

SISTEM PAKAR DETEKSI DINI PENYAKIT STROKE MENGUNAKAN METODE NAÏVE BAYES – *CERTAINTY FACTOR*

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Renaldy Senna Utama

NIM: 135150201111196



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

SISTEM PAKAR DETEKSI DINI PENYAKIT STROKE MENGGUNAKAN METODE NAÏVE
BAYES-CERTAINTY FACTOR

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Renaldy Senna Utama
NIM: 135150201111196

Skripsi ini telah diuji dan dinyatakan lulus pada
12 Januari 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I



Nurul Hidayat, S.Pd, M.Sc
NIP. 19680430 200212 1 001

Dosen Pembimbing II



Edy Santoso, S.Si, M.Kom
NIP. 19740414 200312 1 004

Mengetahui
Ketua Jurusan Teknik Informatika



Tri Astoto Kurniawan, S.T, M.T, Ph.D
NIP. 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 12 Januari 2018



Renaldy Senna Utama

NIM: 135150201111196



KATA PENGANTAR

Dengan memanjatkan puji syukur kehadirat Allah SWT, atas limpahan rahmat dan hidayah-Nya serta Nabi Muhammad SAW yang telah memberikan syafaat kepada umatnya sehingga penulis dapat menyajikan Laporan Skripsi yang berjudul **SISTEM PAKAR DETEKSI DINI PENYAKIT STROKE MENGGUNAKAN METODE NAÏVE BAYES-CERTAINTY FACTOR**. Skripsi ini diajukan untuk memenuhi persyaratan memperoleh gelas Sarjana Komputer pada Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya. Saya mengucapkan terima kasih kepada.

1. Nurul Hidayat, S.Pd., M.Sc., selaku dosen pembimbing skripsi pertama penulis yang telah meluangkan waktu untuk memberikan arahan dan bimbingan bagi penulis.
2. Edy Santoso, S.Si., M.Kom., selaku dosen pembimbing skripsi kedua penulis yang telah meluangkan waktu dan juga arahan bagi penulis.
3. Seluruh bapak dan ibu dosen yang telah memberikan ilmunya selama penulis menempuh pendidikan di Fakultas Ilmu Komputer.
4. Orang tua penulis, Bapak Hamdan dan Ibu Ratna yang telah memberikan dukungan moril maupun materil kepada penulis selama penulisan skripsi. Penulis mengucapkan banyak terima kasih karena telah mendidik dan memberikan kasih sayang secara tulus sebagai dukungan dalam proses menyelesaikan skripsi.
5. Teman-teman seperjuangan kontrakan. Khabib, Risky, Radit, Sanjaya, Endar yang telah menemani dan mendengarkan keluh kesah penulis selama mengerjakan skripsi.
6. Seluruh teman-teman TIF angkatan 2013 yang tidak bisa disebutkan semuanya. Terima kasih atas semangat dan dukungan dari kalian.

Malang, 12 Januari 2018

Penulis

sennahutama@gmail.com

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xii
DAFTAR KODE PROGRAM	xiii
DAFTAR LAMPIRAN	xiv
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan	3
1.4 Manfaat.....	3
1.5 Batasan masalah	3
1.6 Sistematika pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Kajian Pustaka	5
2.2 Sistem Pakar.....	6
2.2.1 Konsep Dasar Sistem Pakar	7
2.2.2 Tujuan Sistem Pakar	8
2.2.3 Bentuk Sistem Pakar	9
2.2.4 Ciri-ciri Sistem Pakar	9
2.2.5 Keuntungan Sistem Pakar	9
2.2.6 Kelemahan Sistem Pakar.....	10
2.2.7 Struktur Sistem Pakar.....	10
2.2.8 Konsep Sistem Pakar	11
2.3 Metode <i>Naïve Bayes Classifier</i>	13
2.4 <i>Certainty Factor</i>	13
2.5 <i>Naïve Bayes-Certainty Factor</i>	15

2.6 Sistem Operasi Android	16
2.6.1 Komponen Aplikasi Android.....	16
2.7 Eclipse, Java Development Kit, Android SDK	17
2.7.1 <i>Eclipse</i>	17
2.7.2 <i>Java Development kit</i>	17
2.7.3 <i>Android SDK</i>	18
2.8 Penyakit Stroke	18
2.8.1 Definisi.....	18
2.8.2 Jenis-jenis	18
2.8.3 Faktor Penyebab Stroke	19
BAB 3 METODOLOGI	22
3.1 Studi Literatur	22
3.2 Pengumpulan Data	22
3.3 Analisis dan Perancangan	23
3.3.1 Diagram <i>Flow Chart</i> Sistem.....	24
3.4 Implementasi	24
3.5 Pengujian	24
3.6 Pengambilan Keputusan	25
BAB 4 PERANCANGAN.....	26
4.1 Analisis Kebutuhan Perangkat Lunak.....	27
4.1.1 Identifikasi Aktor	27
4.1.2 Analisis Kebutuhan Masukkan	27
4.1.3 Analisis Kebutuhan Proses	33
4.1.4 Analisis Kebutuhan Keluaran	33
4.2 Perancangan Perangkat Lunak	34
4.2.1 <i>Use Case Diagram</i>	34
4.2.2 <i>Use Case</i> Skenario	34
4.2.3 <i>Sequence Diagram</i>	37
4.3 Perancangan Sistem Pakar.....	39
4.3.1 Akuisisi Pengetahuan	40
4.3.2 Basis Pengetahuan	41
4.3.3 Mesin Inferensi.....	43

4.3.4 <i>Blackboard</i>	57
4.3.5 Fasilitas Penjelas	57
4.3.6 Antarmuka Pengguna.....	57
4.4 Perancangan Pengujian	60
4.4.1 Pengujian Blackbox	60
4.4.2 Pengujian Usability.....	61
4.4.3 Pengujian Akurasi.....	61
BAB 5 IMPLEMENTASI	61
5.1 Spesifikasi Sistem	62
5.1.1 Spesifikasi Perangkat Keras.....	62
5.1.2 Spesifikasi Perangkat Lunak	63
5.2 Batasan Implementasi	63
5.3 Implementasi Sistem Pakar	64
5.3.1 Implementasi Basis Pengetahuan	64
5.3.2 Implementasi Mesin Inferensi	64
5.3.3 Implementasi Antarmuka	70
BAB 6 PENGUJIAN DAN ANALISIS.....	73
6.1 <i>Blackbox Testing</i>	73
6.1.1 Kasus Uji Validasi.....	73
6.1.2 Hasil Uji Validasi	76
6.1.3 Analisis Pengujian <i>Blackbox</i>	77
6.2 Pengujian <i>Usability</i>	77
6.2.1 Analisis Pengujian <i>Usability</i>	79
6.3 Pengujian Akurasi	79
6.3.1 Analisis Pengujian Akurasi.....	86
BAB 7 PENUTUP	87
7.1 Kesimpulan.....	87
7.2 Saran	87
DAFTAR PUSTAKA.....	88



DAFTAR TABEL

Tabel 2.1 <i>Review</i> Kajian Pustaka	5
Tabel 2.2 Data Faktor Resiko Penyakit Stroke	7
Tabel 2.3 Nilai <i>Evidence</i> Tingkat Keyakinan Pakar	14
Tabel 2.4 Penentuan Diet.....	21
Tabel 3.1 Pengujian Akurasi	25
Tabel 4.1 <i>Identifikasi</i> Aktor	27
Tabel 4.2 Kebutuhan Fungsional.....	27
Tabel 4.3 Faktor-faktor Resiko Penyakit Stroke.....	28
Tabel 4.4 Tekanan Darah Sistolik dan Diastolik	29
Tabel 4.5 Aturan Tingkat Tekanan Darah.....	29
Tabel 4.6 Nilai Skor Tekanan Darah	31
Tabel 4.7 Nilai Skor Merokok	31
Tabel 4.8 Nilai Skor Riwayat Keluarga.....	31
Tabel 4.9 Nilai Skor Aktifitas Fisik	31
Tabel 4.10 Nilai Skor Riwayat Fibrilasi	32
Tabel 4.11 Data Latih	32
Tabel 4.12 Data Uji.....	33
Tabel 4.13 Mengakses Halaman Mulai Deteksi	34
Tabel 4.14 Mengakses Halaman Info Penyakit	35
Tabel 4.15 Mengakses Halaman Info Pengembang	35
Tabel 4.16 Mengakses Halaman Info Perangkat.....	36
Tabel 4.17 Mengakses Halaman Hasil Deteksi.....	36
Tabel 4.18 Jumlah Maing-masing Tingkat Resiko Pada Data Latih	41
Tabel 4.19 Data Latih Tingkat Resiko Terserang Stroke.....	41
Tabel 4.20 Jumlah Jenis Resiko Dari Setiap Tingkat Resiko	42
Tabel 4.21 Bobot Nilai CF Pakar Tingkat Resiko Penyakit Stroke	42
Tabel 4.22 Tabel Perhitungan Tiap Faktor Resiko.....	46
Tabel 4.23 Nilai Prior.....	47
Tabel 4.24 Nilai Perhitungan Likelihood Kelas Stroke Tinggi.....	47
Tabel 4.25 Nilai Perhitungan Likelihood Kelas Stroke Sedang.....	48
Tabel 4.26 Nilai Perhitungan Likelihood Kelas Stroke Rendah	48

Tabel 4.27 Nilai Perhitungan Posterior Kelas Stroke Tinggi.....	49
Tabel 4.28 Nilai Perhitungan Posterior Kelas Stroke Sedang.....	49
Tabel 4.29 Nilai Perhitungan Posterior Kelas Stroke Rendah	50
Tabel 4.30 Nilai Posterior	50
Tabel 4.31 Nilai Posterior Tertinggi.....	51
Tabel 4.32 Nilai CF Pakar Tingkat Resiko Stroke Tinggi.....	51
Tabel 4.33 Nilai CF <i>User</i>	52
Tabel 4.34 Rancangan Pengujian Blackbox	60
Tabel 4.35 Rancangan Pertanyaan Kuisiomer.....	61
Tabel 4.36 Rancangan Pengujian Akurasi	61
Tabel 5.1 Spesifikasi Perangkat Keras	63
Tabel 5.2 Spesifikasi Perangkat Lunak	63
Tabel 6.1 Kasus Uji Validasi Halaman Mulai Deteksi	74
Tabel 6.2 Kasus Uji Validasi Halaman Hasil Deteksi	74
Tabel 6.3 Kasus Uji Validasi Halaman Info Penyakit	75
Tabel 6.4 Kasus Uji Validasi Halaman Info Perangkat	75
Tabel 6.5 Kasus Uji Validasi Halaman Info Pengembang	75
Tabel 6.6 Hasil Uji Validasi	76
Tabel 6.7 Pertanyaan Kuisiomer	78
Tabel 6.8 Hasil Kuisiomer	78
Tabel 6.9 Pengujian Akurasi	80

DAFTAR GAMBAR

Gambar 2.1 Diagram Struktur Sistem Pakar	10
Gambar 3.1 Diagram Metodologi Penelitian	22
Gambar 3.2 Diagram Blog Proses Pengumpulan Data.....	23
Gambar 3.3 Diagram Diagram <i>Flow Chart</i> Sistem	24
Gambar 4.1 Pohon Perancangan	26
Gambar 4.2 <i>Use Case Diagram</i>	34
Gambar 4.3 Sequence Diagram Mengakses Halaman Mulai Deteksi.....	37
Gambar 4.4 Sequence Diagram Mengakses Halaman Info Penyakit.....	38
Gambar 4.5 Sequence Diagram Mengakses Halaman Info Pengembang	38
Gambar 4.6 Sequence Diagram Mengakses Halaman Info Perangkat	39
Gambar 4.7 Sequence Diagram Mengakses Halaman Hasil Deteksi	39
Gambar 4.8 Proses Penelusuran Forward Chaining Tingkat Resiko Stroke.....	43
Gambar 4.9 Diagram alir Metode <i>Naïve Bayes-Certainty Factor</i>	45
Gambar 4.10 <i>Site Map</i> Sistem.....	58
Gambar 4.11 Halaman Menu Utama	58
Gambar 4.12 Halaman Menu Info Penyakit.....	59
Gambar 4.13 Halaman Menu Info Perangkat	59
Gambar 4.14 Halaman Menu Hasil Deteksi	60
Gambar 5.1 Pohon Implementasi	62
Gambar 5.2 Tampilan Antarmuka Halaman Utama.....	70
Gambar 5.3 Tampilan Antarmuka Halaman Info Penyakit	70
Gambar 5.4 Tampilan Antarmuka Halaman Mulai Deteksi	71
Gambar 5.5 Tampilan Antarmuka Halaman Info Perangkat.....	71
Gambar 5.6 Tampilan Antarmuka Halaman Info Pengembang.....	72
Gambar 5.7 Tampilan Antarmuka Halaman Hasil Deteksi.....	72
Gambar 6.1 Pohon Pengujian dan Analisis	73

DAFTAR KODE PROGRAM

Kode Program 5.1 Algoritme <i>Naïve Bayes</i>	64
Kode Program 5.2 Algoritme <i>Certainty Factor</i>	68



DAFTAR LAMPIRAN

LAMPIRAN A DATA LATIH.....	91
LAMPIRAN B DATA UJI	100



BAB 1 PENDAHULUAN

1.1 Latar belakang

Salah satu penyakit yang dapat menyebabkan kematian adalah stroke dimana penyakit ini merupakan penyakit syaraf mematikan yang memasok darah ke otak secara tiba-tiba sehingga mengalami gangguan (Roger et al, 2011). Penyakit ini kerap dijumpai pada usia senja dan menjadi penyebab kematian terbanyak kedua di dunia dan merupakan penyebab kecacatan utama pada usia produktif (Russo et al, 2011). Penyakit ini biasanya muncul ketika usia manusia mulai beranjak tua, namun juga tidak sedikit anak dewasa juga mengidap penyakit ini (Amalia, 2009). Di Indonesia stroke menjadi penyebab kematian nomor tiga terbanyak setelah penyakit jantung dan kanker, setiap 8 dari 100 orang menderita stroke di Indonesia dan dari tujuh orang meninggal satu di antaranya disebabkan oleh stroke. Sebanyak 15,4% dari keseluruhan penyakit di Indonesia yang menyebabkan kematian dan kecacatan (Kemenkes, 2013). Stroke juga menjadi akar penyebab kematian terbanyak kedua di dunia (WHO, 2008).

Dalam beberapa tahun terakhir di Indonesia mengalami peningkatan penyakit stroke cukup tinggi, akan tetapi masih belum ada tindakan ataupun solusi dari pemerintah untuk melakukan proses pencegahan penyakit stroke. Padahal penyakit stroke butuh waktu tidak sedikit dalam melakukan pendeteksian, masalahnya adalah apabila dalam proses penanganannya mengalami penundaan atau keterlambatan bisa mengakibatkan cacat atau bahkan kematian. Penyakit stroke menjadi salah satu penyakit yang harus dengan cepat ditangani, tetapi apabila sudah terserang penyakit stroke bukan berarti penyakit ini tidak bisa disembuhkan, masih ada harapan jika berusaha untuk sembuh. Menurut DR. Heri Aminuddin MD (Dokter Spesialis Bedah Syaraf) memaparkan bahwa "Masih banyak peluang untuk melanjutkan hidup bagi pasien stroke dengan tingkat parahnya stroke tertentu meskipun tingkat kepulihannya tidak bisa 100 persen (Kemenkes, 2013). Dengan adanya permasalahan tersebut maka dibutuhkan sebuah sistem yang dapat mempercepat dan mempermudah dalam mendeteksi resiko untuk mengurangi angka seseorang terserang penyakit stroke. Dan bisa juga dimanfaatkan dalam meringankan tugas seorang dokter dalam mendeteksi penyakit yang diderita oleh pasien.

Penelitian terkait dengan topik yang sudah dilakukan sebelumnya dan sudah dikembangkan sebuah sistem oleh Deby Putri Indraswari (2015) judul "Sistem Pendukung Keputusan Deteksi Dini pada Penyakit Stroke menggunakan metode *Dempster Shafer*". Pada penelitian tersebut dokter spesialis saraf memberikan nilai densitas sebagai bobot yang dipakai untuk proses perhitungan dalam mendiagnosis penyakit stroke. Nilai yang diberikan oleh dokter tersebut masih menghasilkan akurasi yang belum maksimal. Hasil perhitungan akurasi yang diperoleh sebesar 89% dan tingkat ketidakakuratannya sebesar 11% (Indraswari, 2015). Hal ini disebabkan oleh beberapa faktor, misalnya tidak mudahnya bagi

seorang dokter untuk memberikan nilai densitas. Dalam menentukan nilai densitas terjadi beberapa masalah yang dapat menyebabkan nilai densitas tersebut menjadi kurang maksimal. Penelitian selanjutnya yakni oleh Achmad Affan Suprayogi dengan judul “Sistem Pakar Diagnosis Penyakit Kucing Menggunakan Metode *Naïve Bayes-Certainty factor* Berbasis Android”. Pada penelitian tersebut menghasilkan tingkat akurasi sebesar 80% (Affan, 2017).

Sistem pakar merupakan pengetahuan, fakta, dan teknik penalaran dalam memecahkan masalah berbasis komputer yang hanya dapat dipecahkan oleh seorang pakar dalam bidangnya, namun terdapat beberapa unsur ketidakpastian dan kesamaran. Untuk mengatasi ketidakpastian dan kesamaran tersebut metode *Naïve Bayes* dan *Certainty factor* bisa memberikan solusinya, *Naïve Bayes* merupakan teknik probabilitas hipotesis yang mampu menangani masalah ketidakpastian dengan menggunakan konsep probabilitas hipotesis dan *evidence* (Hardika, 2014), metode ini digunakan karena dalam pendeteksiannya memperhatikan seluruh atribut di data latih sehingga dalam mengklasifikasinya lebih optimal. Sistem ini menerima masukkan dari pengguna berupa faktor-faktor resiko yang ada dan menghasilkan keluaran berupa tingkat resiko penyakit stroke. Sedangkan *certainty factor* juga termasuk dalam metode yang dapat menyelesaikan permasalahan ketidakpastian dalam ilmu sistem pakar (Hidayati, 2013), di mana metode ini menyatakan bahwa kepercayaan dalam sebuah kejadian sesuai dengan bukti penilaian sebuah pakar yang mengasumsikan derajat keyakinan seorang pakar (Ardhitama, 2014). Metode *Certainty factor* ini menerima hasil keluaran dari perhitungan *naïve bayes* kemudian melakukan perhitungan untuk nilai keyakinan dari hasil akhir tersebut.

Dalam penggunaan sistem pakar berbasis Android diharapkan dapat mempermudah bagi pengguna untuk mengetahui dengan cepat dalam mendeteksi secara dini tanpa harus memiliki perangkat komputer atau selalu terkoneksi dengan jaringan komputer (Al Ihsan, 2012). Penggunaan aplikasi ini akan lebih praktis dan efisien selain melalui perangkat *mobile* yang sudah banyak digunakan oleh masyarakat luas, pengguna aplikasi ini juga dapat digunakan dimana saja dan kapan saja (Ardiyanto et al, 2012). Oleh karena itu pada penelitian ini akan dikembangkan sebuah sistem pakar yang dapat mendeteksi secara dini resiko penyakit stroke dengan menggabungkan antara metode *Naïve Bayes* dan *Certainty factor* dengan harapan sistem pakar ini dapat memberikan solusi yang tepat yang menghasilkan akurasi yang tinggi dalam proses pendeteksiannya.

1.2 Rumusan masalah

Berdasarkan latar belakang di atas, maka dapat dirumuskan permasalahan yang akan dijadikan objek penelitian, yaitu:

1. Bagaimana merancang sistem pakar untuk mendeteksi dini penyakit stroke dengan menggunakan metode *Naïve Bayes-Certainty factor*?

2. Bagaimana hasil pengujian pada sistem pakar deteksi dini penyakit stroke metode *Naïve Bayes-Certainty factor*?

1.3 Tujuan

Adapun tujuan penelitian ini sebagai berikut:

1. Merancang sistem pakar untuk mendeteksi dini penyakit stroke menggunakan metode *naïve bayes-certainty factor* yang dapat membantu masyarakat dalam mendeteksi dini penyakit stroke.
2. Memperoleh hasil pengujian dalam mendeteksi dini penyakit stroke menggunakan metode *naïve bayes-certainty factor*.

1.4 Manfaat

Manfaat yang didapatkan dalam penelitian ini adalah sebagai berikut:

1. Membantu pasien untuk mengetahui secara cepat dan tepat dalam mendeteksi tingkat resiko penyakit stroke.
2. Memudahkan tugas dari ahli medis atau dokter dan juga pemerintah dalam mendeteksi tingkat resiko penyakit stroke.

