

BAB III

METODE PENELITIAN

3.1 Objek Penelitian

Objek penelitian yang digunakan dalam penelitian ini yaitu pendeteksi penyakit jantung dengan algoritma *certainty factor*. Batasan masalah dalam penelitian ini yaitu jenis penyakit jantung yang dapat di deteksi meliputi jantung koroner, aritmia dan gagal jantung.

1.1.1 Bahan Penelitian

Bahan penelitian berupa data yang diperoleh dari hasil wawancara dengan narasumber. Narasumber yaitu seorang dr. Yusak Alfarets Porotu'o, Sp.JP, FIHA sebagai seorang pakar dan pengguna aplikasi sistem pakar. Pengumpulan data diambil dari referensi literatur dari berbagai sumber seperti : jurnal, buku, website, youtube.

1.1.2 Peralatan penelitian

Peralatan yang digunakan dalam penelitian ini meliputi :

1. Perangkat keras yang digunakan dalam penelitian ini yaitu laptop.

Laptop merupakan perangkat keras yang digunakan untuk menunjang proses penelitian ini. Laptop yang digunakan memiliki spesifikasi :

Processor : AMD A9-9420 RADEON R5 64 bit
Layar : 14"
Hdd : 1 Tb
Ram : 4gb

2. Perangkat lunak yang digunakan dalam penelitian ini yaitu :

- a. *Sublime Text*

Dalam penelitian ini sublime text digunakan untuk proses penulisan kode pada pemrograman PHP.

- b. MySQL

MySQL pada penelitian ini digunakan sebagai basis data penyimpanan.

- c. Coreldraw X7

Aplikasi coreldraw difungsikan untuk membuat desain *user interface*.

d. Google Chrome

Didalam penelitian ini aplikasi google chrome digunakan untuk menampilkan preview pemrograman PHP yang dibuat.

e. Microsoft Word

Dalam penelitian ini *software* Microsoft word digunakan untuk pembuatan hasil penelitian yang dituangkan kedalam kata.

f. Android Studio

Android studio pada penelitian ini digunakan untuk melakukan pengkodean pada pembuatan aplikasi sistem pakar. Bahasa yang digunakan yaitu java.

g. Astah Profesional

Software Astah Profesional digunakan untuk membuat perancangan UML yang meliputi *usecase*, *activity diagram* dan *sequence diagram*.

3.2 Lokasi dan Waktu Penelitian

Penelitian akan dilaksanakan selama enam bulan. Lokasi penelitian untuk proses pembuatan aplikasi, validasi, dan revisi dilakukan di Lab Riset program studi Teknik Informatika, Fakultas Teknik dan Ilmu Komputer, Universitas Buana Perjuangan Karawang.

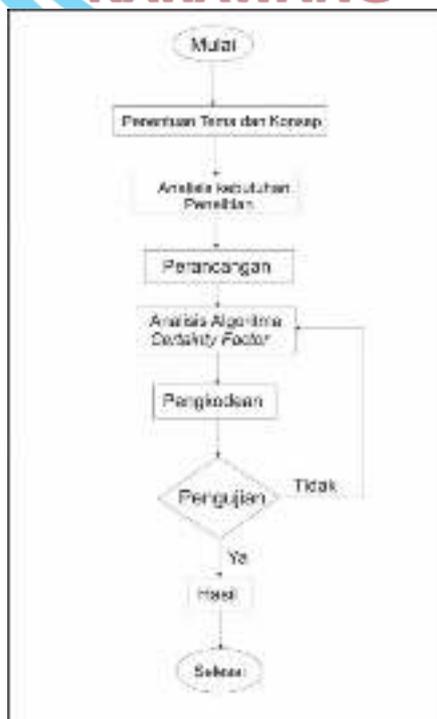
Tabel 3. 1 Tabel Waktu Penelitian

No	Kegiatan	Bulan ke-1				Bulan ke-2				Bulan ke-3				Bulan ke-4				Bulan ke-5				Bulan ke-6			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Penentuan tema dan konsep	■																							
2	Studi Literatur	■																							
3	Analisis data dan kebutuhan	■				■																			
4	Perancangan	■				■				■															

No	Kegiatan	Bulan ke-1				Bulan ke-2				Bulan ke-3				Bulan ke-4				Bulan ke-5				Bulan ke-6			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
5	Implementasi sistem																								
6	Analisis algoritma certainty factor																								
7	Pengkodean																								
8	Pengujian																								
9	Evaluasi																								

3.3 Prosedur Penelitian

Rancangan prosedur penelitian ini terdiri dari 7 (tujuh) tahapan meliputi : penentuan tema dan konsep, analisis kebutuhan penelitian, perancangan, analisis algoritma *certainty factor*, pengkodean, pengujian dan hasil. Prosedur penelitian digambarkan pada Gambar 3.1.



