

**PENERAPAN METODE MINKOWSKI DISTANCE
SIMILARITY PADA CASE BASED REASONING UNTUK
DIAGNOSIS PENYAKIT STROKE**

Skripsi



oleh:

**ANGELINA RUMUY
71190506**

**PROGRAM STUDI INFORMATIKA FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA**

2023

**PENERAPAN METODE MINKOWSKI DISTANCE
SIMILARITY PADA CASE BASED REASONING UNTUK
DIAGNOSIS PENYAKIT STROKE**

Skripsi



Diajukan kepada Program Studi Informatika Fakultas Teknologi Informasi
Universitas Kristen Duta Wacana
Sebagai Salah Satu Syarat dalam Memperoleh Gelar
Sarjana Komputer

Disusun oleh

ANGELINA RUMUY

71190506

PROGRAM STUDI INFORMATIKA FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA

2023

PERNYATAAN KEASLIAN SKRIPSI

PERNYATAAN KEASLIAN SKRIPSI

Saya menyatakan dengan sesungguhnya bahwa skripsi dengan judul:

PENERAPAN METODE MINKOWSKI DISTANCE SIMILARITY PADA CASEBASED REASONING UNTUK DIAGNOSIS PENYAKIT STROKE

yang saya kerjakan untuk melengkapi sebagian persyaratan menjadi Sarjana Komputer pada pendidikan Sarjana Program Studi Informatika Fakultas Teknologi Informasi Universitas Kristen Duta Wacana, bukan merupakan tiruan atau duplikasi dari skripsi keserjanaan di lingkungan Universitas Kristen Duta Wacana maupun di Perguruan Tinggi atau instansi manapun, kecuali bagian yang sumber informasinya dicantumkan sebagaimana mestinya.

Jika dikemudian hari didapati bahwa hasil skripsi ini adalah hasil plagiasi atau tiruan dari skripsi lain, saya bersedia dikenai sanksi yakni pencabutan gelar keserjanaan saya.

Yogyakarta, 8 Juni 2023


ANGELINA RUMUY
71190506

DUTA WACANA

HALAMAN PERSETUJUAN

HALAMAN PERSETUJUAN

Judul Skripsi : PENERAPAN METODE MINKOWSKI DISTANCE
SIMILARITY PADA CASE BASED REASONING
UNTUK DIAGNOSIS PENYAKIT STROKE

Nama Mahasiswa : ANGELINA RUMUY

N I M : 71190506

Matakuliah : Skripsi (Tugas Akhir)

Kode : TI0366

Semester : Genap

Tahun Akademik : 2022/2023

Telah diperiksa dan disetujui di
Yogyakarta,
Pada tanggal 27 Juni 2023

Dosen Pembimbing I



Rosa Delima, Dr. S.Kom., M.Kom.

Dosen Pembimbing II



Laurentius Kuncoro Probo Saputra,
S.T., M.Eng.

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
SKRIPSI/TESIS/DISERTASI UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademika Universitas Kristen Duta Wacana, saya yang bertanda tangan di bawah ini:

Nama : Angelina Rumuy
NIM : 71190506
Program studi : Informatika
Fakultas : Teknologi Informasi
Jenis Karya : Skripsi

demikian pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Kristen Duta Wacana **Hak Bebas Royalti Noneksklusif** (*None-exclusive Royalty Free Right*) atas karya ilmiah saya yang berjudul:

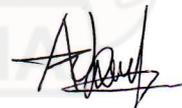
**“PENERAPAN METODE MINKOWSKI DISTANCE SIMILARITY PADA
CASE BASED REASONING UNTUK DIAGNOSIS PENYAKIT STROKE”**

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti/Noneksklusif ini Universitas Kristen Duta Wacana berhak menyimpan, mengalih media/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama kami sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Yogyakarta
Pada Tanggal : 4 Juli 2023

Yang menyatakan



(Angelina Rumuy)

NIM.71190506

HALAMAN PENGESAHAN

HALAMAN PENGESAHAN

PENERAPAN METODE MINKOWSKI DISTANCE SIMILARITY PADA CASE BASED REASONING UNTUK DIAGNOSIS PENYAKIT STROKE

Oleh: ANGELINA RUMUY / 71190506

Dipertahankan di depan Dewan Penguji Skripsi
Program Studi Informatika Fakultas Teknologi Informasi
Universitas Kristen Duta Wacana - Yogyakarta
Dan dinyatakan diterima untuk memenuhi salah satu syarat memperoleh gelar
Sarjana Komputer
pada tanggal 16 Juni 2023

Yogyakarta, 27 Juni 2023
Mengesahkan,

Dewan Penguji:

1. Rosa Delima, Dr. S.Kom., M.Kom.
2. Laurentius Kuncoro Probo Saputra, S.T.,
M.Eng.
3. R. Gunawan Santosa, Drs. M.Si.
4. Yuan Lukito, S.Kom., M.Cs.




Dekan
(Restyandito, S.Kom., MSIS., Ph.D.)

Ketua Program Studi

(Gloria Virginia, Ph.D.)

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS
SECARA ONLINE
UNIVERSITAS KRISTEN DUTA WACANA YOGYAKARTA**

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS
SECARA ONLINE
UNIVERSITAS KRISTEN DUTA WACANA YOGYAKARTA**

Saya yang bertanda tangan di bawah ini:

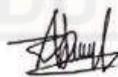
NIM : 71190506
Nama : Angelina Rumuy
Prodi / Fakultas : Teknologi Informasi / Informatika
Judul Tugas Akhir : Penerapan Metode Minkowski Distance
Similarity Pada Case Based Reasoning Untuk
Diagnosis Penyakit Stroke

bersedia menyerahkan Tugas Akhir kepada Universitas melalui Perpustakaan untuk keperluan akademis dan memberikan Hak Bebas Royalti Non Eksklusif (*Non-exclusive Royalty-free Right*) serta bersedia Tugas Akhirmya dipublikasikan secara online dan dapat diakses secara lengkap (*full access*).

Dengan Hak Bebas Royalti Noneklusif ini Perpustakaan Universitas Kristen Duta Wacana berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk *database*, merawat, dan mempublikasikan Tugas Akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta. Demikian pernyataan ini saya buat dengan sebenar-benarnya.

Yogyakarta, 8 Juni 2023

Yang menyatakan,



72290506 – Angelina Rumuy



Karya sederhana ini dipersembahkan
kepada Tuhan, Keluarga Tercinta,
dan Kedua Orang Tua



Segala sesuatu indah pada waktu-Nya

Anonim

Perjalanan ribuan mil dimulai dari langkah satu mil

(Pepatah Kuno)

KATA PENGANTAR

Segala puji dan syukur kepada Tuhan yang maha kasih, karena atas segala rahmat, bimbingan, dan bantuan-Nya sehingga penulis dapat menyelesaikan penelitian tugas akhir ini dengan baik.

Penulis memperoleh banyak bantuan dari kerja sama baik secara moral maupun spiritual dalam penulisan Skripsi ini, untuk itu tak lupa penulis ucapkan terima kasih yang sebesar-besarnya kepada:

1. Tuhan yang maha kasih, atas rahmat dan karunia-Nya yang menguatkan penulis dalam menyelesaikan skripsi ini.
2. Orang tua tercinta yang selama ini telah membantu penulis dalam bentuk perhatian, semangat, dan doa yang tidak henti-hentinya.
3. Dosen Pembimbing 1 selaku Dosen Pembimbing 1, yang telah memberikan ilmunya dan dengan penuh kesabaran membimbing penulis untuk menyelesaikan skripsi ini.
4. Dosen Pembimbing 2, selaku Dosen Pembimbing 2 yang telah memberikan ilmu dan kesabaran dalam membimbing penulis,
5. Saudara Nelson yang telah memberikan izin kepada penulis untuk menggunakan datanya didalam penelitian ini.
6. Keluarga tercinta yang selalu memberikan doa dan dukungan sehingga penulis dapat menyelesaikan skripsi ini.
7. Teman-teman seperjuangan dan sahabat-sahabat yang telah memberikan dukungan dan semangat kepada penulis dalam menyelesaikan skripsi ini.
8. Lain-lain yang telah mendukung moral, spiritual, dan dana untuk belajar selama ini.

Akhir kata, penulis meminta maaf jika ada kesalahan selama penulisan skripsi ini. Semoga skripsi ini dapat bermanfaat bagi pembaca semua.

Yogyakarta, 06 Juni 2023

Penulis

DAFTAR ISI

PERNYATAAN KEASLIAN SKRIPSI.....	iii
HALAMAN PERSETUJUAN.....	iv
HALAMAN PENGESAHAN.....	v
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS SECARA ONLINE	v
UNIVERSITAS KRISTEN DUTA WACANA YOGYAKARTA	vi
KATA PENGANTAR	ix
DAFTAR ISI.....	x
DAFTAR TABEL.....	xiii
DAFTAR GAMBAR	xv
INTISARI.....	xvii
ABSTRACT.....	xix
BAB I	1
PENDAHULUAN	1
1.1. Latar Belakang Masalah	1
1.2. Perumusan Masalah.....	2
1.3. Batasan Masalah.....	2
1.4. Tujuan Penelitian.....	2
1.5. Manfaat Penelitian.....	3
1.6. Metodologi Penelitian	3
1. Studi Pustaka.....	3
2. Pengumpulan Data.....	3
3. Perancangan Sistem.....	3
4. Implementasi Sistem.....	3
5. Pengujian Sistem.....	4
1.7. Sistematika Penulisan.....	4
BAB II.....	5
TINJAUAN PUSTAKA DAN DASAR TEORI	5

2.1	Tinjauan Pustaka	5
2.2	Landasan Teori	9
BAB III		26
METODOLOGI PENELITIAN.....		26
3.1	Analisis Kebutuhan Sistem	26
3.2	Perancangan Penelitian.....	26
1.	Pengumpulan Data.....	26
2.	Perancangan Sistem.....	26
3.	Implementasi Sistem.....	27
4.	Pengujian Sistem.....	27
3.3	Diagram Alir.....	27
3.4	Use Case Diagram	29
3.5	Perancangan Basis Data	30
3.6	Perancangan Antarmuka Pengguna.....	37
3.7	Perancangan Pengujian Sistem.....	41
BAB IV		43
IMPLEMENTASI DAN PEMBAHASAN.....		43
4.1	Implementasi Antar Muka Sistem.....	43
4.2	Implementasi Case Based Reasoning.....	46
4.3	Pengujian dan Analisis	55
4.4	Pembahasan	74
BAB V.....		77
KESIMPULAN DAN SARAN.....		77
4.5	Kesimpulan.....	77
4.6	Saran.....	77
DAFTAR PUSTAKA		78
LAMPIRAN A		81
DATA REKAM MEDIK PENYAKIT STROKE RUMAH SAKIT DKT DR. SOETARTO, YOGYAKARTA (2015 – 2016).....		81
LAMPIRAN B		104

KARTU KONSULTASI DOSEN 1.....	104
LAMPIRAN C	105
KARTU KONSULTASI DOSEN 2.....	105
LAMPIRAN D.....	106
KODE PROGRAM.....	106



DAFTAR TABEL

Tabel 2.1: Rangkuman Tinjauan Pustaka.....	6
Tabel 2.2: Confusion Matrix	12
Tabel 2.3: Pengkodean Gejala Stroke (Nelson R et al., 2018).....	15
Tabel 2.4: Faktor Risiko Stroke (Nelson R et al., 2018).....	18
Tabel 2.5: Contoh Representasi Kasus (K00007) Menggunakan Model Frame (Nelson R et al., 2018)	20
Tabel 2.6: Data Kasus Baru	23
Tabel 4.1: Hasil pengujian oleh dokter (<i>stroke</i> iskemik).....	57
Tabel 4.2: Hasil pengujian oleh dokter (<i>stroke</i> hemoragik).....	57
Tabel 4.3: Rincian Hasil Pengujian pada <i>Threshold</i> = 60.....	58
Tabel 4.4: Perhitungan Nilai <i>Accuracy</i> , <i>Recall</i> , dan <i>precision</i> pada <i>Threshold</i> = 60	59
Tabel 4.5: Rincian Hasil Pengujian pada <i>Threshold</i> = 65.....	60
Tabel 4.6: Perhitungan Nilai <i>Accuracy</i> , <i>Recall</i> , dan <i>Precision</i> pada <i>Threshold</i> = 65	61
Tabel 4.7: Rincian Hasil Pengujian pada <i>Threshold</i> = 70.....	62
Tabel 4.8: Perhitungan Nilai <i>Accuracy</i> , <i>Recall</i> , dan <i>precision</i> pada <i>Threshold</i> = 70	62
Tabel 4.9: Rincian Hasil Pengujian pada <i>Threshold</i> = 75.....	63
Tabel 4.10: Perhitungan Nilai <i>Accuracy</i> , <i>Recall</i> , dan <i>precision</i> pada <i>Threshold</i> = 75.....	64
Tabel 4.11: Rincian Hasil Pengujian pada <i>Threshold</i> = 80.....	65
Tabel 4.12: Perhitungan Nilai <i>Accuracy</i> , <i>Recall</i> , dan <i>precision</i> pada <i>Threshold</i> = 80.....	65
Tabel 4.13: Rincian Hasil Pengujian pada <i>Threshold</i> = 85.....	66
Tabel 4.14: Perhitungan Nilai <i>Accuracy</i> , <i>Recall</i> , dan <i>precision</i> pada <i>Threshold</i> = 85.....	67
Tabel 4.15: Rincian Hasil Pengujian pada <i>Threshold</i> = 90.....	68

Tabel 4.16: Perhitungan Nilai <i>Accuracy</i> , <i>Recall</i> , dan <i>precision</i> pada <i>Threshold</i> = 90.....	69
Tabel 4.17: Rincian Hasil Pengujian pada <i>Threshold</i> = 95.....	70
Tabel 4.18: Perhitungan Nilai <i>Accuracy</i> , <i>Recall</i> , dan <i>precision</i> pada <i>Threshold</i> = 95.....	70
Tabel 4.19: Nilai Akurasi Tertinggi pada Setiap <i>Threshold</i>	71



DAFTAR GAMBAR

Gambar 2.1: Struktur Sistem Pakar (Mankad, 2015).....	10
Gambar 2.2: CBR Cycle (Mendes & Ramos, 2012).....	11
Gambar 3.1: Diagram alir sistem CBR penyakit <i>stroke</i>	28
Gambar 3.2: Diagram alir perhitungan similaritas global.....	29
Gambar 3.3: <i>Usecase diagram</i> sistem pakar diagnosis penyakit <i>stroke</i>	30
Gambar 3.4: ER Diagram Sistem Pakar untuk Mendiagnosa Penyakit Stroke	31
Gambar 3.5: Rancangan Relasi Basis Data Sistem Pakar Diagnosis Stroke	32
Gambar 3.6: Rancangan Antarmuka Halaman Login	38
Gambar 3.7: Rancangan Antarmuka Halaman Diagnosis.....	39
Gambar 3.8: Rancangan Antarmuka Halaman Hasil Diagnosa	39
Gambar 3.9: Rancangan Antarmuka Halaman User	40
Gambar 3.10: Rancangan Antarmuka Halaman Basis Pengetahuan	40
Gambar 3.11: Rancangan Antarmuka Halaman Edit Basis Kasus	41
Gambar 3.12: Rancangan Antarmuka Halaman Tambah Basis Kasus	41
Gambar 4.1: Halaman <i>login</i>	43
Gambar 4.2: Halaman diagnosis	44
Gambar 4.3: Halaman hasil diagnosis.....	45
Gambar 4.4: Halaman manajemen pengguna	45
Gambar 4.5: Halaman pengolahan basis pengetahuan.....	46
Gambar 4.6: Antarmuka proses <i>retrieve</i>	51
Gambar 4.7: Antarmuka proses <i>revise</i> (data pasien, gejala, faktor risiko)	53
Gambar 4.8: Antarmuka proses <i>revise</i> (hasil diagnosis dan keyakinan pakar).....	53
Gambar 4.9: Grafik hasil pengujian sistem pada <i>threshold</i> = 60.....	60
Gambar 4.10: Grafik hasil pengujian sistem pada <i>threshold</i> = 65.....	61
Gambar 4.11: Grafik hasil pengujian sistem pada <i>threshold</i> = 70.....	63
Gambar 4.12: Grafik hasil pengujian sistem pada <i>threshold</i> = 75.....	64
Gambar 4.13: Grafik hasil pengujian sistem pada <i>threshold</i> = 80.....	66
Gambar 4.14: Grafik hasil pengujian sistem pada <i>threshold</i> = 85.....	68
Gambar 4.15: Grafik hasil pengujian sistem pada <i>threshold</i> = 90.....	69

Gambar 4.16: Grafik hasil pengujian sistem pada <i>threshold</i> = 95.....	71
Gambar 4.17: Grafik tingkat akurasi pada setiap nilai <i>threshold</i>	72
Gambar 4.18: Grafik nilai <i>recall</i> pada setiap nilai <i>threshold</i>	73
Gambar 4.19: Grafik tingkat presisi pada setiap nilai <i>threshold</i>	73
Gambar 4.20: Grafik perbandingan nilai akurasi, <i>recall</i> dan presisi pada setiap nilai <i>threshold</i>	74
Gambar 4.21. Perbandingan tingkat akurasi dan <i>recall</i> antara <i>Jacard Coeficient</i> & <i>Minkowski Distance</i>	75



INTISARI

PENERAPAN METODE MINKOWSKI DISTANCE SIMILARITY PADA CASE BASED REASONING UNTUK DIAGNOSIS PENYAKIT STROKE

Oleh

ANGELINA RUMUY

71190506

Stroke merupakan penyakit serebrovaskuler yang ditandai dengan gangguan fungsi otak karena adanya kerusakan atau kematian jaringan otak akibat berkurang atau tersumbatnya aliran darah dan oksigen ke otak. Sistem pakar dapat digunakan sebagai alat bantu belajar bagi mahasiswa kedokteran dalam pembelajaran untuk mendiagnosis penyakit stroke. Rekam medis kasus *stroke* dapat digunakan kembali sebagai acuan untuk mendiagnosis penyakit *stroke* yang diderita ketika ada kasus baru, disebut sebagai metode *case-based reasoning* (CBR).

Penelitian ini mengimplementasikan metode *minkowski distance similarity* pada CBR untuk menghitung nilai kemiripan antar kasus, dimana setiap kasus yang mirip memiliki solusi yang sama. Penelitian ini bermaksud untuk mendapatkan nilai r yang paling optimal dan nilai *threshold* yang paling tepat dalam sistem pakar untuk diagnosis penyakit *stroke* dengan menggunakan metode *minkowski distance similarity* pada CBR.

Proses diagnosis dilakukan dengan memasukan kondisi, gejala dan faktor risiko pasien kemudian sistem akan menghitung nilai similaritas dan mengambil kasus dengan nilai similaritas tertinggi sebagai solusi dengan ketentuan nilai similaritas harus lebih besar atau sama dengan nilai *threshold*. Berdasarkan pengujian sistem nilai akurasi terbaik dicapai pada penerapan nilai *threshold* 75 pada nilai $r = 3$ atau $r = 4$ dengan tingkat akurasi sistem sebesar 88,89%, *recall* sebesar 88% dan *precision* sebesar 100%.

Kata-kata kunci: sistem pakar, *stroke*, diagnosis, *case-based reasoning*, *minkowski distance similarity*, *threshold*, akurasi.



ABSTRACT

APPLICATION OF MINKOWSKI DISTANCE SIMILARITY METHODS ON CASED REASONING FOR STROKE DISEASES DIAGNOSIS

By

ANGELINA RUMUY

71190506

Stroke is a cerebrovascular disease characterized by impaired brain function due to damage or death of brain tissue caused by reduced or blocked blood and oxygen flow to the brain. The challenge of stroke management in Indonesia is that not all hospitals have integrated diagnostic devices. Expert systems can be used as learning aids for medical students in learning to diagnose stroke. Medical records of stroke cases can be reused as a reference for diagnosing stroke when there are new cases, which is known as the case-based reasoning (CBR) method.

This study implements the minkowski distance similarity method in CBR to calculate the similarity value between cases, where each similar case has the same solution. This study aims to obtain the most optimal value of r and the most appropriate threshold value in the expert system for the diagnosis of stroke using the minkowski distance similarity method in CBR.

The diagnosis process is carried out by inputting the patient's condition, symptoms, and risk factors, and then the system will calculate the similarity value and take the case with the highest similarity value as the solution with the provision that the similarity value must be greater than or equal to the threshold value. Based on system testing, the best accuracy value was achieved at the application of a threshold value of 75 with an r value of 3 or 4, which has an accuracy rate of 88.89%, a recall value of 88%, and a precision of 100%.

Keywords: expert system, stroke, diagnosis, case-based reasoning, minkowski distance similarity, threshold, accuracy.



BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Stroke adalah penyakit pembuluh darah otak atau serebrovaskuler yang terjadi karena mati atau rusaknya jaringan otak. Kerusakan atau kematian pada jaringan otak ini disebabkan karena tersumbatnya aliran darah dan oksigen yang menuju ke otak. Menurut P2PTM Kemenkes RI (2018), *stroke* dapat terjadi apabila pembuluh darah otak pecah atau tersumbat. Pembuluh darah otak yang mengalami penyumbatan, penyempitan, atau perdarahan karena pecahnya pembuluh darah menyebabkan aliran darah ke otak berkurang (Indrawati & Catur, 2016).

Saat ini, teknologi telah berkembang pesat sebagai solusi untuk meningkatkan efisiensi dan produktivitas diberbagai bidang, salah satu teknologi yang terus berkembang adalah sistem pakar. Sistem pakar merupakan bagian dari kecerdasan buatan (AI) yang berfokus pada pembangunan sistem cerdas dengan cara meniru atau menyamai kemampuan seorang pakar. Dalam bidang kesehatan, sistem pakar dapat digunakan sebagai alat untuk mendiagnosis penyakit seperti penyakit *stroke*, sedangkan dalam bidang akademik, sistem pakar dapat digunakan sebagai alat bantu dalam proses pembelajaran dalam mendiagnosis penyakit *stroke*. Rekam medis kasus *stroke* dapat digunakan kembali sebagai acuan untuk mendiagnosis penyakit *stroke* yang diderita ketika ada kasus baru, disebut sebagai metode *case-based reasoning* (CBR).

Pada penelitian ini, digunakan metode *minkowski distance similarity* yang merupakan bentuk generalisasi dari metode *euclidean distance* dan *Manhattan distance*, dimana hanya terdapat perbedaan pada nilai r yang merupakan konstanta pangkat minkowski. Penentuan nilai r dan nilai *threshold* pada sistem sangat mempengaruhi akurasi dari sistem yang dibangun.

Berdasarkan uraian tersebut, penelitian ini bermaksud untuk mendapatkan nilai r yang paling optimal dan nilai *threshold* yang paling tepat dalam sistem pakar

diagnosis penyakit *stroke* menggunakan metode *minkowski distance similarity* pada CBR.

1.2. Perumusan Masalah

Berdasarkan latar belakang diatas, maka rumusan masalah dalam penelitian ini adalah:

1. Berapakah nilai *threshold* yang tepat untuk sistem diagnosis penyakit *stroke* menggunakan metode *minkowski distance similarity* pada CBR?
2. Berapakah nilai *r* yang paling optimal dalam metode *minkowski distance similarity* pada CBR untuk diagnosis penyakit *stroke*?

1.3. Batasan Masalah

Adapun batasan masalah yang dikaji dalam penelitian ini adalah sebagai berikut:

1. Data yang digunakan adalah data dari penelitian Nelson et al. pada tahun 2018 yang berjudul *Case-Based Reasoning for Stroke Diseases Diagnosis* yang berupa data rekam medis dari pasien penderita *stroke* selama tahun 2015-2016 di Rumah Sakit DKT Dr. Soetarto, Yogyakarta.
2. Pada penelitian ini, *stroke* diklasifikasikan menjadi empat jenis berdasarkan penyebab dan patologi anatominya, yaitu *embolic stroke*, *thrombotic stroke*, *Subarachnoid hemorrhage stroke*, dan *Intraceberal hemorrhage stroke*.
3. Sistem pakar diagnosis penyakit *stroke* ini dirancang untuk mendukung proses pembelajaran terkait penyakit *stroke*.

1.4. Tujuan Penelitian

Tujuan yang hendak dicapai dari penelitian ini adalah untuk mendapatkan nilai *r* dan nilai *threshold* yang tepat untuk sistem diagnosis penyakit *stroke* menggunakan metode *minkowski distance similarity* pada CBR.

1.5. Manfaat Penelitian

Penelitian ini diharapkan dapat membantu dan mempermudah proses pembelajaran terkait diagnosis penyakit stroke berdasarkan gejala, kondisi, dan faktor risiko pasien.

1.6. Metodologi Penelitian

1. Studi Pustaka

Pada tahap ini penulis mencari dan membaca referensi dari jurnal ilmiah, *website*, buku dan sumber lainnya yang dapat memberikan informasi yang relevan dengan masalah yang diteliti.

2. Pengumpulan Data

Penelitian ini menggunakan data sekunder yang diperoleh dari penelitian Nelson dkk. pada tahun 2018 yang berjudul *Case-Based Reasoning for Stroke Diseases Diagnosis* yang berupa data rekam medis dari pasien penderita *stroke* selama tahun 2015-2016 di Rumah Sakit DKT Dr. Soetarto, Yogyakarta.

