**RANCANG BANGUN SISTEM PAKAR DIAGNOSA DINI  
 PENYAKIT KOLESTROL MENGGUNAKAN   
METODE FORWARD CHAINING   
BERBASIS WEBSITE**

**TUGAS AKHIR**

Sebagai salah satu syarat untuk menyelesaikan

Program Diploma III Teknik Informatika

Politeknik Negeri Indramayu



**Oleh :**

**DILA TRIYANI**

**NIM 2003009**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**JURUSAN TEKNIK INFORMATIKA**

**POLITEKNIK NEGERI INDRAMAYU**

**JULI 2023**

# 

# HALAMAN PENGESAHAN

Tugas Akhir ini diajukan oleh :

|  |  |  |
| --- | --- | --- |
| Nama | : DILA TRIYANI | |
| NIM | : 2003009 | |
| Program Studi | : Diploma III Teknik Informatika | |
| Judul | : Rancang Bangun Sistem Pakar Diagnosa Dini Penyakit Kolestrol Menggunakan Metode Forward Chaining Berbasis Website | |
| Pembimbing | : Eka Ismantohadi, S.Kom., M.Eng. NIP 198107092021211005 | **……………..** |

Telah berhasil dipertahankan dihadapan dewan penguji pada tanggal ……………. dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Ahli Madya Program Studi Diploma III Teknik Mesin, Jurusan Teknik Informatika, Politeknik Negeri Indramayu.

DEWAN PENGUJI

|  |  |  |
| --- | --- | --- |
| Ketua Penguji | : | ..……..………….. |
| Anggota Penguji I | : | ………………….. |
| Anggota Penguji I | : | ………………..… |

|  |
| --- |
| Indramayu, Agustus 2023 Ketua Jurusan Teknik Informatika |
|  |
| Eka Ismantohadi, S.Kom., M.Eng. NIP 198107092021211005 |

# ABSTRAK

# ABSTRAK

# KATA PENGANTAR

Assalamu’alaikum Warahmatullahi Wabarakatuh.

Puji syukur senantiasa penulis panjatkan atas rahmat dan hidayah yang telah Allah SWT berikan, sehingga penulis dapat menyelesaikan tugas akhir yang berjudul Rancang Bangun Aplikasi Sistem Pakar Diagnosa Dini Penyakit Kolestrol Menggunakan Metode *Forward Cahining* Berbasis *Website* dengan baik dan tepat waktu. Pembuatan aplikasi ini di tujukan untuk masyarakat umum untuk mendeteksi gejala awal penyakit kolestrol.

Tidak lupa juga penulis mengucapkan banyak terima kasih kepada pihak yang telah bekerjasama dalam Menyusun laporan tugas akhir sehingga dapat terlaksana dengan baik dan lancar.

# DAFTAR ISI

# DAFTAR TABEL

# DAFTAR GAMBAR

# 

# BAB I PENDAHULUAN

## **Latar Belakang**

Dewasa ini perkembangan teknologi sangatlah pesat sehingga memudahkan manusia dalam melakukan aktivitas. Salah satunya yaitu dengan membuat aplikasi sistem pakar. Sistem pakar merupakan aplikasi berbasis komputer yang dirancang untuk meniru kecerdasan manusia dalam menyelesaikan masalah dan mengambil suatu keputusan. Sistem pakar menggunakan pengetahuan dari pakar atau seseorang yang ahli dibidangnya. Pakar merupakan seseorang yang memiliki kemampuan atau pengetahuan, dan pengalaman di bidangnya serta mampu memberikan solusi.

Berdasarkan ahli dari *Clinical Research Support Unit* (CRSU) Fakultas Kedokteran Universitas Indonesia , Dr. Nafrialdi, PhD menyatakan bahwa 35% penduduk Indonesia memiliki kadar kolestrol lebih tinggi dari batas normal.(Hasan P., dkk., 2019).

Untuk membantu mencegah terjadinya penyakit kolestrol, salah satu langkah yang dapat dilakukan yaitu dengan melakukan diagnosa. Diagnosa yaitu suatu tindakan yang dilakukan untuk mengetahui seseorang mengidap suatu penyakit atau tidak. Dengan dilakukannya suatu diagnosa seseorang dapat mengetahui penyakit yang di deritanya sebelum penyakit itu semakin parah. Maka dari itu penulis membuat sistem pakar diagnosa dini penyakit kolestrol menggunakan metode *forward chaining,* sistem ini dibangun guna membantu masyarakat dalam megetahui gejala-gejala awal kolestrol tanpa harus megeluarkan biaya untuk pergi konsultasi ke dokter.

Sistem pakar diagnosa dini penyakit kolestrol ini merupakan salah satu solusi yang dapat digunakan untuk membantu dalam melakukan diagnosa dini penyakit kolestrol. Sistem pakar tersebut dirancang dengan menggunakan kecerdasan buatan dan logika inferensi untuk membantu mengevaluasi gejala-gejala yang dialami dan memberikan diagnosa yang tepat.

Metode *forward chaining* pada diagnosa kolestrol digunakan untuk membantu dalam menentuka keputusan yang diawali dengan proses pencarian fakta terlebih dahulu kemudian fakta-fakta yang ditemukan akan digunakan untuk menguji nilai kebenaran dari dugaan yang ada. Dengan dibuatnya sistem pakar diagnosa dini penyakit kolestrol menggunakan metode *forward chaining* ini diharapkan dapat memberikan manfaat bagi penggunanya.

## **Rumusan Masalah**

Berdasarkan latar belakang yang dijelaskan diatas maka diperoleh beberapa rumusan masalah , yaitu sebagai berikut:

1. Bagaimana merancang dan membangun aplikasi sistem pakar diagnosa dini penyakit kolestrol menggunakan metode *forward chaining* ?
2. Bagaimana sistem pakar ini dapat membantu pengguna dalam diagnosis awal penyakit kolestrol ?
3. Bagaimana sistem pakar ini dapat membantu pengguna dalam memperoleh solusi atau pencegahan penyakit kolestrol?