Gambar 3. 1 *Flow Chart* Rancangan Penelitian.

3.3.1 Penentuan Tema Dan Konsep.

Tahapan ini berisi kegiatan menghimpun informasi, ide dan permasalahan yang ada di lingkungan sekitar. Ide dan informasi bersumber dari literatur, jurnal, penelitian dan buku yang relevan. Tema penelitian adalah sistem pakar untuk mendeteksi penyakit dimana latar belakang telah diuraikan di Bab I. Sistem pakar pendeteksi penyakit di aplikasikan ke dalam pemrograman android yang dihitung menggunakan algoritma *certainty factor*.

3.3.2 Analisis Kebutuhan Penelitian.

Kegiatan menghimpun kebutuhan bahan dan alat penelitian kemudian akan dikelompokkan sesuai fungsi dan tugasnya. Tahapan ini juga meliputi pengumpulan data. Teknik pengumpulan data dilakukan dengan cara wawancara, merupakan pengumpulan data yang dilakukan dengan cara wawancara dengan pakar dan pengguna. Wawancara bertujuan untuk mendapatkan data yang akurat sesuai dengan kebutuhan pengguna.

3.3.3 Perancangan.

Setelah dilakukan analisis kebutuhan langkah selanjutnya adalah perancangan. Proses perancangan merupakan kegiatan merancang *user interface* untuk kebutuhan antar muka pengguna, *database* untuk kebutuhan penyimpanan dan relasi data, UML (*Unified Modelling Language*) dan analisa program. Berikut merupakan rancangan *user interface* :

1. Halaman *Dashboard user*

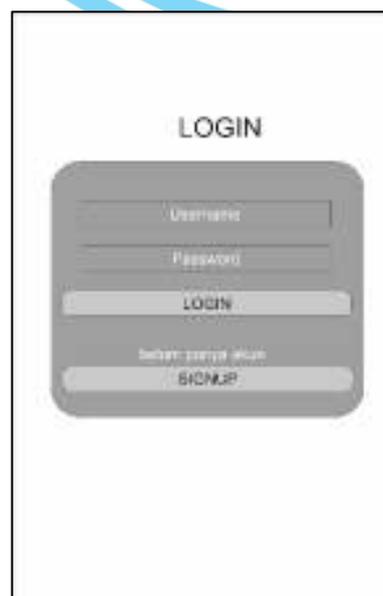
Halaman *dashboard* berisi menu yang tersedia pada aplikasi yang dibuat. Pada bagian atas *content* terdapat *text* nama aplikasi yang dibuat agar membuat *interface* menjadi menarik. Terdapat empat menu yang tersedia, meliputi : cek diagnosa, *about*, penyakit dan riwayat. Desain halaman *dashboard* ditampilkan pada Gambar 3.2.



Gambar 3. 2 Desain Halaman *Dashboard*

2. Halaman *Login*.

Halaman *login* berisi *form* login yang meliputi *form username* dan *password*. Di bagian bawah *form* login terdapat dua buah *button login* dan *signup*. Halaman *login* akan tampil setelah pengguna membuka aplikasi pada perangkat android. Desain halaman *login* ditampilkan pada Gambar 3.3.



Gambar 3. 3 Desain Halaman *Login*

3. Halaman *Signup*.

Halaman *signup* merupakan halaman yang berisi *form* untuk mendaftar menjadi pengguna aplikasi. Halaman *signup* meliputi *form* nama lengkap, *username*, dan *password*. Terdapat dua buah *button* untuk memproses tindakan pendaftaran dan *login*.

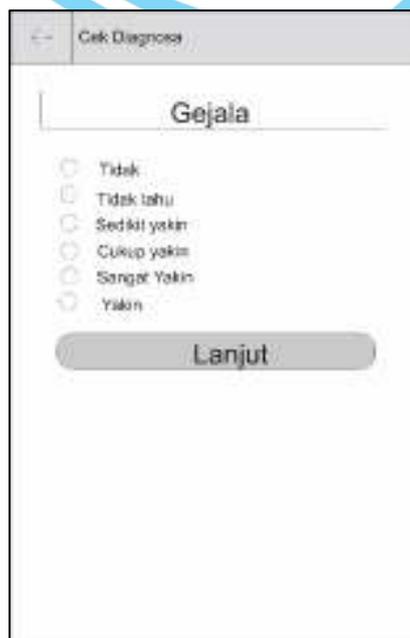


The image shows a vertical form for user registration. It contains three input fields: 'Nama lengkap', 'Username', and 'Password'. Below these fields are two buttons: 'Daftar' (Register) and 'Login'. A small text 'Silahkan pilih salah satu' is positioned above the 'Login' button.