3. Perancangan Sistem

Melakukan perancangan alur sistem berupa *flowchart* yang menggambarkan alur kerja sistem CBR dan perhitungan similaritas antarkasus, perancangan proses yang ada dalam sistem menggunakan *use case*, dan perancangan basis data berdasarkan data rekam medik penyakit *stroke* dan perancangan antarmuka pengguna.

4. Implementasi Sistem

Proses pembangunan sistem menjadi perangkat lunak berbasis web menggunakan HTML/CSS dan PHP dan memasukan basis pengetahuan ke dalam sistem.

5. Pengujian Sistem

Pengujian terhadap sistem yang telah dibangun menggunakan metode *confusion matrix* untuk mengetahui tingkat akurasi dari sistem yang telah dibangun.

1.7. Sistematika Penulisan

Penulisan laporan skripsi ini terdiri dari empat bab, yaitu sebagai berikut:

BAB I. PENDAHULUAN

Bab pendahuluan berisi tentang latar belakang masalah, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan pernyataan keaslian skripsi.

BAB II. TINJAUAN PUSTAKA DAN LANDASAN TEORI

Bab ini berisi tinjauan pustaka tentang penelitian-penelitian terkait, dan teori-teori yang digunakan dalam penelitian ini, yaitu tentang penyakit *stroke*, sistem pakar, metode CBR, metode *blackbox testing*, metode *confusion matrix*, dan metode *minkowski distance* (pengukuran similaritas global).

BAB III. METODOLOGI PENELITIAN

Bab ini berisi metode-metode yang digunakan penulis dalam membangun sistem, yaitu analisis kebutuhan sistem, perancangan penelitian, diagram alir, *use case diagram*, perancangan basis data, perancangan antarmuka pengguna, dan perancangan pengujian sistem.

BAB IV. IMPLEMENTASI DAN PEMBAHASAN

Bab ini berisi implementasi rancangan sistem menjadi sebuah aplikasi web dan pembahasan mengenai hasil implementasi yang telah dilakukan.

BAB V. KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dari penelitian yang telah dilakukan dan juga diberikan saran-saran yang dapat dipertimbangkan sebagai bahan pemikiran bagi pembaca yang ingin mengembangkannya.

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Berdasarkan referensi, penelitian mengenai CBR dengan berbagai metode sudah banyak dilakukan, salah satunya yaitu penelitian tentang *Case Based Reasoning for Stroke Diseases Diagnosis* yang dilakukan oleh Nelson et al., (2018) dengan menggunakan metode *Jaccard coefficient* dengan menerapkan *siriraj* skor sebagai pembeda antara stroke jenis *ischemic* dan *hemorrhagi*, kemudian diterapkan *indexing* menggunakan *dense indexing*. Setelah itu, dilakukan pengujian sistem menggunakan 45 kasus sebagai data uji dan 135 kasus sebagai basis kasus yang menunjukkan nilai *threshold* 0.7 menghasilkan sensitivitas dan akurasi yang lebih baik daripada nilai *threshold* (0,8), (0,9), dan 1, dengan tingkat sensitivitas sebesar 86,95% dan akurasi (84,44% tanpa *indexing*, 81,67% dengan *indexing*). Data yang digunakan Nelson et al. (2018) ini merupakan data yang sama yang akan digunakan oleh penulis dalam menyelesaikan penelitiannya yaitu data rekam medis dari pasien penderita stroke selama tahun 2015-2016 di Rumah Sakit DKT Dr. Soetarto, Yogyakarta.

Zainuddin et al. (2016) pernah melakukan penelitian tentang Penerapan *Case Based Reasoning* (CBR) untuk Mendiagnosis Penyakit Stroke Menggunakan Algoritma *K-Nearest Neighbor* dengan nilai *threshold* sebesar 80%, yang menyatakan bahwa dari 15 kasus yang diuji, sistem mampu mendiagnosis dengan akurat sesuai dengan hasil diagnosis dari pakar sebesar 93,3%. Adawiyah (2020) juga pernah melakukan penelitian tentang Implementasi Metode *Minkowski Distance* untuk Deteksi Kelahiran Bayi Prematur Berbasis *Case Based Reasoning*. Pengujian akurasi sistem dilakukan dengan menggunakan 20 data uji, 7 data dengan diagnosis normal dan 13 data dengan diagnosis prematur. Dari pengujian didapatkan 2 data yang hasilnya tidak sesuai karena nilainya berada dibawah *threshold* yaitu $\leq 60\%$. Hasil dari perhitungan akurasi sistem menunjukkan tingkat akurasi sistem sebesar 90% dalam mendeteksi kelahiran bayi prematur.

Berdasarkan uraian dari beberapa penelitian diatas, menunjukkan bahwa metode CBR telah banyak diterapkan untuk mendiagnosis penyakit dengan menggunakan berbagai metode similaritas yang menghasilkan tingkat akurasi yang berbeda-beda. Namun berdasarkan studi literatur yang telah dilakukan, penelitian tentang *case-based reasoning* dalam mendiagnosis penyakit stroke menggunakan metode *minkowski distance similarity* belum pernah dilakukan. Untuk itu, penelitian ini akan menerapkan metode *minkowski distance similarity* pada CBR untuk diagnosis penyakit stroke.

Ringkasan dari penelitian-penelitian yang telah dijelaskan disajikan pada Tabel 2.1.

Tabel 2.1: Rangkuman Tinjauan Pustaka

No	Penulis	Tahun	Judul/ Domain Masalah Penelitian	Metode	Hasil
1	Zainuddin, et al.	2016	Penerapan <i>Case Based Reasoning</i> (Cbr) untuk Mendiagnosis Penyakit Stroke Menggunakan Algoritma <i>K-Nearest Neighbor</i>	<i>K-Nearest Neighbor</i>	Sistem mampu mendeteksi dengan tepat sebesar 93,3% dengan menggunakan nilai similaritas 80%.
2	Warman	2017	Sistem Pakar Identifikasi Penyakit Tanaman Padi Menggunakan <i>Case-Based Reasoning</i>	<i>K-Nearest Neighbor</i>	Pengujian sensitivitas dan akurasi sistem dilakukan dengan menggunakan 52 data uji dengan nilai <i>threshold</i> 70% yang menghasilkan sensitivitas sistem sebesar 100% dan

					tingkat akurasi 82,69%.
3	Nelson, et al.	2018	<i>Case Based Reasoning for Stroke Diseases Diagnosis</i>	<i>Jaccard coefficient</i>	Tingkat sensitivitas sistem 86,95% dan akurasi (84,44% tanpa menggunakan metode <i>indexing</i> , 81,67% menggunakan metode <i>indexing</i>) dengan nilai <i>threshold</i> 0,7.
4	Salmin & Hartati	2018	<i>Case Based Reasoning</i> untuk Diagnosis Penyakit Infeksi Saluran Pernapasan Akut	<i>Minkowsky Distance Similarity</i>	Metode <i>K-fold cross validation</i> digunakan dalam pengujian dimana $k=5$ (5-fold) dengan 140 kasus penyakit ISPA dimana kasus dibagi ke dalam 5 subset. Hasil akurasi sistem terbaik sebesar 99,29% dengan menggunakan nilai <i>threshold</i> $\geq 60\%$.
5	Luh Putu Merawati et al.	2018	Sistem Rekomendasi Topik Skripsi menggunakan Metode <i>Case Based Reasoning</i>	<i>Manhattan distance</i>	Pengujian dilakukan dengan 280 data kasus menggunakan <i>K-fold cross validation</i> dengan variasi nilai K yaitu 7, 10 dan 13 dimana akurasi terbaik 94,34% (K=13) diikuti 93,99% (K=10) dan 93,95% (K7).

6	Adawiyah	2020	Implementasi Metode <i>Minkowsky Distance</i> untuk Deteksi Kelahiran Bayi Prematur Berbasis <i>Case Based Reasoning</i>	<i>Minkowsky Distance Similarity</i>	Sistem dapat mendeteksi kelahiran bayi <i>premature</i> dengan tingkat akurasi 90%.
7	Arifa & Fitriana	2020	Diagnosis <i>Case-Based Reasoning</i> Penyakit Hipertensi Menggunakan Metode <i>Minkowsky Distance Similarity</i>	<i>Minkowsky Distance Similarity</i>	Pengujian akurasi sistem menggunakan 172 kasus dengan <i>10-fold Cross Validation</i> yang menunjukkan tingkat akurasi dari sistem sebesar 94,71%.
8	Gunawan et al.	2021	Implementasi Metode <i>Case-Based Reasoning</i> dan <i>Similarity Jaccard Coefficient</i> dalam Identifikasi Kerusakan Laptop	<i>Jaccard Coefficient</i>	Pengujian dilakukan 5 kali dimana 25 pengguna akan mengidentifikasi kerusakan laptop dalam sekali uji. Pengujian menunjukkan akurasi sistem sebesar 90.4% dengan menggunakan nilai <i>threshold</i> yaitu 75%.
9	Via et al.	2021	Penerapan Algoritma <i>Case Based Reasoning</i> dan	<i>K-Nearest Neighbour</i>	Pengujian dilakukan menggunakan 15 data uji, diperoleh 12 data yang didiagnosa benar

			<i>K-Nearest Neighbor</i> untuk Diagnosa Penyakit Ayam		dan 3 data didiagnosa tidak tepat, sehingga diperoleh akurasi sistem sebesar 80%
10	Mubarak et al.	2022	Penalaran Berbasis Kasus untuk Diagnosis Penyakit Malaria dengan Menggunakan Metode <i>Minkowsky Distance</i>	<i>Minkowsky Distance Similarity</i>	Pengujian akurasi sistem dilakukan dengan 25 data uji dan 58 data <i>training</i> yang menunjukkan nilai akurasi sistem sebesar 92% dengan <i>threshold</i> 80%.

2.2 Landasan Teori

2.2.1 Sistem Pakar (*Expert System*)

Sistem pakar adalah sistem cerdas (*expert system*) yang merupakan salah satu cabang dari cabang *Artificial Intelligence* (AI) dengan berbasis pengetahuan. Menurut (Rizky et al., 2020) sistem pakar adalah sistem yang bekerja dengan cara menerapkan pengetahuan dari seorang pakar ke dalam komputer, dengan demikian komputer tersebut bisa menemukan solusi yang tepat untuk menyelesaikan masalah seperti yang dilakukan oleh seorang pakar.

Komponen yang dibutuhkan dalam membangun sistem pakar yaitu:

1) Basis pengetahuan (*knowledge base*)

knowledge base dibutuhkan untuk menyelesaikan masalah. Sistem pakar tersusun dari fakta dan aturan (*rule*), di mana fakta merupakan keadaan atau informasi mengenai objek dan *rule* merupakan aturan-aturan untuk mendapatkan fakta yang baru dari fakta yang sudah diketahui (Pratiwi, 2019).

2) Mesin inferensi

Mesin inferensi merupakan inti dalam sistem pakar yang berfungsi mencari keterkaitan antara *rule* pada basis pengetahuan dengan fakta-fakta yang diinputkan pengguna.

3) Memory kerja (*working memory*)

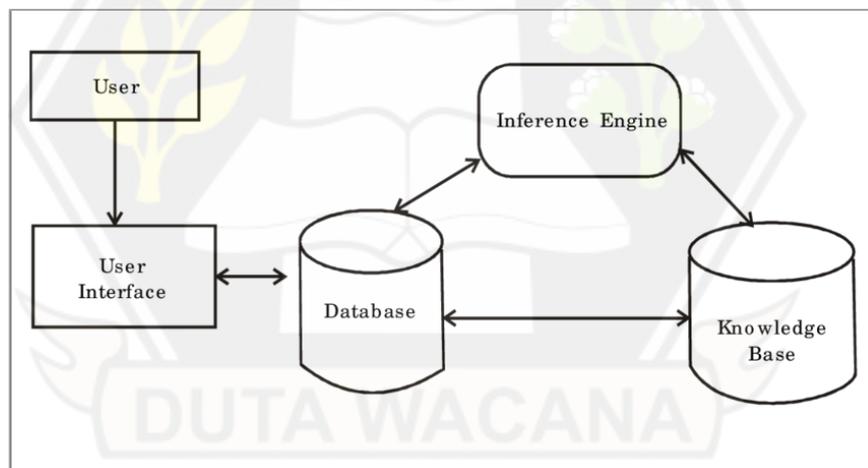
Memori kerja berisi data yang diterima dari inputan pengguna selama sesi sistem pakar, kemudian sistem akan mencocokkan informasi yang diinputkan dengan pengetahuan yang berada di basis pengetahuan untuk mendapatkan solusi atau fakta baru.

4) Basis data

Basis data berisi semua fakta-fakta, hasil observasi dan data-data yang diperlukan untuk pengolahan.

5) Tampilan pengguna (*user interface*)

Tampilan pengguna adalah bagian dari sistem pakar yang menyediakan fasilitas untuk interaksi antara pengguna dengan sistem.



Gambar 2.1: Struktur Sistem Pakar (Mankad, 2015)

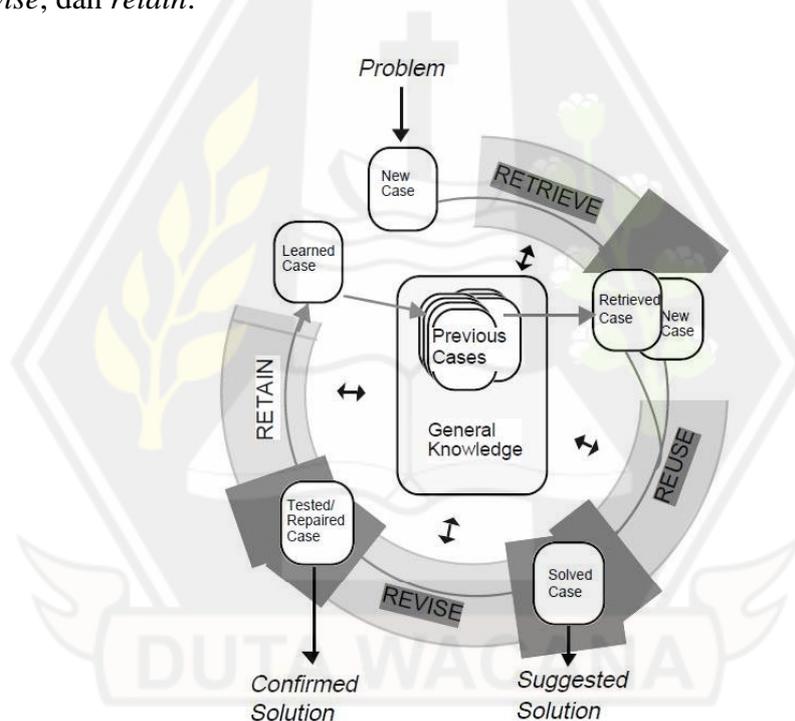
2.2.2 Metode Case Based-Reasoning (CBR)

Case-based reasoning (CBR) merupakan metode untuk memecahkan sebuah permasalahan dengan cara mengingat dan menggunakan kembali pengetahuan dan pengalaman sebelumnya (Xiao et al., 2017). Dalam proses

CBR, jika masalah telah berhasil di pecahkan, maka masalah tersebut akan di simpan untuk memecahkan masalah baru. Namun, jika masalah tersebut gagal di pecahkan, maka masalah tersebut akan diidentifikasi dan di simpan sehingga kesalahan tersebut tidak terjadi lagi dimasa yang akan datang (Subbotin & Voskoglou, 2012).

Banyak bidang yang menerapkan CBR untuk memecahkan kasus. Dalam bidang geografis, Dou, et al. melakukan penelitian untuk mendeteksi longsor menggunakan CBR (Dou et al., 2015), kemudian Tempola, et al. melakukan penelitian tentang penerapan CBR untuk menentukan kelayakan mahasiswa penerima beasiswa (Tempola et al., 2015).

Dalam CBR terdapat 4 tahapan penyelesaian masalah yaitu *retrieve*, *reuse*, *revise*, dan *retain*.



Gambar 2.2: CBR Cycle (Mendes & Ramos, 2012)

1) *Retrieve*

Pada proses *retrieve* dilakukan identifikasi masalah dan pencocokan kasus. Pencocokan kasus dilakukan dengan menghitung nilai similaritas

antara kasus baru dan basis kasus, kemudian diperoleh kasus lama memiliki nilai similaritas tertinggi terhadap kasus baru. Bagian terpenting dalam tahap ini yaitu menentukan tingkat kemiripan atau *similarity* antar kasus (Gunawan et al., 2021).

2) *Reuse*

Pada proses *reuse* terjadi proses penggunaan kembali informasi dan pengetahuan dari kasus lama sebagai solusi untuk kasus baru. Kasus lama yang memiliki nilai similaritas diatas nilai *threshold* dan yang nilainya paling tinggi yang digunakan kembali sebagai solusi untuk memecahkan kasus baru.

3) *Revise*

Proses *revise* merupakan proses perbaikan terhadap solusi yang diberikan yang dilakukan oleh pakar. Jika berhasil, maka kasus baru akan disimpan dalam basis data bersama-sama dengan solusi baru.

4) *Retain*

Retain merupakan proses menyimpan kasus baru yang berhasil diselesaikan dan telah menemukan solusi ke dalam *database*, sehingga bisa digunakan kembali sebagai solusi pada kasus baru dimasa depan.

2.2.3 Metode Confusion Matrix

Confusion matrix merupakan tabel hasil ringkasan dari jumlah data uji benar dan salah (Normawati & Prayogi, 2021). Setiap baris pada *confusion matrix* mewakili kelas yang sebenarnya, sedangkan setiap kolom mewakili kelas yang diprediksi. Matrix akan membandingkan antara nilai sebenarnya dengan hasil prediksi dengan menghitung nilai *accuracy*, *prediction*, dan *recall*.

Tabel 2.2: Confusion Matrix

Nilai Prediksi	Nilai Aktual	
	Positive	Negatif
Positive	True Positive (TP)	False Negative (FP)

	Negative	False Negative (FN)	True Negative (TN)
--	----------	------------------------	-----------------------

1) Accuracy

Accuracy menunjukkan tingkat akurasi model dalam melakukan klasifikasi dengan benar. Ditunjukkan dengan persamaan 2.1.

$$accuracy = \frac{TP + TN}{Total} \quad (2.1)$$

2) Precision

Precision menunjukkan keakuratan antara data yang diminta dengan hasil prediksi dari model (Anggreany, n.d.). Ditunjukkan dengan persamaan 2.2.

$$precision = \frac{TP}{TP + FP} \quad (2.2)$$

3) Recall

Recall menunjukkan keberhasilan sistem dalam menemukan kembali sebuah informasi. Ditunjukkan dengan persamaan 2.3.

$$recall = \frac{TP}{TP + FN} \quad (2.3)$$

Keterangan:

TP (*True positive*) = Prediksi positif dan benar

TN (*True Negative*) = Prediksi negatif dan benar

FP (*False positive*) = Prediksi positif dan salah

FN (*False negative*) = Prediksi negatif dan salah

2.2.4 Penyakit Stroke

Penyakit stroke termasuk ke dalam penyakit tidak menular (PTM) dengan risiko kematian paling tinggi yang terjadi karena suplai darah ke otak tersumbat. Jika darah yang mengalir ke otak tersumbat, maka sel-sel otak akan rusak dan

mati, sehingga menyebabkan kerusakan otak. Stroke jarang terjadi pada orang di bawah 40 tahun, namun stroke bisa saja terjadi pada sekitar 8% anak dengan penyakit *sickle cell* (World Health Organization, 2022).

2.2.5 Klasifikasi Penyakit Stroke

Penyakit stroke dapat dibedakan menjadi dua kelompok berdasarkan penyebabnya, yaitu *stroke* hemoragik dan *stroke* iskemik.

a) Stroke Iskemik (*Ischemic Stroke*)

Stroke iskemik merupakan *stroke* yang disebabkan karena adanya gumpalan darah yang menghalangi aliran darah ke sel-sel otak. Endapan lemak yang melapisi dinding pembuluh, yang disebut aterosklerosis, adalah penyebab utama *stroke* ini. Ada dua jenis *stroke* iskemik, yaitu:

1) *Thrombotic Stroke*

Stroke thrombosis adalah *stroke* yang terjadi karena bekuan darah (trombus) yang berkembang pada pembuluh darah otak. (American Stroke Association, 2022). Sehingga sel-sel otak akan berhenti berfungsi dan mati karena bekuan darah tersebut menghalangi aliran darah ke otak.

2) *Embolic Stroke*

Stroke embolik adalah *stroke* yang disebabkan oleh *fibrilasi atrium* (detak jantung yang tidak teratur) (American Stroke Association, 2022). Hal tersebut mengakibatkan terbentuknya bekuan darah di jantung, yang dapat terbawa oleh aliran darah sampai ke otak dan menyumbat pembuluh darah di otak.

b) Stroke Hemoragik (*Hemorrhagic Stroke*)

Penyebab terjadinya *stroke* hemoragik yaitu karena lemah dan pecahnya pembuluh darah yang mengakibatkan pendarahan di sekitar otak. Kemudian darah terkumpul dan menekan jaringan-jaringan disekitar otak. *Stroke* hemoragik dibagi menjadi dua jenis, yaitu:

1) Pendarahan Subarakhnoid (*Subarachnoid hemorrhage Stroke*)

Stroke pendarahan subarakhnoid adalah *stroke* yang terjadi karena pecahnya pembuluh darah yang berada di permukaan otak yang mengakibatkan

pendarahan di ruang antara tengkorak dan otak (American Stroke Association, 2022).

2) Pendarahan Intraseberal (*Intracerebral hemorrhage Stroke*)

Stroke pendarahan intraseberal terjadi karena pecahnya pembuluh darah yang mengakibatkan pendarahan pada jaringan di dalam otak (American Stroke Association, 2022).

2.2.6 Gejala dan Faktor Risiko Penyakit Stroke

Stroke merupakan penyakit yang perlu ditangani secepatnya karena dapat menyebabkan kematian sel-sel otak hanya dalam hitungan menit. Fakto-faktor penyebab *stroke* yaitu faktor kesehatan, gaya hidup, dan herediter.

a) Gejala penyakit *stroke*

Menurut *American Stroke association* (2022), gejala *stroke* pada umumnya yaitu *face dropping* (salah satu sisi dari wajah mati rasa), salah satu lengan menjadi lemah atau mati rasa, kesulitan dalam berbicara, kesulitan dalam berjalan, masalah penglihatan, kebingungan, kelemahan pada salah satu bagian tubuh, dan sakit kepala tanpa tahu penyebabnya.

Pengkodean gejala-gejala dalam mendiagnosis penyakit *stroke* disajikan pada Tabel 2.3.

Tabel 2.3: Pengkodean Gejala Stroke (Nelson R et al., 2018)

No	Kode Gejala	Nama Gejala
1	G1	Gangguan bicara (Afasia Ekspresif)
2	G2	Status Fungsional
3	G3	Keseimbangan
4	G4	Pusing
5	G5	Mual
6	G6	Muntah
7	G7	Lemas
8	G8	Bicara Pelo
9	G9	Memegang Pinggiran Kursi saat duduk?
10	G10	Nyeri/kram saat berjalan (Claudicatio Intermitten)

11	G11	Perlu penerjemah/susah mengerti (Afasia Reseptif)
12	G12	Sulit berbicara (Disatria)
13	G13	Sesak Nafas
14	G14	Penurunan kesadaran
15	G15	Kelemahan/kelumpuhan anggota gerak kanan/kiri (Hemiplegia)
16	G16	Onset saat beraktivitas (tiba-tiba)
17	G17	Tidak dapat mengendalikan rangsang defeksi (BAB)
18	G18	Tidak dapat mengendalikan rangsang berkemih (BAK)
19	G19	Tidak dapat membersihkan diri sendiri (cuci muka, sikat gigi, sisir rambut)
20	G20	Onset saat istirahat
21	G21	Tidak dapat makan sendiri
22	G22	Tidak dapat berubah sikap dari berbaring ke duduk
23	G23	Tidak mampu berjalan (Ataksia)
24	G24	Tidak dapat memakai pakaian sendiri
25	G25	Tidak mampu naik turun tangga
26	G26	Tidak mampu mandi sendiri
27	G27	Wajah bagian mata dan mulut agak turun sebelah/ wajah merot (Hemiparesis)
28	G28	Sering kesemutan (Parestesia)
29	G29	Gangguan penglihatan (Defisit penglihatan)
30	G30	Telinga berdenging
31	G31	Kesulitan dalam menelan makanan (Disfagia)
32	G32	Bingung (disorientasi)
33	G33	Wajah murung
34	G34	Gangguan memori/pikun (deficit kognitif)
35	G35	Susah menggerakkan kaki atau tangan
36	G36	Kejang
37	G37	Nafsu makan berkurang
38	G38	Cemas/gelisah (Defisit emosional)
39	G39	Batuk
40	G40	Nyeri dibagian dada (Angina Pectoris)
41	G41	Kaku pada bibir/mulut/lidah
42	G42	Kaku pada leher

b) Faktor risiko penyakit *stroke*

Faktor risiko adalah variabel-variabel yang mempengaruhi tingkat risiko seseorang terkena *stroke*. Faktor risiko penyakit *stroke* menurut Centers for Disease Control and Prevention Organization (2022) meliputi:

- Riwayat penyakit darah tinggi (hipertensi)

Hipertensi merupakan penyebab utama *stroke*, tingginya tekanan didalam pembuluh darah dalam jangka waktu yang lama, dapat merusak dinding arteri dan membuatnya menjadi lebih lemah, sehingga dapat menyebabkan *stroke*.