1.5 Batasan masalah

Agar penelian ini lebih berfokus dan tidak meluas, maka batasan penelitian ini, sebagai berikut:

1. Data yang digunakan dalam penelitian ini diperoleh dari Rumah Sakit Syaiful Anwar Kota Malang, Jawa Timur.
2. Jenis data yang diteliti adalah penyebab resiko terserang penyakit stroke.
3. Penyebab-penyebab resiko terserang penyakit stroke yang digunakan adalah tekanan darah, diabetes, riwayat keluarga, merokok, kolesterol, aktifitas fisik, diet, berat badan, tinggi badan, dan riwayat fibrilasi atrium.
4. Pengujian yang dilakukan yakni pengujian validasi (*blackbox testing*), pengujian *usability*, dan pengujian akurasi.
5. Sistem pakar yang dikembangkan merupakan sistem berbasis android.

1.6 Sistematika Pembahasan

Sistematika penulisan laporan ditunjukkan untuk memberikan gambaran dan uraian dari proposal projek secara garis besar yang meliputi beberapa bab, sebagai berikut:

BAB I PENDAHULUAN

Bab ini berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, dan sistematika penulisan laporan penelitian sistem pakar untuk deteksi dini penyakit stroke menggunakan metode *Naïve Bayes-Certainty factor*.

BAB II LANDASAN KEPUSTAKAAN

Bab ini membahas uraian dan pembahasan tentang teori, konsep, model, metode, atau sistem dari literatur ilmiah, yang berkaitan dengan tema, masalah, atau pertanyaan penelitian sistem pakar untuk deteksi dini penyakit stroke menggunakan metode *Naïve Bayes-Certainty factor*.

BAB III METODOLOGI PENELITIAN

Bab ini menguraikan metodologi penelitian dan membahas tentang metode, sistematika, konsep, serta aturan-aturan yang digunakan untuk melaksanakan penelitian.

BAB IV PERANCANGAN

Membahas analisis kebutuhan dan perancangan sistem pakar untuk deteksi dini penyakit stroke menggunakan metode *Naïve Bayes-Certainty factor*.

BAB V IMPLEMENTASI

Membahas implementasi perangkat lunak Aplikasi sistem pakar untuk deteksi dini penyakit stroke menggunakan metode *Naïve Bayes-Certainty factor* sesuai dengan perancangan sistem yang telah dibuat.

BAB VI PENGUJIAN DAN ANALISIS

Memuat proses dan hasil pengujian terhadap sistem sistem pakar untuk deteksi dini penyakit stroke menggunakan metode *Naïve Bayes-Certainty factor* yang telah dibangun.

BAB VII PENUTUP

Memuat kesimpulan yang diperoleh dari pembuatan dan pengujian perangkat lunak sistem pakar untuk deteksi dini penyakit stroke menggunakan metode *Naïve Bayes-Certainty factor* berbasis android yang dikembangkan dalam skripsi ini serta saran-saran untuk pengembangan sistem lebih lanjut.

BAB 2 LANDASAN KEPUSTAKAAN

Bab ini membahas tentang landasan kepustakaan yakni kajian pustaka dan dasar teori yang digunakan sistem pakar deteksi dini penyakit stroke menggunakan metode *naïve bayes-certainty factor* di mana terdapat dua penelitian sebelumnya yang dijadikan acuan.

1.1 Kajian Pustaka

Pada bagian ini, peneliti akan membahas tentang beberapa penelitian sebelumnya yang berhubungan dengan. Penelitian-penelitian tersebut dilakukan oleh Deby Putri Indraswari (2015) dan Achmad Affan Suprayogi (2017).

Penelitian pertama yang dilakukan oleh Deby Putri Indraswari pada tahun 2015 dengan judul “Sistem Pendukung Keputusan Deteksi Dini pada Penyakit Stroke menggunakan metode Dempster Shafer”. Pada penelitian tersebut masih terdapat ketidakakuratan dalam hal akurasi yang disebabkan oleh beberapa faktor, seperti tidak mudahnya dokter dalam memberikan nilai densitas. Dalam menentukan nilai densitas terdapat beberapa masalah yang menyebabkan nilai densitas menjadi kurang maksimal. Hasil perhitungan akurasinya yang diperoleh adalah sebesar 89% (Putri, 2015).

Penelitian selanjutnya yakni oleh Achmad Affan Suprayogi dengan judul “Sistem Pakar Diagnosis Penyakit Kucing Menggunakan Metode Naïve Bayes-certainty factor Berbasis Android”. Pada penelitian tersebut menghasilkan tingkat akurasi sebesar 80% (Affan, 2017). Untuk review Kajian Pustaka ditujukan pada Tabel 2.1.

Tabel 2. 1 Review Kajian Pustaka

No	Judul	Objek (Input)	Metode (Proses)	Hasil (Output)
1	Sistem Pendukung Keputusan Deteksi Dini pada Penyakit Stroke menggunakan metode Dempster Shafer (Deby Putri, 2015).	Jenis tingkat resiko penyakit stroke: Tinggi, Sedang, Rendah Penyebab resiko terserang penyakit: Tekanan Darah, Diabetes, Riwayat Keluarga, Merokok, Kolesterol, Aktifitas Fisik, Diet, Berat Badan, Tinggi badan, Riwayat Fibrilasi Atrium	<ol style="list-style-type: none"> 1. Memasukkan data faktor resiko penyakit stroke. 2. Melakukan proses perhitungan data yang telah dimasukkan menggunakan metode Dempster Shafer. 3. Melakukan pencarian nilai tertinggi dari hasil kombinasi terakhir proses perhitungan yang telah dilakukan 4. Memilih nilai tertinggi untuk dijadikan patokan hasil akhir. 	Output sistem berupa kesimpulan deteksi tingkat resiko penyakit stroke

2	Sistem pakar diagnosis penyakit kucing menggunakan metode <i>naïve bayes – certainty factor</i> berbasis android (Achmad Affan, 2017).	Variabel yang digunakan yaitu gejala-gejala yang terdapat pada penyakit kucing yakni sebanyak 30 gejala.	<ol style="list-style-type: none"> 1. Menghitung <i>prior</i> tiap kelas 2. Menghitung <i>likelihood</i> gejala tiap kelas 3. menghitung <i>posterior</i> tiap kelas 4. Menghitung nilai maksimal <i>posterior</i> 5. Menghitung nilai $CF_{Pakar} * CF_{User}$ 6. Menghitung $CF_{Combine}$ 7. Menghitung $CF_{Presentase}$ 	Kesimpulan diagnosa penyakit pada kucing
3	Sistem pakar deteksi dini penyakit stroke menggunakan metode <i>naïve bayes – certainty factor</i> .	<p>Jenis tingkat resiko penyakit stroke: Tinggi, Sedang, Rendah</p> <p>Penyebab resiko terserang penyakit: Tekanan Darah, Diabetes, Riwayat Keluarga, Merokok, Kolesterol, Aktifitas Fisik, Diet, Berat Badan, Tinggi badan, Riwayat Fibrilasi Atrium</p>	<ol style="list-style-type: none"> 1. Menghitung <i>prior</i> tiap kelas 2. Menghitung <i>likelihood</i> gejala tiap kelas 3. menghitung <i>posterior</i> tiap kelas 4. Menghitung nilai maksimal <i>posterior</i> 5. Menghitung nilai $CF_{Pakar} * CF_{User}$ 6. Menghitung $CF_{Combine}$ 7. Menghitung $CF_{Presentase}$ 	Hasil deteksi tingkat resiko penyakit stroke

1.2 Sistem Pakar

Sistem pakar adalah sistem berbasis komputer yang menggunakan pengetahuan, fakta, dan tehnik penalaran dalam memecahkan masalah yang biasanya hanya dapat dipecahkan oleh seorang pakar dalam bidang tersebut Sistem pakar memberikan nilai tambah pada teknologi untuk membantu dalam menangani era informasi yang semakin canggih. Sistem pakar mengkombinasikan kaidah-kaidah penarikan kesimpulan dengan basis pengetahuan tertentu yang kemudian kombinasi tersebut disimpan ke dalam komputer yang selanjutnya akan digunakan dalam proses pengambilan keputusan untuk menyelesaikan masalah



yang lebih spesifik (Prihatini, 2011). Pada Tabel 2.2 terdapat nilai yang akan menunjukkan tingkatan dari penyakit stroke.

Tabel 2.2 Data Faktor Resiko Penyakit Stroke

Faktor resiko	Tinggi	Sedang	Rendah
Tekanan Darah	>140/90	120-139/80-89	<120/80
Diabetes	Ya	Digaris Batas	Tidak
Merokok	Ya	Kadang-kadang	Tidak
Aktifitas Fisik	Tidak	1-2 kali dalam seminggu	3-4 kali dalam seminggu
Riwayat Keluarga	Ya	Tidak Pasti	Tidak
Berat badan	Kelebihan berat badan	Sedikit melebihi berat badan	Berat badan sehat
Fibrilasi atrium	Tidak Beraturan	Tidak Tahu	beraturan

Sumber: (Association, 2009)

Keterangan:

- Stroke tinggi apabila terdapat \geq dari 3 faktor resiko
- Stroke sedang apabila terdapat 4-6 faktor resiko
- Stroke rendah apabila terdapat 6-8 faktor resiko

Alasan mendasar mengapa sistem pakar dikembangkan menyerupai seorang ahli pakar adalah sebagai berikut (Voni, 2010):

1. Dapat menyediakan kepakaran setiap waktu dan berbagai lokasi
2. Secara otomatis mengerjakan tugas-tugas rutin yang membutuhkan seseorang pakar.
3. Seorang pakar akan pensiun atau pergi.
4. Menghadirkan atau menggunakan jasa seorang pakar memerlukan biaya yang cukup mahal.
5. Kepakaran dibutuhkan juga pada lingkungan yang tidak bersahabat.

1.2.1 Konsep Dasar Sistem Pakar (*Expert Sistem*)

Konsep dasar sistem sendiri memiliki arti keseluruhan yang telah tersusun demi berjalannya suatu proses hingga akhir, antara komponen yang satu dengan komponen lainnya.

1. Keahlian

Keahlian merupakan penguasaan ilmu/ pengetahuan pada bidang tertentu yang diperoleh dari pelatihan, membaca ataupun dari pengalamannya.

2. Ahli

Merupakan seorang yang mampu menjelaskan suatu tanggapan, mempelajari hal-hal serupa tentang topik permasalahan, menyusun kembali pengetahuan, memecah aturan-aturan apabila diperlukan dan menentukan relevan tidaknya keahliannya.

3. Pengalihan keahlian

Pengalihan keahlian dari para ahli ke komputer yang kemudian akan dialihkan lagi ke orang lain yang bukan ahli (Tujuan utama sistem pakar). Proses ini membutuhkan empat aktifitas yakni: tambahan pengetahuan (dari para ahli atau dari sumber-sumber lainnya), representasi pengetahuan dan pengalihan pengetahuan ke pengguna.

4. Inferensi

Merupakan suatu perangkat lunak yang dapat melakukan penalaran dengan menggunakan pengetahuan yang ada untuk menghasilkan suatu kesimpulan.

5. Aturan

Merupakan informasi mengenai cara bagaimana memperoleh fakta baru dari fakta yang telah diketahui.

6. Kemampuan menjelaskan

Kemampuan komputer dalam memberikan penjelasan kepada pengguna tentang ssesuatu informasi mengenai tertentu dari pengguna dan dasar yang dapat digunakan oleh komputer untuk dapat menyimpulkan suatu kondisi.

1.2.2 Tujuan Sistem Pakar

Tujuan dari sistem pakar adalah untuk memindahkan kemampuan (*transferring expertise*) dari seorang ahli atau sumber keahlian yang lain ke dalam komputer dan kemudian memindahkannya dari komputer kepada pemakai yang tidak ahli (bukan pakar). Proses ini meliputi empat aktifitas yaitu (Hidayat, 2010) :

1. Akuisi pengetahuan (*knowledge acquisition*) yaitu kegiatan mencari dan mengumpulkan pengetahuan dari para ahli atau sumber keahlian yang lain.
2. Representasi pengetahuan (*knowledge representation*) adalah kegiatan menyimpan dan mengatur penyimpanan pengetahuan yang diperoleh dalam komputer. Pengetahuan berupa fakta dan aturan disimpan dalam komputer sebagai sebuah komponen yang disebut basis pengetahuan.
3. Inferensi pengetahuan (*knowledge inferencing*) adalah kegiatan melakukan inferensi berdasarkan pengetahuan yang telah disimpan didalam komputer.
4. Pemindahan pengetahuan (*knowledge transfer*) adalah kegiatan pemindahan pengetahuan dari komputer ke pemakai yang tidak ahli.

1.2.3 Bentuk Sistem Pakar

Ada empat bentuk sistem pakar, yaitu (Hidayat, 2010):

1. Mandiri, sistem pakar merupakan *software* yang berdiri sendiri tidak tergantung dengan *software* yang lainnya.
2. Tergabung, sistem pakar merupakan bagian program yang terkandung di dalam suatu metode (konvensional), atau merupakan program dimana di dalamnya memanggil algoritme subrutin lain (konvensional).
3. Terhubung, bentuk ini biasanya merupakan sistem pakar yang menghubungkan ke suatu paket program tertentu, misalnya DBMS.
4. Sistem mengabdikan, sistem pakar merupakan bagian dari komputer khusus yang dihubungkan dengan suatu fungsi tertentu. Misalnya sistem pakar yang digunakan untuk membantu menganalisis data radar.

1.2.4 Ciri-ciri Sistem Pakar

Sistem pakar berupa program-program praktis yang menggunakan strategi heuristik yang dikembangkan oleh manusia untuk menyelesaikan masalah-masalah yang spesifik. Heuristik merupakan suatu strategi untuk melakukan pencarian masalah secara selektif yang memandu proses pencarian yang dilakukan di sepanjang jalur yang memiliki kemungkinan sukses paling besar. Karena keheuristikannya dan sifatnya, maka secara umum sistem pakar mempunyai sifat (Silalahi, 2011):

1. Memiliki informasi yang handal, baik dalam menampilkan langkah-langkah antara ataupun dalam menjawab pertanyaan-pertanyaan tentang proses penyelesaian.
2. Mudah dimodifikasi yakni dengan menambah atau menghapus suatu kemampuan dan basis pengetahuannya.
3. Heuristik dalam menggunakan pengetahuan mendapatkan penyelesaiannya.
4. Dapat digunakan dalam berbagai komputer karena kemampuan untuk beradaptasi.

1.2.5 Keuntungan Sistem Pakar

Secara garis besar, banyak manfaat yang dapat diambil dengan adanya sistem pakar, antara lain:

1. Memungkinkan orang awam bisa mengerjakan pekerjaan para ahli.
2. Bisa melakukan proses secara berulang secara otomatis.
3. Menyimpan pengetahuan dan keahlian para pakar.
4. Meningkatkan *output* dan produktivitas.
5. Meningkatkan kualitas.

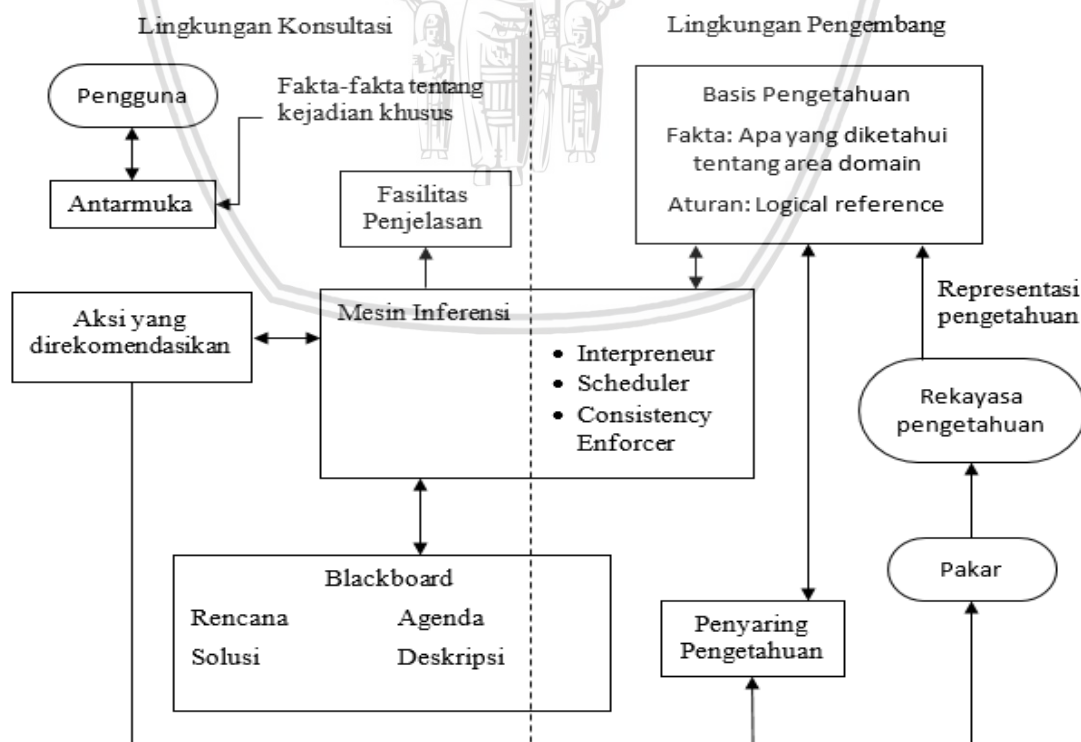
6. Mampu mengambil dan melestarikan keahlian para pakar (terutama yang termasuk keahlian langka).
7. Mampu beroperasi dalam lingkungan yang berbahaya.
8. Memiliki kemampuan untuk mengakses pengetahuan.
9. Memiliki reabilitas. Meningkatkan kapabilitas sistem komputer.
10. Memiliki kemampuan untuk bekerja dengan informasi yang tidak lengkap dan mengandung ketidakpastian.

1.2.6 Kelemahan Sistem Pakar

Di samping memiliki beberapa keuntungan, sistem pakar juga memiliki beberapa kelemahan, antara lain:

1. Biaya yang diperlukan untuk membuat dan memeliharanya sangat mahal.
2. Sulit dikembangkan. Hal ini tentu saja erat kaitannya dengan ketersediaan pakar di bidangnya.
3. Sistem Pakar tidak 100% bernilai benar, karena seseorang yang terlibat dalam pembuatan sistem pakar tidak selalu benar sehingga perlu diuji ulang secara teliti sebelum sistem pakar digunakan.
4. Kepakaran sangat sulit diekstrak dari manusia.

1.2.7 Struktur Sistem Pakar



Gambar 2.1 Diagram Struktur Sistem Pakar

Sumber: (Tambunan, 2010)



Di dalam sistem pakar terdapat dua lingkungan, yakni lingkungan pengembangan dan lingkungan konsultasi seperti pada Gambar 2.1. Lingkungan pengembangan digunakan oleh sistem pakar *builder* dalam membangun komponen dan memasukkan pengetahuan ke dalam basis pengetahuan. Sedangkan lingkungan konsultasi digunakan oleh non pakar dalam memperoleh pengetahuan dan nasihat dari pakar. Kedua lingkungan tersebut dapat dipisahkan setelah sistem lengkap (Tambunan, 2010). Diagram struktur sistem pakar bisa ditunjukkan pada Gambar 2.1.

1.2.8 Konsep Sistem Pakar

Komponen-komponen yang terdapat dalam sistem pakar antara lain: (Tambunan, 2010)

1. Basis Pengetahuan (*Knowledge Base*)

Basis pengetahuan merupakan inti dari suatu sistem pakar, yaitu berupa representasi pengetahuan dari pakar. Basis pengetahuan tersusun atas fakta dan kaidah. Fakta adalah informasi tentang objek, peristiwa, atau situasi. Kaidah adalah cara untuk membangkitkan suatu fakta baru dari fakta yang sudah diketahui.

2. Mesin Inferensi (*Inference Engine*)

Mesin inferensi berperan sebagai otak dari sistem pakar. Mesin inferensi berfungsi untuk memandu proses penalaran terhadap suatu kondisi, berdasarkan pada basis pengetahuan yang tersedia. Di dalam mesin inferensi terjadi proses untuk memanipulasi dan mengarahkan kaidah, model, dan fakta yang disimpan dalam basis pengetahuan dalam rangka mencapai solusi atau kesimpulan. Dalam prosesnya, mesin inferensi menggunakan strategi penalaran dan strategi pengendalian. Strategi penalaran terdiri dari strategi penalaran pasti (*Exact Reasoning*) dan strategi penalaran tak pasti (*Inexact Reasoning*). *Exact reasoning* akan dilakukan jika semua data yang dibutuhkan untuk menarik suatu kesimpulan tersedia, sedangkan *inexact reasoning* dilakukan pada keadaan sebaliknya. Strategi pengendalian berfungsi sebagai panduan arah dalam melakukan proses penalaran. Terdapat tiga teknik pengendalian yang sering digunakan, yaitu *forward chaining*, *backward chaining*, dan gabungan dari kedua teknik pengendalian tersebut.