Gambar 3. 4 Desain Halaman *Signup*

4. Halaman Cek Diagnosa

Halaman cek kesehatan berisi *form* pertanyaan yang harus di isi oleh pengguna. Terdapat dua buah *button* yang berfungsi untuk memproses diagnosa dan *button* kembali ke halaman sebelumnya.



The image shows a mobile app screen titled 'Cek Diagnosa'. It features a section labeled 'Gejala' with a list of radio button options: 'Tidak', 'Tidak tahu', 'Sedikit yakin', 'Cukup yakin', 'Sangat Yakin', and 'Yakin'. At the bottom of the form is a 'Lanjut' button.

Gambar 3. 5 Desain Halaman Cek Diagnosa

5. Halaman Penyakit

Halaman penyakit berisi daftar penyakit yang berkaitan dengan jantung. Kemudian terdapat *button* yang berfungsi untuk kembali ke halaman *dashboard*. Desain halaman tentang penyakit ditampilkan pada Gambar 3.6.



Gambar 3. 6 Desain Halaman Tentang Penyakit

Tampilan daftar penyakit menggunakan *listview*. Nama penyakit dapat dipilih kemudian akan tampil detail penyakit yang dipilih. Desain halaman detail penyakit ditampilkan pada gambar 3.7.



Gambar 3. 7 Desain Halaman Detail Penyakit

6. Halaman Riwayat

Halaman riwayat berisi riwayat hasil diagnosa sistem yang pernah dilakukan. Terdapat *button* yang berfungsi untuk kembali ke halaman sebelumnya dan menghapus riwayat. Desain halaman riwayat ditampilkan pada Gambar 3.8.



Gambar 3. 8 Desain Halaman Riwayat

7. Halaman *Dashboard* Admin.

Halaman *dashboard* admin berisi menu admin yang tersedia pada aplikasi yang dibuat. Pada bagian atas *content* terdapat *text* yang dibuat agar membuat *interface* menjadi menarik. Terdapat empat menu yang tersedia, meliputi : gejala, penyakit, aturan dan *logout*. Desain halaman *dashboard* ditampilkan pada Gambar 3.9.



Gambar 3. 9 Desain Halaman *Dashboard* Admin

8. Halaman Gejala.

Halaman gejala berisi daftar gejala setiap penyakit. Terdapat *button* yang berfungsi untuk kembali ke halaman sebelumnya dan menambah gejala baru. Desain halaman gejala ditampilkan pada Gambar 3.10.



Gambar 3. 10. Desain Halaman Gejala.

Button tambah digunakan untuk menambah gejala baru dan dapat merubah gejala yang sudah ada. Detail tampilan halaman tambah gejala ditampilkan pada Gambar 3.11.

A screenshot of a mobile application interface titled "Tambah Gejala". The page contains three input fields: "Kode gejala", "Nama gejala", and "Nilai Cf". Below these fields is a gray button labeled "Simpan".

Gambar 3. 11 Desain Halaman Tambah Gejala

9. Halaman Penyakit.

Halaman berisi daftar penyakit yang dapat di diagnosa oleh sistem. Terdapat *button* yang berfungsi untuk kembali ke halaman sebelumnya dan menambah penyakit baru. Desain halaman penyakit ditampilkan pada Gambar 3.12.



Gambar 3. 12 Desain Halaman Daftar Penyakit Admin

Button tambah digunakan untuk menambah penyakit baru dan dapat merubah detail yang sudah ada. Detail tampilan halaman tambah gejala ditampilkan pada Gambar 3.13.

Gambar 3. 13. Desain Halaman Tambah Penyakit

10. Halaman Aturan

Halaman berisi daftar aturan gejala dari setiap penyakit. Terdapat *button* yang berfungsi untuk kembali ke halaman sebelumnya dan menambah atau merubah aturan baru. Desain halaman aturan ditampilkan pada Gambar 3.14.

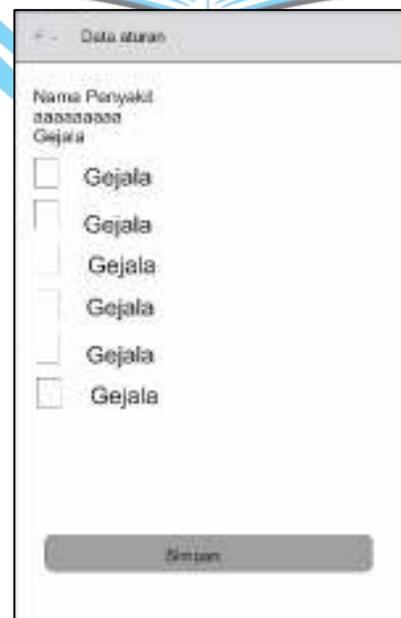
Gambar 3. 14. Desain Halaman Data Aturan.

Setiap daftar aturan dapat dipilih untuk menuju ke halaman detail aturan. Desain halaman detail aturan ditampilkan pada Gambar 3.15.