- Riwayat penyakit kolesterol

Tingginya kadar kolesterol dalam darah dapat menyebabkan timbunan lemak menumpuk pada arteri. Hal ini membuat arteri menjadi kaku dan menyempit sehingga menyumbat aliran darah yang dapat mengakibatkan terjadinya *stroke*.

- Riwayat penyakit jantung

Kondisi jantung seperti detak jantung yang tidak teratur, cacat pada katup jantung, dan ruang jantung yang membesar bisa menyebabkan pembekuan darah dan menyebabkan *stroke*.

- Riwayat penyakit diabetes

Diabetes dapat meningkatkan risiko terkena *stroke* dikarenakan diabetes menyebabkan gula menumpuk didalam darah sehingga menghalangi nutrisi masuk ke berbagai bagian tubuh, termasuk otak.

- Obesitas

Obesitas adalah kelebihan lemak pada tubuh yang dapat mengakibatkan hipertensi dan diabetes, dimana hal ini dapat mengakibatkan seseorang terserang *stroke*.

- Merokok

Salah satu kandungan pada rokok adalah tembakau dimana tembakau dapat meningkatkan risiko *stroke*. Zat lainnya yaitu nikotin yang dapat meningkatkan tekanan darah sehingga meningkatkan risiko terkena *stroke*.

- Alkohol

Alkohol merupakan salah satu faktor risiko *stroke* karena dapat meningkatkan tekanan darah dan trigliserida yang dapat meningkatkan risiko terserang *stroke*.

- Hereditas

Hereditas adalah warisan genetik yang diturunkan oleh orang tua biologis melalui gen. Faktor genetik berhubungan dengan tekanan darah tinggi, *stroke*, dan kondisi terkait lainnya. Beberapa kelainan genetik dapat menyebabkan *stroke*, termasuk penyakit sel sabit.

Dalam penelitian ini faktor risiko hereditas dan kolesterol diabaikan karena didalam basis kasus kedua faktor risiko tersebut tidak disertakan. Faktor risiko penyakit *stroke* dan pengkodeannya dapat dilihat pada tabel 2.4.

Tabel 2.4: Faktor Risiko Stroke (Nelson R et al., 2018)

No	Kode Faktor Risiko	Nama Faktor Risiko
1	R1	Riwayat penyakit jantung
2	R2	Riwayat penyakit hipertensi
3	R3	Riwayat penyakit diabetes melitus
4	R4	Riwayat penyakit hiperkolesterolemia
5	R5	Obesitas
6	R6	Riwayat penyakit <i>stroke</i>
7	R7	Riwayat penyakit <i>stroke</i> dalam keluarga
8	R8	Riwayat penyakit hipertensi dalam keluarga
9	R9	Riwayat penyakit diabetes melitus dalam keluarga
10	R10	Ketergantungan obat-obatan antikoagulan
11	R11	Ketergantungan rokok
12	R12	Ketergantungan alkohol

2.2.7 Representasi Kasus

Kasus harus disimpan ke basis kasus agar bisa digunakan lagi dalam menyelesaikan kasus baru. Cara menyimpan kasus didalam basis kasus sangat mempengaruhi kinerja sistem dalam melakukan pencarian dan proses *retrieval* dalam memecahkan kasus baru. Oleh karena itu, sebuah kasus perlu di representasikan ke dalam bentuk tertentu. Beberapa model representasi kasus

yaitu jaringan semantik, *frame*, *rules*, dan *script*. Data sekunder yang berupa catatan rekam medik pasien penderita *stroke* di Rumah Sakit DKT Dr. Soetarto, Yogyakarta yang penulis dapatkan dari penelitian Nelson, et al. (2018) akan digunakan sebagai basis kasus. Data yang telah diperoleh oleh Nelson, et al. (2018) telah dikonsultasikan dengan dokter spesialis syaraf yang merupakan seorang pakar dalam hal ini, kemudian dipilih data yang dapat digunakan sebagai fitur kasus.

Fitur-fitur yang digunakan dalam penelitian ini yaitu:

1) Usia

Risiko terkena *stroke* meningkat seiring dengan bertambahnya usia seseorang. Hal ini berkaitan dengan proses alamiah yang disebut degenerasi (penuaan) yang terjadi pada lanjut usia dimana terdapat plak yang membuat pembuluh darah menjadi lebih kaku (Noviyanti, 2013).

2) Jenis Kelamin

Berdasarkan penelitian yang dilakukan oleh Noviyanti (2014), dibandingkan perempuan, laki-laki lebih beresiko terkena *stroke*. Hal ini ada hubungannya dengan laki-laki yang cenderung merokok, karena merokok berdampak buruk bagi lapisan dari pembuluh darah.

3) Pemeriksaan Fisik

Merupakan pemeriksaan terkait dengan tanda-tanda vital pasien seperti tingkat kesadaran, tekanan darah, nadi, pernafasan, dan suhu pasien. Hasil dari pemeriksaan ini sangat membantu untuk mendiagnosis penyakit *stroke*.

4) Faktor Gejala

Faktor gejala *stroke* merupakan gejala-gejala yang dialami pasien seperti nyeri atau kram saat berjalan, sulit berbicara, sesak napas, penurunan kesadaran, dan lain sebagainya yang sangat penting dalam menentukan jenis penyakit *stroke*.

5) Faktor Risiko

Faktor risiko *stroke* merupakan variabel-variabel yang berhubungan dengan tingkat risiko seseorang terkena *stroke* yang berpengaruh terhadap jenis penyakit *stroke*. Faktor risiko *stroke* menurut pakar dalam hal ini adalah dokter

spesialis syaraf, yaitu riwayat penyakit (jantung, hipertensi, *diabetes melitus*, *hiperkolesterolemia*, obesitas, dan *stroke*), riwayat penyakit dalam keluarga (*stroke*, hipertensi, dan diabetes melitus), dan ketergantungan (obat-obatan antikoagulan, rokok, dan alkohol).

Tabel 2.5: Contoh Representasi Kasus (K00007) Menggunakan Model Frame (Nelson R et al., 2018)

Basis Kasus		
Kode pasien:		K00007
Kondisi umum:		
1	Usia	60
2	Jenis kelamin	Laki-laki
3	Kesadaran	Compos Mentis
Gejala:		
G1	Gangguan berbicara	Tidak
G2	Status fungsionalitas	Tidak
G3	Keseimbangan	Tidak
Gn	Gejala ke-n	...
Faktor resiko:		
FR1	Jantung	Tidak
FR2	Hipertensi	Ya
FRn	Faktor risiko ke-n	...
Diagnosis:	Embolism Stroke	

2.2.8 Tingkat Keyakinan

a. Tingkat Keyakinan Pakar

Tingkat keyakinan pakar menunjukkan seberapa besar keyakinan seorang pakar terhadap diagnosis. Tingkat keyakinan pakar ditentukan sendiri oleh seorang pakar, dimana tinggi keyakinan berbanding lurus dengan nilai kepastian suatu kasus.

b. Tingkat Keyakinan Terhadap Kasus Baru

Setiap kasus baru yang sudah mendapatkan solusi yang akan disimpan pada basis kasus harus memiliki nilai keyakinan. Rumus yang digunakan untuk menghitung nilai keyakinan kasus baru menggunakan persamaan 2.4.

$$P_k(S_k, T_k) = P_k(S_k) * \frac{J(S_k, T_k)}{J(T_k)} \quad (2.4)$$

Keterangan:

$P_k(S_k, T_k)$: Tingkat keyakinan antara kasus *target case* (T) dan *source case* (S).

$P_k(S_k)$: Tingkat keyakinan dari kasus ke-k di dalam *source case* (S)

$J(S_k, T_k)$: Total fitur pada *target case* yang muncul pada *source case*

$J(T_k)$: Total fitur pada *target case*.

2.2.9 Pengukuran Similaritas

Pengukuran similaritas bertujuan untuk menghitung tingkat kemiripan antar dua objek.

1. Pengukuran Similaritas Lokal

Similaritas lokal digunakan untuk mendapatkan nilai kemiripan antara atribut permasalahan terhadap atribut yang sama dari sebuah kasus. Similaritas lokal dihitung berdasarkan tipe data pada fiturnya (Wahyudi & Pradasari, 2018). Setiap fitur harus memiliki nilai, jika fitur yang dihitung tidak memiliki nilai, maka fitur tersebut bernilai 0.

1) Tipe Data Numerik

Ditunjukkan dengan persamaan 2.5.

$$f(s, t) = 1 - \frac{|s - t|}{R} \quad (2.5)$$

dimana s,t adalah nilai dari fitur yang dibandingkan sedangkan R adalah range nilai dari fitur tersebut (Jha et al., 2013).

2) Tipe Data Boolean

Fungsi similaritas lokal pada tipe data boolean ditunjukkan pada persamaan 2.6 (Nurdiansyah & Hartati, 2014).

$$f(s, t) = \begin{cases} 1, & \text{jika } s = t \\ 0, & \text{jika } s \neq t \end{cases} \quad \text{dimana } s, t \{ \text{benar, salah} \} \quad (2.6)$$

2. Pengukuran Similaritas Global

Similaritas global digunakan untuk melakukan perhitungan kemiripan antara permasalahan dengan kasus dalam basis kasus, dapat dilihat pada persamaan 2.7.

$$E(C_i, C_j) = \left[\frac{\sum_{k=1}^n w_k^r * |d_k(C_{ik}, C_{jk})|^r}{\sum_{k=1}^n w_k^r} \right]^{1/r} * T(C_j) * \frac{n(C_i, C_j)}{n(C_i)} \quad (2.7)$$

Dimana,

$E(C_i, C_j)$: Similaritas global antara *target case* (C_i) dan *source case* (C_j)

w_k : Nilai bobot atribut k

$d_k(C_{ik}, C_{jk})$: Nilai similaritas lokal antara atribut *target case* ke k dan atribut *source case* ke k

r : Faktor *minkowski* (positif integer)

$T(C_j)$: Tingkat keyakinan kasus dalam basis kasus

$n(C_i, C_j)$: Total atribut dari *target case* (C_i) yang muncul pada *source case* (C_j)

$n(C_i)$: Total jumlah atribut pada *target case* (C_i)

(Faizal & Hamdani, 2018)

Berikut adalah contoh perhitungan similaritas lokal dan similaritas global (*minkowski distance similarity*) menggunakan kasus lama (K00007) dan kasus baru.

Tabel 2.6: Data Kasus Baru

Kasus Baru			
Data pasien		Keterangan	Nilai
1	Usia		62
2	Jenis kelamin	Laki-laki	1
3	Kesadaran	Compos metris	0
5	Kondisi	Baik	0
6	Nyeri kepala	Nyeri	2
6	Tekanan darah sistolik	mmHg	155
7	Tekanan darah diastolik	mmHg	100
8	Suhu badan	°C	36,5
9	Napas	Kali/menit	20
10	Nadi	Kali/menit	88
Gejala:			
G2	Status fungsional		1
G8	Bicara pelo		1
G10	Nyeri/kram saat berjalan (Claudicatio Intermittent)		1
G15	Kelemahan/kelumpuhan anggota gerak kanan/kiri (Hemiplegia)		1
G25	Tidak mampu naik turun tangga		1
G35	Susah menggerakkan kaki atau tangan		1
G38	Cemas/gelisah (Defisit emosional)		1
Faktor resiko:			
R2	Riwayat Hipertensi		1
R11	Merokok		1

1) Perhitungan similaritas lokal

- Kedekatan Usia

$$f(s, t) = 1 - \frac{|60 - 62|}{86 - 30} = 1 - \frac{2}{56} = 1 - 0,036 = 0,964$$

- Jenis kelamin bernilai 1 karena antara kasus baru dan kasus lama (K00007) bernilai sama.

- Fitur kesadaran (range 3)

$$f(s, t) = 1 - \frac{|0 - 0|}{3} = 1 - \frac{0}{3} = 1 - 0 = 1$$

- Fitur kondisi (range 3)

$$f(s, t) = 1 - \frac{|0 - 0|}{3} = 1 - \frac{0}{3} = 1 - 0 = 1$$

- Fitur nyeri kepala (range 1 sd. 10)

$$f(s, t) = 1 - \frac{|0 - 2|}{10} = 1 - \frac{2}{10} = 1 - 0,2 = 0,8$$

- Fitur tekanan darah sistolik

$$f(s, t) = 1 - \frac{|150 - 155|}{230 - 100} = 1 - \frac{5}{130} = 1 - 0,04 = 0,96$$

- Fitur tekanan darah diastolik

$$f(s, t) = 1 - \frac{|100 - 100|}{110 - 60} = 1 - \frac{0}{50} = 1 - 0 = 1$$

- Fitur suhu badan

$$f(s, t) = 1 - \frac{|36,5 - 36,5|}{39,2 - 35} = 1 - \frac{0}{4,2} = 1 - 0 = 1$$

- Fitur napas

$$f(s, t) = 1 - \frac{|20 - 20|}{28 - 16} = 1 - \frac{0}{12} = 1 - 0 = 1$$

- Fitur nadi

$$f(s, t) = 1 - \frac{|88 - 88|}{120 - 73} = 1 - \frac{0}{47} = 1 - 0 = 1$$

- Gejala-gejala yang berada pada kedua kasus yaitu gejala G8, G10, G15, G25, G35 dan G38, sehingga gejala-gejala tersebut bernilai 1, selain itu bernilai 0.

- Faktor risiko

Faktor risiko yang berada pada kedua kasus yaitu R2 dan R11 sehingga faktor resiko tersebut bernilai 1, selain itu bernilai 0.

- 2) Perhitungan similaritas global (*minkowski distance similarity*) dengan $r = 3$ menggunakan similaritas lokal yang telah dihitung sebelumnya.

$$E(C_i, C_j) = \left[\frac{\begin{aligned} &usia(4 * 0,964)^3 + jk(4 * 1)^3 + kesad(4 * 1)^3 \\ &+nyeri_kepala(5 * 0,8)^3 + ts(7 * 0,96)^3 + td(8 * 1)^3 \\ &+kondisi(4 * 1)^3 + napas(4 * 1)^3 \\ &+nadi(4 * 1)^3 + suhu(4 * 1)^3 \\ &+ G2(5 * 0)^3 + G8(9 * 1)^3 + G7(5 * 0) \\ &G10(9 * 1)^3 + G15(9 * 1)^3 + G25(7 * 1)^3 \\ &+ G35(9 * 1)^3 + G38(8 * 1)^3 + FR2(9 * 1)^3 + FR11(7 * 1)^3 \end{aligned}}{\begin{aligned} &4^3 + 4^3 + 4^3 + 5^3 + 7^3 + 8^3 + 4^3 \\ &+ 4^3 + 4^3 + 4^3 + 5^3 + \\ &9^3 + 5^3 + 9^3 + 9^3 + 7^3 \\ &+ 9^3 + 8^3 + 9^3 + 7^3 \end{aligned}} \right]^{1/3} * 90 * \frac{18}{19}$$

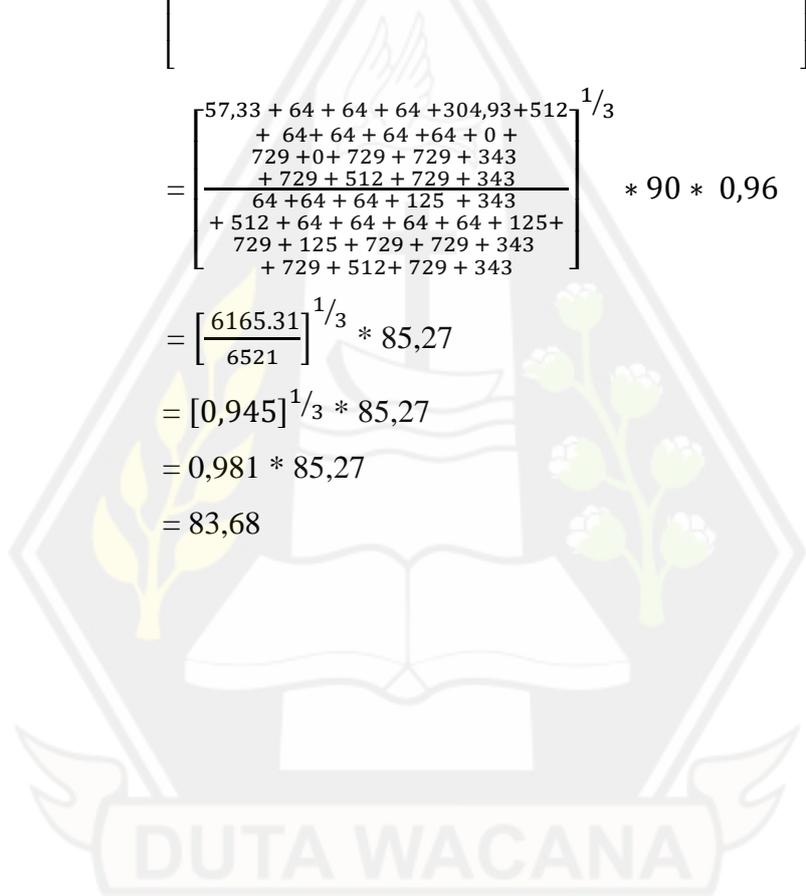
$$= \left[\frac{57,33 + 64 + 64 + 64 + 304,93 + 512}{64 + 64 + 64 + 125 + 343} \right]^{1/3} * 90 * 0,96$$

$$= \left[\frac{6165,31}{6521} \right]^{1/3} * 85,27$$

$$= [0,945]^{1/3} * 85,27$$

$$= 0,981 * 85,27$$

$$= 83,68$$



BAB III

METODOLOGI PENELITIAN

3.1 Analisis Kebutuhan Sistem

1. Analisis Kebutuhan Perangkat Lunak

- a. Visual Studio Code
- b. Xampp

2. Analisis Kebutuhan Perangkat Keras

Berikut adalah spesifikasi dari perangkat keras berupa laptop yang digunakan dalam membangun sistem yaitu:

- a. OS Windows 11 64-bit
- b. Processor: Intel® core™ i7
- c. Memory: 8GB

3.2 Perancangan Penelitian

1. Pengumpulan Data

Data yang digunakan dalam penelitian ini adalah data dari penelitian Nelson et al., pada tahun 2018 yang berjudul *Case-Based Reasoning for Stroke Diseases Diagnosis* yang berupa data rekam medis dari pasien penderita *stroke* selama tahun 2015-2016 di Rumah Sakit DKT Dr. Soetarto, Yogyakarta. Data ini akan digunakan sebagai basis kasus dalam sistem.

2. Perancangan Sistem

Pada tahap ini dilakukan perancangan alur sistem berupa *flowchart* yang menggambarkan alur kerja sistem CBR menggunakan metode *minkowski distance* untuk perhitungan similaritas antarkasus hingga menemukan solusi. Kemudian proses yang ada dalam sistem dirancang menggunakan *use case*. Pada tahap ini juga dilakukan perancangan basis data

berdasarkan data rekam medik penyakit *stroke* dan perancangan antarmuka pengguna.

3. Implementasi Sistem

Pada tahap ini dilakukan pembangunan sistem menjadi perangkat lunak berbasis web menggunakan HTML/CSS dan PHP dengan *apache* sebagai *webserver* dan *mysql* untuk *database*-nya.

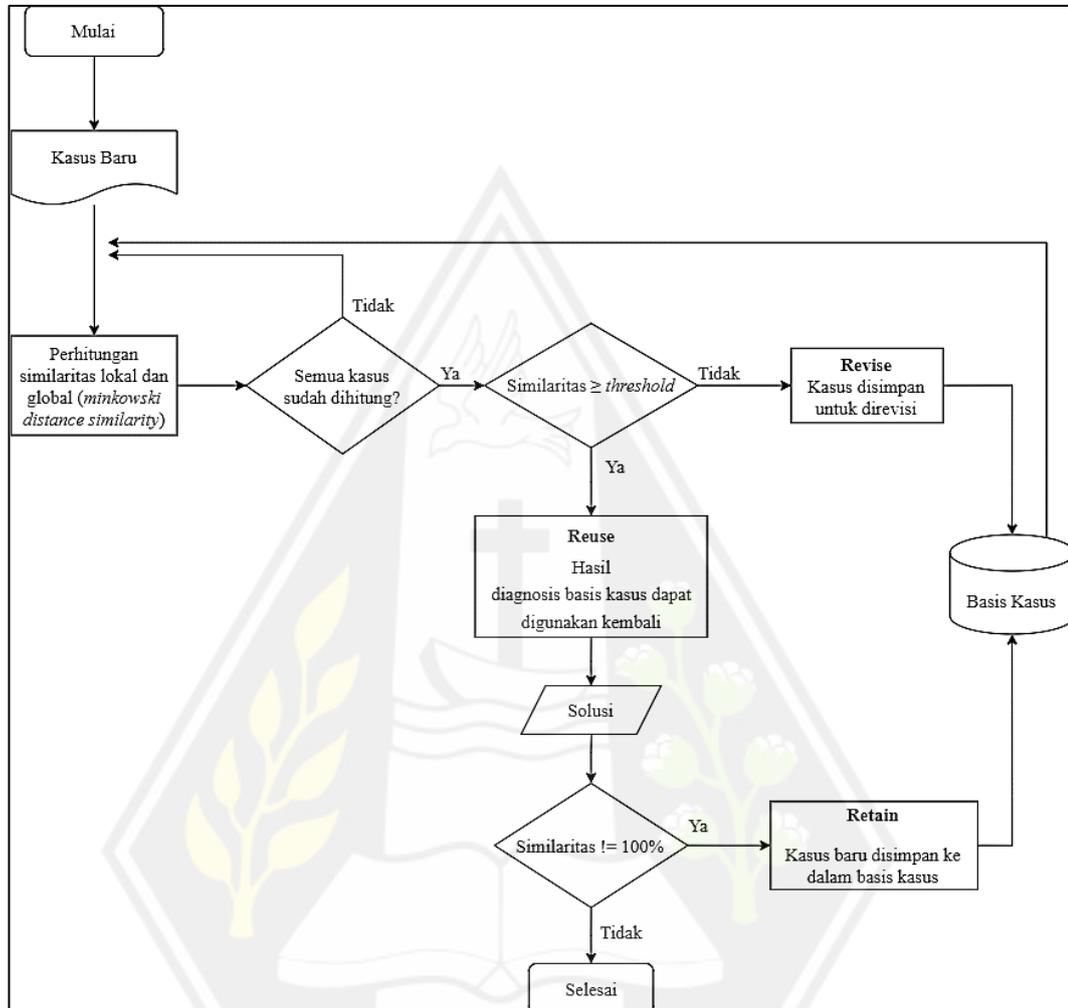
4. Pengujian Sistem

Pada tahap ini dilakukan pengujian terhadap sistem yang sudah dibangun menggunakan metode *confusion matrix* yang akan menghasilkan nilai *accuracy*, *recall* atau *sensitivity*, dan *precision*. Pengujian dilakukan dengan menggunakan data dari penelitian Nelson R et al., (2018) yang berupa data rekam medis dari pasien penderita *stroke* selama tahun 2015-2016 di Rumah Sakit DKT Dr. Soetarto, Yogyakarta yang terdiri dari 180 kasus, dimana 30% dari kasus tersebut yaitu 54 kasus akan dijadikan data uji.