3. Antarmuka Pemakai (*User Interface*)

Merupakan media komunikasi bagi pengguna dengan sistem. Antarmuka menerima informasi dari pemakai dan mengubahnya ke bentuk yang dapat diterima oleh sistem. Selain itu antarmuka menerima informasi dari sistem dan menyajikannya ke dalam bentuk yang dapat dimengerti oleh pemakai. Pada bagian ini terjadi dialog antara sistem dengan pemakai sehingga memungkinkan sistem pakar menerima intruksi dan informasi dalam bentuk input dan sistem juga memberikan informasi berupa output kepada pemakai

4. Akuisisi Pengetahuan / *Knowledge Acquisition*

Akuisisi pengetahuan adalah akumulasi, *transfer*, dan transformasi keahlian dalam menyelesaikan masalah dari sumber pengetahuan ke dalam program komputer. Dalam tahap ini *knowledge engineer* berusaha menyerap pengetahuan untuk selanjutnya ditransfer ke dalam basis pengetahuan. Pengetahuan diperoleh dari pakar, dilengkapi dengan buku, basis data, laporan penelitian dan pengalaman pemakai.

Metode akuisisi pengetahuan:

- a. Wawancara
Metode yang paling banyak digunakan, yang melibatkan pembicaraan dengan pakar secara langsung dalam suatu wawancara.
- b. Analisis protokol dalam metode ini pakar diminta untuk melakukan suatu pekerjaan dan mengungkapkan proses pemikirannya dengan menggunakan kata-kata. Pekerjaan tersebut direkam, dituliskan, dan dianalisis.
- c. Observasi pada pekerjaan pakar Pekerjaan dalam bidang tertentu yang dilakukan pakar direkam dan diobservasi.
- d. Induksi aturan dari contoh Induksi adalah suatu proses penalaran dari khusus ke umum. Suatu sistem induksi aturan diberi contoh-contoh dari suatu masalah yang hasilnya telah diketahui. Setelah diberikan beberapa contoh, sistem induksi aturan tersebut dapat membuat aturan yang benar untuk kasus-kasus contoh. Selanjutnya aturan dapat digunakan untuk menilai kasus lain yang hasilnya tidak diketahui.

5. *Blackboard*

Blackboard merupakan sebuah area kerja memori tempat pendeskripsian masalah yang diberikan oleh data input, juga dapat digunakan untuk menyimpan *intermediate results*. Tiga macam pilihan yang dapat disimpan dalam *Blackboard* adalah:

- *Plan* : berisi tentang cara menyelesaikan masalah
- *Agenda* : aksi-aksi potensial yang membutuhkan eksekusi
- *Solution* : pilihan hipotesis dan alternatif jawaban yang telah dikumpulkan oleh sistem

6. Fasilitas Penjelasan

Fasilitas penjelasan adalah komponen tambahan yang akan meningkatkan kemampuan sistem pakar karena menggambarkan penalaran sistem kepada pemakai. Fasilitas penjelasan dapat menjelaskan perilaku sistem pakar dengan menjawab pertanyaan sebagai berikut:

1. Mengapa pertanyaan tertentu ditanyakan oleh sistem pakar?
2. Bagaimana kesimpulan tertentu diperoleh?
3. Mengapa alternatif tertentu ditolak?

4. Apa rencana untuk memperoleh penyelesaian?

7. Perbaiki Pengetahuan

Seorang pakar memiliki kemampuan untuk menganalisis dan meningkatkan kinerjanya serta kemampuan untuk belajar dari kinerjanya. Kemampuan tersebut adalah penting dalam pembelajaran terkomputerisasi, sehingga program akan mampu menganalisis penyebab kesuksesan dan kegagalan yang dialaminya.

1.3 Metode Naive Bayes Classifier

Menurut (Prasetyo, 2012) *Naive Bayes* merupakan suatu *classifier perobabilistic simple* yang berdasarkan pada teorema Bayes pada umumnya. Inferensi Bayes khususnya dengan asumsi independensi yang kuat. Pada prosesnya *Naive Bayes Classifier* ini mengasumsikan bahwa ada atau tidak adanya suatu fitur pada suatu kelas tidak berhubungan dengan ada atau tidak adanya fitur lain di kelas yang sama. Teorema Bayes merupakan suatu pendekatan untuk sebuah ketidakpastian yang diukur dengan probabilitas. Prediksi Bayes yang berdasarkan pada teorema bayes memiliki rumus pada Persamaan 2.1.

$$P(H|E) = \frac{P(E|H)P(H)}{P(E)} \quad (2.1)$$

Penjelasan dari rumus 2.1 adalah sebagai berikut:

1. Probabilitas akhir (*posterior*) bersyarat sebuah hipotesis H terjadi jika diberikan bukti (*evidence*) E terjadi. *Posterior* dinotasikan sebagai P(H|E).
2. Probabilitas suatu bukti E terjadi maka memengaruhi hipotesis H (*likelihood*). *Likelihood* memiliki rumus pada persamaan 2.2.

$$P(E|H) = \text{Peluang gejala } e, \text{ jika diasumsikan bahwa hipotesis benar} \quad (2.2)$$

3. Probabilitas awal (*prior*) hipotesis H terjadi tanpa melihat bukti apapun. *Prior* memiliki rumus pada persamaan 2.3.

$$P(H) = \text{Peluang dari hipotesis } h \text{ (Penyakit)} \quad (2.3)$$

4. P(E) Probabilitas awal bukti E terjadi tanpa melihat hipotesis yang lainnya.

Yang menjadi ide dasar dari aturan bayes adalah bahwa hasil dari hipotesis P(H) bisa diperkirakan sesuai dengan beberapa bukti (E) yang diamati. Adapun beberapa hal yang berdasarkan pada teorema bayes, yakni (Prasetyo, 2012):

1. Sebuah probabilitas awal/ *prior* H atau P(H) merupakan probabilitas dari suatu hipotesis sebelum bukti diamati.
2. Sebuah probabilitas akhir H atau P(H|E) merupakan probabilitas dari suatu hipotesis setelah bukti diamati.

2.4 Certainty factor

Menurut Sutojo, dkk (2010:194) awal mula Teori *certainty factor* (CF) diusulkan oleh Shortlife dan Buchanan pada 1975 untuk mengakomodasi ketidakpastian pemikiran seorang pakar. Seorang pakar/ahli dalam hal ini biasanya dokter sering

kali menganalisis informasi yang ada dengan ungkapan seperti “mungkin”, “kemungkinan besar”, “hampir pasti”. Untuk mengakomodasi hal ini kita menggunakan *certainty factor* guna menggambarkan tingkat keyakinan pakar terhadap masalah yang sedang dihadapi.

Ada dua cara dalam mendapatkan *certainty factor* (CF) dari sebuah *rule*, yaitu menggunakan Persamaan 2.4 dan 2.5 :

1. Metode “*Net Belief*” yang diusulkan oleh E.H. Shortlife dan B.G. Buchanan

$$CF (Rule) = MB(H,E) - MD(H,E)$$

$$MB(H, E) = \frac{\max[(H|E),P(H)]-P(H)}{\max[1,0]-P(H)} \quad P(H) = 1, \text{ lainnya} \quad (2.4)$$

$$MD(H, E) = \frac{\max[P(H|E),P(H)]-P(H)}{\min[1,0]-P(H)} \quad P(H) = 0, \text{ lainnya} \quad (2.5)$$

Di mana :

CF(Rule) = Faktor Kepastian

MB(H,E) = *Measure of Belief* (ukuran kepercayaan) terhadap hipotesis H, jika diberikan *evidence* E (antara 0 dan 1)

MD(H,E) = *Measure of Disbelief* (ukuran ketidakpercayaan) terhadap *evidence* H, jika diberikan *evidence* E (antara 0 dan 1)

P(H) = Probabilitas kebenaran hipotesis H

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

2. Dengan cara mewawancarai seorang pakar/ahli.

Nilai CF (*Rule*) didapat dari interpretasi “*term*” dari pakar, yang dirubah menjadi nilai CF tertentu. Sebagai contoh dapat dilihat pada Tabel 2.3, yakni *uncertain term* dari seorang pakar dikonversi menjadi sebuah nilai CF.

Tabel 2.3 Nilai *evidence* tingkat keyakinan pakar

<i>Uncertain Term</i>	CF
<i>Definitely Not</i> (Pasti Tidak)	-1.0
<i>Almost Certainly Not</i> (Hampir Pasti Tidak)	-0.8
<i>Probably Not</i> (Kemungkinan Besar Tidak)	-0.6
<i>Maybe Not</i> (Mungkin Tidak)	-0.4
<i>Unknown</i> (Tidak Tahu)	-0.2 to 0.2
<i>Maybe</i> (Mungkin)	0.4
<i>Probably</i> (Kemungkinan Besar)	0.6
<i>Almost Certainly</i> (Hampir Pasti)	0.8
<i>Definitely</i> (Pasti)	1

Sumber : Buku Kecerdasan Buatan (Sutojo, 2010).



2.5 Naïve Bayes-Certainty factor

Teorema Bayes ini dikemukakan oleh Thomas Bayes yakni seorang ilmuwan pada abad ke 18. Pada saat klasifikasi pendekatan Bayes akan menghasilkan label kategori yang paling tinggi probabilitasnya. Proses perhitungan metode Naïve Bayes dapat dilakukan dengan menggunakan langkah-langkah berikut ini (Prasetyo, 2012):

1. Mencari nilai *prior* untuk tiap-tiap kelas dengan menghitung rata-rata tiap kelas dengan menggunakan Persamaan (2.6).

$$P = \frac{x}{A} \quad (2.6)$$

Dimana,

P = Nilai *Prior*

X = Jumlah data tiap kelas

A = Jumlah data seluruh kelas

2. Mencari nilai *likelihood* untuk tiap-tiap kelas dengan menggunakan Persamaan (2.7).

$$L = \frac{F}{B} \quad (2.7)$$

Di mana,

P = Nilai *Likelihood*

X = Jumlah data fitur tiap kelas

A = Jumlah seluruh data tiap kelas

3. Mencari nilai *posterior* dari tiap kelas yang dengan menggunakan Persamaan (2.8).

$$P(c) \text{ III } P(a|c) \quad (2.8)$$

Di mana,

P(c) = Nilai *prior* tiap kelas

P(a|c) = nilai *likelihood*

Hasil klasifikasi dengan menggunakan metode *Naive bayes* dilakukan dengan membandingkan nilai *posterior* dari kelas-kelas yang ada. Nilai *posterior* yang paling tinggi yang akan terpilih sebagai hasil klasifikasi. Setelah diperoleh hasil perhitungan *naïve bayes*, selanjutnya dilakukan proses untuk mencari nilai keyakinan dari hasil itu sendiri, yakni dengan menggunakan metode *certainty factor* pada proses perhitungannya mengalikan antara CF Pakar (diperoleh langsung dari pakar) dengan CF user (sesuai masukkan pengguna). Kombinasi *certainty factor* yang digunakan untuk mengdiagnosa penyakit adalah (Turban: 2005):

1. *Certainty factor* untuk kaidah dengan premis/gejala tunggal (single premis *rules*): $CF_{\text{gejala}} = CF_{\text{pengguna}} * CF_{\text{pakar}}$ (2.9)

2. Apabila terdapat kaidah dengan kesimpulan yang serupa (*similiary concluded rules*) atau lebih dari satu gejala, maka CF selanjutnya dihitung dengan Persamaan:

CF *Combine* digunakan jika gejala lebih dari satu gejala:

$$CF\ combine = CF\ old + CF\ gejala * (1 - CF\ old) \quad (2.10)$$

Keterangan:

CF old= Merupakan nilai hasil dari penjumlahan dari CF gejala 1 dan CF gejala 2

CF gejala= Merupakan nilai bobot yang diberikan oleh pakar

3. Sedangkan untuk menghitung persentase terhadap penyakit, digunakan

$$\text{Persamaan: } CF\ persentase = CF\ combine * 100 \quad (2.11)$$

2.6 Sistem Operasi Android

Android merupakan sebuah software untuk perangkat mobile yang mencakup sistem operasi, *middleware* dan aplikasi kunci. SDK (*Software Development Kit*) Android menyediakan alat dan API (*Application Programming Interface*) yang diperlukan untuk mengembangkan aplikasi pada *platform* Android dengan menggunakan bahasa pemrograman Java. Android ini menyedekana *platform* terbuka bagi pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam *mobile devices* (Huda, 2013).

2.6.1 Komponen Aplikasi Android

Komponen aplikasi merupakan bagian penting dari sebuah aplikasi android. Setiap komponen memiliki fungsi yang berbeda satu sama lainnya. Berikut merupakan komponen yang setidaknya perlu diketahui (Huda, 2013).

1. *Activity* merupakan sebuah *Activity* yang berfungsi untuk menampilkan antarmuka aplikasi di layar, sebagai contoh ketika kita membuka sebuah aplikasi maka akan muncul tampilan dari aplikasi tersebut. *Activity* diimplementasikan sebagai *subclass* dari class *Activity* sebagai contoh:
2. *Content Providers*, merupakan penyedia konten dari satu aplikasi ke aplikasi lainnya atas perintah tertentu. perintah tersebut ditangani oleh sebuah *method* dari class *ContentResolver*. Data dapat disimpan dalam sistem *file*, *database* atau di tempat lainnya. *ContentProvider* diimplementasikan sebagai *subclass* dari class *ContentProvider* dan harus menerapkan satu set standar yang memungkinkan aplikasi lain untuk melakukan transaksi.
3. *Broadcast Receivers*, berfungsi untuk menerima pesan internet dari aplikasi lain. Sebagai contoh, suatu aplikasi mrngirim pesan berisi perintah tertentu untuk aplikasi lain bahwa beberapa data telah diunduh ke perangkat dan tersedia bagi mereka untuk menggunakan aplikasi tersebut, jadi *broadcast*

ini yang akan menangani komunikasi ini dan akan melakukan tindakan yang tepat. Sebuah *broadcast receiver* diimplementasikan sebagai *subclass* dari class *broadcastReceiver* dan setiap pesan akan dikirim dan diterima sebagai objek *intent*.

2.7 Eclipse, Java Development Kit, Android SDK

Untuk membangun and mengembangkan sebuah aplikasi perangkat bergerak Android, terdapat tiga jenis komponen aplikasi yang dibutuhkan di mana komponen ini saling berhubungan dalam membuat aplikasi berbasis Android.

2.7.1 Eclipse

Eclipse merupakan sebuah IDE (*Integrated Development Environment*) Untuk mengembangkan perangkat lunak dan dapat dijalankan di semua *platform(Platform Independent)*. *Eclipse* ini selalu dilengkapi dengan JDT (*Java Development Tools*) yakni sebuah plug in yang membuat *Eclipse* dapat dipakai untuk mengembangkan program java. *Eclipse* beserta *plug in* nya diimplementasikan dalam bahasa pemrograman java. *Eclipse* dikembangkan dengan konsep sebagai IDE yang terbuka dan mudah diperluas untuk apa saja. Jadi, *eclipse* tidak hanya untuk mengembangkan program java melainkan dapat digunakan untuk pemrograman lainnya seperti dengan cara memasang *plug in* yang diperlukan(Wahana, 2013).

2.7.2 Java Development Kit

Java Development Kit (JDK) merupakan sebuah produk yang dikembangkan oleh Oracle yang ditujukan untuk para *developer* java. JDK ini merupakan *Development Kit* yang paling sering digunakan. Pada tanggal 17 November 2006, Sun mengatakan akan merilis JDK open source menggunakan lisensi GNU *General Public License* (GPL), artinya bahwa *software* ini sekarang menjadi *free software*. Dan pada tanggal 8 Mei 2007, Sun juga mengontribusikan Kode Programnya ke OpenJDK (Wahana, 2013).

JDK juga berisikan paket *Java Runtime Environment* yang komplit dan biasa disebut *private* run time karena biasanya terpisah dari RE *regular* dan konten tambahan yakni terdiri dari *Java Virtual Machine* dan semua *class library* yang ada di *environment* produk. Selain itu termasuk juga *library* tambahan yang berguna untuk *developer*, misalnya *library internationalization* dan *library IDL*.

Salinan dari JDK juga berisi beberapa contoh program yang menjelaskan penerapan porsi dari API Java. *Form* JDK merupakan bagian dari *software developmentKit* (SDK). Untuk java SE, EE, dan ME, Sun mengatakan bahwa JDK merupakan bagian dari SDK yang memiliki tanggung jawab untuk penulisan dan penggabungan program java yang sedang berjalan. Bagian SDK lainnya adalah *application server*, *debugger*, dan dokumentasi (Wahana, 2013).

2.7.3 Android SDK

Android SDK ini juga merupakan komponen terpenting untuk disandingkan dengan *eclipse* dalam perograman aplikasi *native* android (Wahana, 2013). Android SDK adalah tools API yang mulai digunakan untuk mengembangkan aplikasi pada platform android menggunakan bahasa pemrograman java.

Di dalam android SDK disertakan pula yang namanya Android *Virtual Device* (AVD) *manager* yang digunakan untuk mmbuat device virtual android menggunakan API yang dibutuhkan untuk membuat sebuah aplikasi berbasis android, sehingga dalam pembuatan dan pengembangan dari aplikasi android bisa lebih diterapkan.

2.8 Penyakit Stroke

2.8.1 Definisi

Penyakit stroke adalah penyakit yang ditandai dengan tanda gejala kehilangan fungsi otak karena terhentinya suplai darah ke otak. Stroke merupakan peringkat kedua dari penyebab kematian dengan mortalitas 18%-37%. Penyakit stroke ini merupakan salah satu dari adanya penyebab kematian dan juga kecacatan neurologis yang utama yang ada di Indonesia.

Serangan dari tanda gejala penyakit stroke yang berkaitan dengan tekanan darah tinggi yang memengaruhi dari munculnya suatu kerusakan dinding pembuluh darah sehingga dinding pembuluh d arah ini menjadi tidak merata. Akibatnya, zat-zat yang larut ini seperti kolesterol, kalsium dan lain sebagainya akan mengendap pada dinding pembuluh darah terjadi dalam waktu lama, akan mengakibatkan suplai darah ke otak menjadi berkurang, bahkan terhenti yang selanjutnya menimbulkan stroke (Farida, 2009).

2.8.2 Jenis-jenis

1. Stroke Iskemik:

Stroke iskemik terjadi bila pembuluh darah yang memasok darah ke otak tersumbat. Jenis stroke ini yang paling umum (hampir 90% stroke adalah iskemik).

Kondisi yang mendasari stroke iskemik adalah penumpukan lemak yang melapisi dinding pembuluh darah (disebut aterosklerosis). Kolesterol, *homocysteine* dan zat lainnya dapat melekat pada dinding arteri, membentuk zat lengket yang disebut plak. Seiring waktu, plak menumpuk. Hal ini sering membuat darah sulit mengalir dengan baik dan menyebabkan bekuan darah (trombus) (Farida, 2009).

Stroke iskemik dibedakan berdasarkan penyebab sumbatan arteri, yaitu:

1. **Stroke trombotik.** Sumbatan disebabkan trombus yang berkembang di dalam arteri otak yang sudah sangat sempit.

2. **Stroke embolik.** Sumbatan disebabkan trombus, gelembung udara atau pecahan lemak (emboli) yang terbentuk di bagian tubuh lain seperti jantung dan pembuluh aorta di dada dan leher, yang terbawa aliran darah ke otak. Kelainan jantung yang disebut fibrilasi atrium dapat menciptakan kondisi di mana trombus yang terbentuk di jantung terpompa dan beredar menuju otak.

2. Stroke Haemorrhagic

Stroke hemoragik disebabkan oleh pembuluh darah yang bocor atau pecah didalam atau di sekitar otak sehingga menghentikan suplai darah ke jaringan otak yang dituju. Selain itu, darah membanjiri dan memampatkan jaringan otak sekitarnya sehingga mengganggu atau mematikan fungsinya (Farida, 2009).

2.8.3 Faktor Penyebab Stroke

Sebagian besar stroke terjadi akibat kombinasi faktor penyebab medis seperti meningkatnya tekanan darah dan faktor penyebab perilaku seperti merokok. Penyebab-penyebab ini disebut faktor resiko.

1. Tekanan Darah

Hipertensi merupakan penyakit kelainan jantung yang ditandai oleh meningkatnya tekanan darah dalam tubuh. Seseorang yang terjangkit penyakit ini biasanya berpotensi mengalami penyakit-penyakit lain, seperti stroke dan penyakit jantung. Tekanan darah dalam tubuh pada dasarnya merupakan ukuran tekanan atau gaya didalam arteri yang harus seimbang dengan denyut jantung, melalui denyut jantung darah akan dipompa melalui pembuluh darah kemudian dibawa keseluruh bagian tubuh. Tekanan darah dipengaruhi volume darah dan elastisitas pembuluh darah (Rusdi, 2009).