## **Batasan Masalah**

Untuk memfokuskan pembahasan pada masalah ini dapat disimpulkan beberapa Batasan masalah diantaranya sebagai berikut:

1. Aplikasi sistem pakar diagnosa dini penyakit kolestrol ini dibangun dengan menggukan framework Laravel dan berbasis website.
2. Pada aplikasi sistem pakar diagnosa dini penyakit kolestrol ini hanya menampilkan gejala-gejala awal penyakit kolestrol.
3. Aplikasi sistem pakar diagnosa dini penyakit kolestrol ini hanya menampilkan solusi akhir dari diagnosa.

## **Tujuan Penelitian**

Penelitian ini memiliki beberapa tujuan yaitu sebagai berikut :

1. Membangun atau merancang apliksi sistem pakar untuk mempermudah pengguna dalam mengetahui gejala awal penyakit kolestrol.
2. Meningkatkan pengetahuan dan kesadaran pengguna tentang penyakit kolestrol.
3. Dapat mengimplemntasikan ilmu yang telah didapatkan penulis salama menempuh Pendidikan di Politeknik Negeri Indramayu.

## **Manfaat Penelitian**

Penelitian ini memiliki beberapa tujuan yaitu sebagai berikut :

1. Membangun atau merancang apliksi sistem pakar untuk mempermudah pengguna dalam mengetahui gejala awal penyakit kolestrol.
2. Meningkatkan pengetahuan dan kesadaran pengguna tentang penyakit kolestrol.
3. Dapat mengimplemntasikan ilmu yang telah didapatkan penulis salama menempuh Pendidikan di Politeknik Negeri Indramayu.

## **Sistematika Penulisan**

Di dalam sistematika penulisan laporan tugas akhir ini dibagi kedalam beberapa bab, untuk mempermudah dalam penulisan tugas akhir isi dari babnya memiliki pembahasan yang berbeda namun saling berkaitan. Adapun urutan penulisannya sebagai berikut :

**BAB I PENDAHULUAN**

Pada bab satu ini membahas latar belakang dilakukannya penelitian, rumusan masalah berisikan masalah-masalah yang ada, Batasan masalah berisikan Batasan-batasan yang ditentukan pada penelitian, tujuan dilakukannya penelitaian, manfaat dilakukannya penelitian, dan sistematika penulisan laporan tugas akhir.

**BAB II LANDASAN TEORI**

Pada bab dua ini membahas teori-teori yang berkaitan dengan permasalahan yang diteliti seperti teori tentang Sistem Pakar, Kecerdasan Buatan, Diagnosa, Definisi Kolestrol, Laravel, PHP, Website, Bootsrap, Flowchart, MySQL, *Unified Modeling Language* (*UML*), *Entity Relationship Diagram* (ERD), Visual Studio Code, Balsamiq mockup, Black box Testing, dan Kuesioner.

**BAB III METODE PELAKSANAAN**

Pada bab ini menjelaskan tentang metode yang digunakan untuk penelitian seperti metode pelaksanaan, metode pengumpulan data, flowchart, analisis kebutuhan sistem, perancangan sistem, perancangan *database*, dan perancangan antarmuka.

**BAB IV HASIL DAN PEMBAHASAN**

Di dalam bab empat berisi hasil penelitian dan pembahasan terkait implementasi sistem pakar untuk mendiagnosa dini gejala awal kolestrol, implementasi desain antarmuka, implementasi basis data, dan pengujian aplikasi menggunakan metode *black box testing*.

**BAB V PENUTUP**

Pada bab terakhir atau bab lima ini berisi kesimpulandan saran dari pnulis terkait penelitian tugas akhir dengan topik sistem pakar untuk mendiagnosa dini penyakit kolestrol.

# 

# BAB II LANDASAN TEORI

## **Siatem Pakar**

Sistem pakar adalah program kecerdasan buatan yang menggabungkan basis pengetahuan (*knowledge base)* dengan sistem inferensi.infrerensi berasal dari Bahasa Inggris yaitu *infrerence*. Inferensi adalah suatu proses memperoleh pengetahuan berdasarkan pengalaman yang terjadi (Andi Kristanto, 2004:19).

Pengetahuan sistem pakar dibentuk dari kaidah atau pengalaman tentang perilaku element dari domain bidang tertentu . pengetahuan sistem pakar diperoleh dari orang yang mempunyai pengetahuan pada suatu bidang (pakarbidang tertentu), buku-buku, jurnal imiah, majalah, maupundokumentasi yang tercetak lainnya. Sumber pengetahuantersebut biasa dikenal dengan sumber keahlian. Pengetahuan-pengetahuan tersebut direpresentasikan dalam format tertentu, dan dihimpun dalam suatu basis pengetahuan. Basis pengetahuan ini selanjutnya dipakai sistem pakar untuk menentukan penalaran atas problema yang dihadapi (Sri Hartati & Sari Iswanti 2008:3).

## **Kecerdasan Buatan**

Artifical intelegence atau kecerdasan buaan merupakan merupakan bagian dari ilmu pengetahuan komputer khusus yang ditujukan dalam perancangan otomatisasi tingkah laku cerdas dalam sistem kecerdasan komputer.sistem memperlihatkan sifat khas yang dihubungkan dengan kecerdasan dalam kelakukan atau tindak-tanduk yang sepenuhnya bisa menirukan beberapa fungsi otak manusia ,seperti pengertianbahasa, pengetahuan, pemikiran, pemecahan masalah dan lain sebagainya (Andi Kristanto, 2004:2).

