan sebagai alternatif di dalam melakukan diagnosa penyakit katarak dengan memasukkan nilai – nilai kepastian certainty factor terhadap gejala yang dirasakan oleh pengguna. Nilai tersebut merupakan nilai numeric yang telah diubah menjadi nilai linguistik sehingga dapat dipahami oleh pengguna pada umumnya. Sistem pakar yang dihasilkan dapat menyimpulkan dalam bentuk persentase kemungkinan penyakit katarak yang diderita, serta memberikan solusi terhadap penyakit yang diderita oleh seorang pengguna.

**5.2 Saran**

Peneliti menyadari masih banyak kekurangan didalam melakukan penelitian ini. Salah satu yang harus diperhatikan adalah penggunaan platform berbasis web yang belum ramah terhadap teknologi mobile (Smartphone), sehingga dapat mengurangi minat dari pengguna untuk menggunakan sistem pakar ini. Untuk pengembangan berikutnya diharapkan sistem memiliki kemampuan multiplatform sehingga penggunaan sistem pakar dapat lebih dioptimalkan.

**DaFTAR PUSTAKA**

Agus, f., Wulandari, h. e. And Astuti, i. f. 2018. Expert System With Certainty Factor For Early Diagnosis Of Red Chili Peppers Diseases, Journal Of Applied Intelligent System, 2(2), Pp. 52–66. Doi: 10.33633/Jais.v2i2.1455.

Anwar, k. Dkk.. 2016. Kalkulasi Bantuan Korban Bencana Alam Menggunakan Sistem Pakar ( Help Victims Of Natural Disasters Calculation Using Expert System ), Juita, Iv, Pp. 17–21. Arifin, m., Slamin, s. And Retnani, w. e. y. 2017.

Penerapan Metode Certainty Factor Untuk Sistem Pakar Diagnosis Hama Dan Penyakit Pada Tanaman Tembakau, Berkala Sainstek, 5(1), p. 21. Doi: 10.19184/Bst.v5i1.5370.

Atmojo, s. And Utami, r. 2017. Web Sistem Pakar Pendiagnosa Penyakit Anak Umum Dan Hfmd Menggunakan Certainty Factor, Jurnal Ilmiah Edutic, 3(2), Pp. 83–90.

Wulandari, f., & Yuliandri, i. 2014. Diagnosa Gangguan Gizi Menggunakan Metode Certainty Factor, Jurnal Sains, Teknologi Dan Industri, 11(2), Pp. 305–313.

Giarratano, j. c. Dan Riley, g. d. 2005. Expert Systems Principles And Programming Fourth Edition, Boston, Massachusetts: Thomson Course Technology, Pp. 167–173.

Irawan, g. m., Saerang, j. And Tongku, y. 2015. Katarak Pada Anak Di Poliklinik Mata Blu Prof. Dr. r. d. Kandou Manado Periode Januari 2011 - Desember 2013, Jurnal e-Clinic, 3(April), Pp. 2013–2016.

Kusrini, k. 2006. Sistem Pakar Teori Dan Aplikasi.Mevung, f. i. Dkk.. 2017. Diagnosis Penyakit Kejiwaan Menggunakan Metode Certainty Factor, Prosiding Seminar Ilmu Komputer Dan Teknologi Informasi, 2(1), Pp. 374–380.

Mitra, n. n., Sovia, r. And Permana, r. 2019. Sistem Pakar Mendiagnosis Penyakit Kulit Pada Kucing Jenis Persia Menggunakan Metode Certainty Factor Berbasis Web, In Seminar Nasional Teknologi Komputer & Sains (Sainteks).

Reisa, r. 2014. Sistem Pakar Untuk Diagnosis Penyakit

Mata, Sistem Informasi, 2(2), Pp. 30–35. Setyaputri, k. e. And Fadlil, a. 2018. Analisis Metode Certainty Factor Pada Sistem Pakar Diagnosa Penyakit Tht, Scientific Journal Of Informatics (Sji) Unnes, 10(1), Pp. 30–35.

Sutojo, t., Mulyanto, e. And Suhartono, v. 2011.Kecerdasan Buatan. Andi Offset.

Vlahavas, i. 2008. Artificial Intelligence For Advanced Problem Solving Techniques. Igi Global.

Wahyumi, i., Amra, a. a. And Others (No Date) Prevalensi Katarak Pada Anak Di Rsup h. Adam

Malik Medan Tahun 2012, Majalah Kedokteran Nusantara The Journal Of Medical School, 46(1)