2. Diabetes Melitus

Diabetes Melitus ditandai oleh hiperglikemia serta gangguan-gangguan metabolisme karbohidrat, lemak, dan protein yang bertalian dengan defisiensi absolute atau relatif aktivitas dan/atau sekresi insulin. Gejala-gejala yang khas adalah poliuria, polidipsia, polifagia (WHO, 2013). Diabetes mellitus atau DM merupakan masalah endokrinologis yang menonjol dalam pelayanan kesehatan dan juga sudah terbukti sebagai faktor resiko stroke dengan peningkatan resiko relatif pada stroke iskemik 1.6 sampai 8 kali dan pada stroke hemoragik 1,02 hingga 1,67 kali (Antonios, 2005).

3. Merokok

Merokok adalah penyebab nyata kejadian stroke yang lebih banyak terjadi pada usia dewasa muda ketimbang usia tengah baya atau lebih tua. Perokok memiliki resiko tujuh kali tiga terkena stroke dibandingkan dengan orang yang tidak merokok atau berhenti merokok (Viveca et.al, 2008). Merokok dapat dikenal secara mudah sekali, tinggal menghitung berapa batang rokok yang dihisap tiap hari. Penelitian tentang pengaruh tembakau terhadap kesehatan telah lama dilakukan. Salah satu pengaruh buruk tembakau terhadap kesehatan adalah bahwa merokok merupakan faktor resiko gangguan peredaran darah otak.

4. Kolesterol

Kolesterol merupakan salah satu senyawa lemak yang berasal dari hati yang memiliki berbagai kegunaan, seperti membuat hormon eks, adrenalin, membentuk dinding sel, dan sebagainya (Soeharto, 2004). Hal ini terlihat dari pentingnya kolesterol bagi tubuh kita, tetapi terdapat berbagai masalah akibat dari tingginya asupan kolesterol dalam makanan yang diterima di dalam tubuh yang mengakibatkan kadar dalam darah meningkat. Kelebihan kadar tersebut dapat menyebabkan pembuluh darah mengendap dan menyebabkan penyempitan dan pengerasan yang disebut dengan aterosklerosis (Soeharto, 2004).

Peningkatan kadar kolesterol seiring dengan bertambahnya resiko penyakit jantung koroner (PJK), namun sebaliknya hubungan antara tingkat kolesterol dan resiko stroke masih belum jelas hal ini disebabkan karena beberapa alasan. Pertama, stroke adalah suatu sindrom heterogen dengan etiologi yang berbeda pula seperti abnormalitas lipid penting untuk beberapa orang dengan jenis stroke tertentu namun berbeda untuk orang lain. Dalam sebuah penelitian ditemukan hubungan positif antara jumlah kolesterol dan stroke iskemik. Kedua, subfraksi lipoprotein dianggap sebagai salah satu faktor resiko yang dapat diperhitungkan. Oleh karena itu, perlu dibahas mengenai sub tipe stroke dan subfraksi lipoprotein (Imamamura, et al. 2008).

5. Riwayat Fibrilasi Atrium

Menurut Riset Kesehatan Dasar (Rikesdas) tahun 2007, penyebab utama kematian di Indonesia adalah stroke yaitu sekitar 15,4%. Prevalensi tertinggi kasus stroke iskemik adalah di Kota Surakarta sebesar 0,75% (Dinkes Jateng, 2010). Fibrilasi atrium terjadi pada 2,2 juta orang di Amerika dan prevalensinya meningkat seiring bertambahnya usia. Empat persen pada umur >60 tahun dan 8% pada >80 tahun.

Beberapa faktor pada pasien dengan fibrilasi atrium menjadi lebih beresiko tinggi menderita stroke antara lain umur lebih dari 65 tahun, hipertensi, penyakit jantung rematik, *transien ischemic attack*, diabetes melitus, penyakit jantung kongestif (Josephson and Zimetbaum, 2005).

6. Diet

Diet yang sehat dapat membantu dalam mengurangi resiko terserang penyakit stroke. Dalam proses melakukan diet disesuaikan dengan perhitungan IMT (Indeks Masa Tubuh). Persamaan IMT dan penentuan diet dapat ditunjukkan dalam Persamaan 2.12 dan Tabel 2.4.

$$IMT = \frac{BB}{TB^2}$$

Persamaan (2.12)

Keterangan:

BB : Berat Badan (Kg)

TB : Tinggi Badan (m)

Tabel 2.4 Penentuan Diet

Diet	Rentang
Obesitas	IMT > 30
<i>Overweight</i>	IMT antara 25-30
Normal	IMT < 25

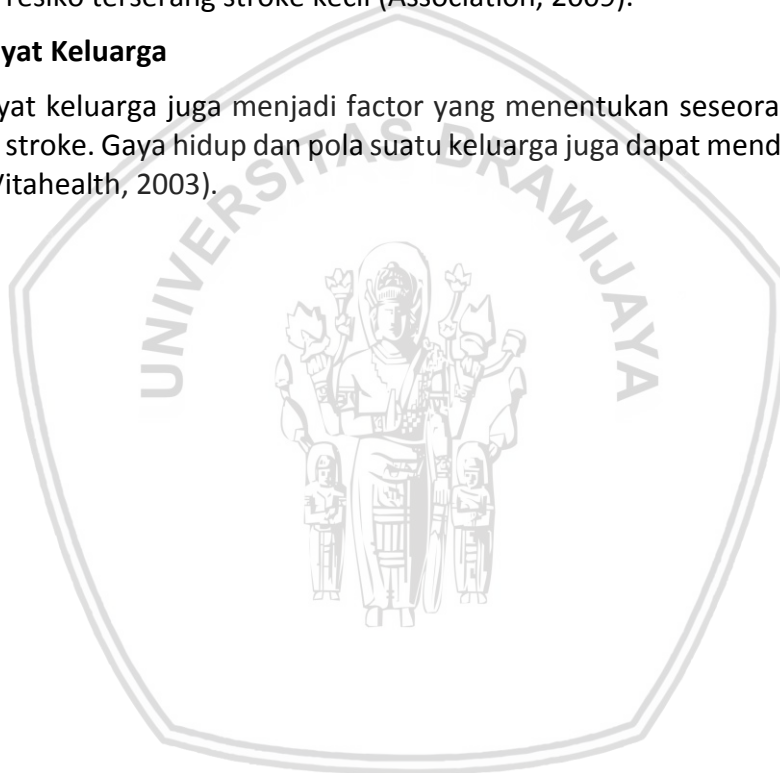
Sumber: (Indraswari, et al, 2015)

7. Aktifitas Fisik

Aktifitas fisik ini meliputi aktifitas yang membuat tubuh kita bergerak. Terdapat penelitian bahwa seseorang yang berolahraga lima kali atau lebih dalam seminggu memiliki resiko terserang stroke kecil (Association, 2009).

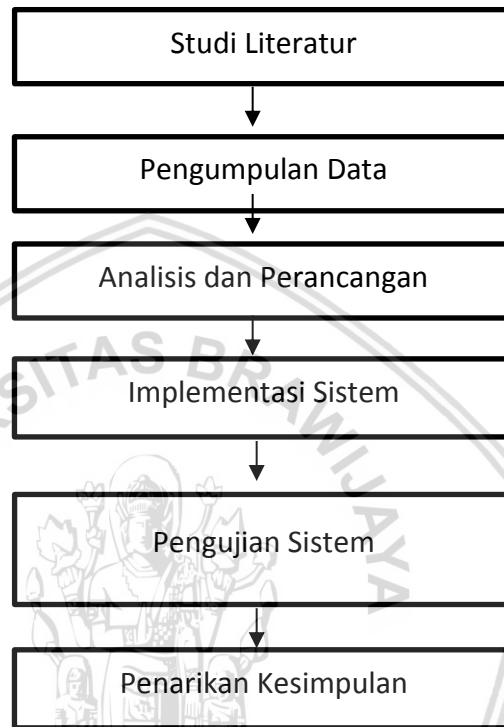
8. Riwayat Keluarga

Riwayat keluarga juga menjadi factor yang menentukan seseorang terserang penyakit stroke. Gaya hidup dan pola suatu keluarga juga dapat mendukung resiko stroke (Vitahealth, 2003).



BAB 3 METODOLOGI PENELITIAN

Metode penelitian yang digunakan dalam penelitian ini akan dibagi dalam beberapa tahap agar proses yang dilakukan berjalan dengan baik. Secara umum, langkah-langkah penelitian yang dilakukan ditunjukkan pada Gambar 3.1.



Gambar 3.1 Diagram Metodologi Penelitian

3.1 Studi Literatur

Studi literatur dilakukan dengan cara mengumpulkan dan mempelajari literatur-literatur yang berkaitan dengan sistem pakar, penyakit stroke, dan metode algoritme *Naïve Bayes-Certainty factor*. Sumber literatur dapat berupa paper hasil penelitian yang sudah ada, jurnal dan skripsi yang sesuai dengan topik penelitian ini yaitu implementasi metode *Naive bayes- Certainty factor* pada sistem pakar khususnya dalam mencari atau mendeteksi dini penyakit stroke.

3.2 Pengumpulan Data

Pengumpulan data diambil dari data sekunder atau *dataset* yaitu data pasien penyakit stroke yang telah dicatat baik dalam hal penyebab hingga hasil deteksi tingkat resiko penyakit stroke. Dari hasil pengumpulan data tersebut didapatkan data penyakit stroke beserta atributnya yaitu:

- Jenis tingkat resiko:
 1. Tinggi



2. Sedang
 3. Rendah
- Atribut / penyebab resiko terserang penyakit:
1. Tekanan Darah
 2. Diabetes
 3. Riwayat Keluarga
 4. Merokok
 5. Kolesterol
 6. Aktifitas Fisik
 7. Diet
 8. Berat Badan
 9. Tinggi badan
 10. Riwayat Fibrilasi Atrium

Proses pengumpulan data tersebut dapat diilustrasikan pada Gambar 3.2:



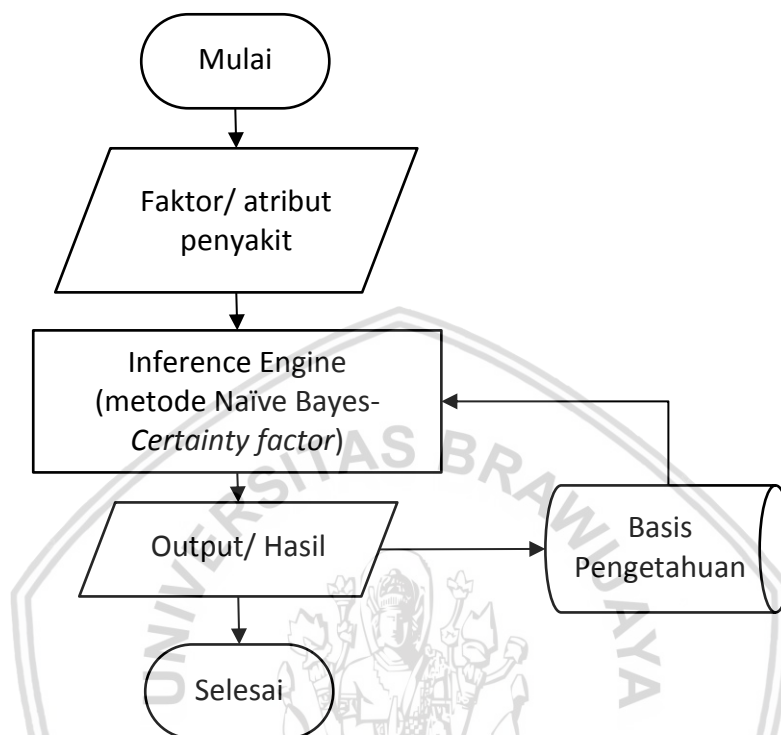
Gambar 3.2 Diagram Blok Proses Pengumpulan Data

3.3 Analisis dan Perancangan

Sistem pakar yang akan dibangun memiliki fungsi untuk mendeteksi dini tingkat resiko penyakit stroke berdasarkan gejala penyebab dari stroke itu sendiri. Admin sistem memiliki peran sebagai pihak yang mengatur dan memasukkan data utama pada sistem ke dalam database. Pengguna sebagai pihak yang melakukan proses mendeteksi penyakit stroke memasukkan gejala-gejala penyebab yang sering dilakukan ke dalam sistem pakar. Metode *Naïve Bayes-Certainty factor* ini akan digunakan untuk mesin inferensi dalam proses perhitungan peluang dan nilai kepercayaan tiap penyebab yang telah dimasukkan oleh pengguna sebelumnya. Pengambilan keputusan diperoleh dari nilai perhitungan peluang dan nilai kepercayaan tertinggi dan selanjutnya akan dipilih sebagai hasil deteksi tingkat penyakit stroke. Hasil keluaran dari sistem pakar terdiri dari tingkat resiko penyakit yang telah dideteksi yakni tinggi, sedang, rendah serta hasil nilai kepercayaannya.

1.3.1 Flow Chart Sistem

Diagram ini menjelaskan aliran logis dari setiap fungsi sistem dan hubungan antar sistem untuk melakukan proses kerja sistem. Diagram ini dapat dilihat pada Gambar 3.3:



Gambar 3.3 Flow Chart Sistem

3.4 Implementasi

Implementasi sistem yang akan dibangun merupakan sistem yang berbasis android sehingga menggunakan bahasa pemrograman JAVA dengan *tools* pendukung lainnya, untuk penyimpanan datanya menggunakan *array*. Sedangkan *user interface* yang akan dibangun adalah *user interface* yang *user friendly*, yang mudah dipahami dan mudah digunakan oleh *user* atau pengguna. Implementasi dari sistem pakar ini meliputi sebagai berikut:

1. Pembuatan antarmuka untuk pengguna yakni halaman berbasis android.
2. Memasukkan data penelitian ke dalam *array* untuk diolah menjadi informasi yang berguna untuk sistem.
3. Penerapan metode *Naïve Bayes-Certainty factor* pada sistem yang dibuat menggunakan bahasa pemrograman java.

3.5 Pengujian

Pengujian sistem dilakukan untuk mengetahui apakah sistem dapat berjalan dengan baik dan sesuai dengan spesifikasi kebutuhan yang telah diterapkan sebelumnya. Pengujian dilakukan untuk menguji kesesuaian antara

perancangan dengan implementasi yang dilakukan dengan sistem. Pengujian yang dilakukan meliputi pengujian akurasi sistem, pengujian *usability*, dan pengujian validasi (*blackbox testing*) pada sistem yang telah dibuat di tahap implementasi. Pengujian akurasi sistem dilakukan dengan cara membandingkan hasil deteksi yang dilakukan oleh sistem dengan hasil yang diperoleh dari pakar. Pada Tabel 3.1 merupakan Contoh dari tabel pengujian akurasi perbandingan antara hasil deteksi sistem dengan hasil deteksi pakar.

Tabel 3.1 Pengujian Akurasi

No	Deteksi Pakar	Deteksi Sistem	Nilai CF

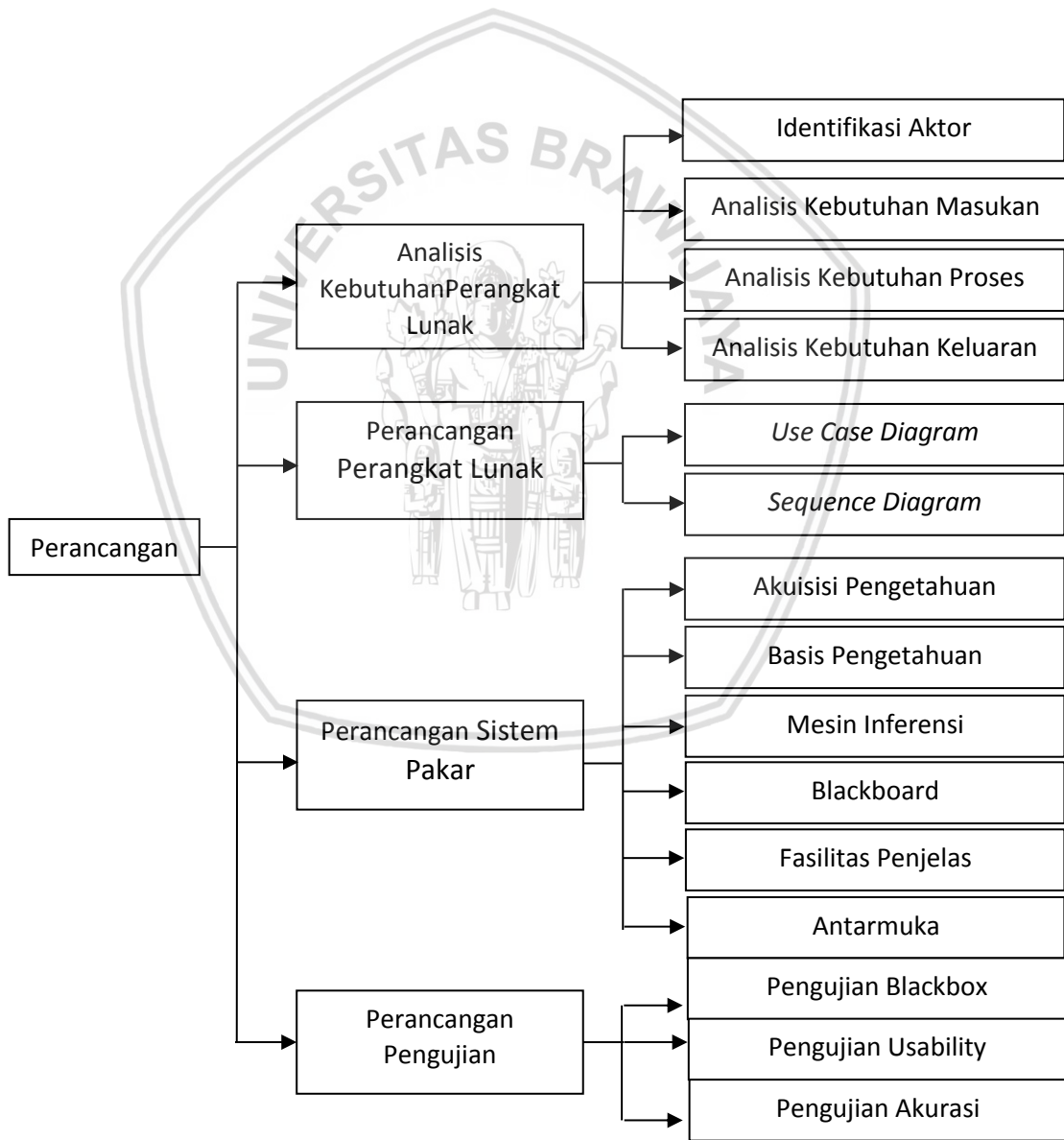
Pengujian validasi (*blackbox testing*) dilakukan dengan cara memeriksa apakah sistem sudah berjalan dengan baik atau tidak ada *error* sesuai dengan daftar kebutuhan di perancangan. Untuk pengujian *usability* ini dilakukan dengan cara memberikan selebaran angket kepada pengguna sistem yang berisi beberapa pertanyaan yang harus diisi oleh pengguna, setelah itu dilakukan proses perhitungan sesuai dengan rumus yang ada untuk mengetahui apakah sistem sudah layak digunakan atau tidak.

3.6 Pengambilan Kesimpulan

Setelah semua tahap perancangan dan implementasi sistem selesai dilaksanakan dan sesuai dalam hal teori dan penerapan, maka dapat dilakukan pengambilan kesimpulan untuk memberikan hasil akhir dari penelitian. Hasil tersebut berupa evaluasi dari penerapan metode yang diterapkan. Setelah didapatkan kesimpulan maka dapat dirumuskan saran sebagai tahap akhir penelitian. Saran ditujukan untuk memperbaiki kesalahan-kesalahan yang terjadi selama penulisan ataupun dalam proses penelitian.

BAB 4 PERANCANGAN

Bab ini membahas mengenai perancangan sistem pakar deteksi dini penyakit stroke dengan metode *naive bayes-certainty factor* yang dilakukan melalui tiga tahap yakni proses analisis kebutuhan perangkat lunak, perancangan sistem pakar dan perancangan perangkat lunak. Tahap analisis kebutuhan pakar terdiri dari identifikasi aktor, analisis kebutuhan input, analisis kebutuhan proses dan analisis kebutuhan output. Perancangan sistem pakar terdiri dari perancangan akuisisi pengetahuan, antarmuka pengguna, fasilitas penjelas, basis pengetahuan, mesin inferensi dan *blackboard*. Perancangan perangkat lunak terdiri dari membuat *use case diagram* dan perancangan algoritme. Pohon perancangan dapat dilihat pada Gambar 4.1.