Jadi tujuan utamadari kecerdasan buatan adalahmembuat computer untuk lebih berguna dan mengerti prinsip-prinsip yang memungkinkan untuk menjadi cerdas. Prinsip-prinsip tersebut termasuk penggunaan struktur data didalam representasi ilmu pengetahuan , algoritma-algoritma yang diperlukan dalam penerapan ilmu pengetahuan,Bahasa serta Teknik-teknik pemrogramanyang digunakan dalam implementasinya. Selain itu tujuan dari kecerdasan buatan adalah untuk mengembangan sistem kerja komputer yang benar-benar mampu melakukan tugas-tugas yang memerlukan kecerdasan tingkat tinggi. Program-programnya tidak diartikan penting untuk meniru selera selera manusi dan proses berpikirnya. Tentusaja dalammelakukantugas-tugas yang berbeda,program-program itu benar-benar melebihi kemampuan manusia. Hal yang penting bahwa semua sistem tersebut mampu melakukan tugas-tugas yang cerdas secara efektif dan efisien (Andi Kristanto, 2004:5).

Terdapat beberapa teknologi kecerdasan buatan yang dikembangkan dan dipelajari diantaranya yaitu sistem pakar,sistem syaraf buatan, penalaran suara, robotika, pengolahan Bahasa, pengenalan pola dan penglihatan komputer.

## **Definisi Diagnosa**

Diagnosa sebagaimana halnya dengan penelitian-penelitian ilmiah, didasarkan atas metode hipotesis. Dengan metode hipotesis ini menjadikan penyakit-penyakit begitu mudah dikenali hanya dengan suatu kesimpulan diagnostik. Diagnosa dimulai sejak permulaan wawancara medis dan berlangsung selama melakukan pemeriksaan fisik. Dari diagnosa tersebut akan diperoleh pertanyaan - pertanyaan yang terarah, perincian pemeriksaan fisik yang dilakukan untuk menentukan pilihan testes serta pemeriksaan khusus yang akan dikerjakan. Data yang berhasil dihimpun akan dipertimbangkan dan diklasifikasikan berdasarkan keluhan-keluhan dari pasien serta hubungannya terhadap penyakit tertentu. Berdasarkan gejala-gejala serta tandatanda yang dialami oleh penderita, maka penegakkan diagnosa akan lebih terpusat pada bagian-bagian tubuh tertentu. Dengan demikian penyebab dari gejala-gejala dan tanda-tanda tersebut dapat diketahui dengan mudah dan akhirnya diperoleh kesimpulan awal mengenai penyakit tertentu (Sutikno,2008). (Hengki T. S., 2014).

## **Forward Chaining**

Metode forward chaining adalah metode pencarian atau teknik pelacakan ke depan yang dimulai dengan informasi yang ada dan penggabungan rule untuk menghasilkan suatu kesimpulan atau tujuan. Pelacakan maju ini sangat baik jika bekerja dengan permasalahan yang dimulai dengan rekaman informasi awal dan ingin dicapai penyelesaian akhir, karena seluruh proses akan dikerjakan secara berurutan maju. Strategi inferensi yang dimulai dengan sekumpulan fakta-fakta pengetahuan, memperoleh fakta-fakta baru menggunakan aturan-aturan dimana premis-premis sesuai dengan fakta-fakta pengetahuan, dan meneruskan prosesnya sampai sebuah tujuan yang ditetapkan telah tercapai (Novita, dkk, 2021)

## **Metode waterfall**

Metode Waterfall adalah sebuah metode pengembangan sistem dimana antar satu fase ke fase yang lain dilakukan secara berurutan. Dalam proses implementasi metode Waterfall ini, sebuah langkah akan diselesaikan terlebih dahulu dimulai dari tahapan yang pertama sebelum melanjutkan ke tahapan yang berikutnya. Adapun keuntungan menggunakan metode waterfall ini yaitu requirement harus didefinisikan lebih mendalam sebelum proses coding dilakukan, selain itu proses implementasinya dilakukan secara bertahap dari tahap pertama hingga tahap terakhir secara berurutan. Disamping itu metode Waterfall ini juga memungkinkan sedikit mungkin perubahan yang dilakukan oleh proyek berlangsung ( Novitasari, 2023 ).

Beberapa tahapan model pengembangan dari metode *waterfall* diantaranya sebagai berikut:

Pengujian / Testing

Implementasi   
program

Pemeliharaan / Maintanance

Desain

Analisis kebutuhan

Analisis kebutuhan perangkat lunak

Pada tahapan ini merupakan tahapan pengumpulan kebutuhan pengguna, tujuan dari analisia kebutuhan ini guna merangkum hal-hal yang dibutuhkan oleh pengguna dan mencari kebutuhan apa saja yang diperlukan untuk pengembangan perangkat lunak.

Desain

Setelah kebutuhan dikumpulkan, tahap desain mulai dilakukan. Desain pada tahap ini dibagi menjadi dua jenis, yaitu desain arsitektur sistem dan desain detail. Desain arsitektur sistem bertujuan untuk membuat model sistem secara umum, sedangkan desain detail menentukan bagaimana setiap komponen akan dibangun.

Implementasi

Tahap implementasi atau *coding* merupakan tahap pembuatan perangkat lunak yang sesuai dengan *design* yang dibuat. Kode program akan ditulis dan di coba apakah berfungsi dengan baik atau tidak.

1. Pengujian (*testing)*

Pada tahap *testing* ini bertujuan untuk mengevaluasi perangkat lunak yang dibangun sesuai dengan spesifikasi yang telah ditentukan sebelumnya. Ada beberapa jenis pengujian yang dilakukan salah satunya yaitupengujian sistem.

1. Pemeliharaan (*Maintenance*)

Pada tahap ini ada ketika perangkat lunak sudah di rilis atau di *hosting* dan digunakan. Pemeliharaan dilakukan untuk memantau perangkat lunak berfungsi dengan baik, jika ada masalah yang muncul selama penggunaan perangkat lunak maka akan segera diperbaiki.