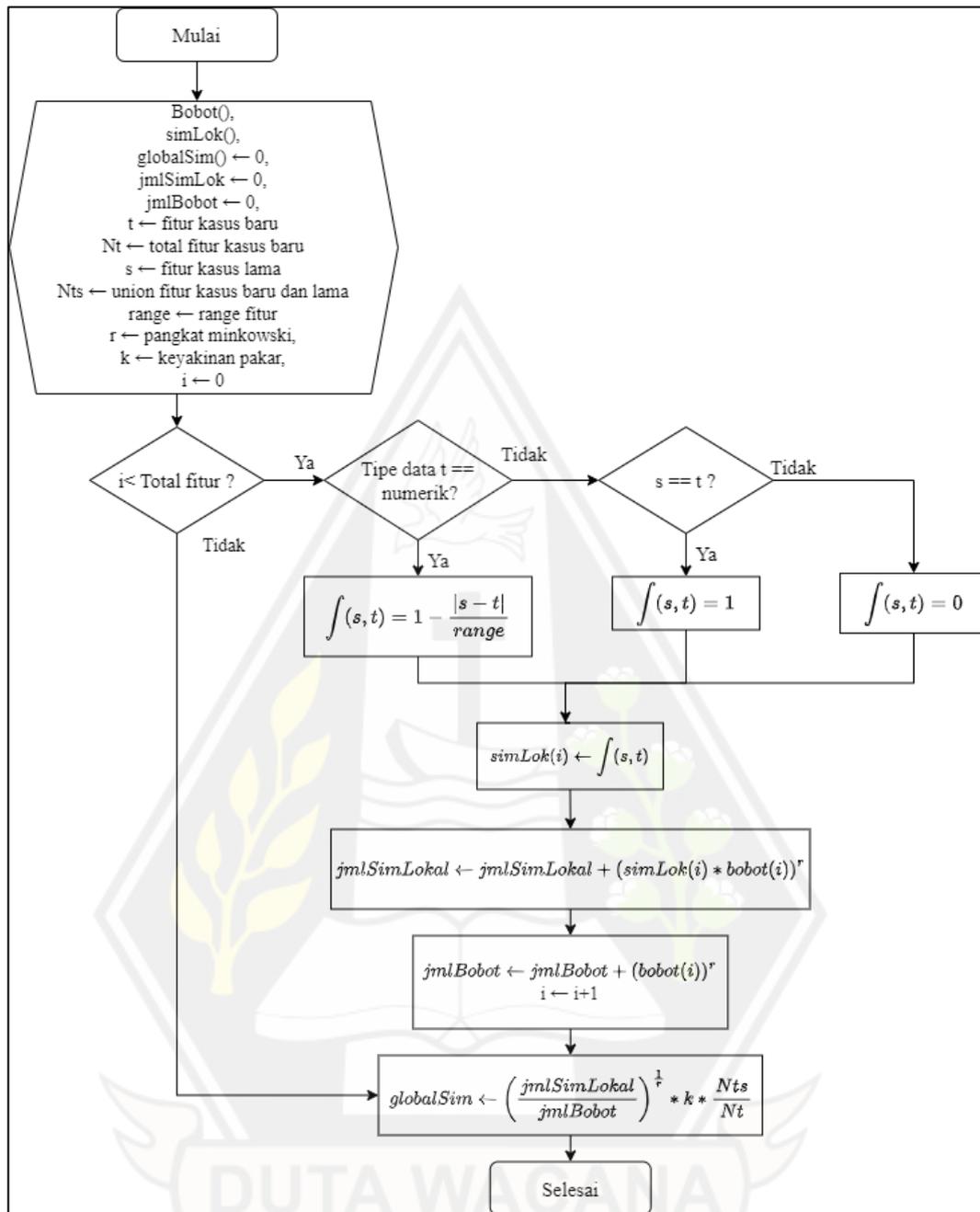
3.3 Diagram Alir

Sistem pakar ini dibangun menggunakan metode CBR dan metode *minkowski distance similarity* untuk melakukan perhitungan kemiripan antara kasus baru dan kasus lama. Sistem akan menerima inputan dari *user* berupa data diri pasien, gejala, dan juga faktor resiko yang dialami. Kemudian sistem akan menghitung nilai similaritas lokal dan similaritas global antara data kasus baru (data yang diinputkan user) dengan kasus-kasus lama yang ada pada basis kasus menggunakan metode *minkowski distance similarity*. Alur perhitungan similaritas global menggunakan metode *minkowski distance similarity* ditunjukkan pada gambar 3.2. Setelah itu, kasus dengan nilai similaritas yang tertinggi akan digunakan sebagai solusi untuk menjawab kasus baru dan nilai similaritasnya harus melebihi *threshold*. Jika nilai similaritasnya tidak melebihi *threshold*, maka kasus akan disimpan ke dalam basis kasus dan akan direvisi oleh pakar. Kemudian, sistem

akan mengeluarkan *output* berupa nama jenis penyakit *stroke* yang diderita oleh pasien. Gambaran alur sistem ditunjukkan pada gambar 3.1.



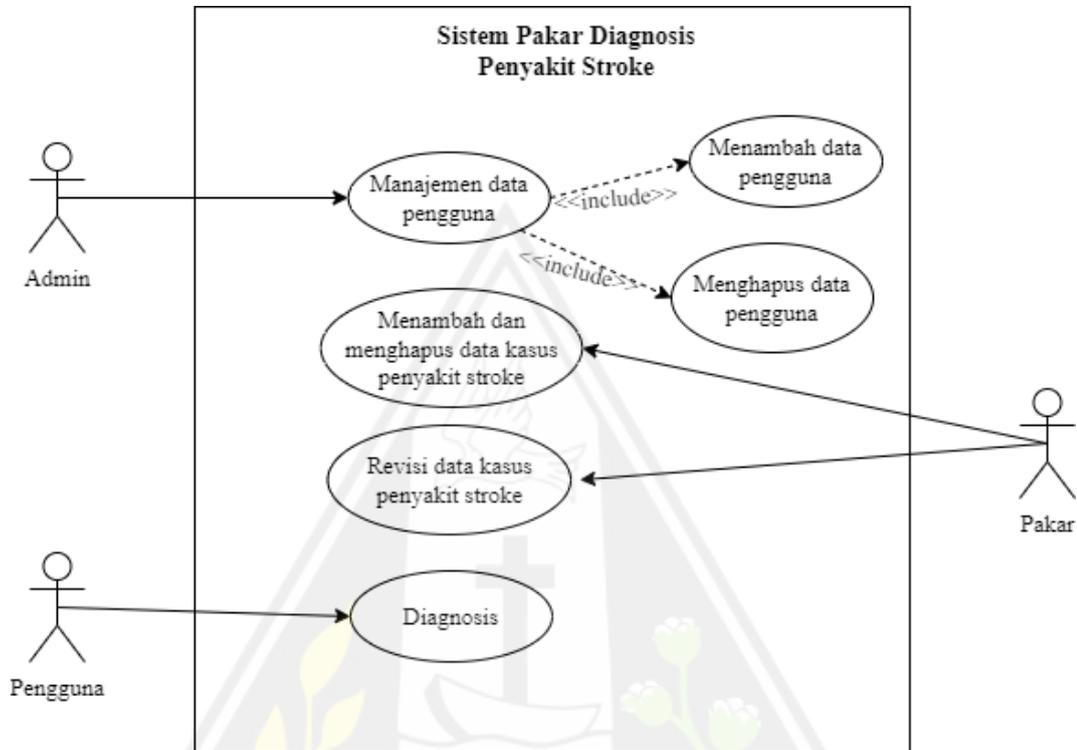
Gambar 3.1: Diagram alir sistem CBR penyakit *stroke*



Gambar 3.2: Diagram alir perhitungan similaritas global

3.4 Use Case Diagram

Usecase diagram merupakan diagram yang digunakan untuk menggambarkan interaksi antar aktor dan juga dengan fitur-fitur sistem. *Usecase diagram* sistem pakar untuk diagnosis penyakit *stroke* dapat dilihat pada gambar 3.3.



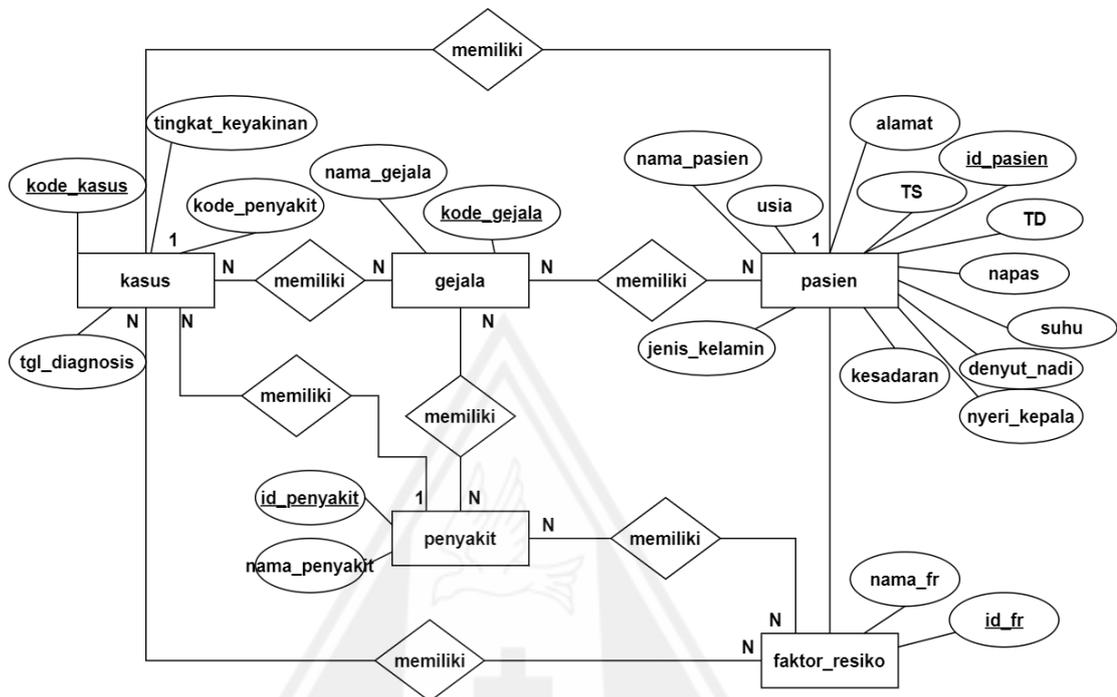
Gambar 3.3: *Usecase diagram* sistem pakar diagnosis penyakit *stroke*

Usecase diagram pada gambar 3.3 menjelaskan apa saja yang dapat dilakukan tiap aktor terkait dengan fitur-fitur sistem. *Admin* dapat menambah dan menghapus data pengguna sistem. *Pakar* dapat melakukan revisi data penyakit *stroke*. Sedangkan pengguna bisa melakukan konsultasi.

3.5 Perancangan Basis Data

i. Entity Relationship Diagram (ERD)

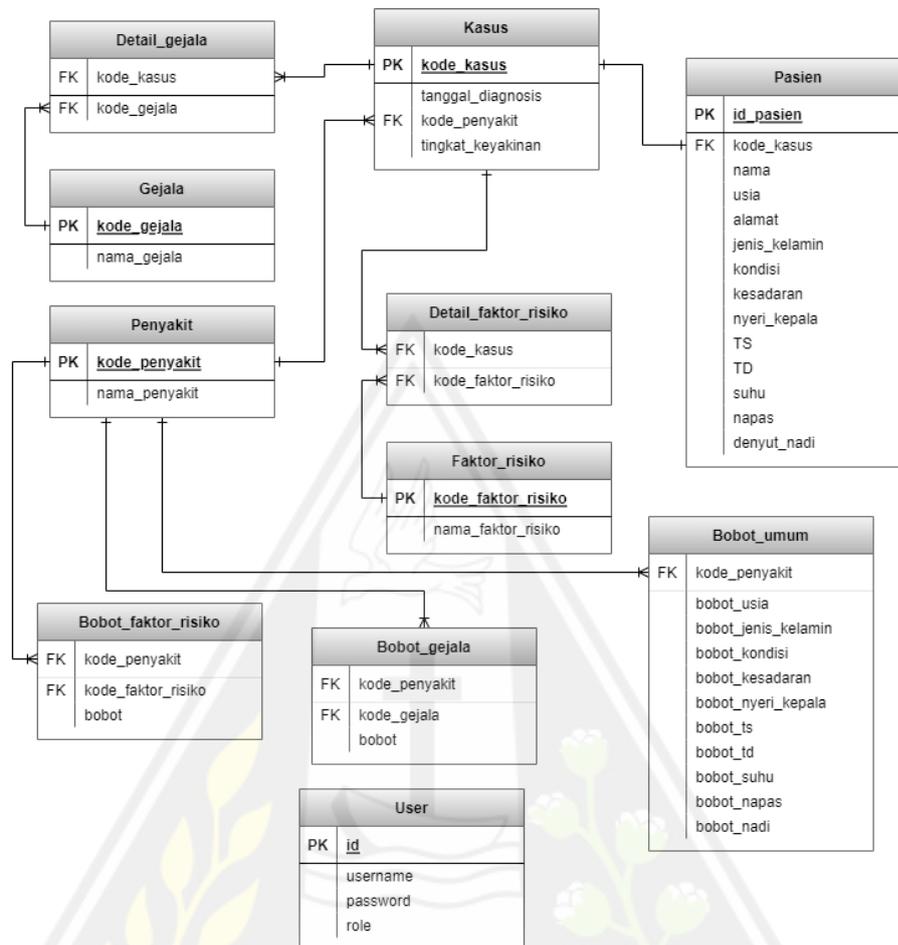
ERD merupakan diagram yang menggambarkan desain *database* dan menunjukkan hubungan antar entitas dan atributnya secara detail. Data berupa kasus-kasus yang digunakan pada penelitian ini akan disimpan ke dalam basis data dengan desain seperti pada gambar 3.4.



Gambar 3.4: ER Diagram Sistem Pakar untuk Mendiagnosa Penyakit Stroke

ii. Relasi antar table

Relasi antar tabel pada basis data untuk diagnosis penyakit *stroke* dapat dilihat pada gambar 3.5.



Gambar 3.5: Rancangan Relasi Basis Data Sistem Pakar Diagnosis Stroke

iii. Struktur tabel

Berikut adalah tabel-tabel dalam database yang digunakan dalam sistem diagnosis penyakit *stroke* beserta keterangannya:

1) Tabel user

Tabel ini menyimpan daftar pengguna yang menggunakan sistem ini. Pengguna terbagi menjadi tiga peran, yaitu paramedis, pakar, dan admin. Pasien adalah pengguna yang akan melakukan konsultasi, pakar bertugas untuk merevisi kasus yang nilai similaritasnya dibawah *threshold*, sedangkan admin bertugas untuk manajemen data pengguna.

Atribut	Tipe Data	Keterangan
id_user	Varchar (5)	Primary key

username	Varchar (25)	Nama pengguna
password	Varchar (25)	Kata sandi untuk login ke sistem
role	Varchar (10)	Hak akses yang terdiri dari admin, pasien, dan pakar

2) Tabel pasien

Tabel ini menyimpan data pasien yang telah didiagnosis yang akan berelasi dengan tabel kasus.

Atribut	Tipe Data	Keterangan
id_pasien	Varchar (5)	<i>Primary key</i>
kode_kasus	Varchar (5)	<i>Foreign_key</i>
nama	Varchar (50)	Nama dari pasien
usia	Int (3)	Kata sandi untuk login ke sistem
alamat	Varchar (200)	Hak akses yang terdiri dari admin, paramedis, dan pakar
jenis_kelamin	Int (1)	Jenis kelamin pasien
kondisi	Int (1)	Kondisi pasien
kesadaran	Int (1)	Kesadaran pasien
nyeri_kepala	Int (2)	Nyeri kepala yang dirasakan pasien
TS	Int (3)	Tekanan darah sistolik pasien
TD	Int (3)	Tekanan darah diastolik pasien
suhu	Double	Suhu pasien

napas	Int (2)	Jumlah napas pasien dalam 1 menit
denyu_nadi	Int (3)	Jumlah denyut nadi pasien dalam 1 menit

3) Tabel kasus

Tabel kasus digunakan untuk menyimpan data kasus lama yang sudah memiliki solusi.

Atribut	Tipe Data	Keterangan
kode_kasus	Varchar (5)	<i>Primary key</i>
tanggal_diagnosis	date	tanggal saat diagnosis
kode_penyakit	Varchar (5)	<i>Foreign key</i>
Tingkat_keyakinan	Float (4)	Tingkat keyakinan terhadap kasus

4) Tabel penyakit

Tabel ini menyimpan data jenis penyakit *stroke* yaitu *embolic stroke*, *thrombotic stroke*, *subarachnoid hemorrhage stroke*, *intracerebral hemorrhage stroke*.

Atribut	Tipe Data	Keterangan
kode_penyakit	Varchar (5)	<i>Primary key</i>
nama_penyakit	Varchar (100)	Nama penyakit <i>stroke</i>

5) Tabel gejala

Tabel gejala digunakan untuk menyimpan data gejala dari penyakit *stroke*.

Atribut	Tipe Data	Keterangan
kode_gejala	Varchar (5)	<i>Primary key</i>
nama_gejala	Varchar (50)	Nama gejala <i>stroke</i>

6) Tabel faktor_risiko

Table ini digunakan untuk menyimpan data faktor-faktor risiko penyakit *stroke*.

Atribut	Tipe Data	Keterangan
kode_faktor_risiko	Varchar (5)	<i>Primary key</i>
nama_faktor_risiko	Varchar (50)	Nama faktor risiko <i>stroke</i>

7) Tabel detail_gejala

Tabel ini merupakan tabel yang digunakan untuk menghubungkan antara tabel kasus dan tabel gejala.

Atribut	Tipe Data	Keterangan
kode_kasus	Varchar (5)	<i>Foreign key</i> (tabel kasus)
kode_gejala	Varchar (5)	<i>Foreign key</i> (tabel gejala)

8) Tabel detail_faktor_risiko

Tabel ini merupakan tabel yang digunakan untuk menghubungkan antara tabel kasus dan tabel faktor risiko. Tabel ini terdiri dari id detail kasus dan id faktor risiko.

Atribut	Tipe Data	Keterangan
kode_kasus	Varchar (5)	<i>Foreign key</i> (tabel kasus)
kode_faktor_risiko	Varchar (5)	<i>Foreign key</i> (tabel faktor risiko)

9) Tabel bobot_gejala

Tabel ini terbentuk dari relasi antara tabel penyakit dan tabel gejala, yang menyimpan bobot dari gejala berdasarkan penyakit yang diderita.

Atribut	Tipe Data	Keterangan
kode_penyakit	Varchar (5)	<i>Foreign key</i> (tabel penyakit)
kode_gejala	Varchar (5)	<i>Foreign key</i> (tabel gejala)
bobot	Int (1)	Nilai bobot gejala terhadap penyakit

10) Tabel bobot_faktor_risiko

Tabel ini terbentuk dari relasi antara tabel penyakit dan tabel faktor_risiko, yang menyimpan bobot dari faktor risiko berdasarkan penyakit yang diderita.

Atribut	Tipe Data	Keterangan
kode_penyakit	Varchar (5)	<i>Foreign key</i> (tabel penyakit)
kode_faktor_risiko	Varchar (50)	<i>Foreign key</i> (tabel faktor_risiko)
bobot	Int (1)	Nilai bobot faktor risiko terhadap penyakit

11) Tabel bobot_umum

Tabel ini terbentuk dari relasi antara tabel penyakit dan tabel pasien, yang menyimpan bobot dari faktor risiko berdasarkan penyakit yang diderita.

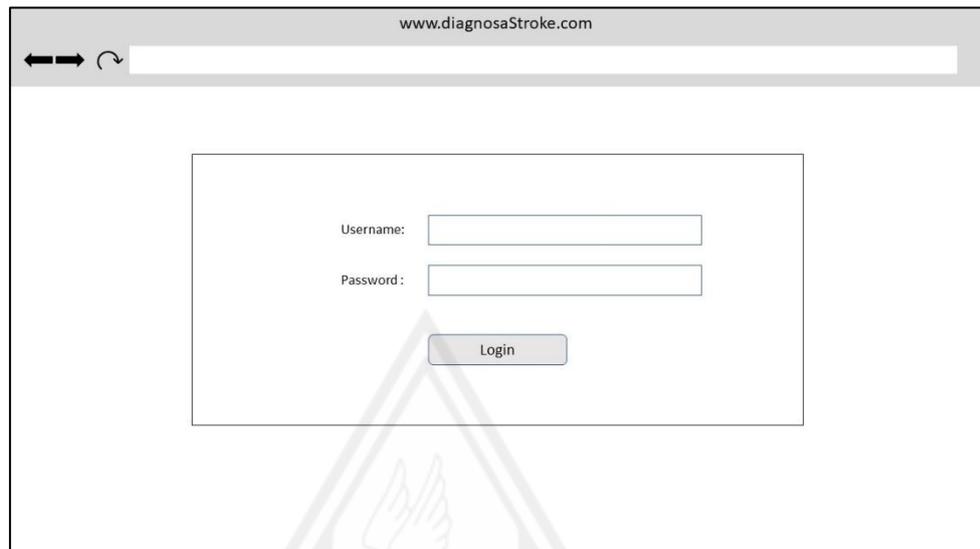
Atribut	Tipe Data	Keterangan
kode_penyakit	Varchar (5)	<i>Foreign key</i> (tabel kasus)
bobot_usia	Int (1)	Bobot untuk usia
bobot_jenis_kelamin	Int (1)	Bobot untuk jenis kelamin

bobot_kondisi	Int (1)	Bobot untuk kondisi
bobot_kesadaran	Int (1)	Bobot untuk kesadaran
bobot_nyeri_kepala	Int (1)	Bobot untuk nyeri kepala
bobot_ts	Int (1)	Bobot untuk tekanan sistolik
bobot_td	Int (1)	Bobot untuk tekanan diastolik
bobot_suhu	Int (1)	Bobot untuk suhu
bobot_napas	Int (1)	Bobot untuk nafas
bobot_nadi	Int (1)	Bobot untuk nadi

3.6 Perancangan Antarmuka Pengguna

3.6.1 Halaman Login

Halaman login adalah halaman pertama saat pengguna memasuki sistem. Pengguna akan dimintai *username* dan *password* agar dapat diverifikasi oleh sistem, jika cocok maka sistem akan mengecek *role* atau hak akses pengguna. Terdapat tiga hak akses, yaitu hak akses sebagai paramedis, hak akses sebagai pakar, dan hak akses sebagai *admin*. Setelah berhasil *login*, maka sistem akan menampilkan halaman awal berdasarkan hak akses pengguna.

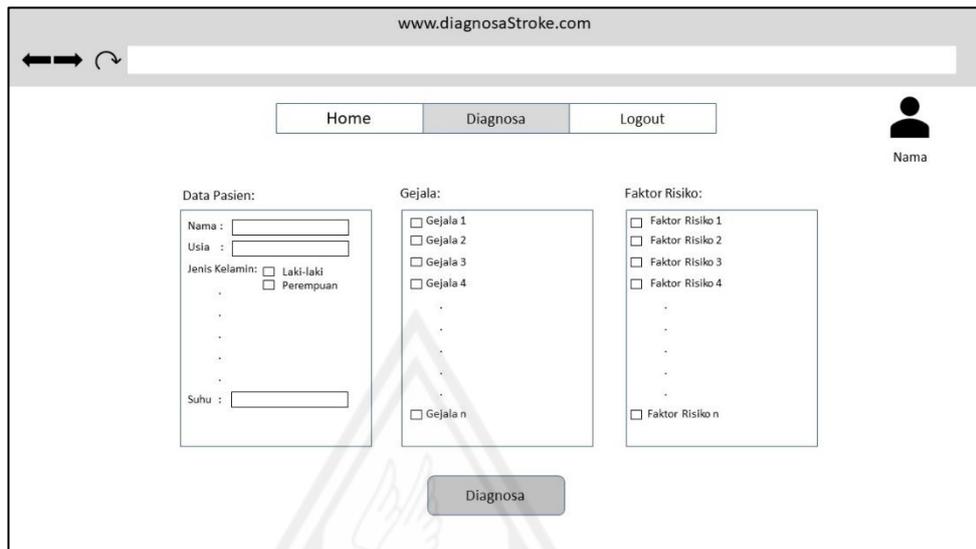


Gambar 3.6: Rancangan Antarmuka Halaman Login

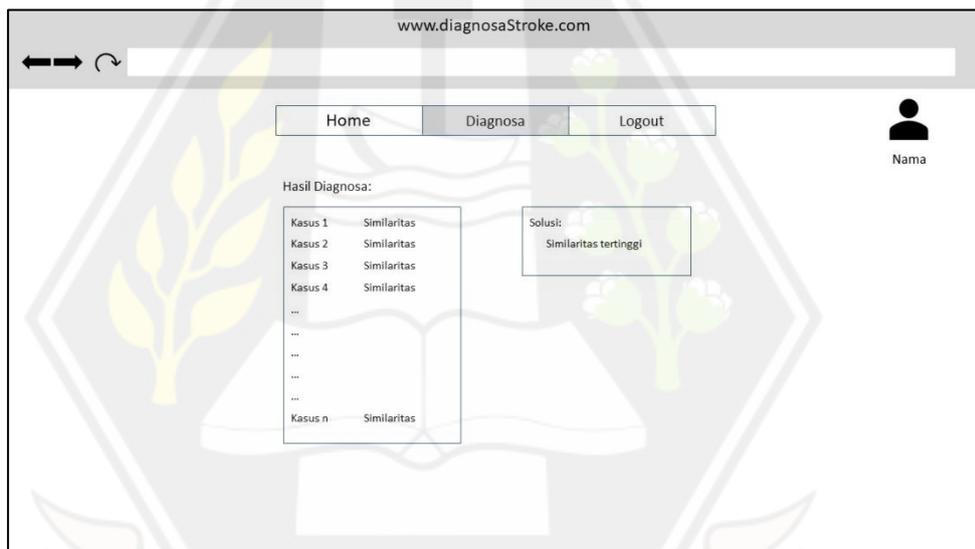
3.6.2 Halaman Home

3.6.3 Halaman Diagnosis

Halaman diagnosis berfungsi sebagai halaman diagnosis, dimana pengguna akan memasukan data pasien, gejala dan faktor risiko untuk didiagnosis dan hanya pengguna dengan hak akses paramedis yang memiliki akses ke halaman ini. Tombol diagnosa berfungsi untuk memulai diagnosa dari data pasien yang telah dimasukan, sistem akan menghitung similaritas antara kasus baru yang dimasukan dan basis kasus. Gambar 3.7 menunjukkan rancangan untuk halaman diagnosis. Kemudian hasil perhitungan similaritas dan hasil diagnosa berupa jenis penyakit *stroke* akan ditampilkan pada halaman hasil diagnosa yang dapat dilihat pada gambar 3.8.