Gambar 4.1 Pohon Perancangan



4.1 Analisis Kebutuhan Perangkat Lunak

Analisis kebutuhan sistem merupakan tahapan awal dalam proses perancangan yang diawali dengan mengidentifikasi pengguna yang dapat menggunakan sistem yang terlibat di dalam sistem. Berikut ini merupakan penjelasan dari tahapan-tahapannya.

4.1.1 Identifikasi Aktor

Tahap ini bertujuan untuk menjelaskan siapa saja yang akan berinteraksi dengan sistem pakar yang akan dibangun. Pada tahapan ini dijelaskan juga hal-hal apa saja yang dapat dilakukan oleh aktor. Tabel indentifikasi aktor ditujukan pada Tabel 4.1.

Tabel 4.1 Identifikasi Aktor

Aktor	Deskripsi Aktor
Pengguna Umum (PU)	Aktor yang dapat menggunakan sistem pakar untuk melakukan proses <i>login</i> , registrasi pengguna, konsultasi, proses , melihat informasi mengenai jenis-jenis resiko yang memengaruhi terserangnya penyakit stroke dan pencegahannya.
<i>Knowledge Engineer</i> (KE)	Aktor yang menyerap sumber pengetahuan dari pakar kemudian ditransformasikan ke basis pengetahuan. Admin dapat menggunakan semua proses yang ada dalam sistem pakar.

1.1.2 Analisis Kebutuhan Masukan

Analisis kebutuhan masukan ini terdiri dari kebutuhan fungsional yang diperlukan dalam proses melakukan interaksi dengan sistem pakar yang dibangun. Kebutuhan fungsional bertujuan untuk menjelaskan kebutuhan yang harus disediakan oleh sistem, pengguna yang dapat berinteraksi dengan sistem tersebut serta nama proses dari masing-masing kebutuhan. Tabel kebutuhan fungsional ditujukan pada Tabel 4.2.

Tabel 4. 2 Kebutuhan Fungsional

ID	<i>Requirements</i>	Entitas	Nama Aliran Data
KF_01	Sistem dapat menyediakan menu untuk melakukan deteksi penyakit stroke	PU, KE	Tampil halaman mulai deteksi
KF_02	Sistem dapat menampilkan hasil deteksi tingkat resiko penyakit stroke	PU, KE	Tampil halaman hasil deteksi
KF_03	Sistem dapat menampilkan halaman info penyakit	PU, KE	Tampil halaman info penyakit

KF_04	Sistem dapat menampilkan halaman info perangkat	PU, KE	Tampil halaman info perangkat
KF_05	Sistem dapat menampilkan halaman info pengembang	PU, KE	Tampil halaman info pengembang

Pada sistem pakar deteksi dini penyakit stroke ini menggunakan delapan faktor resiko terserang penyakit stroke yang ditunjukkan pada Tabel 4.3.

Tabel 4.3 Faktor-faktor Resiko Penyakit Stroke

No	Parameter	Kode Resiko	Keterangan
1	Tekanan Darah	R1	Tekanan darah menjadi salah satu faktor yang menyebabkan pengerasan dan penyumbatan arteri. Penderita tekanan darah ini mempunyai faktor resiko hingga enam kali lipat dibandingkan dengan orang tanpa tekanan darah, sekitar 40 hingga 90 persen stroke ternyata menderita tekanan darah sebelum terkena stroke.
2	Diabetes	R2	Diabetes memiliki resiko tiga kali lipat terkena stroke. Faktor penyebab lain yang bisa menjadikan seseorang terkena penyakit stroke adalah sebanyak 40 persen penderita diabetes yang biasanya juga menderita penyakit tekanan darah.
3	Diet	R3	Diet sehat bisa membantu mengurangi resiko terserang penyakit stroke dengan meningkatkan kesehatan secara menyeluruh dan dapat membantu menyeimbangkan berat badan.
4	Riwayat Keluarga	R4	Faktor genetik ini juga sangat berperan penting untuk terserang penyakit stroke seperti tekanan darah tinggi, diabetes, jantung, dan cacat pembuluh darah.
5	Merokok	R5	Merokok bisa menjadi salah satu faktor yang bisa mudah untuk diubah, untuk seorang perokok berat bisa lebih besar untuk terserang penyakit stroke.
6	Kolesterol	R6	Untuk seseorang yang memiliki kadar kolesterol dibawah 200 mg/dl masih belum rentan untuk terserang penyakit stroke, untuk kadar diatas 240 mg/dl menjadi kondisi yang berbahaya terhadap penyakit stroke.

7	Aktifitas Fisik	R7	Seseorang yang memiliki aktifitas seperti berolahraga lima kali lebih seminggu akan mengurangi tingkat resiko penyakit stroke.
8	Riwayat Fibrilasi Atrium	R8	Fibrilasi atrium merupakan penyakit stroke dengan kondisi denyut jantung yang tidak teratur di bilik kiri atas. Denyut jantung di dalam atrium kiri ini mencapai empat kali lebih cepat dibandingkan dengan bagian-bagian lainnya.

Berdasarkan dengan parameter-parameter yang sudah dijelaskan diatas, berikut merupakan penjelasan dari delapan parameter tersebut dalam mendeteksi dini penyakit stroke.

1. Tekanan Darah

Untuk menentukan tinggi rendahnya tekana darah terdapat tingkatannya, tekanan darah dapat dilihat dari dual hal yakni sistolik dan diastolik. Untuk lebih jelasnya dapat dilihat pada Tabel 4.4 dan 4.5.

Tabel 4.4 Tekanan Darah Sistolik dan Diastolik

Kode	Tekanan Darah	Jarak
S1	Sistolik Tinggi	>140
S2	Sistolik Sedang	120-140
S3	Sistolik Rendah	<120
D1	Diastolik Tinggi	>90
D2	Diastolik Sedang	80-90
D3	Diastolik Rendah	<80

Sumber: (Nurfaizzah, et al., 2015)

Tabel 4.5 Aturan Tingkatan Tekanan Darah

Aturan	Tekanan Darah
A1	IF S1 AND D1 THEN TEKANAN DARAH TINGGI
A2	IF S1 AND D2 THEN TEKANAN DARAH TINGGI
A3	IF S1 AND D3 THEN TEKANAN DARAH TINGGI
A4	IF S2 AND D1 THEN TEKANAN DARAH TINGGI
A5	IF S2 AND D2 THEN TEKANAN DARAH SEDANG
A6	IF S2 AND D3 THEN TEKANAN DARAH SEDANG
A7	IF S3 AND D1 THEN TEKANAN DARAH TINGGI



A8	IF S3 AND D2 THEN TEKANAN DARAH SEDANG
A9	IF S3 AND D3 THEN TEKANAN DARAH RENDAH

Sumber: (Nurfaizah, et al., 2015)

Untuk beberapa aturan bila nilainya tinggi pada sistolik/diastolik maka masuk kedalam kategori tinggi. Bila terdapat sistolik/diastolik sedang dan rendah maka masuk ke dalam sedang, bila terdapat sistolik/diastolik rendah dan rendah maka masuk ke dalam tekanan darah rendah.

2. Diabetes

Untuk parameter diabetes ditentukan sesuai dengan gula darah acak atau gula darah sewaktu. Untuk penentuan tingkat diabetesnya adalah sebagai berikut:

- Diabetes dengan gulah darah acak >200
- Diabetes dengan gula darah acak antara 160-200
- Diabetes dengan gula darah acak <160

3. Merokok

Merokok menjadi salah faktor yang menyebabkan stroke. Merokok ini dapat dibedakan menjadi merokok, kadang-kadang, dan tidak merokok.

4. Aktifitas Fisik

Semakin banyak atau teraturnya berolahraga akan semakin menurunkan tingkat terserang penyakit stroke. Aktifitas fisik dapat dibedakan menjadi rutin berolahraga, kadang-kadang, dan tidak pernah berolahraga.

5. Kolesterol

untuk kadar kolesterol dibawah 200 mg/dl dikategorikan sebagai kolesterol rendah, untuk antara 200-240 termasuk kolesterol sedang, untuk 240 mg/dl ke atas dikategorikan kolesterol tinggi.

6. Riwayat Keluarga

Seseorang yang memiliki riwayat keluarga dengan penyakit stroke maka tingkat terserang penyakit stroke lebih tinggi, untuk yang tidak memiliki riwayat keluarga dengan penyakit stroke, penentuan tingkat resikonya tergantung dengan kriteria lainnya.

7. Diet

Dalam proses melakukan diet disesuaikan dengan perhitungan IMT (Indeks Masa Tubuh) yang sesuai pada Persamaan IMT 2.6 dan penentuan diet sesuai pada Tabel 2.3.

8. Riwayat Fibrilasi Atrium

Parameter ini dibedakan menjadi jantung teratur, tidak tahu/ tidak merasakan detak jantung, dan jantung tidak beraturan.

Untuk parameter yang telah dijelaskan di atas merupakan jenis parameter yang memiliki nilai, untuk beberapa parameter yang tidak memiliki nilai akan diberikan sebuah skor. Berikut merupakan nilai skor untuk parameter tersebut:

1. Tekanan Darah

Tabel skor untuk tekanan darah ditunjukkan pada Tabel 4.6.

Tabel 4.6 Nilai Skor Tekanan Darah

Tingkat Gejala	Kategori	Skor
Tekanan Darah > 140/90	Tinggi	3
Tekanan Darah 120-140/80-90	Sedang	2
Tekanan Darah < 120/80	Rendah	1

2. Merokok

Tabel skor untuk merokok ditunjukkan pada Tabel 4.7.

Tabel 4.7 Nilai Skor Merokok

Tingkat Gejala	Skor
Ya (merokok)	3
Kadang-kadang	2
Tidak	1

3. Riwayat Keluarga

Tabel skor untuk riwayat Keluarga ditunjukkan pada Tabel 4.8.

Tabel 4.8 Nilai Skor Riwayat Keluarga

Tingkat Gejala	Skor
Ada (Stroke)	3
Tidak Tahu	2
Tidak Ada	1

4. Aktifitas Fisik

Tabel skor untuk riwayat Keluarga ditunjukkan pada Tabel 4.9.

Tabel 4.9 Nilai Skor Aktifitas Fisik

Tingkat Gejala	Skor
Tidak Pernah Olahraga	3
Kadang-kadang	2
Olahraga teratur	1

5. Riwayat Fibrilasi Atrium

Tabel skor untuk riwayat Keluarga ditunjukkan pada Tabel 4.10.

Tabel 4.10 Nilai Skor Riwayat Fibrilasi Atrium

Tingkat Gejala	Skor
Ya	3
Tidak Tahu	2
Tidak	1

Tabel data latih yang digunakan dalam proses untuk mendeteksi dini penyakit stroke menggunakan metode *naive bayes – certainty factor* terdapat pada Tabel 4.11.

Tabel 4.11 Data Latih

No	R1	R2	R3	R4	R5	R6	R7	R8	Tingkat Resiko
1	1	120	3	1	183	1	20,57	3	Tinggi
2	3	220	2	3	243	3	33,2	1	Tinggi
3	3	230	2	3	251	2	32,44	1	Tinggi
.
.
.
17	3	305	2	3	326	3	21,3	1	Sedang
18	2	240	2	2	248	2	33,4	2	Sedang
19	1	250	2	2	259	2	33,24	2	Sedang
.
.
.
18	1	193	1	1	192	1	17,22	3	Rendah
19	2	123	1	1	190	1	15,57	3	Rendah
20	1	143	1	1	195	1	18,93	3	Rendah

Tabel data uji yang digunakan dalam proses untuk mendeteksi dini penyakit stroke menggunakan metode *naive bayes – certainty factor* terdapat pada Tabel 4.12.



Tabel 4.12 Data Uji

No	R1	R2	R3	R4	R5	R6	R7	R8	Tingkat Resiko
1	1	180	3	1	100	1	20.3	3	Tinggi
2	3	240	1	3	150	1	30.1	1	Tinggi
3	1	220	2	3	260	3	31.1	1	Tinggi
4	3	250	1	3	300	3	32.7	1	Tinggi
6	1	300	2	2	310	2	32,9	1	Tinggi
7	1	180	2	2	220	3	33,5	2	Sedang
8	1	190	1	1	360	3	33,6	3	Sedang
9	1	210	2	3	145	2	32,7	2	Tinggi
10	1	206	2	2	210	3	24,6	2	Tinggi
11	1	180	2	2	330	3	24,7	3	Sedang
12	1	190	2	2	216	2	30,1	3	Sedang
13	2	190	2	2	218	2	30,1	1	Sedang
14	1	170	2	3	220	3	32,2	1	Sedang
15	1	180	2	2	220	3	26,7	1	Sedang
16	1	178	1	1	120	2	20,0	1	Sedang
.
.
.
25	1	300	1	1	230	1	33,9	1	Tinggi

1.1.3 Analisis Kebutuhan Proses

Proses inti dari sistem ini adalah proses penalaran. Sistem akan melakukan penalaran untuk menentukan tingkat resiko penyakit stroke berdasarkan faktor resiko yang dimasukkan oleh pakar. Data aturan pada basis pengetahuan akan digunakan sebagai data *training* metode *naive bayes - certainty factor*.

1.1.4 Analisis Kebutuhan Keluaran

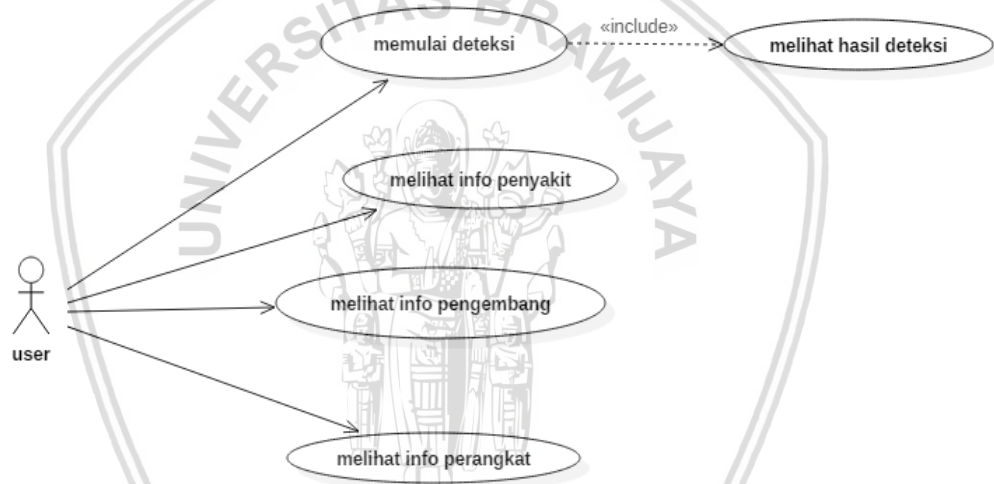
Data keluaran dari sistem pakar ini adalah berupa hasil deteksi tingkat resiko (tinggi, sedang, rendah) dari penyakit stroke yang diderita oleh pengguna berdasarkan perhitungan metode *naive bayes*, nilai *presentase* keyakinan hasil untuk mendeteksi dengan perhitungan metode *certainty factor*, dan informasi tentang jenis-jenis resiko yang menentukan tingkat resiko dari penyakit stroke yang dideteksi. Hasil deteksi tersebut diperoleh sesuai dengan jenis-jenis resiko yang telah diinputkan oleh pengguna pada saat proses deteksi tingkat resiko penyakit stroke.

1.2 Perancangan Perangkat Lunak

Perancangan perangkat lunak ini menjelaskan mengenai pola hubungan antar komponen-komponen detail dalam sistem sehingga mampu membentuk sebuah fungsi yang mampu memberikan pelayanan terhadap kebutuhan pengguna. Perancangan perangkat lunak menggunakan *Use Case Diagram* sebagai pemodelan perangkat lunaknya.

1.2.1 Use Case Diagram

Merupakan salah satu diagram yang berfungsi untuk menjelaskan secara detail elemen-elemen penting di dalam sebuah sistem. Selain itu *use case* juga menjelaskan proses dimana sistem akan digabungkan dengan kebutuhan fungsional yang telah dijelaskan pada bab sebelumnya. *Use case diagram* menggambarkan kegiatan yang dilakukan oleh pengguna sistem. *Use case diagram* sistem pakar deteksi dini penyakit stroke ditunjukkan pada Gambar 4.2.



Gambar 4.2 Use Case Diagram

1.2.2 Use Case Skenario

Untuk masing-masing dari fungsi use case diagram akan dijelaskan pada use case scenario berikut ini.

1.2.2.1 Use Case Skenario Halaman Mulai Deteksi

Use Case Skenario untuk halaman mulai deteksi digunakan untuk menjelaskan pengguna masuk ke dalam halaman mulai deteksi yang ditunjukkan pada Tabel 4.13.

Tabel 4.13 Mengakses Halaman Mulai Deteksi

Mengakses Halaman Mulai Deteksi	
Actors	Pengguna (PU, KE)



Pre-Condition	Pengguna masuk ke halaman utama
Main Flow	<ol style="list-style-type: none"> 1. Pengguna memilih menu mulai deteksi. 2. Sistem menampilkan halaman mulai deteksi yang terdapat beberapa pertanyaan resiko yang harus diisi.
Alternative Flow	-
Post Condition	Pengguna masuk ke dalam halaman mulai deteksi

1.2.2.2 Use Case Skenario Halaman Info Penyakit

Use Case Skenario untuk halaman info penyakit digunakan untuk menjelaskan pengguna masuk ke dalam halaman info penyakit yang ditunjukkan pada Tabel 4.14.

Tabel 4.14 Mengakses Halaman Info Penyakit

Mengakses Halaman Info Penyakit	
Actors	Pengguna (PU, KE)
Pre-Condition	Pengguna masuk ke halaman utama
Main Flow	<ol style="list-style-type: none"> 1. Pengguna memilih menu info penyakit. 2. Sistem menampilkan halaman info penyakit yang terdapat beberapa informasi seputar penyakit stroke serta resiko-resiko yang ada di dalam penyakit stroke.
Alternative Flow	-
Post Condition	Pengguna masuk ke dalam halaman info penyakit.

1.2.2.3 Use Case Skenario Halaman Info Pengembang

Use Case Skenario untuk halaman info pengembang digunakan untuk menjelaskan pengguna masuk ke dalam halaman info pengembang yang ditunjukkan pada Tabel 4.15.

Tabel 4.15 Mengakses Halaman Info Pengembang

Mengakses Halaman Info Pengembang	
Actors	Pengguna (PU, KE)
Pre-Condition	Pengguna masuk ke halaman utama

<i>Main Flow</i>	<ol style="list-style-type: none"> 3. Pengguna memilih menu info pengembang. 4. Sistem menampilkan halaman info pengembang yang terdapat beberapa informasi tentang profil pengembang dari sistem pakar deteksi dini penyakit stroke.
<i>Alternative Flow</i>	-
<i>Post Condition</i>	Pengguna masuk ke dalam halaman info pengembang.

1.2.2.4 Use Case Skenario Halaman Info Perangkat

Use Case Skenario untuk halaman info pengembang digunakan untuk menjelaskan pengguna masuk ke dalam halaman info pengembang yang ditunjukkan pada Tabel 4.16.

Tabel 4.16 Mengakses Halaman Info Perangkat

Mengakses Halaman Info Perangkat	
<i>Actors</i>	Pengguna (PU, KE)
<i>Pre-Condition</i>	Pengguna masuk ke halaman utama
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Pengguna memilih menu info perangkat. 2. Sistem menampilkan halaman info perangkat yang terdapat beberapa informasi tentang perangkat dari sistem pakar deteksi dini penyakit stroke.
<i>Alternative Flow</i>	-
<i>Post Condition</i>	Pengguna masuk ke dalam halaman info perangkat.

1.2.2.5 Use Case Skenario Halaman Hasil Deteksi

Use Case Skenario untuk halaman hasil deteksi digunakan untuk menjelaskan pengguna masuk ke dalam halaman hasil deteksi yang ditunjukkan pada Tabel 4.17.