## **Definisi kolestrol**

Kolesterol adalah lemak yang terdapat di dalam aliran darah atau sel tubuh yang sebenarnya dibutuhkan untuk pembentukan dinding sel dan sebagai bahan baku beberapa harmon. Kolestrol yang normal harus di bawah 200 mg/dl. Apabila di atas 240 mg/dl, maka berisiko tinggi terkena penyakit seperti serangan Jantung atau Stroke, Kolestrol secara alami bisa dibentuk oleh tubuh, selebihnya didapat dari makanan hewani, seperti daging, unggas, ikan, margarin, keju, dan susu. Kolestrol tidal larut dalam darah sehingga perlu berikatan dengan pengangkutnya, yaitu lipoprotein. Oleh karena itu pula kolestrol dibedakan menjadi *Low-Density Lipopprotein* (LDL) dan *High-Denzsity Lipoprotein* (HDL) (Hengki T. S., 2014).

## **Framework Laravel**

Laravel merupakan salah satu framework yang cukup populer di Indonesia. Framework ini berbasis PHP dan menggunakan metode MVC (Model, View, Controller). Laravel akan mempermudah developer dalam membangun sebuah website, karena developer tidak harus membuatnya dari awal (Muljono, 2020).

Keunggulan menggunakan framework Laravel yaitu framework open source yang dapat digunakan secara gratis yang memungkinkan kamu untuk membuat web aplikasi yang besar dan kompleks dengan mudah. Laravel mempunyai fitur utama yaitu migration, dengan migration dapat mempertahankan struktur database yang dimiliki tanpa harus membuatnya kembali (Muljono, 2020). Pada aplikasi sistem pakar diagnosa dini kolestrol ini penulis menggunakan framework Laravel untuk memudahkan dalam mengerjakan program.

## **XAMPP**

XAMPP adalah *software web server* lokal untuk membangun *website,* aplikasi, hingga database secara offline. Lho, kok offline? Sebagian orang cenderung membuat website secara offline di komputer. Tujuannya,  untuk memastikan semuanya berjalan lancar. Baru kemudian mengupload *website,* aplikasi, atau *database* itu ke *server* ( Arrifudin, 2022 ).

XAMPP yaitu *software* bebas yang mendukung banyak sistem operasi yang berfungsi sebagai server yang berdiri sendiri *(localhost)*, yang terdiri atas Apache, HTTP Server, MySQL database, dan penerjemah bahasayang dirangkai dengan bahasa pemrograman PHP dan Perl. Server lokal menyimpan file web, yang biasa digunakan adalah htdocs (berisi bahasa pemrograman) dan mysql (Dedi, Iqbal, & Fahroji, 2019).

## **HTML**

HMTL merupakan singkatan dari *Hypertext Markup Language* yaitu bahasa standar web yang dikelola penggunaannya oleh W3C (*World Wide Web Consortium*) berupa tag – tag yang menyusun setiap elemen dari Website. HTML berperan sebagai penyusun struktur halaman *Website* yang menempatkan setiap elemen *Website* sesuai *layout* yang diinginkan. HTML biasanya disimpan dalam sebuah file berektensi .html. Untuk mengetikkan skrip HTML dapat menggunakan text editor seperti Notepad sebagai bentuk paling sederhana atau text editor khusus yang dapat yang berbeda sehingga mudah dibaca, seperti Notepad++, Sublime Text, dan masih banyak lagi aplikasi yang sejenis. (Rohi Abdulloh, 2018).

## **MYSQL**

MySQL adalah *Database Management System* (*DBMS*) yang *open source* dengan dua bentuk lisensi, yaitu *Free Software* (perangkat lunak bebas) dan *shareware* (perangkat lunak berpemilik yang penggunaanya terbatas). Jadi MySQL adalah *database server* yang gratis dengan lisensi GNU *General Public License* (GPL) sehingga dapat dipakai untuk keperluan pribadi atau komersial tanpa harus membayar lisensi. MySQL masuk ke dalam jenis RDBMS (*Relational Database Management System)* maka dari itu, istilah semacam baris, kolom, tabel dipakai pada MySQL (Fitri, 2020).

## ***Website***

Terdapat banyak informasi yang dibutuhkan dan juga dapat diakses secara cepat sehingga dalam pembuatan web diperlukan suatu *website.* Dalam dunia teknologi yang pesat diperlukan suatu jaringan yang mempermudah serta mempercepat penyampaian informasi secara luas, dan dapat dengan mudah oleh siapapun yang mendapatkan akses internet. (Sutrisno, 2020).

*Website* atau situs dapat diartikan sebagai kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara, dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk suatu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman (Sutrisno, 2020).

## **Visual studio code**

Visual Studio Code (VS Code) adalah sebuah text editor ringan dan handal yang dibuat oleh Microsoft untuk sistem operasi multiplatform, artinya tersedia juga untuk versi Linux, Mac, dan Windows. Text editor ini secara langsung mendukung bahasa pemrograman JavaScript, Typescript, dan Node.js, serta bahasa pemrograman lainnya dengan bantuan plugin yang dapat dipasang via marketplace Visual Studio Code (seperti C++, C#, Phython, Go, Java, dst) (Salamah, 2021:1).

Teks Editor VS Code juga bersifat *open source*, yang mana kode sumbernya dapat kalian lihat dan kalian dapat berkontribusi untuk pengembangannya. Hal ini juga yang membuat VS Code menjadi favorit para pengembang aplikasi, karena para pengembang aplikasi bisa ikut serta dalam proses pengembangan VS Code ke depannya (Salamah, 2021).

## **Php**

Menurut Hidayatullah dan Kawistara (2014:231) dalam Fridayanthie W. E., Mahdiati T. (2016). PHP singkatan dari *Perl Hypertext Preprocessor* yaitu bahasa pemrograman *web server-side* yang bersifat *open source*. PHP merupakan *script* yang berintergrasi dengan HTML dan berada pada *server* (*server side* HTML *embedded scripting*). PHP adalah *script* yang digunakan untuk membuat halaman web dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*. Mekanisme ini menyebabkan informasi yang diterima *client* selalu yang terbaru atau *up to date*. Semua *script* PHP dieksekusi pada *server* dimana *script* tersebut dijalankan.