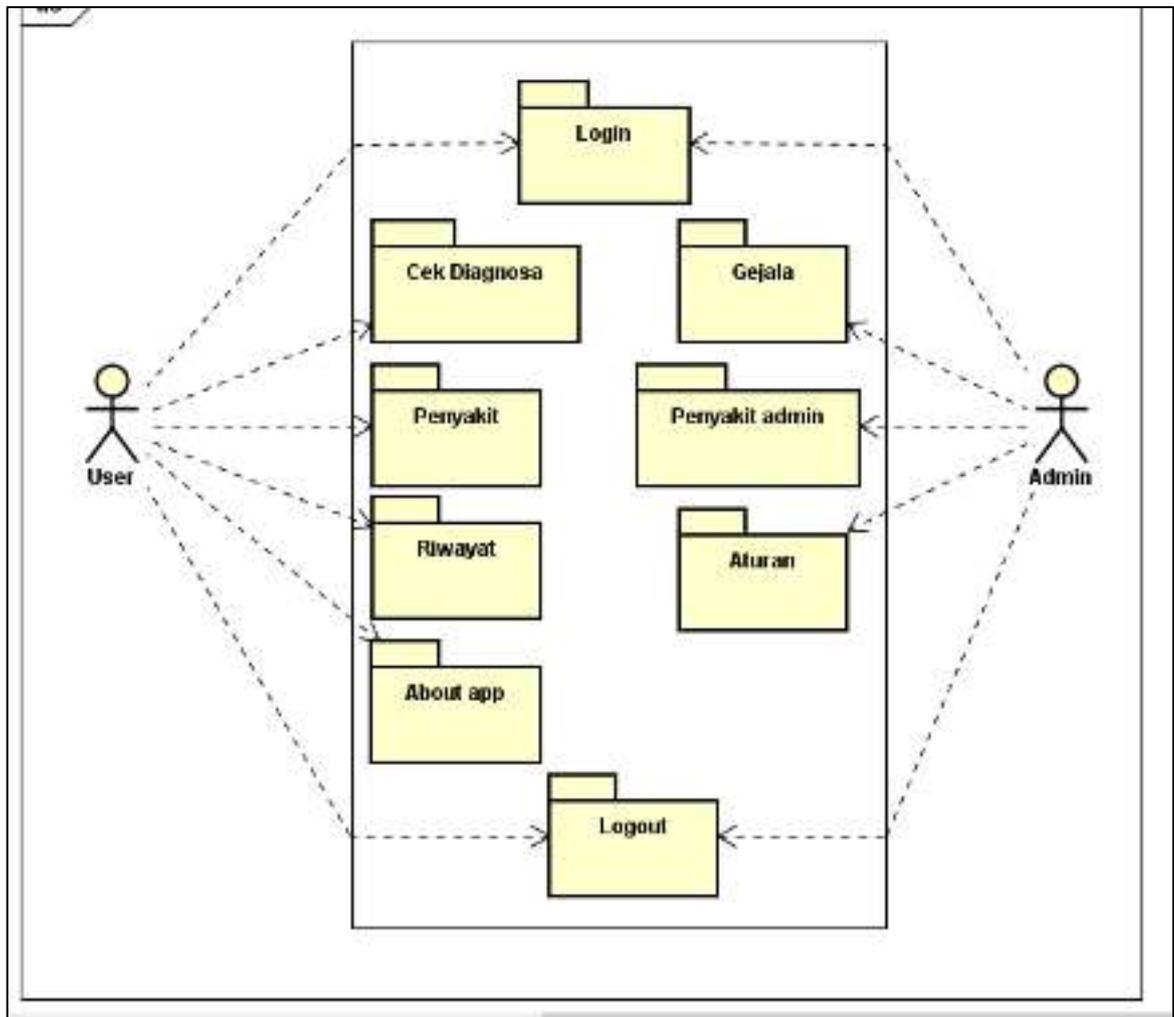
Gambar 3. 15. Desain Halaman Detail Aturan

Admin dapat mengatur ulang aturan gejala setiap penyakit pada halaman atur ulang aturan. Desain tampilan halaman atur ulang aturan ditampilkan pada Gambar 3.16.



Gambar 3. 16. Desain Halaman Atur Ulang Aturan.

Berikut merupakan perancangan UML yang ditunjukkan ke dalam bentuk *package diagram*.



Gambar 3. 17 Perancangan *Unified Modelling Language (UML)*

Berikut merupakan perancangan *database* yang dituangkan ke dalam tabel :

1. Tabel pengguna.

Tabel pengguna berfungsi untuk menyimpan data pengguna aplikasi sistem pakar. Tabel pengguna berisi id, nama, *username*, *password*, email.