Tabel 4.17 Mengakses Halaman Hasil Deteksi

Mengakses Halaman Hasil Deteksi	
<i>Actors</i>	Pengguna (PU, KE)
<i>Pre-Condition</i>	Pengguna masuk ke halaman utama

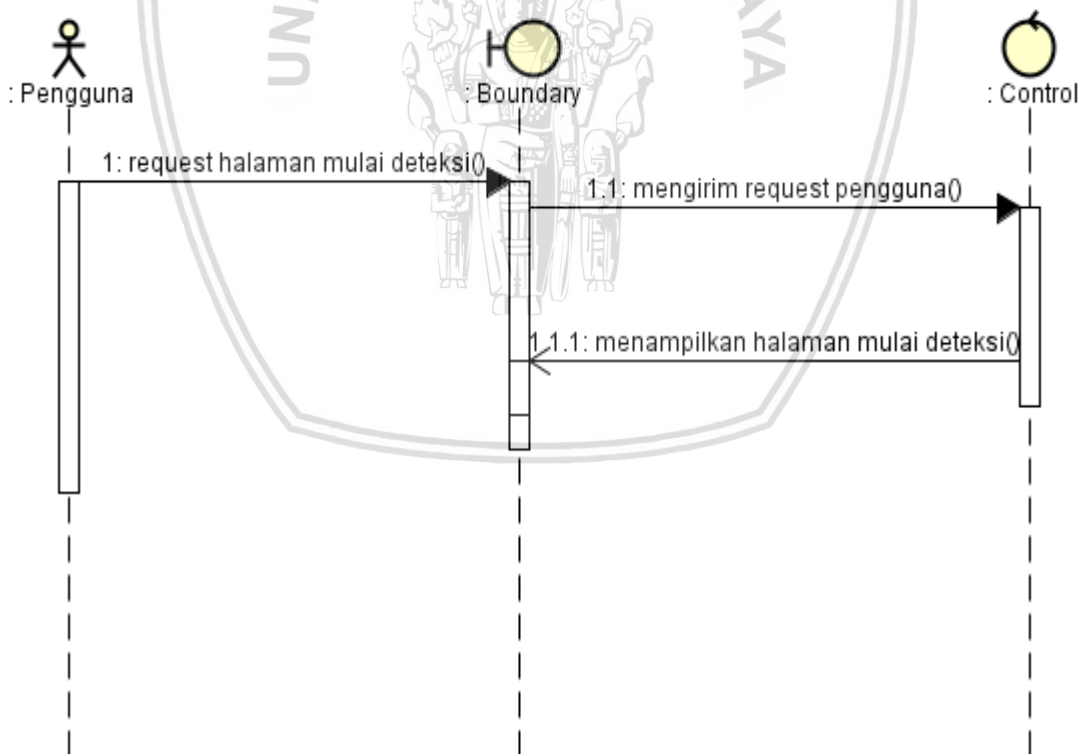
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Pengguna memilih menu mulai deteksi. 2. Pengguna menekan tombol Mula Deteksi 3. Sistem menampilkan halaman hasil deteksi yang berisi hasil deteksi dari sistem berupa tingkat resiko dan juga nilai keyakinanya.
<i>Alternative Flow</i>	-
<i>Post Condition</i>	Pengguna masuk ke dalam halaman hasil deteksi.

1.3.3 Sequence Diagram

Sequence diagram digunakan untuk menunjukkan rangkaian pesan yang dikirimkan antar objek juga antara objek lainnya.

1.3.3.1 Sequence Diagram Mengakses Halaman Mulai Deteksi

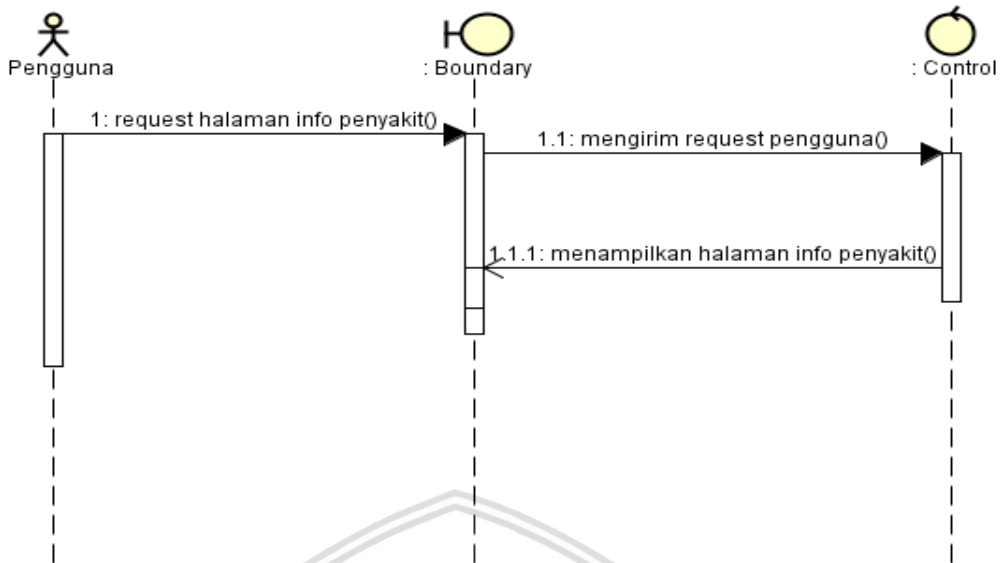
Sequence diagram untuk mengakses halaman mulai deteksi ditunjukkan pada Gambar 4.3.



Gambar 4.3 *Sequence Diagram* Mengakses Halaman Mulai Deteksi

1.3.3.2 Sequence Diagram Mengakses Halaman Info Penyakit

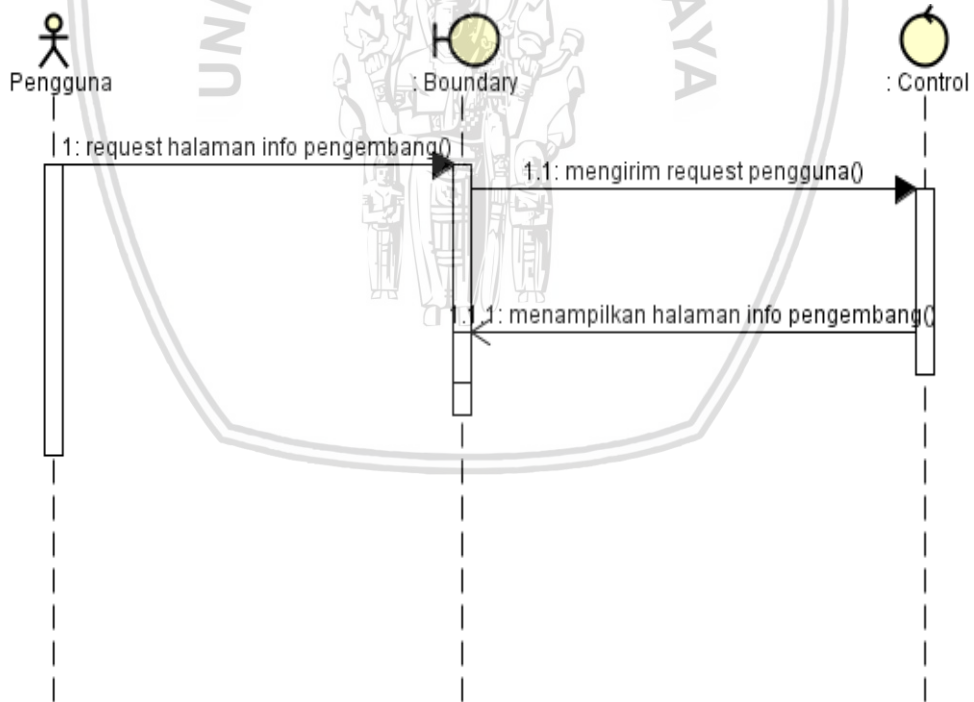
Sequence diagram untuk mengakses halaman info penyakit ditunjukkan pada Gambar 4.4.



Gambar 4.4 *Sequence Diagram* Mengakses Halaman Info Penyakit

1.3.3.3 *Sequence Diagram* Mengakses Halaman Info Pengembang

Sequence diagram untuk mengakses halaman info pengembang ditunjukkan pada Gambar 4.5.

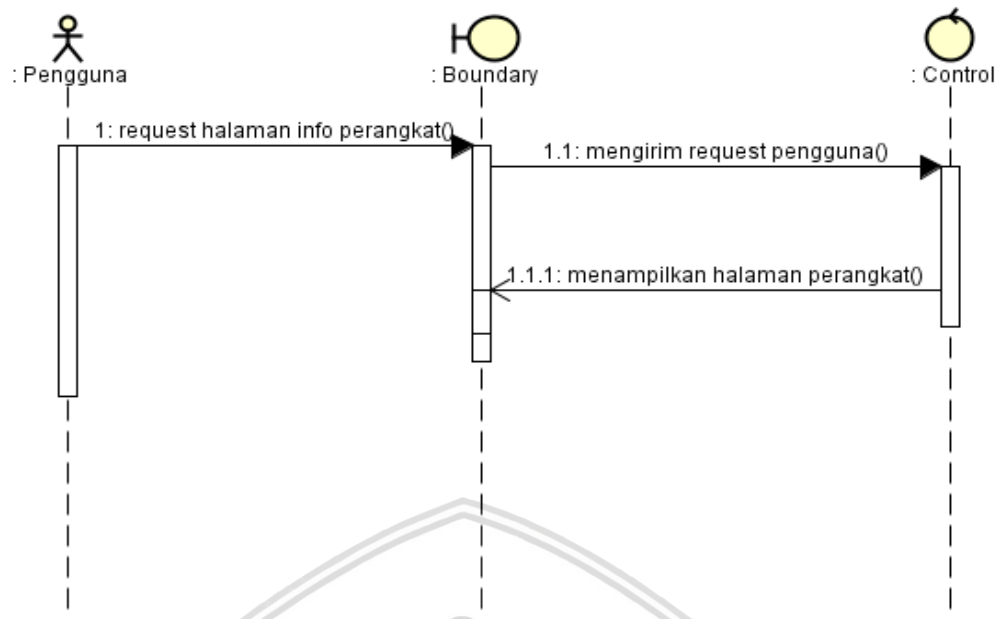


Gambar 4.5 *Sequence Diagram* Mengakses Halaman Info Pengembang

1.3.3.4 *Sequence Diagram* Mengakses Halaman Info Perangkat

Sequence diagram untuk mengakses halaman info perangkat ditunjukkan pada Gambar 4.6.

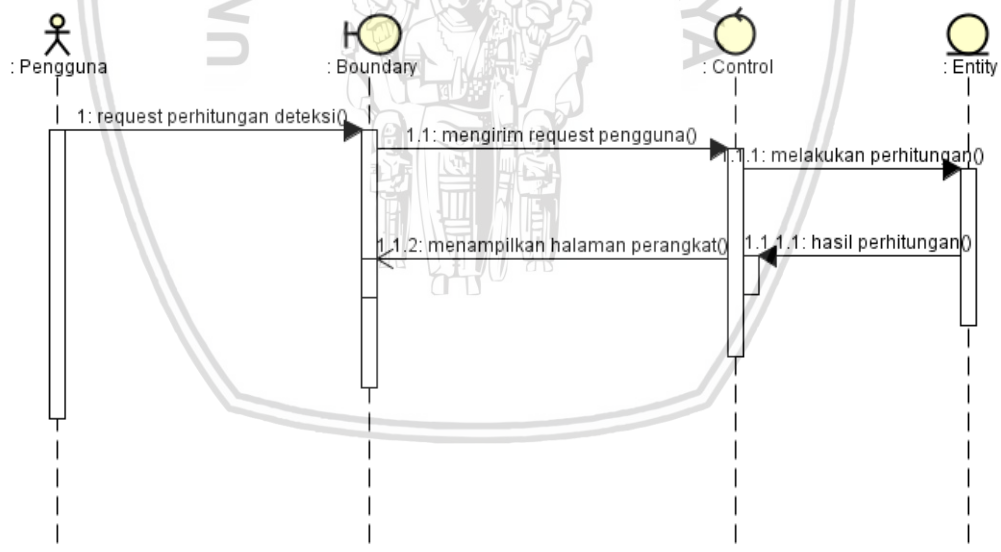




Gambar 4.6 *Sequence Diagram* Mengakses Halaman Info Perangkat

1.3.3.5 *Sequence Diagram* Mengakses Halaman Hasil Deteksi

Sequence diagram untuk mengakses halaman hasil deteksi ditunjukkan pada Gambar 4.7.



Gambar 4.7 *Sequence Diagram* Mengakses Halaman Hasil Deteksi

4.3 Perancangan Sistem Pakar

Sistem pakar yang akan dibangun ini bertujuan untuk mendeteksi tingkat resiko penyakit stroke. Metode *Naive bayes* digunakan untuk proses perhitungan pada nilai probabilitas di setiap classnya dan diambil nilai yang terbesar sebagai hasil deteksi. Untuk metode *Certainty factor* digunakan untuk proses perhitungan nilai presentase keyakinan terhadap hasil deteksi pada metode *naive bayes*. Pada proses perhitungan metode *Certainty factor* dilakukan perhitungan dengan

memasukkan nilai keyakinan pakar dan nilai keyakinan dari pengguna yang sesuai dengan masukkan jenis-jenis penyakit yang memengaruhi terserangnya penyakit stroke oleh pengguna.

Tahap yang biasa dilakukan baik oleh orang awam maupun seorang pakar dalam melakukan identifikasi atau mendeteksi adalah dengan melihat jenis-jenis penyakit yang dialami oleh seseorang. Semakin banyak dan jelas jenis-jenis penyakit yang diamati maka akan semakin besar tingkat keyakinannya. Konsep sistem pakar yang akan dibangun dengan metode *Naive bayes – Certainty factor* merupakan sistem pengambil keputusan berdasarkan data sample atau hasil observasi. Sistem menerima masukan dari *user* terhadap jenis-jenis penyakit yang dialami pada orang tersebut, kemudian hasil dari masukan *user* tersebut akan dilakukan proses perhitungan dengan menggunakan rumus *Naive bayes*. Dari perhitungan nilai probabilitas tersebut maka akan menghasilkan hasil deteksi yang kemudian akan dihitung kembali dengan rumus *Certainty factor* untuk memperoleh nilai keyakinan dari hasil deteksi tersebut.

Bagian-bagian atau komponen yang membangun sistem pakar berbasis android untuk mendeteksi dini tingkat resiko penyakit stroke meliputi:

1.3.1 Akuisisi Pengetahuan

Merupakan proses mengambil pengetahuan dari pakar atau sumber pengetahuan lainnya ke dalam sistem komputer untuk membangun basis pengetahuan. Tahap-tahap yang dilakukan antara lain:

1. Wawancara

Wawancara adalah metode akuisisi yang paling banyak digunakan. Metode ini melibatkan pembicaraan dengan pakar secara langsung dalam suatu wawancara. Tujuan wawancara ini adalah memperoleh keterangan terperinci dan mendalam mengenai pandangan pakar terhadap permasalahan tertentu sehingga kesimpulan akhir dapat diperoleh.

2. Analisis Protokol

Pada analisis protokol pakar diminta untuk melakukan suatu pekerjaan dan mengungkapkan proses pemikirannya dengan menggunakan kata-kata. Proses ini melakukan deteksi dini tingkat resiko penyakit stroke. Informasi dari pemikiran pakar tersebut nantinya akan digunakan sebagai pengetahuan tentang bagaimana mendeteksi tingkat resiko penyakit stroke berdasarkan faktor resiko yang dialami seseorang.

Pada proses akuisisi ini peneliti diminta untuk memberikan pengetahuan dan hasil penelitian selama ini mengenai faktor resiko terserang penyakit stroke yang nantinya dapat digunakan sebagai acuan dalam mengambil keputusan. Dalam menentukan suatu pengambilan keputusan, pakar menggunakan kemampuan yang dimiliki. Hasil penelitian dan observasi mengenai tingkat resiko terserang stroke yang dilakukan pakar tersebut nantinya akan digunakan oleh penulis sebagai data pendukung dalam perhitungan *naive bayes – certainty factor*.

4.3.2 Basis Pengetahuan

Basis pengetahuan berisi pengetahuan tentang data aturan yang diperlukan untuk memformulasikan, memahami, dan memecahkan masalah. Basis pengetahuan tersebut mencakup dua elemen dasar yaitu fakta dan aturan khusus yang mengarahkan pengguna untuk memecahkan persoalan khusus yang mengarahkan pengguna pengetahuan untuk memecahkan persoalan khusus dalam domain tertentu. Basis pengetahuan merupakan inti program dari sistem pakar dimana basis pengetahuan ini merupakan representasi pengetahuan dari seorang pakar. Penalaran berdasarkan pada basis pengetahuan yang ada, manipulasi dan mengarahkan sesuai kaidah, model, dan fakta yang disimpan hingga dicapai suatu kesimpulan.

Berdasarkan data latih yang ada maka dapat diperoleh jumlah masing-masing tingkat resiko yang ada pada data latih, dan jumlah masing-masing faktor resiko yang memengaruhi seseorang terserang penyakit stroke. Data-data ini akan digunakan dalam proses pengambilan keputusan metode *Naive bayes*. Jumlah masing-masing tingkat resiko yang ada di dalam data latih dan jumlah masing-masing jenis resiko yang ada akan ditunjukkan pada Tabel 4.18, 4.19, dan Tabel 4.20. Untuk bobot nilai CF yang digunakan dalam perhitungan metode *Certainty factor* diperoleh berdasarkan nilai keyakinan yang diberikan oleh pakar. Nilai tersebut didasarkan pada jenis-jenis dari tiap tingkat resiko yang ada. Bobot nilai *Certainty factor* pakar ditunjukkan pada Tabel 4.21.

Tabel 4.18 Jumlah masing-masing Tingkat Resiko Pada Data Latih

No	Tingkat Resiko	Jumlah
1	Tinggi	16
2	Sedang	72
3	Rendah	32
Jumlah Keseluruhan Data		120

Tabel 4.19 Data Latih Tingkat Resiko Terserang Stroke

No	R1	R2	R3	R4	R5	R6	R7	R10	Tingkat Resiko
1	Rendah	120	Iya	Tidak	183	Rutin	20,5 7	Iya	Tinggi
2	Tinggi	220	Tidak Tahu	Iya	243	Tidak	33,2	Tidak	Tinggi
3	Tinggi	230	Tidak Tahu	Iya	251	Kadang	32,4 4	Tidak	Tinggi
4	Tinggi	210	Tidak Tahu	Iya	241	Tidak	32,1 8	Tidak Tahu	Tinggi

5	Tinggi	250	Tidak Tahu	Iya	249	Kadang	26,13	Tidak Tahu	Tinggi
6	Rendah	120	Iya	Tidak	183	Rutin	20,57	Iya	Tinggi
.									
.									
.									
120	Rendah	143	Tidak Ada	Tidak	195	Rutin	18,93	Iya	Rendah

Tabel 4.20 Jumlah Jenis Resiko Dari Setiap Penyakit

No	Tingkat Resiko	Jenis Resiko									
		R1	R2	R3	R4	R5	R6	R7	R8	R9	R10
1	Tinggi	14	1	1	14	0	11	7	6	3	9
2	Sedang	24	50	71	72	35	70	33	38	56	65
3	Rendah	16	2	22	12	0	32	71	14	3	0

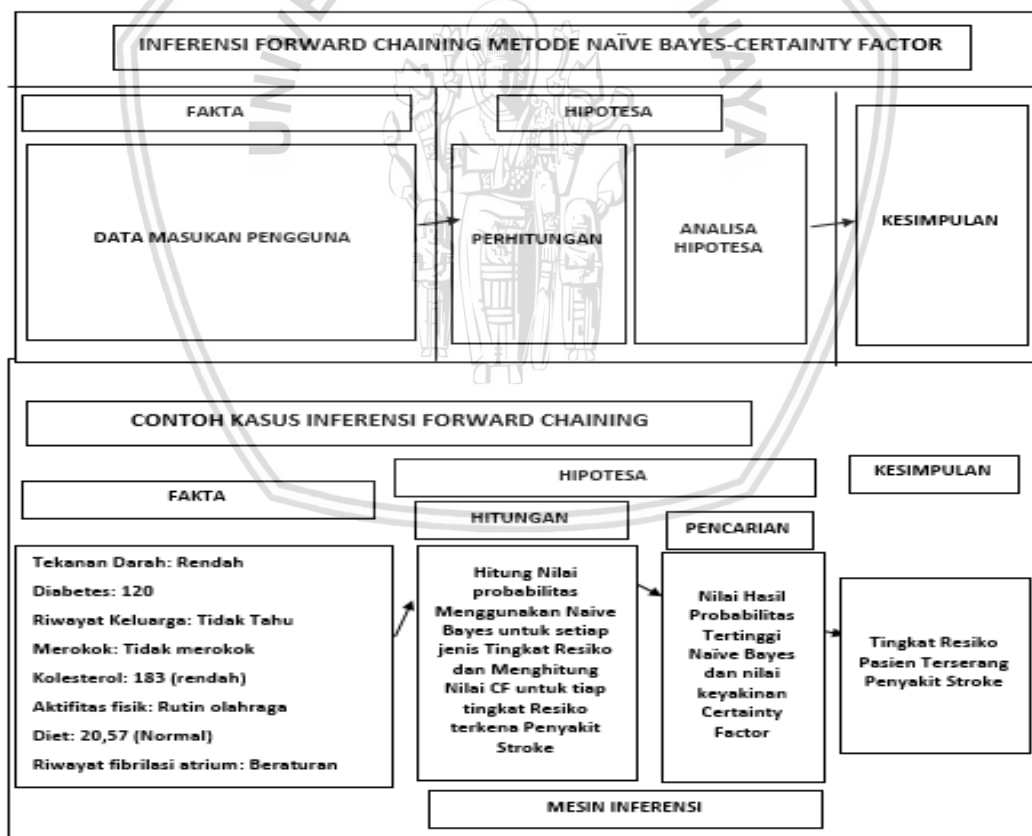
Tabel 4.21 Bobot Nilai CF Pakar Tingkat Resiko Penyakit Stroke