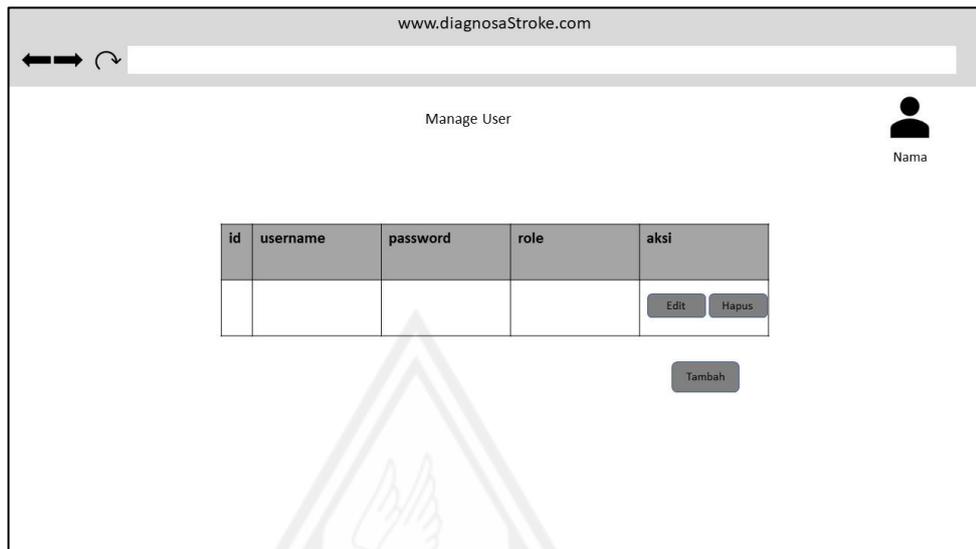
Gambar 3.7: Rancangan Antarmuka Halaman Diagnosis



Gambar 3.8: Rancangan Antarmuka Halaman Hasil Diagnosa

3.6.4 Halaman User

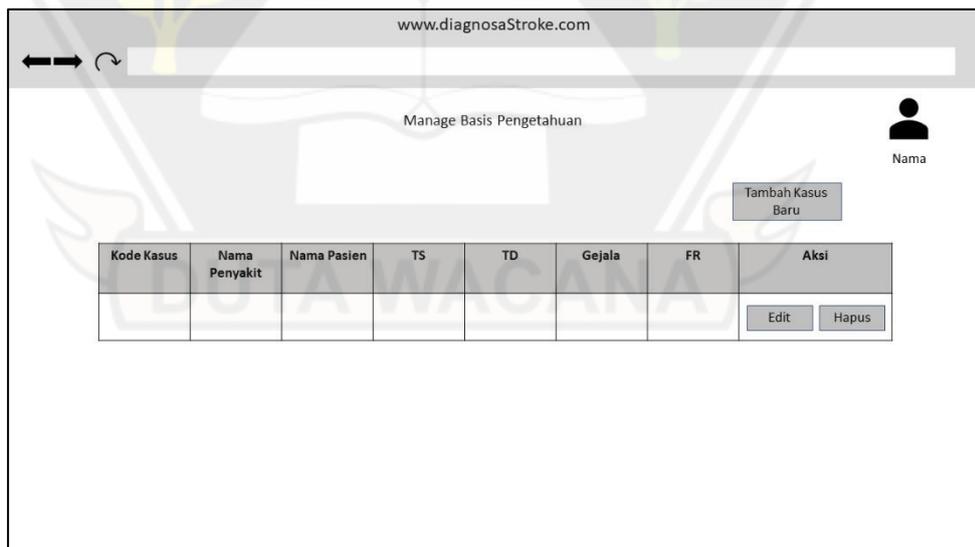
Halaman *user* adalah halaman untuk manajemen data pengguna dan hanya pengguna dengan hak akses admin yang dapat mengakses halaman ini. Seorang admin dapat menghapus, mengedit, dan menambah *user*. Rancangan halaman *user* dapat dilihat pada gambar 3.9.



Gambar 3.9: Rancangan Antarmuka Halaman User

3.6.5 Halaman Basis Pengetahuan

Halaman ini berfungsi sebagai halaman untuk manajemen data basis kasus dan hanya dapat diakses oleh pengguna yang memiliki hak akses sebagai pakar. Seorang pakar dapat melakukan revisi terhadap data kasus, menghapus kasus baru, dan menambah kasus. Rancangan halaman basis pengetahuan ditunjukkan pada gambar 3.9.



Gambar 3.10: Rancangan Antarmuka Halaman Basis Pengetahuan

Gambar 3.11: Rancangan Antarmuka Halaman Edit Basis Kasus

Gambar 3.12: Rancangan Antarmuka Halaman Tambah Basis Kasus

3.7 Perancangan Pengujian Sistem

Pengujian terhadap sistem yang telah dibangun menggunakan metode *confusion matrix* yang akan menghasilkan nilai *accuracy*, *recall* atau *sensitivity*, dan *precision*. Pengujian dilakukan dengan menggunakan data dari penelitian Nelson R et al., (2018) yang berupa data rekam medis dari pasien penderita *stroke* selama tahun 2015-2016 di Rumah Sakit DKT Dr. Soetarto, Yogyakarta yang

terdiri dari 180 kasus, dimana 30% dari kasus tersebut yaitu 54 kasus akan dijadikan data uji. Setelah itu sistem akan diuji menggunakan nilai *threshold* yang berbeda-beda yaitu (0,7), (0,75), (0,8), (0,85), (0,9), (0,95), (1,0) dan dihitung nilai akurasi sistem terhadap setiap nilai *threshold* tersebut. Untuk mendapatkan nilai *r* yang paling optimal, dilakukan pengujian dengan menggunakan nilai *r* yang berbeda-beda, dimulai dengan $r = 1$ dan akan terus bertambah 1 hingga nilai similaritas yang dihasilkan tidak menunjukkan selisih yang signifikan. Sehingga nilai *r* yang menghasilkan similaritas tertinggi yang akan digunakan.



BAB IV

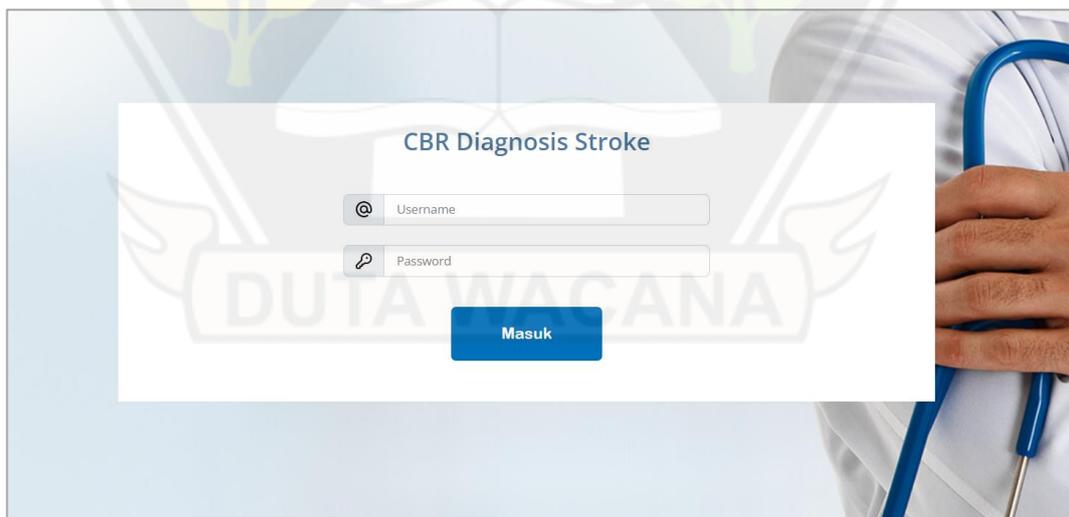
IMPLEMENTASI DAN PEMBAHASAN

4.1 Implementasi Antar Muka Sistem

Implementasi antar muka sistem terdiri dari halaman *login*, halaman diagnosis, halaman hasil diagnosis, halaman manage pengguna (admin), halaman manage kasus (pakar).

4.1.1 Implementasi halaman login

Halaman *login* merupakan halaman pertama yang diakses oleh pengguna untuk dapat menggunakan sistem. Pada halaman *login*, pengguna akan diminta untuk memasukkan *username* dan *password*. Setelah pengguna menekan tombol masuk, maka sistem akan mengecek apakah *username* dan *password* yang dimasukkan benar atau tidak, jika benar maka sistem akan mengecek hak akses dari pengguna untuk menentukan halaman yang dapat diakses berdasarkan hak akses pengguna. Implementasi dari halaman *login* dapat dilihat pada gambar 4.1.



Gambar 4.1: Halaman *login*

4.1.2 Implementasi halaman diagnosis

Halaman diagnosis merupakan halaman yang hanya dapat diakses oleh pengguna dengan hak akses paramedis. Halaman ini merupakan halaman untuk melakukan diagnosis sehingga pengguna akan diminta untuk memasukan data pasien, gejala yang dirasakan pasien, dan faktor risiko pasien. Tombol diagnosis berfungsi untuk memulai diagnosis dimana sistem akan memulai proses *retrieve*, kemudian menghitung nilai similaritas lokal dan global, dan proses *reuse*. Implementasi halaman diagnosis dapat dilihat pada gambar 4.2.

CBS CBR Diagnosis Stroke

Home About Angel Rumuy

Data Pasien:

Nama :
Masukkan Nama Pasien

Alamat :
Masukkan Alamat Pasien

Usia :
Masukkan Usia Pasien

Jenis Kelamin :
-

Kesadaran :
-

Gejala:

Gangguan Motorik

- G1 - Gangguan Bicara (Afasia Ekspresif)
- G10 - Nyeri atau kram saat berjalan (Cladiocatio Intermitten)
- G12 - Sulit berbicara (Disatria)
- G15 - Kelemahan anggota gerak kiri atau kanan (Hemiplegia)
- G16 - Onset saat beraktivitas (tiba-tiba)
- G19 - Tidak dapat membersihkan diri sendiri (cuci muka, sikat gigi, sisir rambut)
- G2 - Status Fungsional perlu bantuan
- G20 - Onset saat istirahat

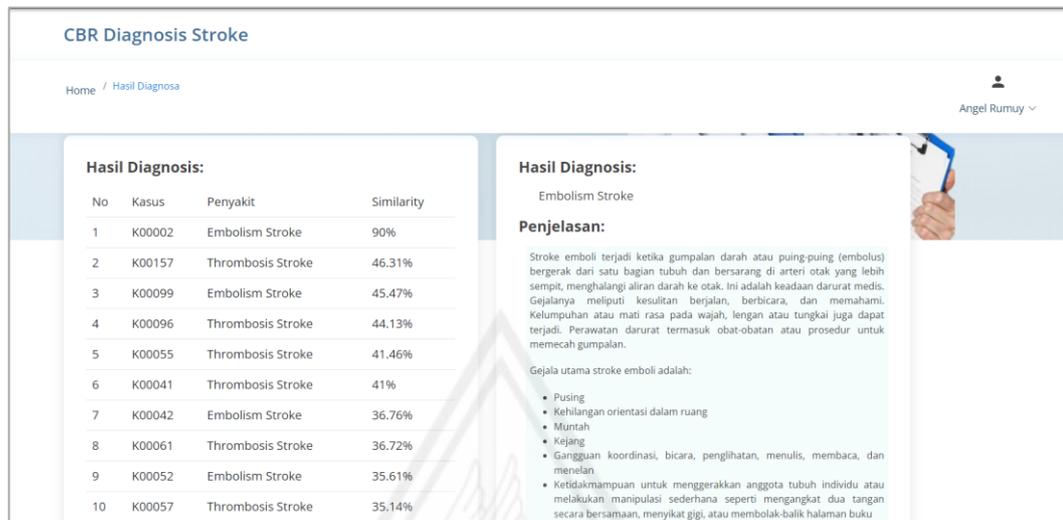
Faktor Risiko:

Riwayat Penyakit

- FR1 - Riwayat penyakit Jantung
- FR2 - Riwayat penyakit Hipertensi
- FR3 - Riwayat penyakit Diabetes Mellitus
- FR4 - Riwayat Hiperkolesterolemia
- FR6 - Riwayat penyakit Stroke
- FR7 - Riwayat penyakit Stroke dalam keluarga
- FR8 - Riwayat penyakit Hipertensi dalam keluarga
- FR9 - Riwayat penyakit Diabetes Mellitus dalam keluarga

Gambar 4.2: Halaman diagnosis

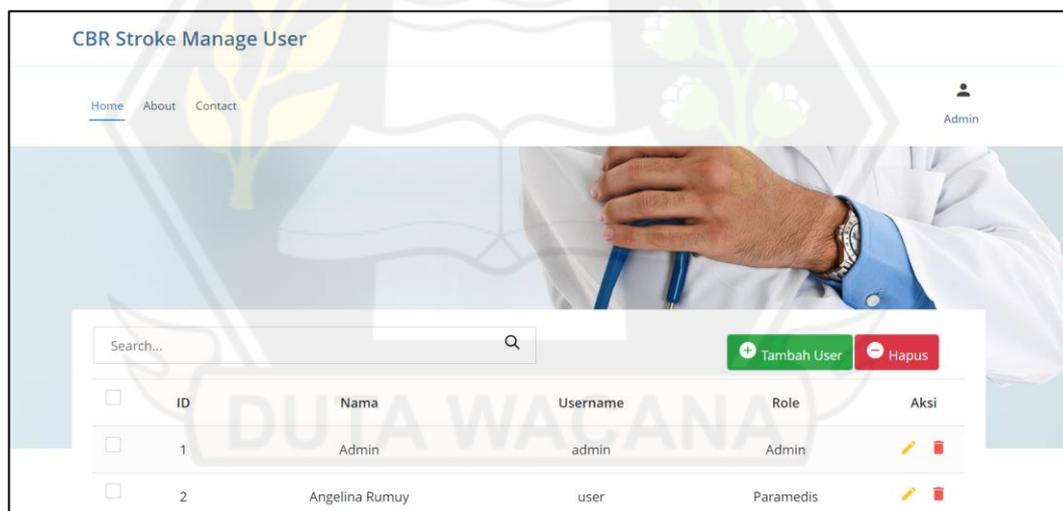
Setelah pengguna menekan tombol diagnosis, maka hasil diagnosis akan ditampilkan pada halaman hasil diagnosis. Pada halaman ini akan ditampilkan solusi berupa nama penyakit dan juga sepuluh basis kasus yang memiliki similaritas tertinggi terhadap kasus baru. Implementasi dari halaman hasil diagnosis dapat dilihat pada gambar 4.3.



Gambar 4.3: Halaman hasil diagnosis

4.1.3 Implementasi halaman manajemen pengguna

Halaman admin merupakan halaman untuk menambah, menghapus, dan mengedit pengguna sistem. Halaman ini hanya dapat diakses oleh pengguna dengan hak akses admin. Implementasi dari halaman admin dapat dilihat pada gambar 4.4.

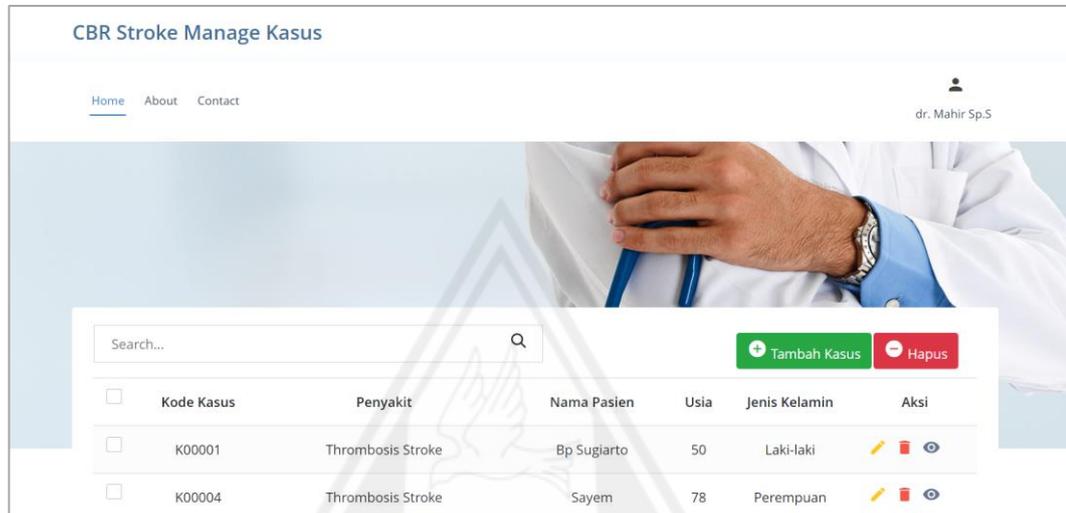


Gambar 4.4: Halaman manajemen pengguna

4.1.4 Implementasi halaman pengolahan basis pengetahuan

Halaman pakar merupakan halaman untuk menambah, menghapus, dan merevisi data kasus seperti data pasien, data gejala, dan data faktor risiko. Halaman

ini hanya bisa diakses oleh pengguna dengan hak akses pakar. Implementasi dari halaman pakar dapat dilihat pada gambar 4.5.



Gambar 4.5: Halaman pengolahan basis pengetahuan

4.2 Implementasi Case Based Reasoning

4.2.1 Implementasi Proses Retrieve

Proses *retrieve* adalah proses pencarian kasus yang mirip dengan kasus baru yang dilakukan dengan cara menghitung tingkat similaritas antara kasus baru dan basis kasus. Perhitungan similaritas menggunakan metode *minkowski distance similarity*, dengan tahap sebagai berikut.

1. Perhitungan similaritas lokal

Perhitungan similaritas lokal dilakukan dengan menghitung tingkat kemiripan antara fitur kasus baru dan basis kasus berdasarkan tipe data dari fitur. Fitur usia, jenis kelamin, tekanan darah sistolik, tekanan darah diastolik, nyeri kepala, suhu, nadi, kondisi, dan tingkat kesadaran memiliki tipe data numerik sehingga dihitung menggunakan persamaan 2.5, sedangkan fitur gejala dan faktor risiko memiliki tipe data boolean sehingga dihitung menggunakan persamaan 2.6. Implementasi perhitungan similaritas lokal dapat dilihat pada *Source code* 4.1 sampai dengan *Source code* 4.4.

```

function sim_local_numeric($source_case, $target_case,
$range) {
    $result = 1 - (abs($source_case - $target_case)/$range);
    return $result;
}

function sim_local_boolean($source_case, $target_case){
    if($source_case == $target_case){
        return 1;
    }
}

```

Source code 4.1: Fungsi untuk menghitung similaritas lokal

```

$sim_local_fitur_value = array();
//mengambil data dari db
$range_usia = $max_usia['max_usia'] -
$min_usia['min_usia'];
$range_kesadaran = 3;
$range_kondisi = 3;
$range_nyeri_kepala = 10;
$range_ts = $max_ts['max_ts'] - $min_ts['min_ts'];
$range_td = $max_td['max_td'] - $min_td['min_td'];
$range_suhu = $max_suhu['max_suhu'] -
$min_suhu['min_suhu'];
$range_napas = $max_napas['max_napas'] -
$min_napas['min_napas'];
$range_nadi = $max_nadi['max_nadi'] -
$min_nadi['min_nadi'];

//memanggil fungsi similaritas lokal
$sim_lok_usia = sim_local_numeric($row['Usia'], $usia,
$range_usia);
$sim_local_fitur_value['usia'] = $sim_lok_usia;

$sim_lok_jk =
sim_local_boolean($jk,$row['Jenis_kelamin']);
$sim_local_fitur_value['jk'] = $sim_lok_jk;
...

```

Source code 4.2: Perhitungan similaritas lokal untuk data pasien

```

$sql_gejala = "SELECT detail_gejala.kode_gejala,detail_
              gejala.nilai FROM detail_gejala WHERE
              detail_gejala.kode_kasus =
              '$row['Kode_kasus'].''";
$det_gejala = mysqli_query($conn, $sql_gejala);
while($getail_gejala = $det_gejala->fetch_assoc()){
    if(in_array($getail_gejala['kode_gejala'],$array_value
    _post) and $getail_gejala['nilai'] == 1){
        $sim_local_fitur_value[$getail_gejala['kode_gejala']] =
1;
    }elseif($getail_gejala['nilai'] == 1){
        array_push($union_fitur,
        $getail_gejala['kode_gejala']);
        $sim_local_fitur_value[$getail_gejala
        ['kode_gejala']] = 0 ;
    }else{
        $sim_local_fitur_value[$getail_gejala
        ['kode_gejala']] = 0;
    }
    if($getail_gejala['nilai'] == 1){
        array_push($fitur_sc, $getail_gejala['kode_gejala']);
    }
}

```

Source code 4.3: Perhitungan similaritas lokal fitur gejala

```

$sql_fr = "SELECT detail_fr.kode_FR, detail_fr.nilai FROM
           detail_fr WHERE detail_fr.kode_kasus =
           '$row['Kode_kasus'].'' ";
$det_fr = mysqli_query($conn, $sql_fr);
while($detail_fr = $det_fr->fetch_assoc()){
    if(in_array($detail_fr['kode_FR'], $array_value_post) and
    $detail_fr['nilai'] == 1){
        $sim_local_fitur_value[$detail_fr['kode_FR']] = 1;
        array_push($union_fitur, $detail_fr['kode_FR']);
    }elseif($detail_fr['nilai'] == 1){
        array_push($union_fitur, $detail_fr['nilai']);
        $sim_local_fitur_value[$detail_fr['kode_FR']] = 0 ;
    }else{
        $sim_local_fitur_value[$detail_fr['kode_FR']] = 0 ;
    }
}

```

Source code 4.4: Perhitungan similaritas lokal fitur gejala

2. Proses perhitungan similaritas global (*minkowski distance similarity*). perhitungan similaritas global dilakukan dengan mengambil bobot dari setiap fitur menggunakan fungsi pada *Source code 4.5*. Setelah itu, sistem akan mengambil tingkat keyakinan pakar terhadap kasus dan menghitung

nilai similaritas global (*minkowski distance similarity*), dapat dilihat pada *Source code* 4.6 dan 4.7.

```
function get_bobot_fitur($kode_penyakit, $conn){
    //mengambil bobot umum
    $sql_bobot_umum = "SELECT pembobotan_umum.bbt_usia,
pembobotan_umum.bbt_jk,pembobotan_umum.bbt_kondisi,pembobot
an_umum.bbt_kesadaran,pembobotan_umum.bbt_nyeri_kepala,pemb
obotan_umum.bbt_ts,pembobotan_umum.bbt_td,pembobotan_umum.b
bt_suhu, pembobotan_umum.bbt_napas,pembobotan_umum.bbt_nadi
FROM pembobotan_umum WHERE pembobotan_umum.kode_penyakit=
'".$kode_penyakit."' ";
    $bobot_um = mysqli_query($conn, $sql_bobot_umum);
    $bobot= $bobot_um->fetch_assoc();

    //mengambil bobot gejala
    $sql_bobot_gejala = "SELECT
pembobotan_gejala.kode_gejala, pembobotan_gejala.bobot FROM
pembobotan_gejala WHERE pembobotan_gejala.kode_penyakit =
'".$kode_penyakit."' ";
    $bobot_gej = mysqli_query($conn, $sql_bobot_gejala);
    while($r = $bobot_gej->fetch_assoc()){
        $key_bbt_gejala = 'bbt_' . $r['kode_gejala'];
        $bobot[$key_bbt_gejala] = $r['bobot'];
    }
    //mengambil bobot fr
    $sql_bobot_fr = "SELECT pembobotan_fr.kode_FR,
pembobotan_fr.bobot FROM pembobotan_fr WHERE
pembobotan_fr.kode_penyakit = '".$kode_penyakit."' ";
    $fr_bobot = mysqli_query($conn, $sql_bobot_fr);
    while($rFr = $fr_bobot->fetch_assoc()){
        $key_bbt_fr = 'bbt_' . $rFr['kode_FR'];
        $bobot[$key_bbt_fr] = $rFr['bobot'];
    }
    return $bobot;
}
```

Source code 4.5: Fungsi untuk mengambil nilai bobot dari basis data

```
$sql_keyakinan_pakar = mysqli_query($conn, "SELECT
keyakinan_pakar FROM kasus WHERE
Kode_kasus='".$row['Kode_kasus']."' ");

$tingkat_keyakinan_pakar =
mysqli_fetch_assoc($sql_keyakinan_pakar);

$nilai_keyakinan_pakar =
$tingkat_keyakinan_pakar['keyakinan_pakar'];
```

Source code 4.6: Pengambilan tingkat keyakinan pakar dari basis data

```

function sim_global($arr_bobot, $sim_loc, $union_fitur,
$CiCj, $ci, $r, $keyakinan_pakar) {
    $bobot_kali_sim_lokal_pangkat_r = array();
    $arr_bobot_pangkat = array();
    foreach(array_keys($sim_loc) as $similaritas_loc){
        $key_bobot = "bbt_".$similaritas_loc;
        array_push($bobot_kali_sim_lokal_pangkat_r, pow
($arr_bobot[$key_bobot]*$sim_loc[$similaritas_loc],$r));
        //untuk menghitung jumlah bobot dari fitur2 kedua case
        if(in_array($similaritas_loc, $union_fitur)){
            array_push($arr_bobot_pangkat,
            pow($arr_bobot[$key_bobot], $r));
        }
    }
    $sum_bobot_kali_local_pangkat =
array_sum($bobot_kali_sim_lokal_pangkat_r);
    $sum_bobot_pangkat =
array_sum($arr_bobot_pangkat);
    $pre_result = pow(($sum_bobot_kali_local_pangkat/
$sum_bobot_pangkat), (1/$r));
    $result = round($pre_result * $keyakinan_pakar *
($CiCj/$ci), 2);
    return $result;
}
//jika penyakit adalah non stroke atau "revisi" maka tidak
dihitung
if(($row['Kode_penyakit'] != 'P5') and
($row['Kode_penyakit'] != 'P0')){
    $bobot_fitur = get_bobot_fitur($row['Kode_penyakit'],
$conn);
    $sim_glob = sim_global($bobot_fitur,
$sim_local_fitur_value, $union_fitur, sizeof
($arr_CiCj), sizeof($arr_ci), $r,$nilai_keyakinan_pakar);
    $arr_result_sim_global[$row['Kode_kasus']] = $sim_glob;
}
}

```

Source code 4.7: Perhitungan similaritas global (*minkowski distance similarity*)

Berikut adalah tampilan antarmuka untuk proses *retrieve*.