## **Balsamiq Mockup**

Balsamiq adalah alat wireframing UI dengan ketelitian rendah yang cepat yang mereproduksi pengalaman membuat sketsa di notepad atau papan tulis, tetapi menggunakan computer. Ini benar-benar memaksa pengguna untuk fokus pada struktur dan konten, menghindari diskusi panjang tentang warna dan detail yang akan muncul nanti dalam proses. Wireframes akan menghasilkan lebih banyak ide, sehingga pengguna dapat membuang ide yang buruk dan menemukan solusi terbaik ( balsamiq 2008-2023)

## **Flowchart**

Flowchart atau bagan alur adalah diagram yang menampilkan langkah-langkah dan keputusan untuk melakukan sebuah proses dari suatu program. Setiap langkah digambarkan dalam bentuk diagram dan dihubungkan dengan garis atau arah panah. Flowchart berperan penting dalam memutuskan sebuah langkah atau fungsionalitas dari sebuah proyek pembuatan program yang melibatkan banyak orang sekaligus. Selain itu dengan menggunakan bagan alur proses dari sebuah program akan lebih jelas, ringkas, dan mengurangi kemungkinan untuk salah penafsiran. Penggunaan flowchart dalam dunia pemrograman juga merupakan cara yang bagus untuk menghubungkan antara kebutuhan teknis dan non-teknis (Setiawan, 2021).

Tabel 2.1 simbol-simbol *flowchart* diagram

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama** | **Fungsi** |
|  | Flow Direction | Berfungsi untuk menghubungkan simbol yang satu dengan yang lainnya, menyatakan arus suatu proses. |
|  | Processing | Digunakan untuk menunjukkan pengolahan yang akan dilakukan dalam komputer. |
|  | Connector | Digunakan untuk menyatakan sambungan dari proses yang satu ke proses berikutnya di halaman yang sama. |
|  | Decision | Digunakan untuk memilih proses yang akan dilakukan berdasarkan kondisi tertenntu. |
|  | Input / Output | Digunakan untuk menyatakan input dan output tanpa melihat jenisnya. |
|  | Document | Digunakan untuk menyatakan masukan dan keluaran yang berasal dari dokumen. |

Sumber : (Rosaly, 2019)

## **Unified Modeling Language**

*Unified Modeling Language* (*UML*) merupakan salah satu metode pemodelan visual yang dapat digunakan dalam perancangan dan pembuatan sebuah *software* yang berorientasi pada objek. *UML* merupakan sebuah standar penulisan atau semacam *blue print* dimana didalamnya termasuk sebuah bisnis proses, penulisan kelas-kelas dalam sebuah bahasa yang spesifik. Terdapat beberapa diagram *UML* yang sering digunakan dalam pengembangan sebuah sistem (Priyandoho, 2018).

Berikut beberapa diagram uml yang sering digunakan :

### ***Use Case Diagram***

Use case diagram adalah diagram yang bagian dari Unified Modelling Language (UML) yang merupakan bentuk awal dari sebuah sistem atau perangkat lunak untuk sebuah perangkat lunak baru yang belum dikembangkan. Diagram ini juga digunakan sebagai penggambaran grafis antara sistem dan pengguna (actor) dengan mendeskripsikan apa yang dilakukan sistem dan bagaimana pengguna menggunakan sistem. Diagram use case dapat membantu Anda dalam menentukan perilaku yang diharapkan (how) dan bukan metode yang tepat untuk mewujudkannya (how). Simbol-simbol dari *Use Case Diagram* dapat dilihat pada table dibawah ini

Tabel 2.2 Simbol-simbol *Use Case Diagram*.

|  |  |  |
| --- | --- | --- |
| **Gambar** | **Nama** | **Keterangan** |
|  | Aktor | Aktor adalah peran pengguna, sistem yang lain, atau alat yang ketika *actor* sedang berkomunikasi dengan *use case* |
|  | *Use Case* | Abstraksi dan interaksi antara sistem dan aktor. |
|  | *Association* | abstraksi dari penghubung antara aktor dengan use case. |
| <<include>> | *Include* | Menunjukkan bahwa suatu *use case* seluruhnya merupakan bagian dari suatu fungsionalitas dari *use case* lainnya. |
| <<extend>> | *Extend* | Menunjukkan bahwa suatu *use case*  merupakan tambahan fungsional dari *use case* lainnya jika suatu kondisi terpenuhi. |

Sumber: (Purwo, 2021)

* + 1. ***Activity* Diagram**

Activity diagram menunjukan kegiatan dan perubahan dari satu aktivitasbke aktivitas lainnya dengan peristiwa yang terjadi dibeberapa dari bagian sistem. Activity diagram merupakan state diagram khusus yang menggambarkan berbagai macam aliran aktivitas sistem yang sedang dirancang, sehingga mengetahui bagaimana masing-masing aliran itu berawal dan berakhir. Bahkan mungkin terjadi *decision* didalamnya( Pratama, 2019).

Simbol-simbol activity diagram dapat dilihat pada table 2.3 dibawah ini.

Tabel 2.3 Simbol-simbol *Activity Diagram*

|  |  |  |
| --- | --- | --- |
| **Gambar** | **Nama** | **Keterangan** |
|  | *Initial Node* | Status awal dari aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal dari aktivitas. |
|  | *Activity* | Aktivitas yang dilakukan sistem, aktivitas umumnya diawali dengan kata kerja |
|  | *Decision* | Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu. |
|  | *Fork Node* | Asosiasi penggabungan yang dimana lebih dari suatu aktivitas digabungkan menjadi satu. |
|  | *Activity Final Node* | Status akhir yang dilakukan sistem, diagram aktivitas memiliki sebuah status akhir dari aktivitas. |
|  | *Swimline* | Memisahkan organisasi bisnis yang akan bertanggung jawab terhadap suatu aktivitas yang terjadi. |

Sumber: (Simatupang, 2019)

* + 1. ***Class Diagram***

Class diagram disebut jenis diagram struktur karena menggambarkan apa yang harus ada dalam sistem yang dimodelkan dengan berbagai komponen. Berbagai komponen tersebut dapat mewakili class yang akan diprogram, objek utama, atau interaksi antara *class*dan objek. Class sendiri merupakan istilah yang mendeskripsikan sekelompok objek yang semuanya memiliki peran serupa dalam sistem ( Oktriwina, 2021 ).