Berikut merupakan isi dari tabel pengguna :

Tabel 3. 2 Perancangan Basisdata Tabel pengguna

No	Nama	Type data	Panjang Data	Auto Increment
1	Id_pengguna	int	11	Ya
2	nama_lengkap	varchar	20	
3	username	varchar	30	
4	password	varchar	30	
5	level	enum	(admin, user)	

2. Tabel gejala

Tabel gejala berfungsi untuk menyimpan daftar kode gejala, nama gejala penyakit jantung dan nilai cf masing masing gejala.

Tabel 3. 3 Perancangan Basisdata Tabel Gejala

No	Nama	Type Data	Panjang Data	Auto Increment
1	id_gejala	int	11	Ya
2	Kode_gejala	Varchar	5	
3	Nama_gejala	Text		
4	Nilai_cf	float		

3. Tabel penyakit

Tabel penyakit digunakan untuk menyimpan kode penyakit, nama penyakit dan solusi dari penyakit tersebut. Di dalam tabel penyakit terdapat data id_penyakit, kode_penyakit, nama_penyakit, deskripsi penyakit, dan solusi. Untuk lebih lengkap mengenai tabel penyakit solusi akan di tampilkan pada Tabel 3.4.

Tabel 3. 4 Perancangan Basisdata Tabel penyakit

No	Nama	Type Data	Panjang Data	Auto Increment
1	id_penyakit	int	11	ya
2	Kode_penyakit	Varchar	5	
3	nama_penyakit	Varchar	50	
4	deskripsi	Text		
5	solusi	Text		

4. Tabel aturan

Tabel aturan berfungsi untuk melakukan relasi antar tabel. Tabel relasi berisi id_relasi, kd_gejala, kd_penyakit dan bobot. Untuk lebih lengkap mengenai tabel penyakit solusi akan di tampilkan pada Tabel 3.5.

Tabel 3. 5 Perancangan Basisdata Tabel aturan

No	Nama	Type data	Panjang Data	Auto increment
1	id_aturan	Int	11	Ya
2	id_gejala	Int	11	
3	id_penyakit	Int	11	

5. Tabel Riwayat.

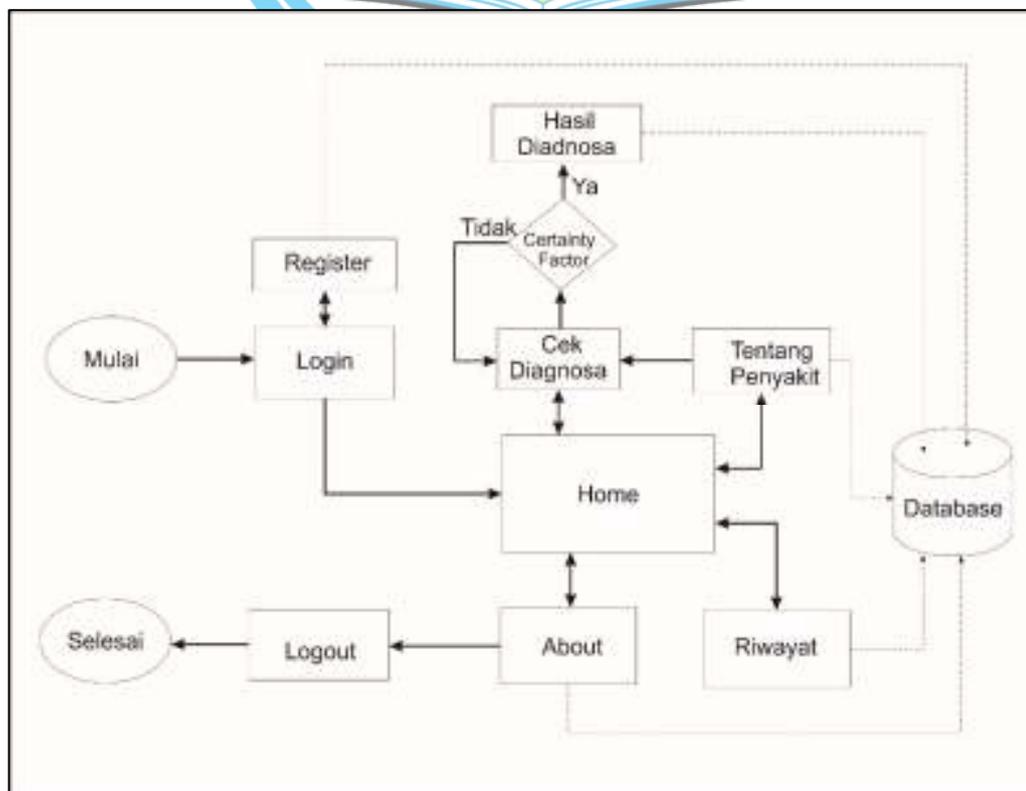
Tabel Riwayat berfungsi untuk menyimpan hasil diagnosa penyakit yang oleh *user*. Tabel Riwayat berisi id penyakit, id pengguna, tanggal, id riwayat dan nilai. Untuk lebih lengkap mengenai tabel penyakit solusi akan di tampilkan pada Tabel 3.6.