CBR Diagnosis Stroke

Home / Hasil Diagnosa

Angel Rumuy

Hasil Diagnosis:

No	Kasus	Penyakit	Similarity
1	K00002	Embolism Stroke	90%
2	K00157	Thrombosis Stroke	46.31%
3	K00099	Embolism Stroke	45.47%
4	K00096	Thrombosis Stroke	44.13%
5	K00055	Thrombosis Stroke	41.46%
6	K00041	Thrombosis Stroke	41%
7	K00042	Embolism Stroke	36.76%
8	K00061	Thrombosis Stroke	36.72%
9	K00052	Embolism Stroke	35.61%
10	K00057	Thrombosis Stroke	35.14%

Hasil Diagnosis:

Embolism Stroke

Penjelasan:

Stroke emboli terjadi ketika gumpalan darah atau puing-puing (embolus) bergerak dari satu bagian tubuh dan bersarang di arteri otak yang lebih sempit, menghalangi aliran darah ke otak. Ini adalah keadaan darurat medis. Gejalanya meliputi kesulitan berjalan, berbicara, dan memahami. Kelumpuhan atau mati rasa pada wajah, lengan atau tungkai juga dapat terjadi. Perawatan darurat termasuk obat-obatan atau prosedur untuk memecah gumpalan.

Gejala utama stroke emboli adalah:

- Pusing
- Kehilangan orientasi dalam ruang
- Muntah
- Kejang
- Gangguan koordinasi, bicara, penglihatan, menulis, membaca, dan menelan
- Ketidakmampuan untuk menggerakkan anggota tubuh individu atau melakukan manipulasi sederhana seperti mengangkat dua tangan secara bersamaan, menyikat gigi, atau membolak-balik halaman buku

Gambar 4.6: Antarmuka proses *retrieve*

4.2.2 Implementasi Proses Reuse

Proses *reuse* adalah proses menggunakan kembali kasus lama yang memiliki nilai similaritas tertinggi terhadap kasus baru. Setelah mendapatkan kasus dengan similaritas tertinggi, kemudian similaritas kasus tersebut akan dibandingkan dengan nilai *threshold*. Jika nilai similaritasnya lebih besar atau sama dengan nilai *threshold*, maka kasus lama tersebut dapat dijadikan sebagai solusi untuk kasus baru, namun jika nilai similaritasnya lebih kecil dari nilai *threshold*, maka kasus akan masuk ke tahap revisi. Implementasi dari tahap *reuse* dapat dilihat pada *Source code* 4.8.

```

function validate_solution($case, $threshold){
    if($case >= $threshold){
        return true;
    }else{
        return false;
    }
}
//sorting
array_multisort($arr_result_sim_global, SORT_DESC);

//pengambilan solusi
$threshold = 80;
$max_sim = max($arr_result_sim_global);
if(validate_solution($max_sim, $threshold)){
    $passed = true;
    $solusi = array_key_first($arr_result_sim_global);
}else{
    $passed = false;
    $solusi = "Need Revise";
}

```

Source code 4.8: Implementasi proses reuse

4.2.3 Implementasi Proses Revise

Revise adalah proses memperbaiki solusi kasus yang diberikan, proses ini dilakukan oleh pakar. Saat pakar menekan tombol revisi, maka sistem akan memperbaharui hasil diagnosis kasus. Implementasi dari proses *revise* dapat dilihat pada *Source code 4.9*.

```

$sql_update_kasus = 'UPDATE kasus SET kasus.tgl_diagnosis =
"'.$_POST['tanggalDiagnosa'].'", kasus.kode_penyakit =
"'.$_POST['penyakit'].'", kasus.keyakinan_pakar =
'.$_POST['keyakinan_pakar'].' WHERE kasus.Kode_kasus =
"'.$_POST['kode_kasus'].'" ';

if(($conn->query($sql_update_kasus)){
    echo "<script type='text/javascript'>
        alert('Revisi Kasus Berhasil!');
        window.location.href='manageKasus.php';
    </script>";
}

```

Source code 4.9: Implementasi proses revise kasus

Gambar 4.7: Antarmuka proses *revise* (data pasien, gejala, faktor risiko)

Gambar 4.8: Antarmuka proses *revise* (hasil diagnosis dan keyakinan pakar)

4.2.4 Implementasi Proses Retain

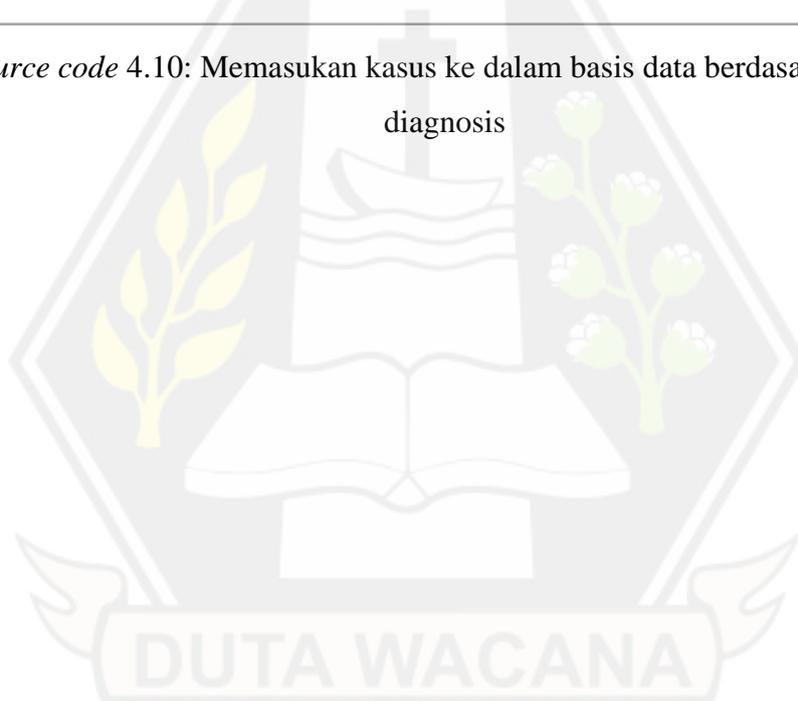
Retain adalah proses untuk menyimpan kasus baru kedalam basis kasus. Saat pengguna menekan tombol simpan, maka kasus tersebut akan disimpan ke dalam basis kasus agar dapat dijadikan solusi untuk kasus baru selanjutnya. Jika kasus masuk ke tahap revisi, maka kasus akan dimasukkan kedalam basis kasus dengan hasil diagnosis adalah *revise* agar direvisi oleh pakar. Implementasi dari proses *revise* dapat dilihat pada *Source code* 4.10 dan 4.11.

```

if($_POST['state'] == "revise"){
    $sql_insert_kasus = "INSERT INTO kasus (Kode_kasus,
    tgl_diagnosis, kode_penyakit, keyakinan_pakar) VALUES
    ('".$_SID_case."', '0000-00-00', 'P0', 0)";
} elseif($_POST['state'] == "passed"){
    $sql_keyakinan_pakar_case_solusi = mysqli_query($conn,
    "SELECT kasus.keyakinan_pakar as keyakinan_pakar FROM
    `kasus` WHERE kasus.Kode_kasus =
    '".$_POST['solusi_kode_kasus']."' ");
    $keyakinan_pakar_case_solusi =
    $sql_keyakinan_pakar_case_solusi->fetch_assoc();
    $keyakinan_pakar_case_solusi_value =
    $keyakinan_pakar_case_solusi['keyakinan_pakar'];
    if($_POST['tingkat_similaritas'] != 100){
        $new_keyakinan = tingkat_keyakinan_newcase
        ($keyakinan_pakar_case_solusi_value);
        $sql_insert_kasus = "INSERT INTO kasus (Kode_kasus,
        tgl_diagnosis, kode_penyakit, keyakinan_pakar) VALUES
        ('".$_SID_case."', '".$_POST['tanggalDiagnosa']."',
        '".$_POST['kode_penyakit']."', '".$_new_keyakinan.'"");
    }
}

```

Source code 4.10: Memasukan kasus ke dalam basis data berdasarkan hasil diagnosis



```

if ($conn->query($sql_insert_kasus)) {
    $sql_pasien = "INSERT INTO `pasien` (`id_pasien`,
`kode_kasus`, `Nama`, `Alamat`, `Usia`, `Jenis_kelamin`,
`Kondisi`, `Kesadaran`, `nyeri_kepala`, `TS`, `TD`, `Suhu`,
`Nafas`, `Denyut_nadi`) VALUES ('".$id_pasien."', '".
$id_case."', '".$_POST['name']."', '".$_POST['alamat']."',
"$_POST['usia']."', "$_POST['jk']."', "$_POST['kondisi']."',
"$_POST['kesadaran']."', "$_POST['nyeri_kepala']."',
"$_POST['ts']."', "$_POST['td']."', "$_POST['suhu']."',
"$_POST['napas']."', "$_POST['nadi']."')";
    if($conn->query($sql_pasien)){
        //insert into detail_gejala
        $checked_gejala = array();
        $checked_fr = array();
        foreach ($_POST['data'] as $item){
            if(strpos($item, "G")!= false){
                array_push($checked_gejala, $item);
            }else if(strpos($item, "FR")!= false){
                array_push($checked_fr, $item);
            }
        }
        while($row = $data_gejala->fetch_assoc()) {
            $nilai_gejala = 0;
            if(in_array($row['kode_gejala'],
            $checked_gejala)){
                $nilai_gejala = 1;
            }
            $sql_insert_gejala = "INSERT INTO
            `detail_gejala` (`kode_kasus`, `kode_gejala`,
            `nilai`) VALUES ('".$id_case."', '". $row
            ['kode_gejala']."' , " . $nilai_gejala . ")";
            if($conn->query($sql_insert_gejala)){
                echo("gejala berhasil");
            }
        }
        //insert into detail_fr
        while($row = $data_FR->fetch_assoc()) {
            $nilai_fr = 0;
            if(in_array($row['kode_FR'], $checked_fr)){
                $nilai_fr = 1;
            }
            $sql_insert_fr = "INSERT INTO `detail_fr`
            (`kode_kasus`, `kode_FR`, `nilai`) VALUES
            ('".$id_case."', '". $row['kode_FR']."' ,
            " . $nilai_fr . ")";
            if($conn->query($sql_insert_fr)){
                echo("fr berhasil");
            }
        }
    }
}

```

Source code 4.11: Memasukan gejala dan faktor risiko kasus ke dalam basis data

4.3 Pengujian dan Analisis

4.3.1 Data uji dan metode pengujian

Pada tahap ini dilakukan pengujian untuk mengukur kinerja dari sistem yang telah dibuat menggunakan metode *confusion matrix*, dimana akan dihitung nilai akurasi, *precision*, dan *recall* atau *sensitivitas* dari sistem. Penelitian ini menggunakan data kasus dari pasien penderita *stroke* selama tahun 2015-2016 di Rumah Sakit DKT Dr. Soetarto, Yogyakarta yang berjumlah 180 kasus. Berikut beberapa hal yang dilakukan dalam proses pengujian sistem.

- Kasus yang digunakan pada penelitian ini berjumlah 180 kasus yang terdiri dari 56 kasus *hemorrhage stroke* (28 kasus *intracerebral stroke* dan 28 kasus *subarachnoid stroke*), 112 kasus *ischemic stroke* (56 kasus *thrombotic stroke* dan 56 kasus *embolic stroke*), dan 12 kasus bukan *stroke*.
- Kasus dibagi menjadi dua bagian, yaitu kasus yang akan menjadi basis kasus (70% dari total kasus yaitu 126 kasus) dan kasus yang akan menjadi data uji (30% dari total kasus yaitu 54 kasus).
- 54 data uji terdiri dari 17 kasus *thrombotic stroke*, 17 kasus *embolic stroke*, 8 kasus *subarachnoid stroke*, 8 kasus *intracerebral stroke*, dan 4 kasus bukan *stroke*.
- Pengujian dilakukan menggunakan metode *convolution matrix*, dimana akan dihitung nilai akurasi menggunakan persamaan 2.1, *precision* menggunakan persamaan 2.2, dan *recall* atau *sensitivitas* menggunakan persamaan 2.3.
- Pengujian dilakukan dengan menerapkan nilai *threshold* yaitu (0,6), (0,65), (0,7), (0,75), (0,8), (0,85) dan (0,9) dimana nilai pangkat *minkowski* (r) yang diterapkan akan terus meningkat mulai dari 1 sampai nilai akurasi sistem tidak mengalami perubahan yang signifikan lagi.
- Pengujian oleh pakar yaitu dokter untuk memvalidasi hasil diagnosis sistem.

Pengujian dilakukan menggunakan skrip *automation* agar lebih efisien, dimana skrip *automation* tersebut akan mengisi data pasien yang akan didiagnosis secara otomatis untuk 54 kasus pada setiap nilai *threshold* dan nilai r . Skrip *automation* akan melakukan *login* ke sistem lalu mengisi data pasien, gejala, dan faktor risiko secara otomatis berdasarkan data uji setiap kasus dan mencatat hasilnya pada file excel dan file pdf.

4.3.2 Hasil Pengujian

a. Hasil pengujian oleh pakar (dokter)

Pengujian sistem dilakukan dengan uji coba masukan data pasien stroke oleh pakar dan memastikan hasil diagnosis yang diberikan sistem sesuai dengan keinginan pakar. Hasil pengujian disajikan pada Tabel 4.1.

Tabel 4.1: Hasil pengujian oleh dokter (*stroke* iskemik)

Fitur	Diagnosis
Prosedur pengujian	<ol style="list-style-type: none">1. Membuka halaman login2. Login sebagai mahasiswa atau pengguna biasa yang akan melakukan diagnosis3. Mengisi data pasien, gejala, dan faktor risiko4. Menekan tombol diagnosis
Masukan	Usia: 48 Jenis kelamin: Laki-laki Kesadaran: Sopor (koma) Keadaan umum: Sedang Nyeri kepala: 6 TS:190 TD:100 Suhu: 38 Napas: 20 Nadi: 120 Gejala: G1, G12, G15, G16, G35, G8, G14, G27, G40, G42, G43, G7, G31, G32 Faktor risiko: FR1, FR7, FR11, FR5
Hasil yang diharapkan	<i>Thrombosis Stroke (Ischemic stroke)</i>
Hasil yang didapat	<i>Thrombosis Stroke</i>
Kesimpulan	Sukses

Tabel 4.2: Hasil pengujian oleh dokter (*stroke* hemoragik)

Fitur	Diagnosis
Prosedur pengujian	<ol style="list-style-type: none">1. Membuka halaman login2. Login sebagai mahasiswa atau pengguna biasa yang akan melakukan diagnosa3. Mengisi data pasien, gejala, dan faktor risiko4. Menekan tombol diagnosis

Masukan	Usia: 58 Jenis kelamin: Perempuan Kesadaran: Somnolen (Apatitis) Keadaan umum: Sedang Nyeri kepala: 6 TS:170 TD:100 Suhu: 37 Napas: 22 Nadi: 88 Gejala: G1, G12, G15, G16, G35, G8, G14, G27, G36, G6, G7, G81, G32 Faktor risiko: FR2, FR8
Hasil yang diharapkan	<i>Hemoragic stroke</i>
Hasil yang didapat	<i>Intracerebral hemorrhage Stroke</i> (termasuk jenis <i>hemoragic stroke</i>)
Kesimpulan	Sukses

Pengujian oleh pakar dilakukan untuk memvalidasi hasil diagnosis dari sistem dimana pakar diminta untuk memasukkan data pasien, gejala dan faktor risiko dari orang yang terkena *stroke* kemudian mencocokkan hasil diagnosa dari pakar dan diagnosa dari sistem. Terdapat dua skenario pengujian yaitu diagnosis *stroke* iskemik dan diagnosis *stroke* hemoragik. Kedua skenario berhasil dimana hasil diagnosis dari pakar dan hasil diagnosis dari sistem sama, disajikan pada Tabel 4.1 dan 4.2. Setelah dilakukan pengujian, ada beberapa saran yang diberikan oleh pakar terkait sistem diagnosis *stroke* yang telah dibangun dalam penelitian ini. Pertama, sistem sebaiknya dapat menampilkan hasil pemeriksaan *CT scan/MRI*. Kedua, perlu adanya pemeriksaan saraf kranial untuk mendukung hasil diagnosis jenis *stroke*. *CT scan/MRI* kepala sangat membantu dalam menentukan penyebab terjadinya *stroke* apakah disebabkan oleh pendarahan atau penyumbatan.

b. Hasil pengujian dengan nilai *threshold* = 60

Tabel 4.3: Rincian Hasil Pengujian pada *Threshold* = 60

Nilai R	1	2	3	4	5	6	7	8	9	10	11	12
---------	---	---	---	---	---	---	---	---	---	----	----	----

TP	42	39	34	34	35	35	35	35	34	34	34	34
FN	8	11	16	16	15	15	15	15	16	16	16	16
TN	4	4	2	1	1	0	0	0	0	0	0	0
FP	0	0	2	3	3	4	4	4	4	4	4	4
Total Data Uji	54											

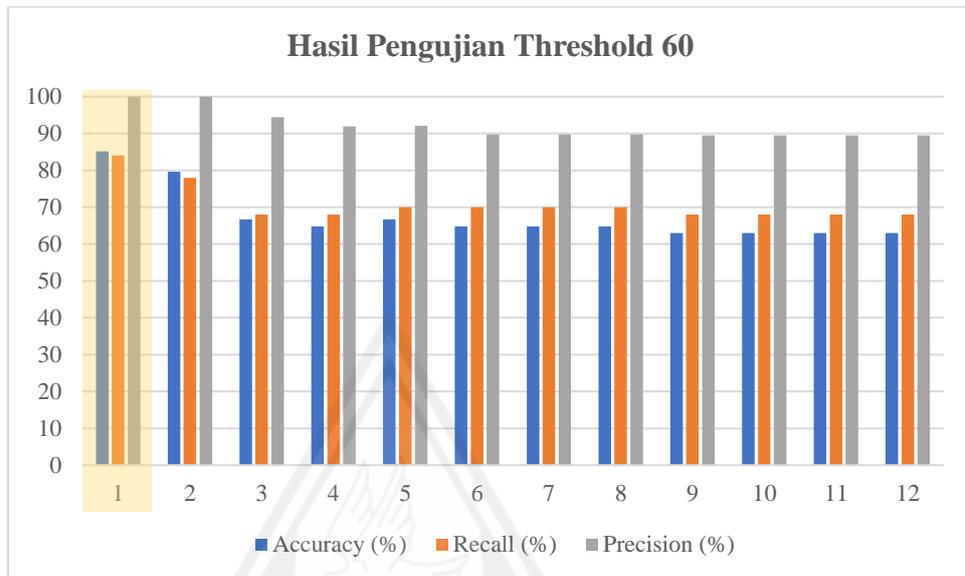
Keterangan:

TP : *True positive*, FP : *False positive*, FN : *False negative*, TN: *True negative*

Dari hasil pengujian tersebut, kemudian dihitung nilai akurasi menggunakan persamaan 2.1, *precision* menggunakan persamaan 2.2, dan *recall* atau *sensitivitas* menggunakan persamaan 2.3. Perhitungan akurasi, *precision*, dan *recall* pada nilai *threshold* 60 disajikan pada Tabel 4.4.

Tabel 4.4: Perhitungan Nilai *Accuracy*, *Recall*, dan *precision* pada *Threshold* = 60

Nilai R	Accuracy (%)	Recall (%)	Precision (%)
	$(TP + TN) / Total$	$TP / (TP + FN)$	$TP / (TP + FP)$
1	85,19	84	100
2	79,63	78	100
3	66,67	68	94,44
4	64,81	68	91,89
5	66,67	70	92,11
6	64,81	70	89,74
7	64,81	70	89,74
8	64,81	70	89,74
9	62,96	68	89,47
10	62,96	68	89,47
11	62,96	68	89,47
12	62,96	68	89,47



Gambar 4.9: Grafik hasil pengujian sistem pada *threshold* = 60

Hasil pengujian menggunakan nilai *threshold* 60 menunjukkan nilai akurasi sistem yang baik yaitu sebesar 85,19% pada nilai $r = 1$ dengan nilai *precision* 100% dan *recall* sebesar 84%, dimana sebanyak 42 kasus *stroke* didiagnosis benar dan memenuhi *threshold*, 8 kasus *stroke* didiagnosis salah dan tidak memenuhi nilai *threshold* (masuk ke tahap revisi), dan 1 kasus bukan penyakit *stroke* yang didiagnosis benar bukan penyakit *stroke*.

c. Hasil pengujian dengan nilai *threshold* = 65

Tabel 4.5: Rincian Hasil Pengujian pada *Threshold* = 65

Nilai R	1	2	3	4	5	6	7	8	9	10	11	12
TP	41	42	36	35	34	34	34	34	33	33	33	33
FN	9	8	14	15	16	16	16	16	17	17	17	17
TN	4	4	4	3	2	1	1	1	1	0	0	0
FP	0	0	0	1	2	3	3	3	3	4	4	4
Total Data Uji	54	54	54	54	54	54	54	54	54	54	54	54

Keterangan:

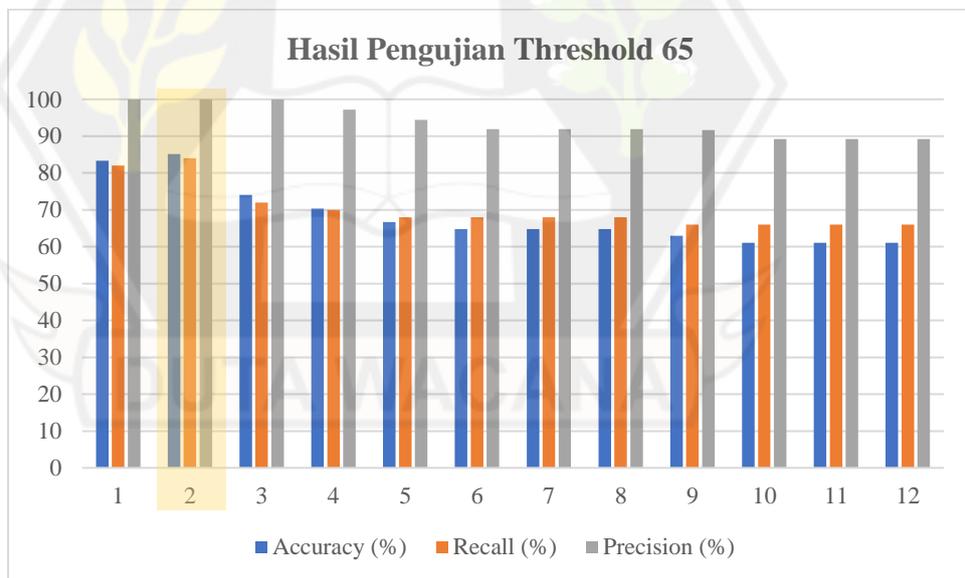
TP : *True positive*, FP : *False positive*, FN : *False negative*, TN: *True negative*

Dari hasil pengujian tersebut, kemudian dihitung nilai akurasi menggunakan persamaan 2.1, *precision* menggunakan persamaan 2.2, dan

recall atau *sensitivitas* menggunakan persamaan 2.3. Perhitungan akurasi, *precision*, dan *recall* pada nilai *threshold* 65 disajikan pada Tabel 4.6.