Simbol-simbol class diagram dapat diliat pada table 2.4 dibawah ini

Tabel 2.5 Simbol-simbol *Class Diagram*

|  |  |  |
| --- | --- | --- |
| **Gambar** | **Nama** | **Keterangan** |
|  | *Class* | Menggambarkan sebuah kelas yang terbagi menjadi 3 bagian. Bagian atas adalah nama kelas. Bagian tengah adalah atribut atau properti dari kelas. Bagian bawah adalah *method* dari kelas. |
|  | *association* | Hubungan statis antar kelas. Yang menggambarkan kelas memiliki atribut berupa kelas lain atau kelas yang harus mengetahui eksistensi dari kelas lain. |
|  | *directed association* | Asosiasi dengan makna kelas yang satu digunakan oleh kelas yang lain. |
|  | *generalization* | Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus). |
|  | *Aggregation* | Hubungan yang menyatakan bahwa suatu kelas merupakan atribut bagi kelas lain, tetapi dua *class* ini dapat berdiri masing-masing. |
|  | *Composition* | Bentuk khusus dari *agregation* yang dimana kelas yang menjadi bagian diciptakan setelah kelas *whole* dibuat. Jika salah satu *class* tidak dapat berdiri tanpa kelas lain. |

Sumber: (Fadholi, 2020)

## **Entity relationship diagram**

**ERD atau *Entity Relationship Diagram* adalah suatu bentuk diagram yang menjelaskan hubungan antar objek-objek data yang mempunyai hubungan antar relasi.**ERD digunakan untuk menyusun struktur data dan hubungan antar data, dan untuk menggambarkannya digunakan notasi, simbol, bagan, dan lain sebagainya (Baharsyah, 2020 ).

Terdapat simbol-simbol dalam membuat *entity relationship diagram* dapat dilihat pada tabel 2.6 dibawah ini

Tabel 2.6 Simbol-simbol ERD

|  |  |  |
| --- | --- | --- |
|  | *Entity* | Suatu entity digambarkan sebagai sebuah persegi panjang yang memiliki nama entity tersebut |
|  | *Weak Entity* | Suatu entity yang tidak dapat diidentifikasikan melalui atributnya dengan sendirinya. Keberadaan weak entity bergantung kepada entity lain yang disebut owner entity |
|  | *Associative Entity* | Entity yang digunakan pada many-to-many relationship (banyak antar banyak ) |
|  | *Attribute* | Dalam notasi Chen, Sebuah atribut digambarkan sebagai sebuah oval yang memuat nama atribut tersebut. |
|  | *Key attribute* | Suatu atribut yang mengidentifikasi suatu *entity* dengan sangat spesifik atau unik. Nama dalam *Key Attribute* selalu di-*underscore.* |
|  | *Multivalued attribute* | Attribute yang dapat memuat lebih dari satu nilai (Multivalued). Multivalued Attribute digambarkan dengan dua oval. |
|  | *Derived attribute* | Suatu attribute di mana nilainya dihitung atau berdasar dari atribut lain. Derived attribute mungkin atau tidak dapat disimpan dalam database. Attribute ini digambarkan dengan oval putus-putus. |
|  | *Strong relationship* | Suatu relationship (hubungan) di mana sebuah keberadaan entity bergantung dengan entity lain, dan PK (Primary Key) dari Child entity (entity anak) tidak memuat komponen PK Parent Entity (entity induk). Strong Relationship digambarkan dengan belah ketupat. |

Sumber : ( Baharsyah, 2020 )

## **Black box testing**

Pengujian kotak hitam atau black box testing adalah teknik pengujian perangkat lunak yang berfokus pada pengujian persyaratan fungsional aplikasi tanpa mengetahui cara kerja internalnya. Istilah *black box*sebenarnya mengacu pada fakta bahwa penguji tidak memiliki akses ke *code internal* atau desain aplikasi yang sedang diuji, melainkan hanya berinteraksi dengan *input*dan *output*sistem. Dalam pengujian kotak hitam, penguji memperhatikan perilaku dan fungsionalitas sistem, bukan *code*yang mendasarinya (Silvia, 2023 ).

## **Kueisioner**

Kuesioner merupakan daftar pertanyaan yang diberikan kepada [responden](https://penerbitdeepublish.com/responden-penelitian/" \t "_blank) secara langsung maupun tidak langsung. Kuesioner termasuk aspek penting dalam penelitian yang terdiri dari serangkaian pertanyaan untuk mengumpulkan informasi dari responden ( Abdhul, 2022 ).

Responden hanya perlu memilih atau menjawab daftar pertanyaan yang ada dalam kuesioner penelitian. Karena itu, kuesioner penelitian sudah bisa dianggap sebagai wawancara tertulis yang ditentukan berdasarkan jawaban responden ( Abdhul, 2022 ).