Tabel 3. 6 Perancangan Basisdata Tabel Riwayat

No	Nama	Type data	Panjang Data	Auto increment
1	id_riwayat	Int	11	Ya
2	id_pengguna	Int	11	
3	tanggal	Date		
4	id_penyakit	Int	11	
5	nilai	float		

3.3.4 Analisis Algoritma *Certainty Factor*.

Tahapan ini merupakan proses kalkulasi data gejala dengan nilai kepercayaan dan ketidakpercayaan yang akan dihitung menggunakan algoritma *certainty factor*. Berikut merupakan skema aplikasi yang ditampilkan pada Gambar 3.9.



Gambar 3. 18 Skema Aplikasi

Analisis dilakukan oleh sistem setelah pengguna memilih gejala yang dirasakan. Kaidah produksi yang digunakan yaitu *IF THEN*. Proses pencarian nilai *certainty factor* (CF) merupakan hasil pengurangan nilai kepercayaan terhadap hipotesis h (MBh) dengan nilai ketidakpercayaan hipotesis h (MDh). Pencarian kepercayaan dan ketidakpercayaan dipengaruhi oleh gejala yang dialami pengguna. Berikut merupakan gambaran proses analisis algoritma *certainty factor* :

Keterangan :

nilaiGejala1 = bobot atau nilai gejala 1 yang diberikan dari pakar

doubleGejala1 = nilai kepercayaan yang berasal dari pengguna

proses penghitungan CFh :

hasilHitungGejala1 (CF1) = nilaiGejala1 * doubleGejala1 ;

hasilHitungGejala2 (CF2) = nilaiGejala2 * doubleGejala2 ;

hasilHitungGejala3 (CF3) = nilaiGejala3 * doubleGejala3 ;

hasilHitungGejala(n) (CFn) = nilaiGejala(n) * doubleGejala(n) ;

proses penghitungan combine CFh (CFH) :

combine_CF1_CF2 = hasilHitungGejala1 (CF1) + hasilHitungGejala2 (CF2) * (1 - hasilHitungGejala1 (CF1));

combine_CF3 = combine_CF1_CF2 + hasilHitungGejala3 (CF3) * (1 - combine_CF1_CF2);

combine_CFn = combine_CF3 + hasilHitungGejala(n) (CFn) * (1 - combine_CF3)

;

Proses penghitungan prosentase CFH

hasilPerhitungan = combine_CFn * 100% ;

Berikut merupakan data gejala dan bobot yang diperoleh berdasarkan wawancara dengan dr. Yusak Alfarets Porotu'o, Sp.JP, FIHA selaku seorang pakar yang ditampilkan pada Tabel 3.6.

Tabel 3. 7 Tabel Kode gejala, Nama gejala Dan Bobot

Kode Gejala	Nama Gejala	Bobot
G001	Sesak napas	0,6
G002	Batuk kering	0,5
G003	Kaki Bengkak	0,4
G004	Dada terasa berat	0,5
G005	Mual	0,8
G006	Begah	0,5
G007	Jantung berdebar	0,8
G008	Pusing	0,6
G009	Kliyengan	0,7
G10	Sakit dada	0,4
G11	Keringat dingin	0,6
G12	Leher terasa tercekik	0,6

Berikut merupakan data penyakit yang dapat di deteksi berdasarkan gejala :