Tabel 4.6: Perhitungan Nilai *Accuracy*, *Recall*, dan *Precision* pada *Threshold* = 65

Nilai R	Accuracy (%)	Recall (%)	Precision (%)
	$(TP + TN) / Total$	$TP / (TP + FN)$	$TP / (TP + FP)$
1	83,33	82	100
2	85,19	84	100
3	74,07	72	100
4	70,37	70	97,22
5	66,67	68	94,44
6	64,81	68	91,89
7	64,81	68	91,89
8	64,81	68	91,89
9	62,96	66	91,67
10	61,11	66	89,19
11	61,11	66	89,19
12	61,11	66	89,19



Gambar 4.10: Grafik hasil pengujian sistem pada *threshold* = 65

Hasil pengujian menggunakan nilai *threshold* 65 menunjukkan nilai akurasi sistem yang sama tinggi dengan *threshold* 60 yaitu sebesar 85,19%

pada penerapan nilai $r = 2$ dengan nilai *precision* 100% dan *recall* sebesar 84%, dimana sebanyak 42 kasus stroke didiagnosis dengan benar dan memenuhi threshold, 3 kasus stroke didiagnosis salah dan dibawah threshold, 5 kasus stroke didiagnosis salah dan memenuhi threshold dan 4 kasus bukan penyakit stroke yang didiagnosis benar bukan penyakit stroke.

d. Hasil pengujian dengan nilai *threshold* = 70

Tabel 4.7: Rincian Hasil Pengujian pada *Threshold* = 70

Nilai R	1	2	3	4	5	6	7	8	9	10	11	12
TP	28	43	42	37	35	36	35	34	33	33	33	33
FN	22	7	8	13	15	14	15	16	17	17	17	17
TN	4	4	4	4	4	4	3	3	2	2	1	1
FP	0	0	0	0	0	0	1	1	2	2	3	3
Total Data Uji	51	54										

Keterangan:

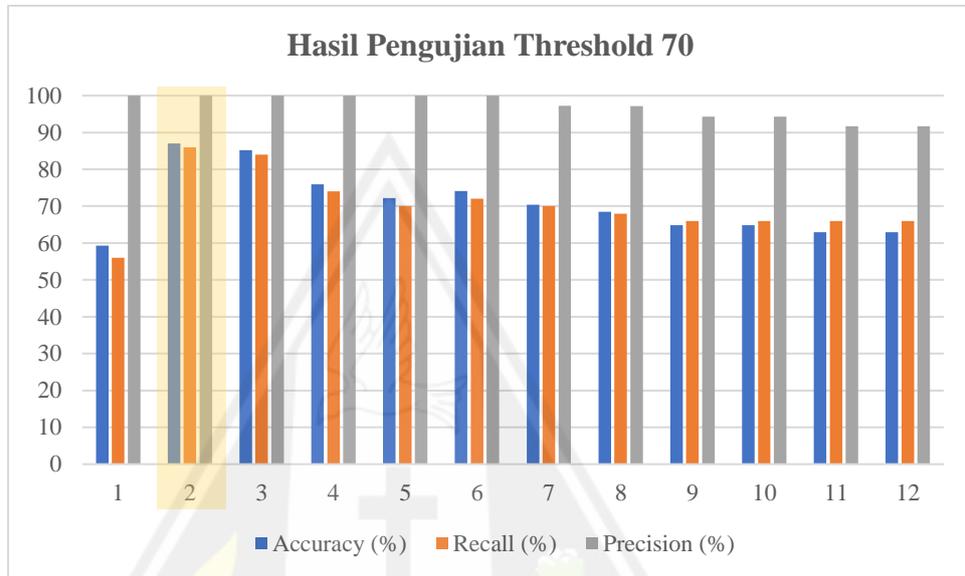
TP : *True positive*, FP : *False positive*, FN : *False negative*, TN: *True negative*

Dari hasil pengujian tersebut, kemudian dihitung nilai akurasi menggunakan persamaan 2.1, *precision* menggunakan persamaan 2.2, dan *recall* atau *sensitivitas* menggunakan persamaan 2.3. Perhitungan akurasi, *precision*, dan *recall* pada nilai *threshold* 70 disajikan pada Tabel 4.8 .

Tabel 4.8: Perhitungan Nilai *Accuracy*, *Recall*, dan *precision* pada *Threshold* = 70

Nilai R	Accuracy (%)	Recall (%)	Precision (%)
	$(TP + TN) / Total$	$TP / (TP + FN)$	$TP / (TP + FP)$
1	59,26	56	100
2	87,04	86	100
3	85,19	84	100
4	75,93	74	100
5	72,22	70	100
6	74,07	72	100
7	70,37	70	97,22
8	68,52	68	97,14
9	64,81	66	94,29

10	64,81	66	94,29
11	62,96	66	91,67
12	62,96	66	91,67



Gambar 4.11: Grafik hasil pengujian sistem pada *threshold* = 70

Hasil pengujian menggunakan nilai *threshold* 70 menunjukkan nilai akurasi sistem yang cukup baik yaitu sebesar 87,04% pada penerapan nilai $r = 2$ dengan nilai *precision* sebesar 100% dengan nilai *recall* sebesar 84%, dimana sebanyak 43 kasus *stroke* didiagnosis dengan benar dan memenuhi *threshold*, 3 kasus *stroke* didiagnosis salah dan memenuhi *threshold*, 4 kasus *stroke* didiagnosis benar tetapi dibawah *threshold* dan 4 kasus bukan penyakit *stroke* yang didiagnosis benar bukan penyakit *stroke*.

e. Hasil pengujian dengan nilai *threshold* = 75

Tabel 4.9: Rincian Hasil Pengujian pada *Threshold* = 75

Nilai R	1	2	3	4	5	6	7	8	9	10	11	12
TP	16	34	44	44	42	39	38	38	34	34	34	34
FN	34	16	6	6	8	11	12	12	16	16	16	16
TN	4	4	4	4	4	4	4	4	4	4	4	3
FP	0	0	0	0	0	0	0	0	0	0	0	1
Total Data Uji	54	54	54	54	54	54	54	54	54	54	54	54

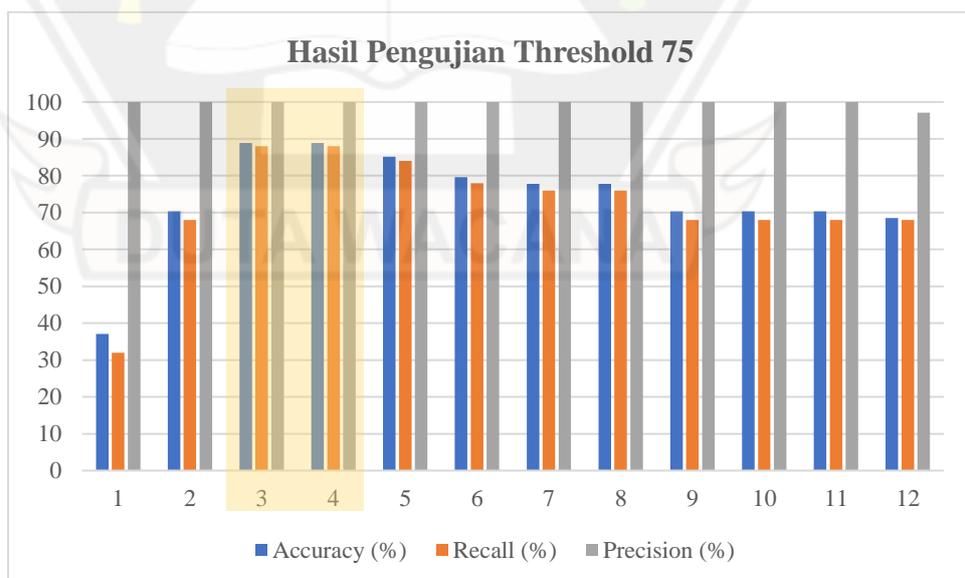
Keterangan:

TP : *True positive*, FP : *False positive*, FN : *False negative*, TN: *True negative*

Dari hasil pengujian tersebut, kemudian dihitung nilai akurasi menggunakan persamaan 2.1, *precision* menggunakan persamaan 2.2, dan *recall* atau *sensitivitas* menggunakan persamaan 2.3. Perhitungan akurasi, *precision*, dan *recall* pada nilai *threshold* 65 disajikan pada Tabel 4.10.

Tabel 4.10: Perhitungan Nilai *Accuracy*, *Recall*, dan *precision* pada *Threshold* = 75

Nilai R	Accuracy (%)	Recall (%)	Precision (%)
	$(TP + TN) / Total$	$TP / (TP + FN)$	$TP / (TP + FP)$
1	37,04	32	100
2	70,37	68	100
3	88,89	88	100
4	88,89	88	100
5	85,19	84	100
6	79,63	78	100
7	77,78	76	100
8	77,78	76	100
9	70,37	68	100
10	70,37	68	100
11	70,37	68	100
12	68,52	68	97,14



Gambar 4.12: Grafik hasil pengujian sistem pada *threshold* = 75

Hasil pengujian menggunakan nilai *threshold* 75 menunjukkan nilai akurasi sistem terbaik sebesar 88,89% yaitu pada nilai $r = 3$ dan $r = 4$ dengan nilai *precision* sebesar 100% dan nilai *recall* sebesar 88%, dimana sebanyak 44 kasus stroke didiagnosis dengan benar dan memenuhi *threshold*, 3 kasus *stroke* didiagnosis benar tetapi berada dibawah nilai *threshold*, 3 kasus *stroke* didiagnosis salah dan memenuhi *threshold* dan 4 kasus bukan penyakit *stroke* yang didiagnosis benar bukan penyakit *stroke*.

f. Hasil pengujian dengan nilai *threshold* = 80

Tabel 4.11: Rincian Hasil Pengujian pada *Threshold* = 80

Nilai R	1	2	3	4	5	6	7	8	9	10	11	12
TP	11	19	29	33	35	36	38	41	37	37	37	38
FN	39	31	21	17	15	14	12	9	13	13	13	12
TN	4	4	4	4	4	4	4	4	4	4	4	4
FP	0	0	0	0	0	0	0	0	0	0	0	0
Total Data Uji	54	54	54	54	54	54	54	54	54	54	54	54

Keterangan:

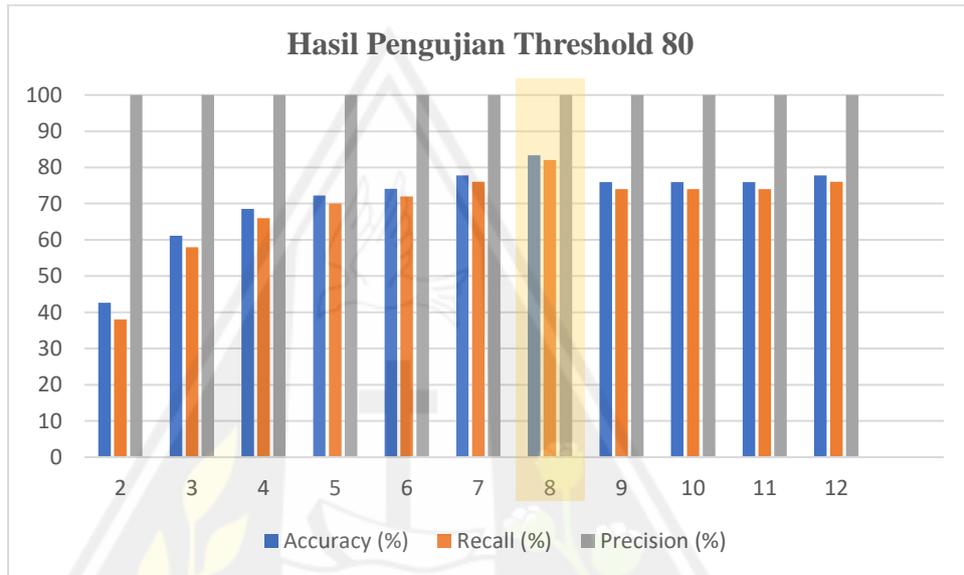
TP : *True positive*, FP : *False positive*, FN : *False negative*, TN: *True negative*

Dari hasil pengujian tersebut, kemudian dihitung nilai akurasi menggunakan persamaan 2.1, *precision* menggunakan persamaan 2.2, dan *recall* atau *sensitivitas* menggunakan persamaan 2.3. Perhitungan akurasi, *precision*, dan *recall* pada nilai *threshold* 80 disajikan pada Tabel 4.12.

Tabel 4.12: Perhitungan Nilai *Accuracy*, *Recall*, dan *precision* pada *Threshold* = 80

Nilai R	Accuracy (%)	Recall (%)	Precision (%)
	(TP + TN) / Total	TP / (TP + FN)	TP / (TP + FP)
1	27,78	22	100
2	42,59	38	100
3	61,11	58	100
4	68,52	66	100
5	72,22	70	100
6	74,07	72	100

7	77,78	76	100
8	83,33	82	100
9	75,93	74	100
10	75,93	74	100
11	75,93	74	100
12	77,78	76	100



Gambar 4.13: Grafik hasil pengujian sistem pada *threshold* = 80

Hasil pengujian menggunakan nilai *threshold* 80 menunjukkan nilai akurasi sistem tertinggi sebesar 83,33% yaitu pada nilai $r = 8$ dengan nilai *precision* sebesar 100% dan nilai *recall* sebesar 82%, dimana sebanyak 41 kasus *stroke* didiagnosis dengan benar dan memenuhi *threshold*, 6 kasus *stroke* didiagnosis benar tetapi berada dibawah nilai *threshold*, 3 kasus didiagnosis salah dan memenuhi nilai *threshold*, dan 4 kasus bukan penyakit *stroke* yang didiagnosis benar bukan penyakit *stroke*.

g. Hasil pengujian dengan nilai *threshold* = 85

Tabel 4.13: Rincian Hasil Pengujian pada *Threshold* = 85

Nilai R	1	2	3	4	5	6	7	8	9	10	11	12
TP	11	13	17	20	22	23	23	25	27	28	30	30
FN	39	37	33	30	28	27	27	25	23	22	20	20
TN	4	4	4	4	4	4	4	4	4	4	4	4
FP	0	0	0	0	0	0	0	0	0	0	0	0
Total Data Uji	54	54	54	54	54	54	54	54	54	54	54	54

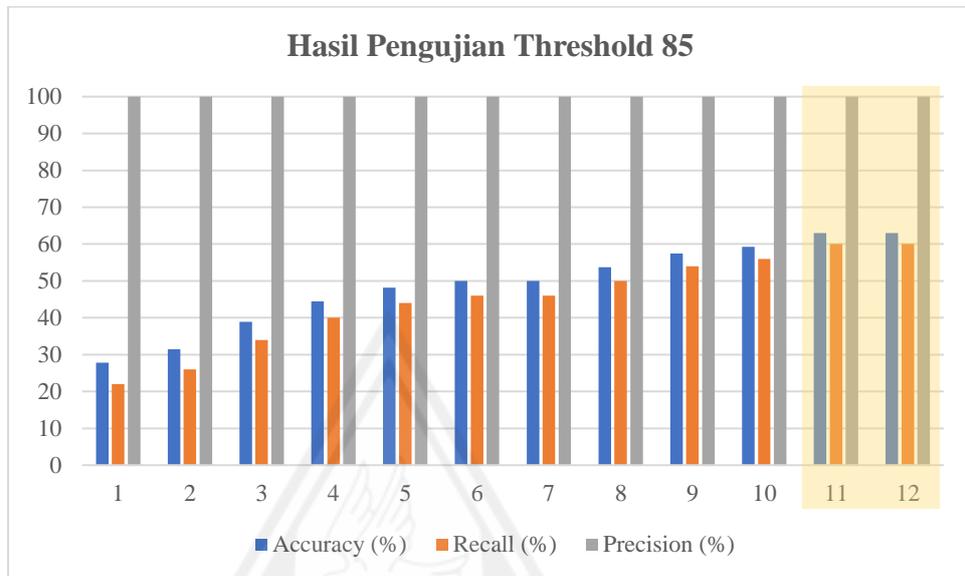
Keterangan:

TP : *True positive*, FP : *False positive*, FN : *False negative*, TN: *True negative*

Dari hasil pengujian tersebut, kemudian dihitung nilai akurasi menggunakan persamaan 2.1, *precision* menggunakan persamaan 2.2, dan *recall* atau *sensitivitas* menggunakan persamaan 2.3. Perhitungan akurasi, *precision*, dan *recall* pada nilai *threshold* 85 disajikan pada Tabel 4.14 .

Tabel 4.14: Perhitungan Nilai *Accuracy*, *Recall*, dan *precision* pada *Threshold* = 85

Nilai R	Accuracy (%)	Recall (%)	Precision (%)
	$(TP + TN) / \text{Total}$	$TP / (TP + FN)$	$TP / (TP + FP)$
1	27,78	22	100
2	31,48	26	100
3	38,89	34	100
4	44,44	40	100
5	48,15	44	100
6	50	46	100
7	50	46	100
8	53,70	50	100
9	57,41	54	100
10	59,26	56	100
11	62,96	60	100
12	62,96	60	100



Gambar 4.14: Grafik hasil pengujian sistem pada *threshold* = 85

Hasil pengujian menggunakan nilai *threshold* 85 menunjukkan nilai akurasi sistem yang cukup rendah yaitu sebesar 62,96% pada nilai $r = 11$ dan $r = 12$ dengan nilai *precision* sebesar 100% dan nilai *recall* sebesar 60%, dimana sebanyak 30 kasus stroke didiagnosis dengan benar dan memenuhi *threshold*, 17 kasus stroke didiagnosis benar tetapi berada dibawah nilai *threshold*, 3 kasus stroke didiagnosis salah dan berada dibawah nilai *threshold*, dan 4 kasus bukan penyakit stroke yang didiagnosis benar bukan penyakit stroke.

h. Hasil pengujian dengan nilai *threshold* = 90

Tabel 4.15: Rincian Hasil Pengujian pada *Threshold* = 90

Nilai R	1	2	3	4	5	6	7	8	9	10	11	12
TP	11	13	15	15	15	15	15	15	16	16	16	17
FN	39	37	35	35	35	35	35	35	34	34	34	33
TN	4	4	4	4	4	4	4	4	4	4	4	4
FP	0	0	0	0	0	0	0	0	0	0	0	0
Total Data Uji	54	54	54	54	54	54	54	54	54	54	54	54

Keterangan:

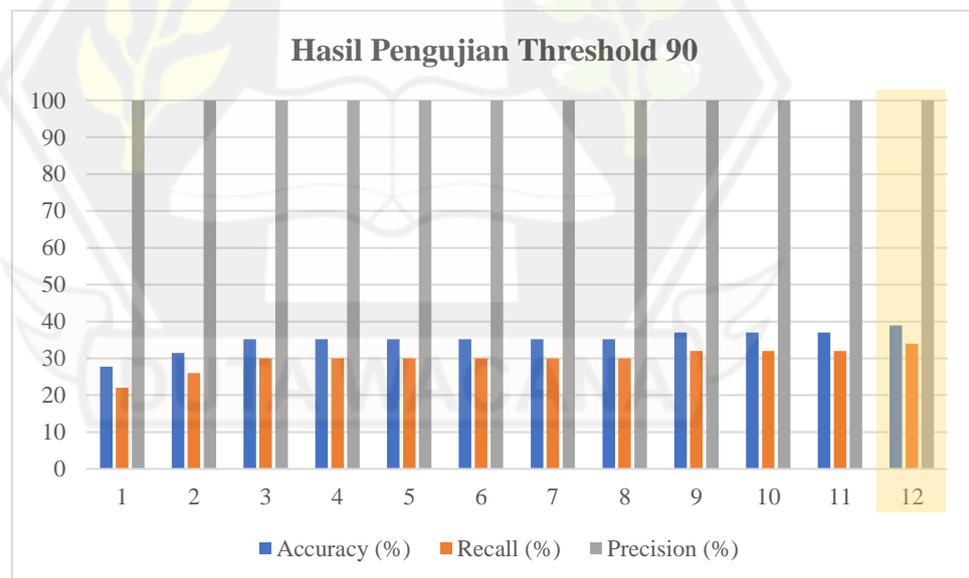
TP : *True positive*, FP : *False positive*, FN : *False negative*, TN: *True negative*

Dari hasil pengujian tersebut, kemudian dihitung nilai akurasi menggunakan persamaan 2.1, *precision* menggunakan persamaan 2.2, dan

recall atau *sensitivitas* menggunakan persamaan 2.3. Perhitungan akurasi, *precision*, dan *recall* pada nilai *threshold* 90 disajikan pada Tabel 4.16.

Tabel 4.16: Perhitungan Nilai *Accuracy*, *Recall*, dan *precision* pada *Threshold* = 90

Nilai R	Accuracy (%)	Recall (%)	Precision (%)
	$(TP + TN) / Total$	$TP / (TP + FN)$	$TP / (TP + FP)$
1	27,78	22	100
2	31,48	26	100
3	35,19	30	100
4	35,19	30	100
5	35,19	30	100
6	35,19	30	100
7	35,19	30	100
8	35,19	30	100
9	37,04	32	100
10	37,04	32	100
11	37,04	32	100
12	38,89	34	100



Gambar 4.15: Grafik hasil pengujian sistem pada *threshold* = 90

Hasil pengujian menggunakan nilai *threshold* 90 menunjukkan nilai akurasi sistem yang sangat rendah, nilai akurasi hanya dapat mencapai 38,89% yaitu pada nilai $r = 12$ dengan nilai *precision* sebesar 100% dan

nilai *recall* sebesar 34%, dimana sebanyak 17 kasus *stroke* didiagnosis dengan benar dan memenuhi *threshold*, 32 kasus *stroke* didiagnosis benar tetapi berada dibawah *threshold*, 1 kasus *stroke* didiagnosis salah dan memenuhi *threshold*, dan 4 kasus bukan penyakit *stroke* yang didiagnosis benar bukan penyakit *stroke*.

i. Hasil pengujian dengan nilai *threshold* = 95

Tabel 4.17: Rincian Hasil Pengujian pada *Threshold* = 95

Nilai R	1	2	3	4	5	6	7	8	9	10	11	12
TP	11	13	15	15	15	15	15	15	16	16	16	16
FN	39	37	35	35	35	35	35	35	34	34	34	34
TN	4	4	4	4	4	4	4	4	4	4	4	4
FP	0	0	0	0	0	0	0	0	0	0	0	0
Total Data Uji	54	54	54	54	54	54	54	54	54	54	54	54

Keterangan:

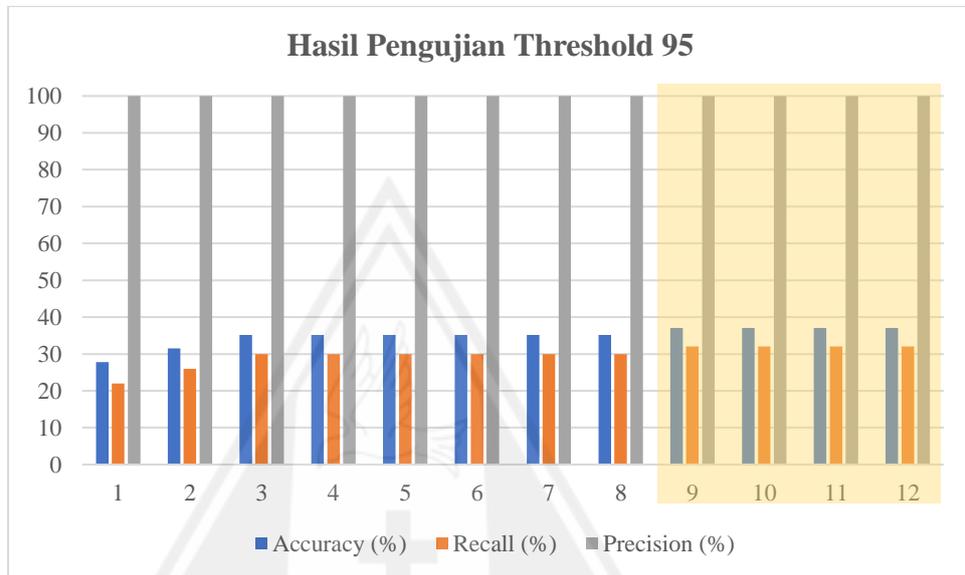
TP : *True positive*, FP : *False positive*, FN : *False negative*, TN: *True negative*

Dari hasil pengujian tersebut, kemudian dihitung nilai akurasi menggunakan persamaan 2.1, *precision* menggunakan persamaan 2.2, dan *recall* atau *sensitivitas* menggunakan persamaan 2.3. Perhitungan akurasi, *precision*, dan *recall* pada nilai *threshold* 90 disajikan pada Tabel 4.18.

Tabel 4.18: Perhitungan Nilai *Accuracy*, *Recall*, dan *precision* pada *Threshold* = 95

Nilai R	Accuracy (%)	Recall (%)	Precision (%)
	$(TP + TN) / Total$	$TP / (TP + FN)$	$TP / (TP + FP)$
1	27,78	22	100
2	31,48	26	100
3	35,19	30	100
4	35,19	30	100
5	35,19	30	100
6	35,19	30	100
7	35,19	30	100
8	35,19	30	100
9	37,04	32	100
10	37,04	32	100

11	37,04	32	100
12	38,89	34	100



Gambar 4.16: Grafik hasil pengujian sistem pada *threshold* = 95

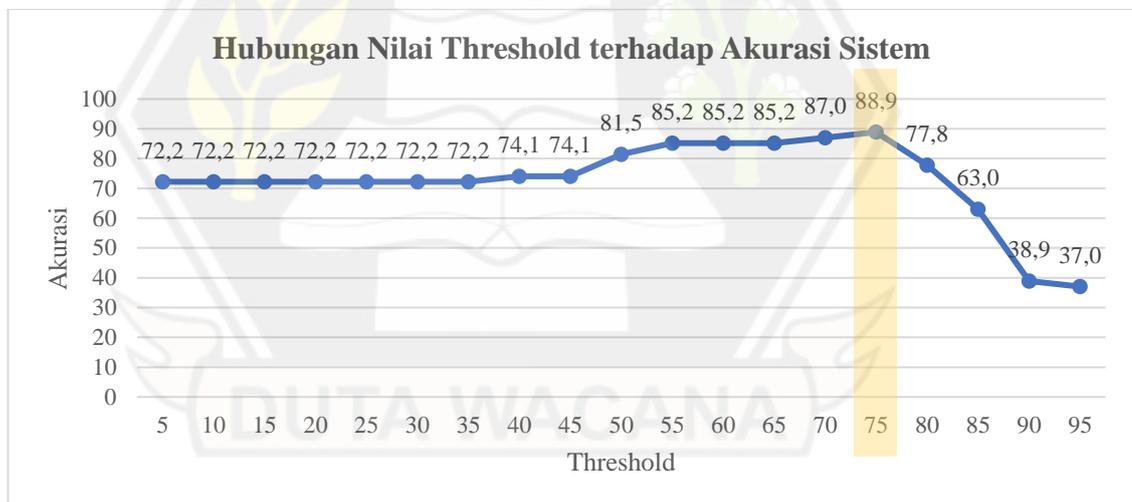
Hasil pengujian menggunakan nilai *threshold* 95 menunjukkan nilai akurasi sistem yang sangat rendah, nilai akurasi hanya dapat mencapai 38,89% yaitu pada nilai $r = 9$ sampai $r = 12$ dengan nilai *precision* sebesar 100% dan nilai *recall* terbesar 32%, dimana sebanyak 16 kasus *stroke* didiagnosis dengan benar dan memenuhi *threshold*, 34 kasus *stroke* didiagnosis benar tetapi berada dibawah *threshold* dan 4 kasus bukan penyakit *stroke* yang didiagnosis benar bukan penyakit *stroke*.