## 

# BAB III METODE PELAKSANAAN

## **Metode Pelaksanaan**

## **Pengumpulan Data**

## **Analisi Data Sistem Pakar**

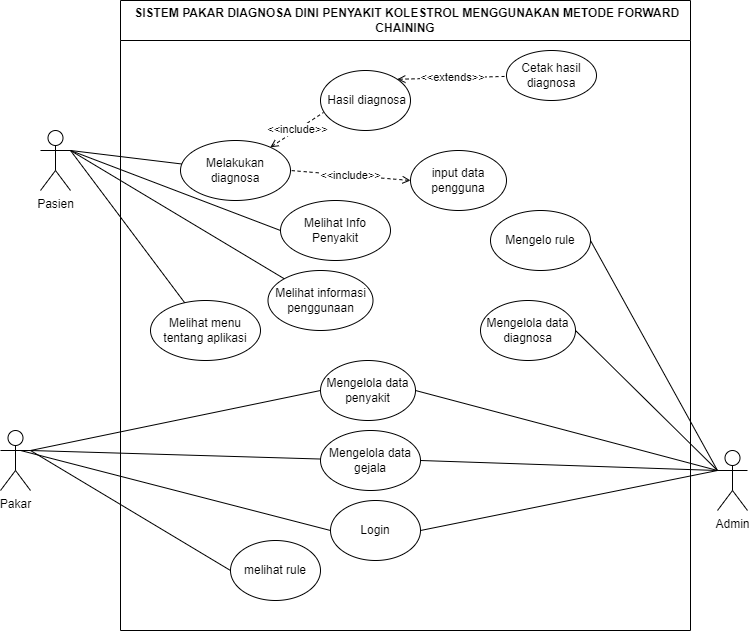
### **Analisis Data Penyakit**

* + 1. **Analisi Data Gejala**
    2. **Analisis Data Rule**

## **Analisis Kebutuhan Sistem**

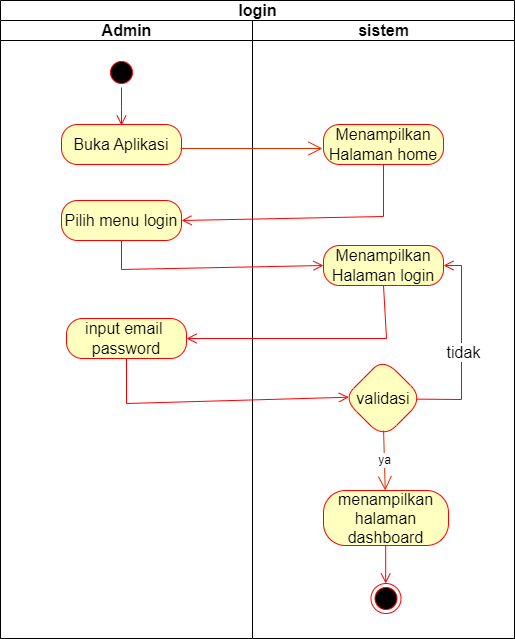
## **Perancangan Sistem**

### **Use Case Diagram**



* + 1. **Activity Diagram**

1. Login



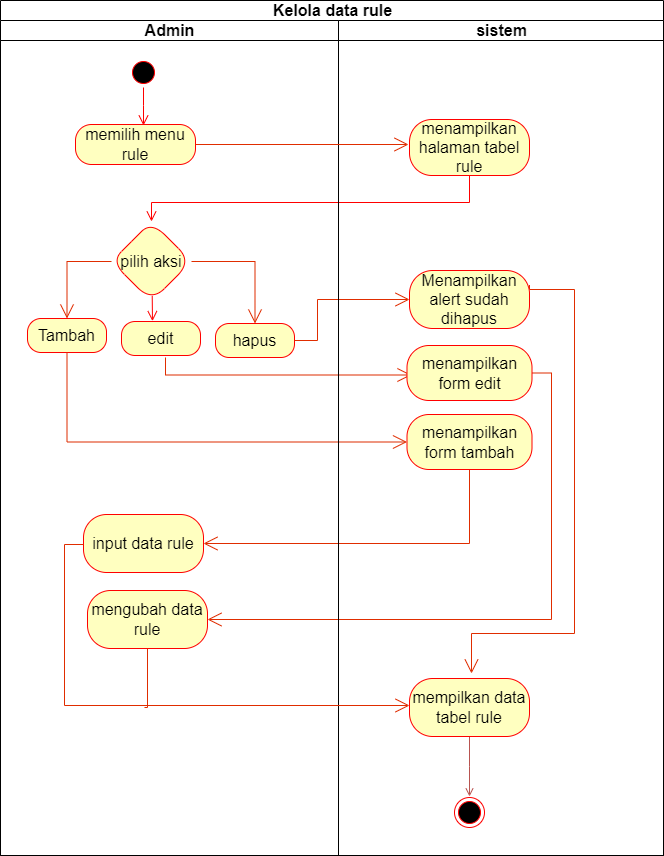
1. Kelola Data Gejala



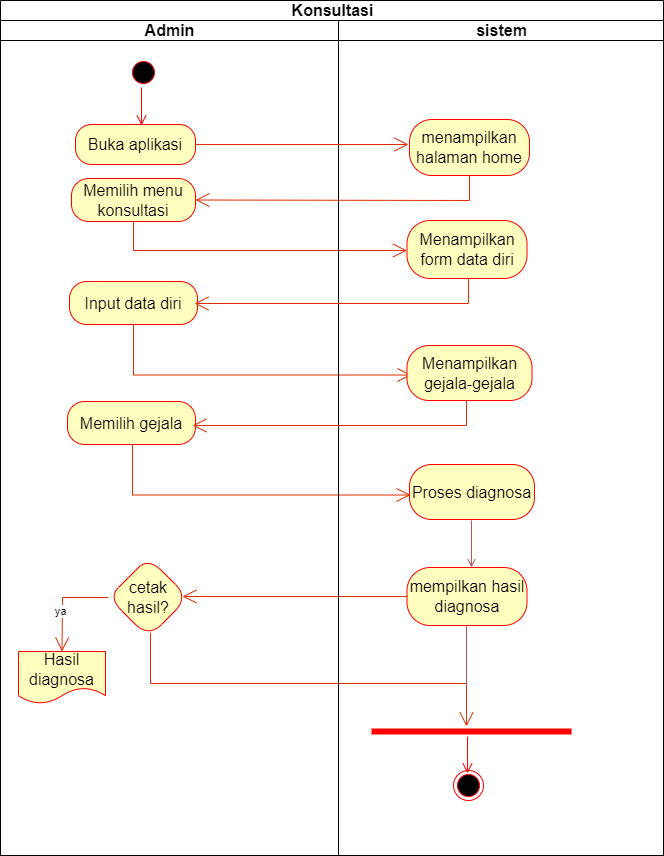
1. Kelola Data Penyakit



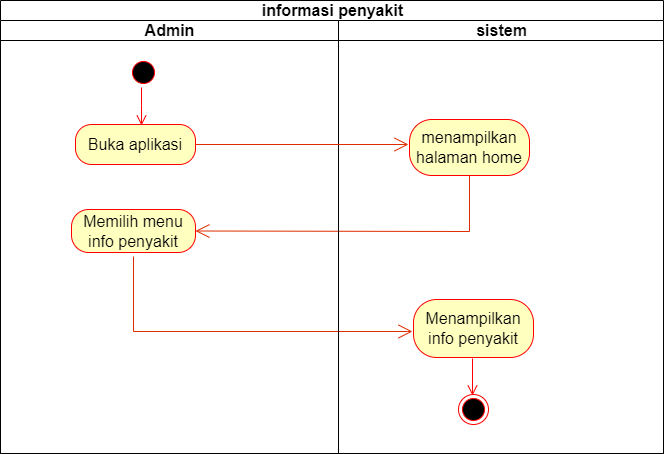
1. Kelola Data Rule



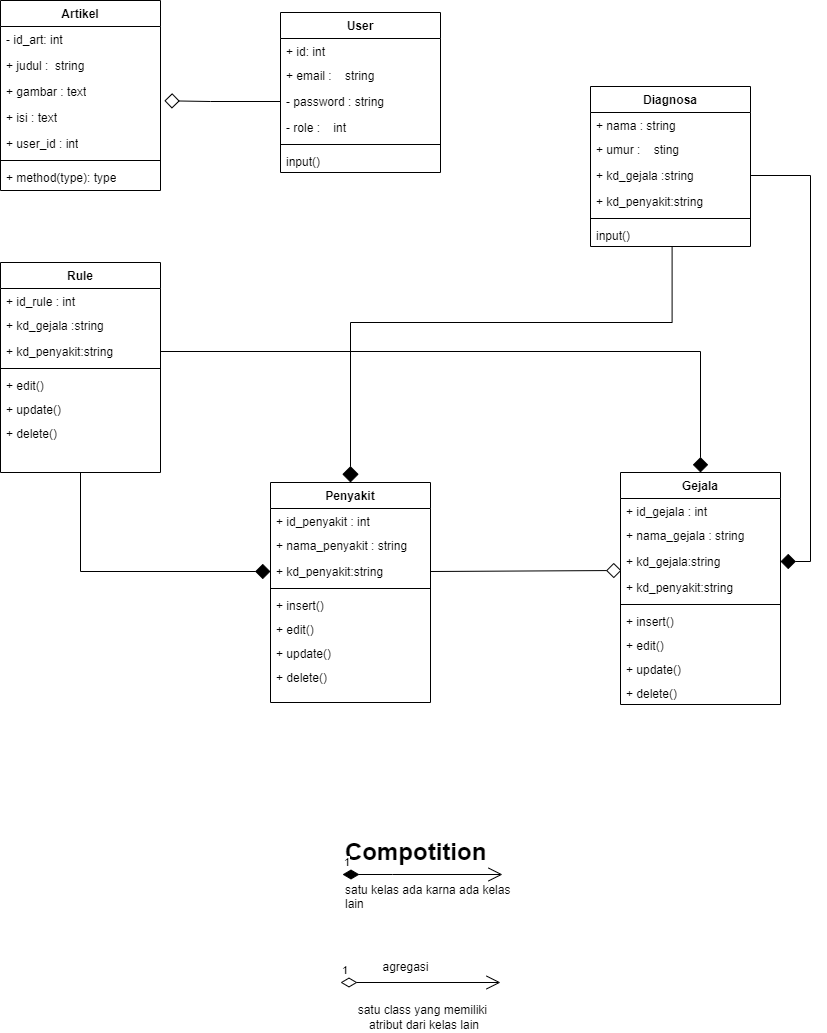
1. Konsultasi



1. Info Penyakit



1. Tentang Aplikasi
   * 1. **Class Diagram**

****

* + 1. **Sequence Diagram**

## **Flowchart**

### **Flowchart Program**

* + 1. **Flowchart sistem**

## **Perancangan ERD**

## **Perancangan Database**

## **Perancangan Antarmuka**

# BAB IV HASIL DAN PEMBAHASAN

## **4.1. Hasil**

## **4.2. Pembahasan**

# BAB V KESIMPULAN DAN SARAN

# Kesimpulan