No	Jenis Resiko	Tingkat Resiko		
		Tinggi	Sedang	Rendah
1	Tekanan Darah(tinggi)	0.5	0.4	0.3
2	Tekanan Darah(sedang)	0.2	0.4	0.3
3	Tekanan Darah(rendah)	0.3	0.4	0
4	Diabetes(tinggi)	0.5	0.6	0
5	Diabetes(sedang)	0.2	0.6	0.3
6	Diabetes(rendah)	0.2	0	0.3
7	riwayat keluarga(iya)	0.2	0	0
8	riwayat keluarga(tidak tahu)	0.4	0.7	0.1
9	riwayat keluarga(tidak)	0.3	0.1	0.5
10	merokok(iya)	0.5	0.2	0
11	merokok(tidak)	0.2	0	0.3
12	merokok(kadang)	0	0.7	0.4
13	kolesterol(tinggi)	0.4	0.6	0



14	kolesterol(sedang)	0.2	0.4	0.4
15	kolesterol(rendah)	0.3	0.5	0.4
16	aktifitas fisik(iya)	0.2	0	0.5
17	aktifitas fisik(kadang)	0.3	0.7	0
18	aktifitas fisik(tidak)	0.5	0.2	0
19	Diet(normal)	0.2	0.2	0.4
20	Diet(sedang)	0	0.5	0.4
21	diet(tinggi)	0.5	0.6	0
22	riwayat fibrilasi(beraturan)	0.2	0.6	0.5
23	riwayat fibrilasi(tidak tahu)	0.3	0.6	0
24	riwayat fibrilasi(tidak beraturan)	0.4	0.2	0

4.3.3 Mesin Inferensi



Gambar 4.8 Proses penelusuran Forward Chaining Tingkat Resiko Stroke

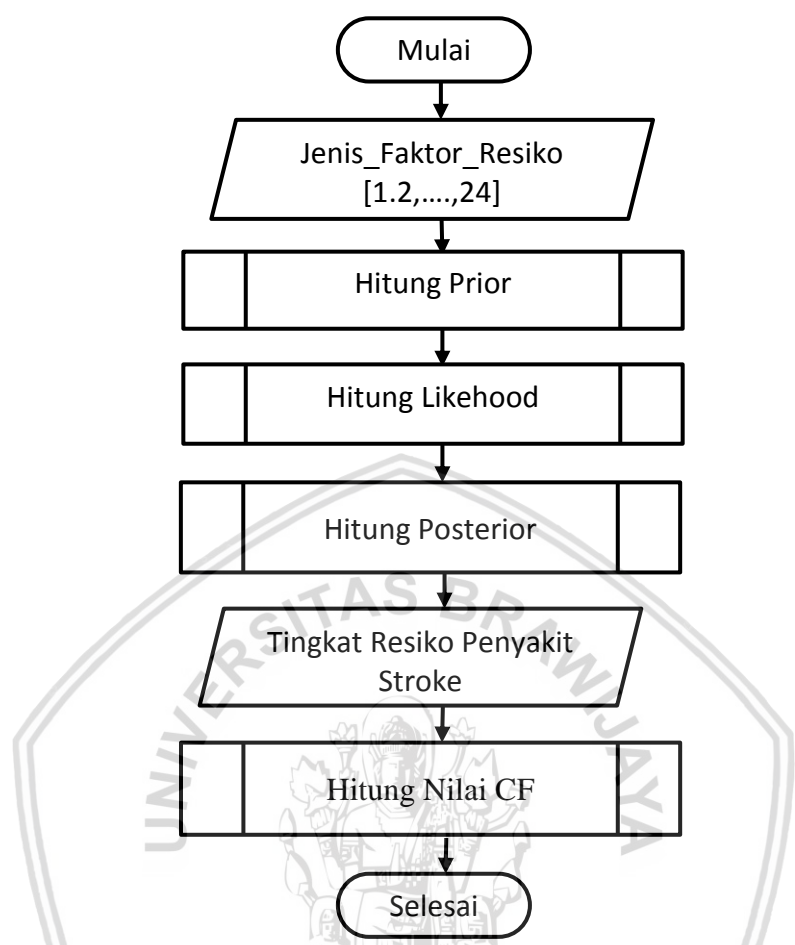
Mesin inferensi adalah sebuah program yang berfungsi untuk memadu proses penalaran terhadap suatu kondisi berdasarkan basis pengetahuan yang ada, memanipulasi dan mengarahkan suatu kaidah, model, dan fakta yang

disimpan dalam basis pengetahuan untuk mencapai suatu solusi dan kesimpulan. Dalam prosesnya, mesin inferensi menggunakan strategi pengendalian yaitu strategi yang berfungsi sebagai panduan arah dalam melakukan suatu proses penalaran.

Sedangkan sebelum masuk ke dalam proses inferensi sistem untuk penelusuran jawabannya menggunakan *forward chaining*. Penelusuran ini dimulai dengan mengumpulkan fakta mengenai jenis-jenis tingkat resiko seseorang terserang penyakit yang diberikan oleh pengguna sebagai masukan oleh sistem, kemudian dilakukan perhitungan dengan menggunakan metode *Naive bayes - Certainty factor* sampai dengan kesimpulan akhir berupa hasil deteksi untuk tingkat resiko kemungkinan terkena penyakit Stroke.

Proses inferensi sistem pakar dapat diilustrasikan pada Gambar 4.8, perhitungan dimulai dengan memasukkan nilai skoring tiap-tiap gejala penyakit Stroke oleh pakar yang kemudian disimpan di dalam basis data sebagai dasar perhitungan. Pengguna akan memasukkan gejala-gejala fakta yang terjadi sesuai dengan rekam medis pasien kedalam program. Didalam program tersebut akan dilakukan proses pencocokan dari gejala yang dimasukkan dengan data yang ada pada basis data sehingga akan didapatkan gejala dan skoring nilai dari tiap gejala tersebut.

Perhitungan *Naive bayes* dimulai dengan menghitung nilai probabilitas *prior* atau peluang kemunculan tingkat resiko penyakit pada data *training*, kemudian sistem akan menghitung probabilitas *likelihood* atau peluang munculnya jenis-jenis penyakit terhadap suatu tingkat resiko penyakit stroke. Setelah itu sistem akan menghitung probabilitas *posterior* atau peluang terakhir, jika semua dari tingkat resiko terserang penyakit stroke (tinggi, sedang, rendah) telah dihitung nilai probabilitas *posterior*nya, maka akan dilakukan perbandingan dimana yang memiliki nilai *posterior* tertinggi yang akan menjadi hasil akhir dari deteksi dari tingkat resiko terserang penyakit stroke. Untuk proses perhitungan *Certainty factor* dilakukan dengan menggunakan rumus *CF combine* untuk mendapatkan nilai di setiap aturan. Perhitungan *CF combine* menggunakan nilai *CF* yang diperoleh oleh pakar. Nilai yang sudah didapatkan tersebut akan digunakan untuk memberikan bobot pada setiap aturan yang ada dimana aturan yang memiliki nilai tertinggi akan dilihat data tingkat resikonya, kemudian setelah data tersebut ditemukan maka bisa disimpulkan bahwa hasil tersebut bisa digunakan sebagai hasil deteksi untuk tingkat resiko terserang penyakit stroke. Diagram alir sistem dengan metode *Naive bayes - Certainty factor* ditunjukkan pada Gambar 4.9.



Gambar 4.9 Diagram Alir Sistem Metode *Naive bayes* – *Certainty factor*

4.3.3.1 Contoh Perhitungan Manual dengan Metode *Naive bayes* – *Certainty factor*

Pada Tabel 4.3 dapat dijadikan acuan sebagai beberapa data masukan berupa fakta jenis penyakit yang memengaruhi resiko terserang penyakit stroke oleh pengguna. Jenis-jenis penyakit yang dimasukkan antara lain:

1. Tekanan Darah
2. Diabetes
3. Riwayat Keluarga
4. Merokok
5. Kolesterol
6. Aktifitas Fisik
7. Diet
8. Berat Badan
9. Tinggi badan



10. Riwayat Fibrilasi Atrium

Sesuai dengan analisis jenis-jenis penyakit yang dimasukkan oleh pengguna maka dapat dilakukan proses perhitungan dengan menggunakan metode *Naive bayes* seperti berikut:

Jika diketahui suatu masukan gejala penyakit stroke seperti berikut:

- Tekanan Darah : Rendah
- Diabetes : 120 (rendah)
- Riwayat Keluarga : Tidak Tahu
- Merokok : Tidak merokok
- Kolesterol : 183 (rendah)
- Aktifitas fisik : Rutin olahraga
- Diet : 20,57 (Normal)
- Riwayat fibrilasi atrium : Beraturan

1. Langkah pertama

Menghitung nilai probabilitas sesuai dengan kondisi derajat keanggotaannya dengan melakukan pencarian nilai derajat anggotanya dengan mencari jumlah pada tiap faktor.

Jumlah keseluruhan data latih yakni 120 yang dapat dilihat pada Tabel 4.11. Berikut merupakan tabel perhitungan untuk metode *naive bayes* yang sesuai dengan Tabel 4.22 sampai dengan Tabel 4.29.

Tabel 4.22 Tabel Perhitungan Tiap Faktor Resiko

Parameter Faktor Resiko $P_{1...P_n}$	Probabilitas	Hasil
P(tekanan_darah rendah)	42/120	0,35
P(diabetes rendah)	17/120	0,142
P(riwayat_klrg tidak_tahu)	81/120	0,675
P(merokok tidak)	14/120	0,117
P(kolesterol rendah)	44/120	0,367
P(aktifitas_fisik rutin_olahraga)	34/120	0,283
P(diet normal)	18/120	0,15
P(riwayat_fb beraturan)	70/120	0,583

2. Langkah Kedua

Menghitung nilai *prior* atau peluang kemunculan suatu penyakit pada data training berdasarkan jenis-jenis penyakit yang dimasukkan. Perhitungan ini sesuai dengan Tabel 4.4 yakni dengan membagi jumlah masing-masing tingkat resiko

dengan jumlah keseluruhan data yang ada pada data latih. Berikut merupakan contoh perhitungan *prior* dengan menggunakan Persamaan (2.6).

$$P(R1) = \text{Jumlah Tingkat Resiko tinggi/Keseluruhan data}$$

$$= 16/120$$

$$= 0,133$$

$$P(R2) = \text{Jumlah Tingkat Resiko sedang/Keseluruhan data}$$

$$= 72/120$$

$$= 0,6$$

$$P(R3) = \text{Jumlah Tingkat Resiko rendah/Keseluruhan data}$$

$$= 32/120$$

$$= 0,2667$$

Berdasarkan perhitungan *prior* dari masing-masing tingkat resiko penyakit diatas, maka hasil dari proses perhitungan nilai *prior* ini ditunjukkan pada Tabel 4.23.

Tabel 4.23 Nilai *Prior*

No.	P(Tingkat Resiko)	Nilai <i>Prior</i>
1	Tinggi	0,133
2	Sedang	0,6
3	Rendah	0,2667

3. Langkah Ketiga

Melakukan pencarian nilai *likelihood* atau peluang munculnya suatu jenis penyakit terhadap tingkat resiko terserang penyakit stroke dari probabilitas yang memengaruhi pada setiap tingkat resikonya. Proses perhitungannya dilakukan dengan membagi jumlah jenis penyakit yang ada pada masing-masing tingkat resiko dengan jumlah masing-masing tingkat resiko yang sesuai dengan Tabel 4.18. berikut adalah contoh perhitungan untuk mencari *likelihood* dengan menggunakan persamaan (2.7) yang ditunjukkan pada Tabel 4.24.

Tabel 4.24 Nilai Perhitungan *Likelihood* kelas Stroke Tinggi

Faktor Resiko Kelas Stroke Tinggi	Jumlah Faktor Resiko Kelas Tinggi	Jumlah Kelas Stroke Tinggi	<i>Likelihood</i> Kelas Stroke Tinggi
Tekanan Darah (rendah)	1	16	1/16= 0,0625
Diabetes (rendah)	1	16	1/16= 0,0625
Riwayat Keluarga (tidak tahu)	9	16	9/16= 0,5625

Merokok (tidak merokok)	1	16	$1/16 = 0,0625$
Kolesterol (rendah)	3	16	$3/16 = 0,1875$
Aktifitas fisik (rutin olahraga)	1	16	$1/16 = 0,0625$
Diet (normal)	1	16	$1/16 = 0,0625$
Riwayat fibrilasi atrium (beraturan)	1	16	$1/16 = 0,0625$

Tabel 4.25 Nilai Perhitungan *Likelihood* kelas Stroke Sedang

Faktor Resiko Kelas Stroke Tinggi	Jumlah Faktor Resiko Kelas Sedang	Jumlah Kelas Stroke Sedang	<i>Likelihood</i> Kelas Stroke Sedang
Tekanan Darah (rendah)	24	72	$1/72 = 0,333$
Diabetes (rendah)	36	72	$1/72 = 0,5$
Riwayat Keluarga (tidak tahu)	72	72	$9/72 = 1$
Merokok (tidak merokok)	0	72	$1/72 = 0$
Kolesterol (rendah)	24	72	$3/72 = 0,333$
Aktifitas fisik (rutin olahraga)	0	72	$1/72 = 0$
Diet (normal)	1	72	$1/72 = 0,0138$
Riwayat fibrilasi atrium (beraturan)	36	72	$1/72 = 0,5$

Tabel 4.26 Nilai Perhitungan *Likelihood* kelas Stroke Rendah

Faktor Resiko Kelas Stroke Tinggi	Jumlah Faktor Resiko Kelas Tinggi	Jumlah Kelas Stroke Tinggi	<i>Likelihood</i> Kelas Stroke Tinggi
Tekanan Darah (rendah)	16	32	$1/32 = 0,5$
Diabetes (rendah)	16	32	$1/32 = 0,5$
Riwayat Keluarga (tidak tahu)	0	32	$9/32 = 0$
Merokok (tidak merokok)	12	32	$1/32 = 0,375$
Kolesterol (rendah)	16	32	$3/32 = 0,5$

Aktifitas fisik (rutin olahraga)	32	32	$1/32= 1$
Diet (normal)	15	32	$1/32= 0,469$
Riwayat fibrilasi atrium (beraturan)	32	32	$1/32= 1$

4. Langkah Keempat

Perhitungan kali ini untuk mencari nilai *posterior* atau probabilitas akhir dari masing-masing penyakit dengan cara mengalikan nilai *prior* dengan nilai *likelihood* dari masing-masing jenis resiko pada setiap tingkat resiko yang sesuai dengan Tabel 4.20. Berikut merupakan contoh perhitungan *posterior* dengan menggunakan Persamaan (2.8).

Berikut ini merupakan proses perhitungan *posterior* untuk tingkat resiko tinggi yang mengacu pada Tabel 4.27.

Tabel 4.27 Nilai Perhitungan *Posterior* kelas Stroke Tinggi

	<i>Prior</i> Resiko Stroke tinggi	Perkalian	<i>Likelihood</i> resiko Stroke Tinggi
<i>Posterior</i> Resiko Stroke Tinggi	0,133	*	0,0625
			0,0625
			0,5625
			0,0625
			0,1875
			0,0625
			0,0625
			0,0625
Hasil	0,000,000,000,838		

Berikut ini merupakan proses perhitungan *posterior* untuk tingkat resiko sedang yang mengacu pada Tabel 4.28.

Tabel 4.28 Nilai Perhitungan *Likelihood* kelas Stroke Sedang

	<i>Prior</i> Resiko Stroke Sedang	Perkalian	<i>Likelihood</i> resiko Stroke Sedang
			0,333



<i>Posterior</i> Resiko Stroke Sedang	0,6	*	0,5
			1
			0
			0,333
			0
			0,0138
			0,5
Hasil	0		

Berikut ini merupakan proses perhitungan *posterior* untuk tingkat resiko rendah yang mengacu pada Tabel 4.29.

Tabel 4.29 Nilai Perhitungan *Posterior* kelas Stroke Rendah

	<i>Prior</i> Resiko Stroke Rendah	Perkalian	<i>Likelihood</i> resiko Stroke Rendah
<i>Posterior</i> Resiko Stroke Rendah	0,2667	*	0,5
			0,5
			0
			0,375
			0,5
			1
			0,469
			1
Hasil	0		

Berdasarkan proses perhitungan *posterior* metode *Naive bayes* sesuai dengan fakta yang telah dimasukkan oleh *user*, maka dapat dihasilkan nilai *posterior* yang ditunjukkan pada Tabel 4.30.

Tabel 4.30 Nilai *Posterior*

No.	Tingkat resiko	Nilai <i>Posterior</i>
1	Tinggi	0,000,000,000,838
2	Sedang	0
3	Rendah	0

Hasil perhitungan *posterior* akan dilakukan pengurutan dari nilai tertinggi ke nilai yang terendah. Hasil pengurutan tersebut dapat dilihat pada Tabel 4.31.

Tabel 4.31 Nilai *Posterior* Tertinggi

No.	Tingkat resiko	Nilai <i>Posterior</i>
1	Tinggi	0,000,000,000,838
2	Sedang	0
3	Rendah	0

Berdasarkan hasil pengurutan nilai *posterior* yang telah dilakukan diatas maka dapat disimpulkan bahwa nilai probabilitas akhir maksimum yang ada pada tingkat resiko tinggi sebesar 0,000,000,000,838. Hasil deteksi tingkat resiko penyakit stroke sesuai dengan fakta jenis resiko untuk tingkat resiko penyakit stroke yakni Tinggi.

Setelah menghasilkan deteksi tingkat resiko penyakit stroke menggunakan metode *Naive bayes* maka selanjutnya akan dilakukan proses perhitungan nilai kepastian dengan menggunakan metode *Certainty factor*. Pada Tabel 4.16 terdapat nilai bobot CF pakar untuk setiap jenis resiko dari masing-masing penyakit. Dari nilai bobot itu akan dihitung nilai CF terbesar sesuai dengan jenis resiko yang ada di setiap tingkat resikonya. Perhitungan ini diharapkan akan menghasilkan hasil keluaran CF maksimal yakni mendekati nilai 1, dan mengetahui *presentase* dari hasil perhitungan metode *Naive Bayes*.

1. Langkah pertama

Tingkat resiko yang akan dihitung nilai kepastiannya merupakan penyakit dari hasil deteksi menggunakan metode *Nive Bayes* yakni tingkat resiko Sedang. Data jenis resiko yang ada diambil dari fakta jenis resiko yang telah dimasukkan oleh *user*. Nilai CF pakar yang digunakan yakni nilai CF pakar pada tingkat resiko Sedang sesuai Tabel 4.21 untuk menghasilkan nilai CF maka nilai CF pakar dikalikan dengan nilai CF *user*. Nilai CF pakar pada tingkat resiko Tinggi ditunjukkan pada Tabel 4.32, untuk nilai CF *user* ditunjukkan pada Tabel 4.33.