Setelah pengujian dilakukan, kemudian diambil hasil pengujian sistem dengan nilai r yang memiliki tingkat akurasi terbaik pada setiap nilai *threshold* yang dapat dilihat pada Tabel 4.19. Perbandingan hasil akurasi, sensitivitas, dan *recall* terbaik pada setiap nilai *threshold* disajikan dalam bentuk grafik pada Gambar 4.20.

Tabel 4.19: Nilai Akurasi Tertinggi pada Setiap *Threshold*

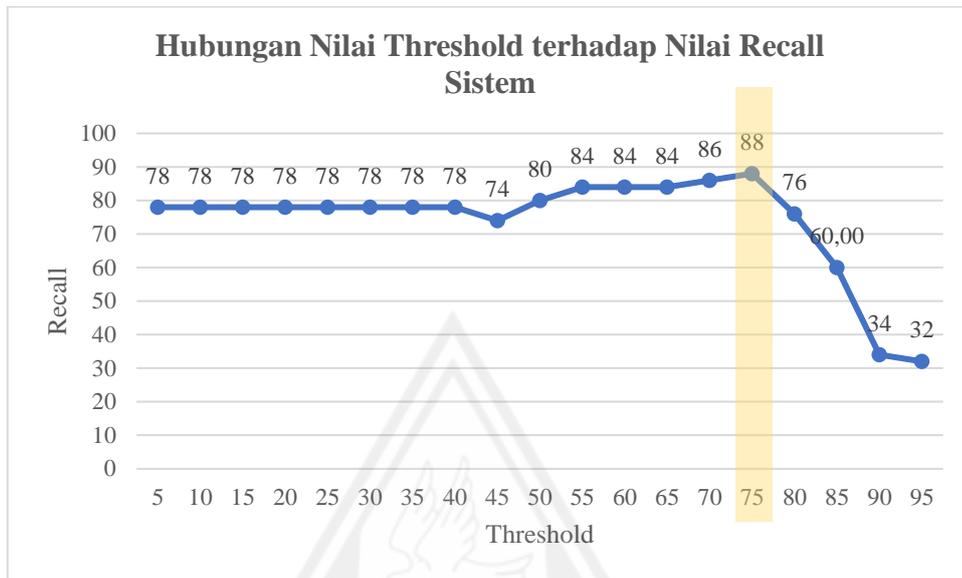
Threshold	Nilai R	Accuracy (%)	Recall (%)	Precision (%)
5	1	72,22	78	90,7
10	1	72,22	78	90,7
15	1	72,22	78	90,7

20	1	72,22	78	90,7
25	1	72,22	78	90,7
30	1	72,22	78	90,7
35	1	72,22	78	90,7
40	1	74,07	78	92,86
45	1	74,07	74	97,37
50	1	81,48	80	100
55	1	85,19	84	100
60	1	85,19	84	100
65	2	85,19	84	100
70	2	87,04	86	100
75	3 & 4	88,89	88	100
80	8	77,78	76	100
85	12	62,96	60	100
90	12	38,89	34	100
95	12	37,04	32	100



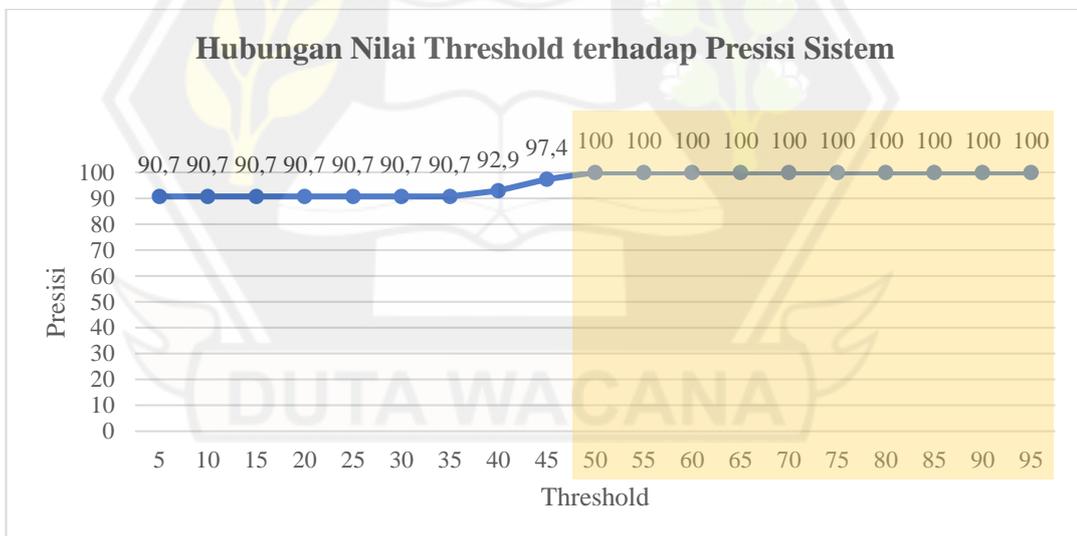
Gambar 4.17: Grafik tingkat akurasi pada setiap nilai *threshold*

Grafik 4.17 menunjukkan nilai akurasi tertinggi dicapai pada penerapan nilai *threshold* 75 dengan $r = 3$ dan $r = 4$ yaitu sebesar 88,89%, kemudian secara berurutan disusul oleh nilai *threshold* 70 sebesar 87,04%, *threshold* 65 dan 60 sebesar 85,19%, sedangkan akurasi terendah pada penerapan nilai *threshold* 95%.



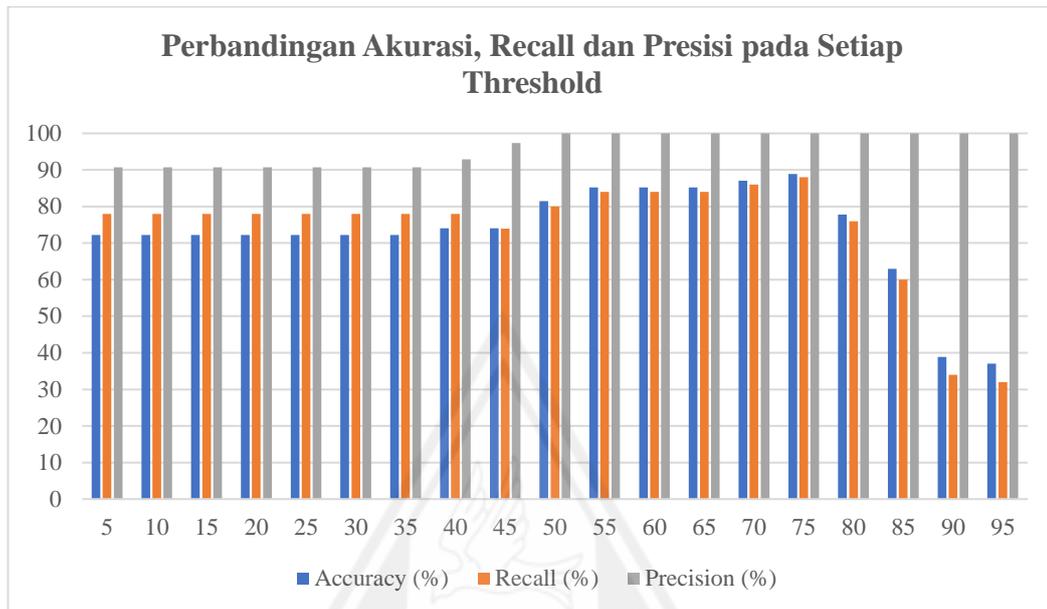
Gambar 4.18: Grafik nilai *recall* pada setiap nilai *threshold*

Grafik 4.18 menunjukkan nilai *recall* tertinggi dicapai pada nilai *threshold* 75 yaitu sebesar 88%, kemudian secara berurutan disusul oleh nilai *threshold* 70 sebesar 86%, *threshold* 65 dan 60 sebesar 84%, sedangkan nilai *recall* terendah pada *threshold* 95.



Gambar 4.19: Grafik tingkat presisi pada setiap nilai *threshold*

Grafik 4.19 menunjukkan nilai *precision* tertinggi dicapai pada nilai *threshold* $\geq 50\%$ sebesar 100%. Dari hasil pengujian, dapat dilihat bahwa semakin tinggi nilai *threshold*, maka semakin tinggi nilai *precision* dari sistem.



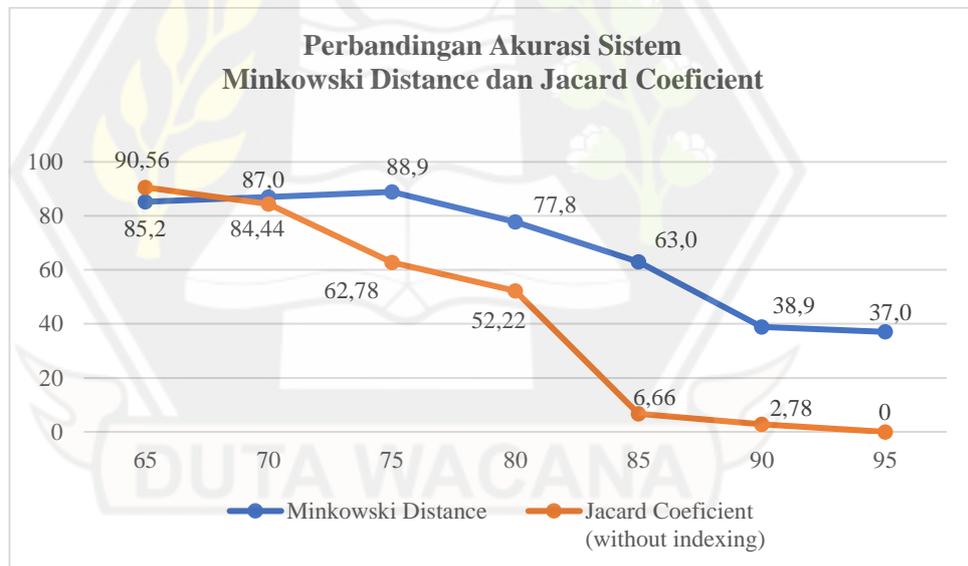
Gambar 4.20: Grafik perbandingan nilai akurasi, *recall* dan presisi pada setiap nilai *threshold*

4.4 Pembahasan

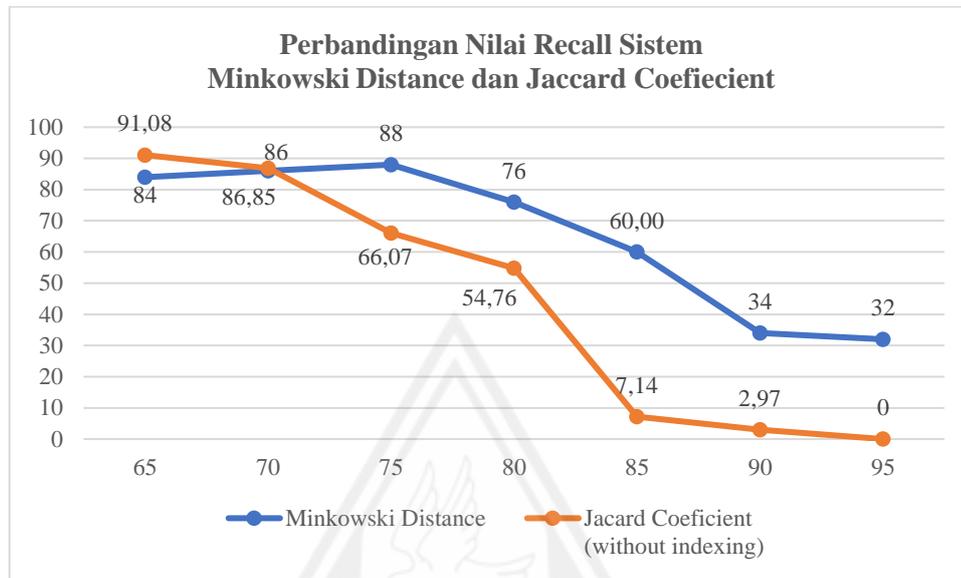
Pengujian sistem menggunakan *confusion matrix* menunjukkan tingkat akurasi, sensitivitas, dan *recall* yang berbeda-beda pada setiap nilai *threshold* dan nilai *r*. Tingkat akurasi tertinggi dicapai pada penerapan nilai *threshold* 75 dengan nilai *r* = 3 dan nilai *r* = 4 sebesar 88,89%. Nilai akurasi menunjukkan seberapa baik sistem dapat mendiagnosis dengan benar, sehingga semakin tinggi nilai akurasi maka semakin akurat hasil diagnosis atau solusi yang diberikan oleh sistem. Akurasi menghitung semua nilai prediksi sebenarnya, tetapi tidak spesifik untuk setiap label yang ada, oleh karena itu akurasi yang lebih tinggi tidak berarti bahwa sistem memiliki kinerja yang baik dalam memprediksi label tertentu sehingga nilai *recall* dan *precision* perlu dihitung. Nilai *recall* digunakan untuk menghitung tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi, sehingga semakin tinggi nilai *recall* maka semakin baik sistem dalam mengidentifikasi kasus positif. Nilai *recall* terbaik dicapai pada penerapan nilai *threshold* 75 dengan nilai *r* = 3 dan *r* = 4 yaitu sebesar 88%. *Precision* adalah metrik yang mengukur keakuratan prediksi positif dimana semakin tinggi nilai *precision* maka semakin

akurat prediksi positif. Nilai *precision* tertinggi dicapai pada nilai *threshold* ≥ 50 yang mencapai 100%.

Berdasarkan hasil pengujian dari sistem yang telah dilakukan, penerapan nilai *threshold* similaritas yang cocok untuk sistem CBR dalam mendiagnosis penyakit *stroke* adalah nilai *threshold* similaritas 75 dan nilai $r = 3$ atau $r = 4$ karena menghasilkan nilai akurasi dan *recall* yang paling baik dibanding nilai *threshold* similaritas lainnya dan memiliki nilai *sensitivitas* 100%. Dalam sistem untuk deteksi penyakit berisiko tinggi (seperti *stroke*), *recall* merupakan metrik evaluasi yang lebih penting daripada *precision*, karena nilai *recall* yang rendah berarti sistem mendiagnosis beberapa pasien yang benar-benar memiliki penyakit *stroke* sebagai pasien yang tidak sakit, dimana hal ini sangat membahayakan nyawa pasien. Oleh karena itu, penerapan nilai *threshold* dan nilai r yang paling optimal untuk sistem CBR diagnosis penyakit *stroke* adalah nilai *threshold* 75 pada nilai $r = 3$ atau $r = 4$ dengan tingkat akurasi sistem sebesar 88,89%, *recall* sebesar 88% dan *precision* sebesar 100%.



Gambar 4.21. Perbandingan tingkat akurasi antara *Minkowski Distance* dan *Jacard Coefficient* (Nelson et al., 2018)



Gambar 4.22. Perbandingan tingkat *recall* antara *Minkowski Distance* dan *Jaccard Coeficient* (Nelson et al., 2018)

Gambar 4.21 menunjukkan tingkat akurasi sistem diagnosis penyakit stroke dengan metode *Jaccard Coeficient* tertinggi sebesar 90,56% tanpa indexing, tetapi menunjukkan banyak kasus yang tidak relevan sehingga *threshold* 0,7 lebih cocok untuk diterapkan dengan sensitivitas (89,88%) dan akurasi (84,44% untuk CBR menggunakan pengindeksan dan 87,78% untuk CBR tanpa pengindeksan) (Nelson et al., 2018). Sistem CBR menggunakan metode *minkowski distance similarity* memiliki nilai akurasi dan *recall* yang lebih tinggi daripada sistem CBR menggunakan *Jaccard Coeficient*.

BAB V

KESIMPULAN DAN SARAN

4.5 Kesimpulan

Berdasarkan penelitian dan hasil pengujian sistem yang telah dilakukan, maka dapat disimpulkan bahwa :

1. Penelitian ini menghasilkan sebuah sistem penalaran berbasis kasus (CBR) untuk mendiagnosis penyakit *stroke* dengan menerapkan metode *minkowski distance similarity* untuk perhitungan similaritas antar kasus.
2. Sistem diagnosis penyakit *stroke* menggunakan metode *minkowski distance similarity* pada CBR yang dikembangkan mampu mendapatkan performa sistem yang terbaik dilihat dari tingkat akurasi tertinggi (88,89%), nilai *recall* tertinggi (88%), dan *precision* yang tinggi (100%) yaitu ketika menggunakan nilai *threshold* 75 dengan nilai $r = 3$ atau $r = 4$.
3. Sistem CBR untuk mendiagnosis penyakit *stroke* dengan menggunakan metode *Minkowski Distance Similarity* menghasilkan tingkat akurasi dan sensitifitas atau *recall* yang lebih tinggi daripada sistem yang menggunakan metode *Jacard Coeficient*.

4.6 Saran

Beberapa saran yang dapat diberikan untuk pengembangan lebih lanjut dari sistem CBR untuk diagnosis penyakit *stroke* ini, antara lain:

1. Perlu menambah kasus yang digunakan sebagai basis pengetahuan, karena semakin banyak basis kasus, maka hasil diagnosis akan semakin akurat.
2. Perlu untuk menambah pemeriksaan penunjang seperti pemeriksaan *CT scan* atau MRI dan pemeriksaan saraf kranial untuk menambah keakuratan dalam mendiagnosis jenis *stroke*.

DAFTAR PUSTAKA

- Adawiyah, R. (2020). IMPLEMENTASI METODE MINKOWSKY DISTANCE UNTUK DETEKSI KELAHIRAN BAYI PREMATUR BERBASIS CASE BASED REASONING. *Jurnal Informatika Dan Komputer) Akreditasi KEMENRISTEKDIKTI*, 3(1). <https://doi.org/10.33387/jiko>
- American Stroke Association. (2022). *American Stroke Association*. <https://www.stroke.org/>.
- Anggreany, M. S. (n.d.). *Confusion Matrix*. BINA NUSANTARA School of Computer Science.
- Arifa, A. B., & Fitriana, G. F. (2020). DIAGNOSIS PENYAKIT HIPERTENSI MENGGUNAKAN METODE CASE-BASED REASONING HYPERTENSION DIAGNOSIS USING CASE-BASED REASONING. In *IJIS Indonesian Journal on Information System* (Vol. 5).
- Centers for Disease Control and Prevention Organization. (2022, April 12). *Know Your Risk for Stroke*. https://www.cdc.gov/stroke/risk_factors.htm.
- Dou, J., Chang, K.-T., Chen, S., Yunus, A., Liu, J.-K., Xia, H., & Zhu, Z. (2015). Automatic Case-Based Reasoning Approach for Landslide Detection: Integration of Object-Oriented Image Analysis and a Genetic Algorithm. *Remote Sensing*, 7(4), 4318–4342. <https://doi.org/10.3390/rs70404318>
- Faizal, E., & Hamdani, H. (2018). Weighted Minkowski Similarity Method with CBR for Diagnosing Cardiovascular Disease. *International Journal of Advanced Computer Science and Applications*, 9(12). <https://doi.org/10.14569/IJACSA.2018.091244>
- Gunawan, A., Suhery, C., & Rismawan, T. (2021). Implementasi Metode Case-Based Reasoning Dan Similarity Jaccard Coefficient Dalam Identifikasi Kerusakan Laptop. *Jurnal Komputer Dan Aplikasi*, 09, 292–305.
- Jha, M. K., Pakhira, D., & Chakraborty, B. (2013). Diabetes Detection and Care Applying CBR Techniques . *IJSCE*, 2(6).
- Luh Putu Merawati, N., Hartati, S., Informatika, T., Komputer, I., Bumigora, S., Gadjah Mada, U., Ismail Marzuki Mataram, J., & Utara Bulaksumur

- Yogyakarta, S. (2018). SISTEM REKOMENDASI TOPIK SKRIPSI MENGGUNAKAN METODE CASE BASED REASONING. In *Jurnal Ilmiah Teknologi Informasi Terapan: Vol. IV* (Issue 3).
- Mankad, K. B. (2015). An Expert System Design to Categorize Multiple Intelligence of Students. *IUP Journal of Computer Sciences, IX*.
- Mendes, J. B., & Ramos, A. (2012). A case-based reasoning framework for knowledge structuring: Structuring the domain module in intelligent tutoring systems. *2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE)*, 427–430.
<https://doi.org/10.1109/CSAE.2012.6272986>
- Mubarak, A., Salmin, M., Fuad, A., & Abdullah, S. Do. (2022). Penalaran Berbasis Kasus Untuk Diagnosis Penyakit Malaria Dengan Menggunakan Metode Minkowsky Distance. *Jurnal Ilmiah ILKOMINFO - Ilmu Komputer & Informatika, 5*(1). <https://doi.org/10.47324/ilkominfo.v4i3.136>
- Nelson, R., Harjoko, A., & Musdholifah, A. (2018). Case-Based Reasoning for Stroke Disease Diagnosis. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems), 12*(1), 33. <https://doi.org/10.22146/ijccs.26331>
- Nelson R, Harjoko, A., & Musdholifah, A. (2018). *Rekam Medik Stroke.xlsx*.
- Normawati, D., & Prayogi, S. A. (2021). Implementasi Naïve Bayes Classifier Dan Confusion Matrix Pada Analisis Sentimen Berbasis Teks Pada Twitter. In *Jurnal Sains Komputer & Informatika (J-SAKTI)* (Vol. 5, Issue 2).
- Noviyanti, R. D. (2013). *FAKTOR RISIKO PENYEBAB MENINGKATNYA KEJADIAN STROKE PADA USIA REMAJA DAN USIA PRODUKTIF. 10*.
- Nurdiansyah, Y., & Hartati, S. (2014). *Case-Based Reasoning untuk Pendukung Diagnosa Gangguan pada Anak Autis*.
- P2PTM Kemenkes RI. (2018, July 4). *Apa itu Stroke*.
[Http://P2ptm.Kemkes.Go.Id/Infographic-P2ptm/Stroke/Apa-Itu-Stroke](http://P2ptm.Kemkes.Go.Id/Infographic-P2ptm/Stroke/Apa-Itu-Stroke).
- Pratiwi, H. (2019). *Buku Ajar Sistem Pakar* (P. Heny, Ed.). Goresan Pena.
- Rizky, R., Wibowo, A. H., Hakim, Z., & Sujai, L. (2020). Sistem Pakar Diagnosis Kerusakan Jaringan Local Area Network (LAN) Menggunakan Metode

- Forward Chaining. *JURNAL TEKNIK INFORMATIKA UNIS*, 7(2), 145–152.
<https://doi.org/10.33592/jutis.v7i2.396>
- Salmin, M., & Hartati, S. (2018). Case Based Reasoning untuk Diagnosis Penyakit Infeksi Saluran Pernapasan Akut. *JIKO (Jurnal Informatika Dan Komputer)*, 1(1), 21–26. <https://doi.org/10.33387/jiko.v1i1.1167>
- Subbotin, I. Ya., & Voskoglou, M. Gr. (2012). *Applications of fuzzy logic to Case-Based Reasoning*. 11, 7–18.
<https://www.researchgate.net/publication/223129950>
- Tempola, F., Musdholifah, A., & Hartati, S. (2015). *Case Based Reasoning Untuk Penentuan Kelayakan Mahasiswa Penerima Beasiswa*.
- Via, Y. V., Anggraeny, F. T., & Jorgie, R. A. (2021). PENERAPAN ALGORITMA CASE BASED REASONING DAN K-NEAREST NEIGHBOR UNTUK DIAGNOSA PENYAKIT AYAM. *Prosiding Seminar Nasional Informatika Bela Negara*, 2, 192–195.
<https://doi.org/10.33005/santika.v2i0.140>
- Wahyudi, E., & Pradasari, N. I. (2018). Case Based Reasoning untuk Diagnosis Penyakit Jantung Menggunakan Metode Minkowski Distance. *INTECOMS: Journal of Information Technology and Computer Science*, 1(1), 119–123.
<https://doi.org/10.31539/intecom.v1i1.170>
- Warman, I. (2017). Sistem Pakar Identifikasi Penyakit Tanaman Padi Menggunakan Case-Based Reasoning. In *Seminar Nasional Aplikasi Teknologi Informasi (SNATi)*.
- World Health Organization. (2022). *Stroke, Cerebrovascular accident*.
- Xiao, X., Skitmore, M., & Hu, X. (2017). Case-based Reasoning and Text Mining for Green Building Decision Making. *Energy Procedia*, 111, 417–425.
<https://doi.org/10.1016/j.egypro.2017.03.203>