P01 : Gagal Jantung

P02 : Aritmia

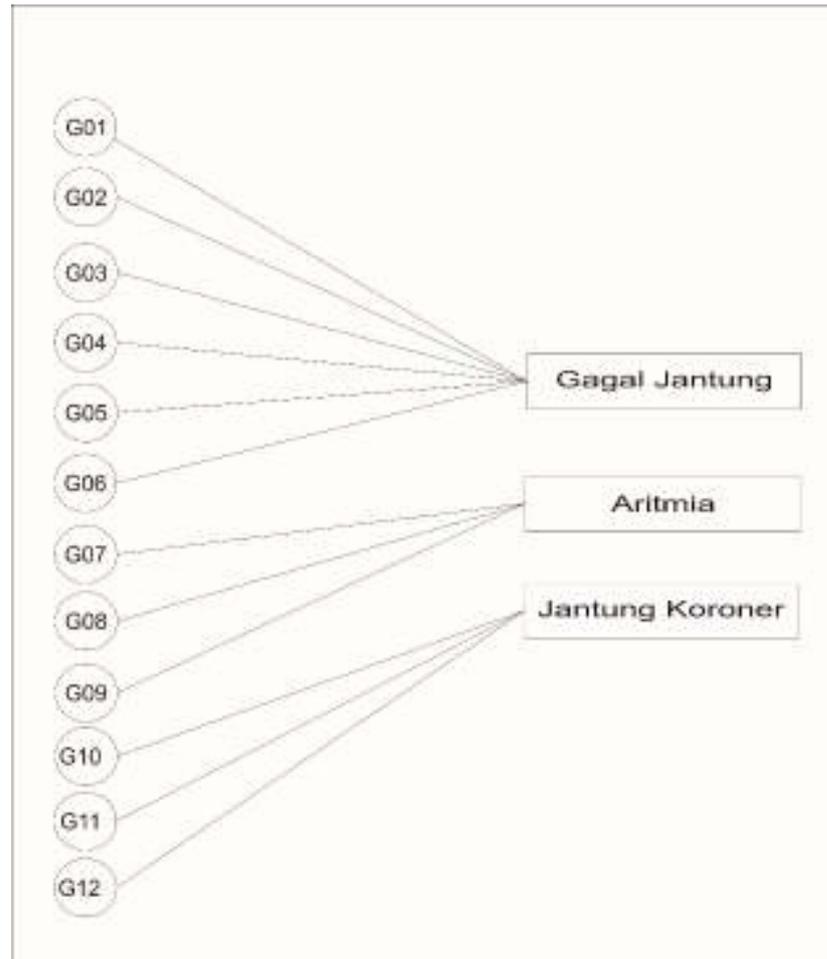
P03 : Jantung Koroner

Tabel 3. 8 Tabel *Rule Base* Penyakit

<i>Rule Base</i>	
Kode penyakit	Relasi Gejala
P01	G001, G002, G003, G004, G005, G006
P02	G007, G008, G009
P03	G10, G11, G12

Tabel 3. 9 Tabel Nilai Pengguna

No	Keterangan	Nilai Pengguna
1	Tidak	0
2	Tidak tahu	0,2
3	Sedikit yakin	0,4
4	Cukup yakin	0,6
5	Yakin	0,8
6	Sangat yakin	1



KARAWANG
Gambar 3.7 Bagan Aturan Penyakit Jantung

3.3.5 Pengkodean.

Tahapan ini berisi implementasi sistem pakar ke dalam Bahasa pemrograman android. Proses pembuatan *source code* aplikasi menggunakan bahasa pemrograman Java pada aplikasi android studio. Bahasa pemrograman disesuaikan dengan kebutuhan yaitu pemrograman android untuk aplikasi sistem pakar dan pemrograman web untuk halaman admin. Tahapan ini juga mengimplementasikan kaidah yang telah dibuat.

3.3.6 Pengujian.

Pengujian sistem dilakukan dengan aturan aturan tertentu dan melalui kondisi kondisi yang memungkinkan dilakukan. Kemudian dilakukan inferensi sesuai dengan aturan yang berlaku sehingga ditemukan semua masalah.

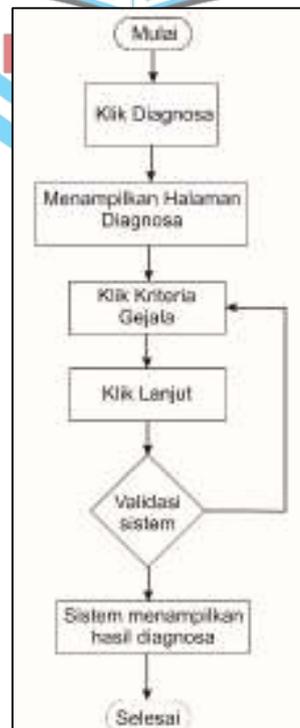
Pada penelitian ini menggunakan pengujian *Whitebox* dan *Blackbox*. Berikut dibawah ini penjelasan tentang pengujian :

A. Pengujian *Whitebox*.

Pengujian *whitebox* merupakan pengujian yang didasarkan pada struktural kode program. Pengujian *whitebox* memungkinkan tester dapat melihat kode program mana yang dipanggil pada setiap fungsi. Hal ini digunakan untuk menguji aliran data dan penanganan kesalahan.

1. Skenario Pengujian *Whitebox*

Dalam penelitian ini teknik pengujian *whitebox* menggunakan teknik *basis path* atau jalur dasar. Metode jalur dasar atau *basis path* merupakan salah satu metode *whitebox testing*, yang mana proses pengujian memerlukan pembuatan *flow graph* dari program *skrip* dan juga menentukan nilai kompleksitas siklomatik. Tes ini mempunyai tujuan untuk menganalisis kebenaran kinerja program dan struktur program (Rahayuda dan Santiari, 2017). Gambar 3.19 merupakan *flow chart* pengujian *whitebox* pada halaman cek diagnosa.