# Saran

# DAFTAR PUSTAKA

Muljono, N. C., Devina Gunadi, & Agus Cahyo Nugroho. (2020). Rancang Bangun Website Pemesanan Makanan Kedai Twins Menggunakan Laravel PHP Framework. Jurnal Sains, Teknologi, Masyarakat dan Jejaring, 3(1), 49.

Sutrisno, Aprih Widayanto, & Muhammad Robbi Syahiri. (2020). Aplikasi Sistem Informasi Pemendek URL (SI SOUP) Berbasis Web. Indonesian Journal on Software Engineering, 6(1), 94-96

Salamah, U. G., & ST, S. (2021). Tutorial Visual Studio Code. Media Sains

Indonesia.

Dedi, Iqbal, M., & Fahroji, W.(2019). Sistem Informasi Administrasi Kependudukan Berbasis Web di Kelurahan Sangiang Jaya. Seminar Nasional APTIKOM (SEMNASTIK), 306-313.

Rosaly, R., & Prasetyo, A. (2019). Pengertian Flowchart Beserta Fungsi dan Simbol-simbol Flowchart yang Paling Umum Digunakan.

<https://pelajarindo.com/metode-waterfall-menurut-sommerville/>

<https://sekawanstudio.com/blog/apa-itu-use-case-diagram/>

<https://glints.com/id/lowongan/author/?authorid=96#.ZE6wkHZBzIU>

<https://www.jagoanhosting.com/blog/erd-apa-sih-itu/>