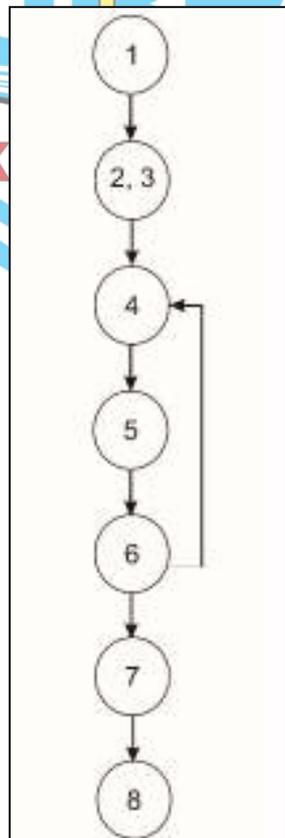
Gambar 3. 19 *Flow Chart* Halaman Cek Diagnosa

2. Tahapan Pengujian *Whitebox*.

Terdapat empat tahapan dalam pengujian *whitebox* menggunakan metode *basic path* yaitu :

- a. Menggambarkan grafik alir (*flow graph*).
- b. Menentukan pengukuran tingkat kompleksifitas (*cyclomatic complexity*).
- c. Menentukan jalur-jalur dasar sesuai dengan jumlah dari *cyclomatic complexity*.
- d. Mendefinisikan kasus-kasus uji untuk setiap jalur dasar yang telah ditentukan.

Setelah proses pembuatan *flowchart*, tahapan selanjutnya yaitu mengubah *flowchart* menjadi *flow graph*. Gambar 3.20 merupakan *flow graph* halaman cek diagnosa.



Gambar 3. 20 *Flow Graph* Halaman Cek Diagnosa

B. Pengujian *Blackbox*.

Pengujian *Blackbox* merupakan suatu pendekatan untuk menguji apakah fungsi yang ada disetiap program dapat berjalan dengan benar. Pengujian *Blackbox* biasanya tanpa memperdulikan stuktur kendali di dalam program. Pengujian tersebut dapat diartikan sebagai pengujian fungsional, karena dapat menguji apakah program berjalan sesuai spesifikasi kebutuhan.

1. Tujuan Pengujian *Blackbox*

Tujuan pengujian *Blackbox* pada sistem untuk melihat apakah sistem dibuat sudah sesuai dengan tujuan awal pembuatan dan layak untuk dipergunakan.

2. Prosedur Pengujian *Blackbox*

Pengujian *Blackbox* memiliki beberapa prosedur pengujian, berikut beberapa prosedur dari Pengujian *Blackbox* :

a. *Equivalence Partitioning*

Pengujian ini dilakukan pada alat dan sistem dengan memasukkan data yang tidak sesuai dengan tipe data atau memasukkan data acak.

b. *Boundary Value Analysis*

Pengujian ini untuk memastikan bahwa masukkan data yang melebihi batas yang sudah ditentukan tidak dapat tersimpan dengan baik pada database.

c. *Comparison Testing*

Membandingkan tampilan interface sistem pada browser yang berbeda.

d. *Sample Testing*

Pengujian ini untuk memastikan nilai yang terpilih dapat menghasilkan data yang baik dan sesuai dengan data masukkan dari user.

e. *Robustness Testing*

Penguji akan memasukkan data acak untuk membuktikan bahwa tidak ada kesalahan jika masukan tidak valid.

f. Behavior Testing

Pengujian ini dilakukan dengan cara membuat data baru secara berkali – kali untuk menghindari data stack.

g. Performance Testing

Pengujian ini mengevaluasi kemampuan program untuk beroperasi dengan benar dipandang dari aliran pemakaian memori.

h. Requirement Testing

Tipe ini hanya melihat spesifikasi kebutuhan dari sistem mulai dari sistem pembuatan sampai pengujian.

i. Endurance Testing

Tipe ini untuk memastikan apakah hasil operasi matematika pada sistem ini benar atau salah.

j. Cause – Effect Relationship

Testing Pengujian yang melibatkan kondisi input dan aliran data mulai dari Input, View, Update dan Delete.

C. Pengujian Pakar.

Pengujian pakar dilakukan untuk mengetahui tingkat akurasi aplikasi sistem pakar. Proses pengujian pakar dilakukan setelah sistem diuji dengan metode *whitebox* dan *blackbox*. Pengujian pakar dilakukan oleh dr. Hervin Ramadhani, Sp.JP, FIHA selaku pakar. Pengujian pakar dilakukan dengan cara membandingkan hasil diagnosa kasus-kasus yang terjadi menggunakan aplikasi sistem pakar dengan hasil diagnosa dokter selaku pakar.

$$\text{Persentase pengujian} = \frac{\text{Jumlah prediksi yang sesuai}}{\text{Jumlah kasus}} \times 100\% \quad (7)$$