Tabel 4.32 Nilai CF Pakar Tingkat Resiko Terserang Stroke Tinggi

No	Jenis Resiko	Nilai CF Pakar
		Tinggi
1	Tekanan Darah(tinggi)	0.5
2	Tekanan Darah(sedang)	0.2
3	Tekanan Darah(rendah)	0.3
4	Diabetes(tinggi)	0.5

5	Diabetes(sedang)	0.2
6	Diabetes(rendah)	0.2
7	riwayat keluarga(iya)	0.2
8	riwayat keluarga(tidak tahu)	0.4
9	riwayat keluarga(tidak)	0.3
10	merokok(iya)	0.5
11	merokok(tidak)	0.2
12	merokok(kadang)	0
13	kolesterol(tinggi)	0.4
14	kolesterol(sedang)	0.2
15	kolesterol(rendah)	0.3
16	aktifitas fisik(iya)	0.2
17	aktifitas fisik(kadang)	0.3
18	aktifitas fisik(tidak)	0.5
19	Diet(normal)	0.2
20	Diet(sedang)	0
21	diet(tinggi)	0.5
22	riwayat fibrilasi(beraturan)	0.2
23	riwayat fibrilasi(tidak tahu)	0.3
24	riwayat fibrilasi(tidak beraturan)	0.4

Tabel 4.33 Nilai CF User

No	Jenis Resiko	Nilai CF User
1	Tekanan Darah(tinggi)	0
2	Tekanan Darah(sedang)	0
3	Tekanan Darah(rendah)	1
4	Diabetes(tinggi)	0
5	Diabetes(sedang)	0
6	Diabetes(rendah)	1
7	riwayat keluarga(iya)	0
8	riwayat keluarga(tidak tahu)	1

9	riwayat keluarga(tidak)	0
10	merokok(iya)	0
11	merokok(tidak)	1
12	merokok(kadang)	0
13	kolesterol(tinggi)	0
14	kolesterol(sedang)	0
15	kolesterol(rendah)	1
16	aktifitas fisik(iya)	1
17	aktifitas fisik(kadang)	0
18	aktifitas fisik(tidak)	0
19	Diet(normal)	1
20	Diet(sedang)	0
21	diet(tinggi)	0
22	riwayat fibrilasi(beraturan)	1
23	riwayat fibrilasi(tidak tahu)	0
24	riwayat fibrilasi(tidak beraturan)	0

Untuk mendapatkan nilai CF maka akan dilakukan proses perkalian antara nilai CF Pakar dengan Nilai CF *User*.

$$CF(H,E) = CF \text{ Pakar} \times CF \text{ User}$$

$$CF(1) = 0,5 \times 0$$

$$= 0$$

$$CF(2) = 0,2 \times 0$$

$$= 0$$

$$CF(3) = 0,3 \times 1$$

$$= 0,3$$

$$CF(4) = 0,5 \times 0$$

$$= 0$$

$$CF(5) = 0,2 \times 0$$

$$= 0$$

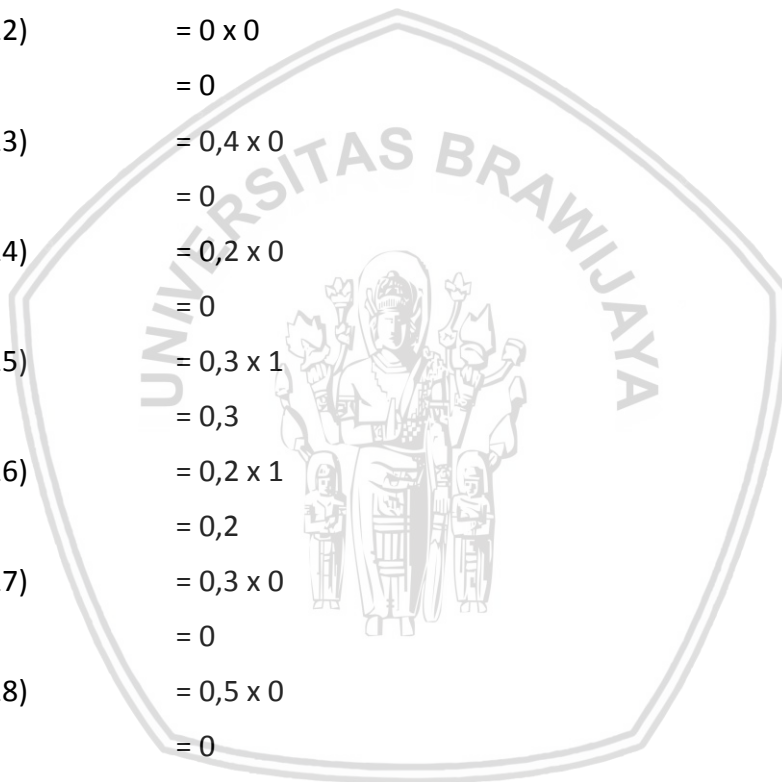
$$CF(6) = 0,2 \times 1$$

$$= 0,2$$

$$CF(7) = 0,2 \times 0$$



- = 0
- CF(8) = 0,4 x 1
- = 0,4
- CF(9) = 0,3 x 0
- = 0
- CF(10) = 0,5 x 0
- = 0
- CF(11) = 0,2 x 1
- = 0,2
- CF(12) = 0 x 0
- = 0
- CF(13) = 0,4 x 0
- = 0
- CF(14) = 0,2 x 0
- = 0
- CF(15) = 0,3 x 1
- = 0,3
- CF(16) = 0,2 x 1
- = 0,2
- CF(17) = 0,3 x 0
- = 0
- CF(18) = 0,5 x 0
- = 0
- CF(19) = 0,2 x 1
- = 0,2
- CF(20) = 0 x 0
- = 0
- CF(21) = 0,5 x 0
- = 0
- CF(22) = 0,2 x 1
- = 0,2
- CF(23) = 0,3 x 0



$$\begin{aligned}
 &= 0 \\
 \text{CF(24)} &= 0,4 \times 0 \\
 &= 0
 \end{aligned}$$

2. Langkah Kedua

Setelah nilai CF telah dihitung langkah selanjutnya adalah dengan menghitung nilai CF *Combine*. Perhitungan CF *Combine* ini menggunakan nilai CF yang didapatkan dari CF Pakar dikali CF *User*, untuk nilai CF dianggap sebagai nilai CF1 dan CF2, setiap ada eksekusi hanya menggunakan dua jenis data saja (CF1 dan CF2). Nilai tersebut digunakan untuk memberikan bobot pada setiap aturan yang berlaku. Berikut merupakan contoh perhitungan CF *Combine*.

$$\begin{aligned}
 \text{CF(A)} &= \text{CF(1)} + \text{CF(2)} * (1 - \text{CF(1)}) \\
 &= 0 + 0 \times (1 - 0) \\
 &= 0
 \end{aligned}$$

$$\begin{aligned}
 \text{CF(B)} &= \text{CF(3)} + \text{CF(A)} * (1 - \text{CF(3)}) \\
 &= 0,3 + 0 \times (1 - 0,3) \\
 &= 0,3
 \end{aligned}$$

$$\begin{aligned}
 \text{CF(C)} &= \text{CF(4)} + \text{CF(B)} * (1 - \text{CF(4)}) \\
 &= 0 + 0,3 \times (1 - 0) \\
 &= 0,3
 \end{aligned}$$

$$\begin{aligned}
 \text{CF(D)} &= \text{CF(5)} + \text{CF(C)} * (1 - \text{CF(5)}) \\
 &= 0 + 0,3 \times (1 - 0) \\
 &= 0,3
 \end{aligned}$$

$$\begin{aligned}
 \text{CF(E)} &= \text{CF(6)} + \text{CF(D)} * (1 - \text{CF(6)}) \\
 &= 0,2 + 0,3 \times (1 - 0,2) \\
 &= 0,44
 \end{aligned}$$

$$\begin{aligned}
 \text{CF(F)} &= \text{CF(7)} + \text{CF(E)} * (1 - \text{CF(7)}) \\
 &= 0 + 0,44 \times (1 - 0) \\
 &= 0,44
 \end{aligned}$$

$$\begin{aligned}
 \text{CF(G)} &= \text{CF(8)} + \text{CF(F)} * (1 - \text{CF(8)}) \\
 &= 0,4 + 0,44 \times (1 - 0,4) \\
 &= 0,664
 \end{aligned}$$

$$\begin{aligned}
 \text{CF(H)} &= \text{CF(9)} + \text{CF(G)} * (1 - \text{CF(9)}) \\
 &= 0 + 0,664 \times (1 - 0) \\
 &= 0,664
 \end{aligned}$$

$$\begin{aligned}CF(I) &= CF(10)+CF(H)*(1-CF(10)) \\ &= 0 + 0,664 \times (1-0) \\ &= 0,664\end{aligned}$$

$$\begin{aligned}CF(J) &= CF(11)+CF(I)*(1-CF(11)) \\ &= 0,2 + 0,664 \times (1-0,2) \\ &= 0,7312\end{aligned}$$

$$\begin{aligned}CF(K) &= CF(12)+CF(J)*(1-CF(12)) \\ &= 0+ 0,7312 \times (1-0) \\ &= 0,7312\end{aligned}$$

$$\begin{aligned}CF(L) &= CF(13)+CF(K)*(1-CF(13)) \\ &= 0 + 0,7312 \times (1-0) \\ &= 0,7312\end{aligned}$$

$$\begin{aligned}CF(M) &= CF(14)+CF(L)*(1-CF(14)) \\ &= 0 + 0,7312 \times (1-0) \\ &= 0,7312\end{aligned}$$

$$\begin{aligned}CF(N) &= CF(15)+CF(M)*(1-CF(15)) \\ &= 0,3 + 0,7312 \times (1-0,3) \\ &= 0,81184\end{aligned}$$

$$\begin{aligned}CF(O) &= CF(16)+CF(N)*(1-CF(16)) \\ &= 0,2 + 0,81184 \times (1-0,2) \\ &= 0,849\end{aligned}$$

$$\begin{aligned}CF(P) &= CF(17)+CF(O)*(1-CF(17)) \\ &= 0 + 0,849 \times (1-0) \\ &= 0,849\end{aligned}$$

$$\begin{aligned}CF(Q) &= CF(18)+CF(P)*(1-CF(18)) \\ &= 0 + 0,849 \times (1-0) \\ &= 0,849\end{aligned}$$

$$\begin{aligned}CF(R) &= CF(19)+CF(Q)*(1-CF(19)) \\ &= 0,2 + 0,849 \times (1-0,2) \\ &= 0,879\end{aligned}$$

$$\begin{aligned}CF(S) &= CF(20)+CF(R)*(1-CF(20)) \\ &= 0 + 0,879 \times (1-0)\end{aligned}$$

$$\begin{aligned}
 &= 0,879 \\
 \text{CF(T)} &= \text{CF(21)} + \text{CF(S)} * (1 - \text{CF(21)}) \\
 &= 0 + 0,879 * (1 - 0) \\
 &= 0,879 \\
 \text{CF(U)} &= \text{CF(22)} + \text{CF(T)} * (1 - \text{CF(22)}) \\
 &= 0,2 + 0,879 * (1 - 0,2) \\
 &= 0,9032 \\
 \text{CF(V)} &= \text{CF(23)} + \text{CF(U)} * (1 - \text{CF(23)}) \\
 &= 0 + 0,9032 * (1 - 0) \\
 &= 0,9032 \\
 \text{CF(W)} &= \text{CF(24)} + \text{CF(V)} * (1 - \text{CF(24)}) \\
 &= 0 + 0,9032 * (1 - 0) \\
 &= 0,9032
 \end{aligned}$$

Berdasarkan hasil perhitungan CF *Combine* maka didapatkan nilai keyakinan Tingkat Resiko seseorang terserang penyakit stroke sedang adalah sebesar 0,9032 atau 90,3%.

4.3.4 Blackboard

Daerah kerja (*Blackboard*) merupakan area memori yang berfungsi sebagai basis data untuk merekam hasil sementara suatu keputusan, dengan menyertakan hasil perhitungan akhir sebelum sistem memutuskan kesimpulannya. Pada sistem pakar deteksi dini penyakit stroke dengan menggunakan *metode Naive bayes – Certainty factor*, yang disimpan adalah hasil perhitungan sementara dari metode *Naive bayes – Certainty factor*.

4.3.5 Fasilitas Penjelas

Pada umumnya, fasilitas penjelas berisi tuntunan penggunaan aplikasi sistem pakar dan bagaimana kesimpulan bisa diambil. Fasilitas penjelas yang akan diterapkan pada sistem pakar deteksi dini penyakit stroke dengan menggunakan metode *Naive bayes – Certainty factor* yaitu dengan memberikan informasi tentang cara penggunaan aplikasi sistem pakar dan informasi hasil perhitungan dari hasil proses deteksinya. Fasilitas penjelas ini penting untuk memberikan informasi kepada pengguna mengenai cara penggunaan dan fitur-fitur apa saja yang ada pada aplikasi sistem pakar.

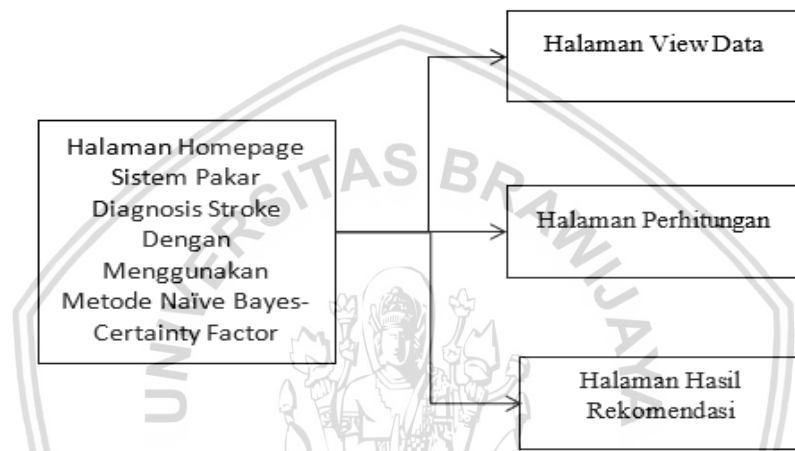
4.3.6 Antarmuka Pengguna

Antarmuka merupakan mekanisme yang digunakan oleh pengguna untuk berinteraksi dengan sistem. Antarmuka menerima informasi dari pemakai dan merubahnya kedalam bentuk yang dapat diterima oleh sistem. Antarmuka juga menerima informasi dari sistem dan menyajikan ke dalam bentuk yang dapat

dimengerti oleh pengguna. Pada bagian ini akan dijelaskan mengenai spesifikasi rancangan antarmuka sistem pakar deteksi dini penyakit stroke dengan menggunakan metode *Naive bayes – Certainty factor*.

Subsistem Antarmuka Pengguna merupakan subsistem yang bertujuan sebagai sarana komunikasi antara pengguna dan sistem. Desain antarmuka berperan penting pada sebuah sistem karena bagian ini membuat gambaran awal mengenai halaman antarmuka kepada pengguna. Perancangan antarmuka pengguna sistem ini dijelaskan pada alur *sitemap* dan desain antarmuka tiap-tiap halaman.

Sistem ini terbagi atas empat halaman sesuai banyaknya proses yang berjalan. Halaman *homepage*, halaman view data, halaman perhitungan, dan halaman hasil rekomendasi. *Site map* sistem ditunjukkan pada Gambar 4.10.



Gambar 4.10 Site Map Sistem

Penjelasan dari Sitemap sistem akan dijelaskan sebagai berikut:

1. Rancangan Menu Utama

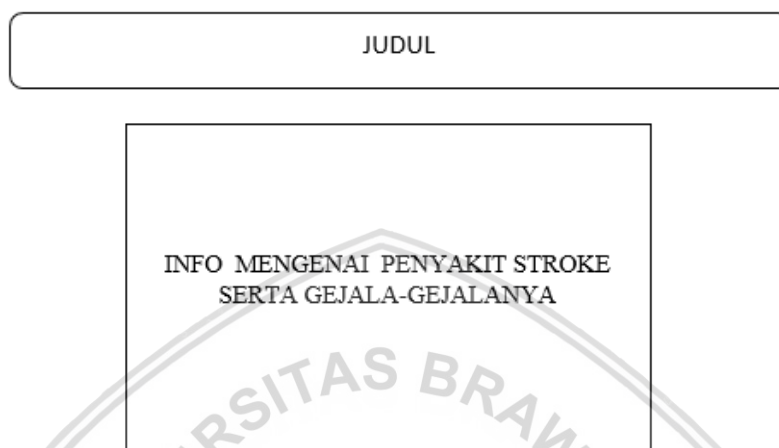
Menu utama adalah *form* halaman pertama yang akan tampil ketika membuka aplikasi Sistem Pakar Deteksi Dini Penyakit Stroke dengan Menggunakan Metode *Naive bayes - Certainty factor*. *User* dapat melakukan aktifitas seperti melihat info-info, mencoba mendeteksi penyakit, dan juga melihat info-info tentang penyakit stroke. Rancangan halaman utama ditunjukkan pada Gambar 4.11.



Gambar 4.11 Halaman Menu Utama

2. Menu Info Penyakit

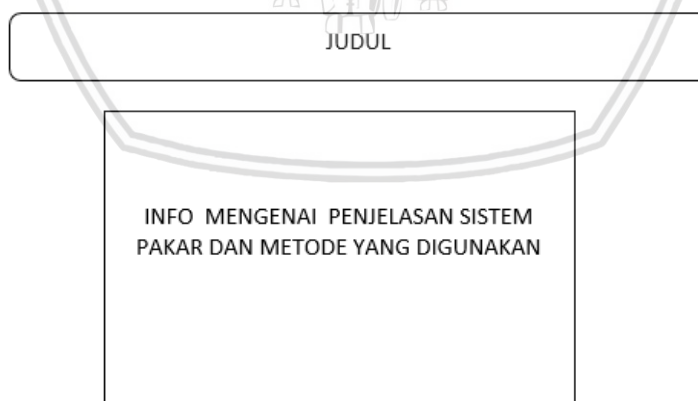
Halaman info penyakit ini berisi tentang informasi penyakit dari penyakit stroke itu sendiri dan juga informasi dari beberapa penyebab seseorang terserang penyakit stroke. Rancangan halaman info penyakit ditunjukkan pada Gambar 4.12.



Gambar 4.12 Halaman Menu Info Penyakit

3. Rancangan Menu Info Perangkat

Menu info perangkat ini digunakan oleh *user* untuk mengetahui informasi tentang aplikasi sistem pakar deteksi dini penyakit stroke yang dibangun. Di dalam menu tentang ini pengertian tentang apa itu sistem pakar, dan juga metode apa yang digunakan untuk membangun aplikasi tersebut. Rancangan halaman info perangkat ditunjukkan pada Gambar 4.13.



Gambar 4.13 Halaman Menu Info Perangkat

4. Menu Halaman Mulai Deteksi

Pada menu ini digunakan oleh *user* untuk mendeteksi tingkat resiko penyakit yang diderita oleh pengguna. *User* akan mengklik mulai deteksi untuk memulai deteksi, setelah itu akan muncul form seperti kuisioner, yaitu menampilkan jenis

resiko yang harus dipilih oleh *user*, ada beberapa pertanyaan yang harus diisi oleh *user*, dengan cara memilih salah satu gejala yang dialami oleh *user*. Setelah menjawab semua pertanyaan, maka ketika dipilih selesai program akan menampilkan hasil deteksi berupa form yang berisikan nama *user*, jenis-jenis resiko yang dialami, penyebab dan solusi yang diberikan pakar. Rancangan halaman info penyakit ditunjukkan pada Gambar 4.14.

JUDUL

1. Pertanyaan 1 :

- Pilihan 1
- Pilihan 2
- Pilihan 3

2. Pertanyaan 2 :

- Pilihan 1
- Pilihan 2
- Pilihan 3

Dst

Gambar 4.14 Halaman Menu Halaman Deteksi

4.4 Perancangan Pengujian

Dalam proses membangun perangkat lunak untuk memperoleh hasil yang baik maka dilakukan pengujian. Pengujian yang dilakukan dalam membangun sistem pakar deteksi dini penyakit stroke dibagi menjadi tiga, yakni pengujian *blackbox*, pengujian *usability*, dan pengujian akurasi.

4.4.1 Pengujian *Blackbox*

Pada pengujian *blackbox* ini digunakan untuk mengetahui apakah sistem yang dibangun sesuai atau tidak dengan kebutuhan. Proses pengujian ini dilakukan dengan menggunakan kasus uji untuk mengetahui apakah kinerja sistem sesuai dengan daftar kebutuhan. Untuk rancangan pengujian *blackbox* ditunjukkan pada Tabel 4.34.

Tabel 4.34 Rancangan Pengujian *Blackbox*

Nomor Kasus Uji	
Nama Kasus Uji	
Objek Kasus Uji	
Tujuan Pengujian	
prosedur Pengujian	
Hasil yang diharapkan	

4.4.2 Pengujian Usability

Pada pengujian *usability* ini ditunjukkan kepada para pengguna yang sudah menguji atau menggunakan sistem dengan cara memberikan angket atau kuisisioner. Setiap kuisisioner ini berisi beberapa pertanyaan yang akan diberikan sebuah nilai yakni satu sampai lima dan totalnya nanti akan dibagi dengan jumlah pertanyaan yang ada. Untuk rancangan pertanyaan kuisisioner ditunjukkan pada Tabel 4.35.

Tabel 4.35 Rancangan Pertanyaan Kuisisioner

No	Pertanyaan	Pilihan				
		SS	S	KS	TS	STS
1	Sistem ini sangat mudah digunakan					
2	Saya suka dengan tampilan sistem ini					
3	Sistem ini mudah dioperasikan					
4	Bahasa yang digunakan didalam sistem mudah dimenegerti					
5	Menu yang ada didalam sistem mudah dimengerti					

Keterangan:

SS (Sangat Setuju): 5

S (Setuju): 4

KS (Kurang Setuju): 3

TS (Tidak Setuju): 2

STS (Sangat Tidak Setuju): 1

4.4.3 Pengujian Akurasi

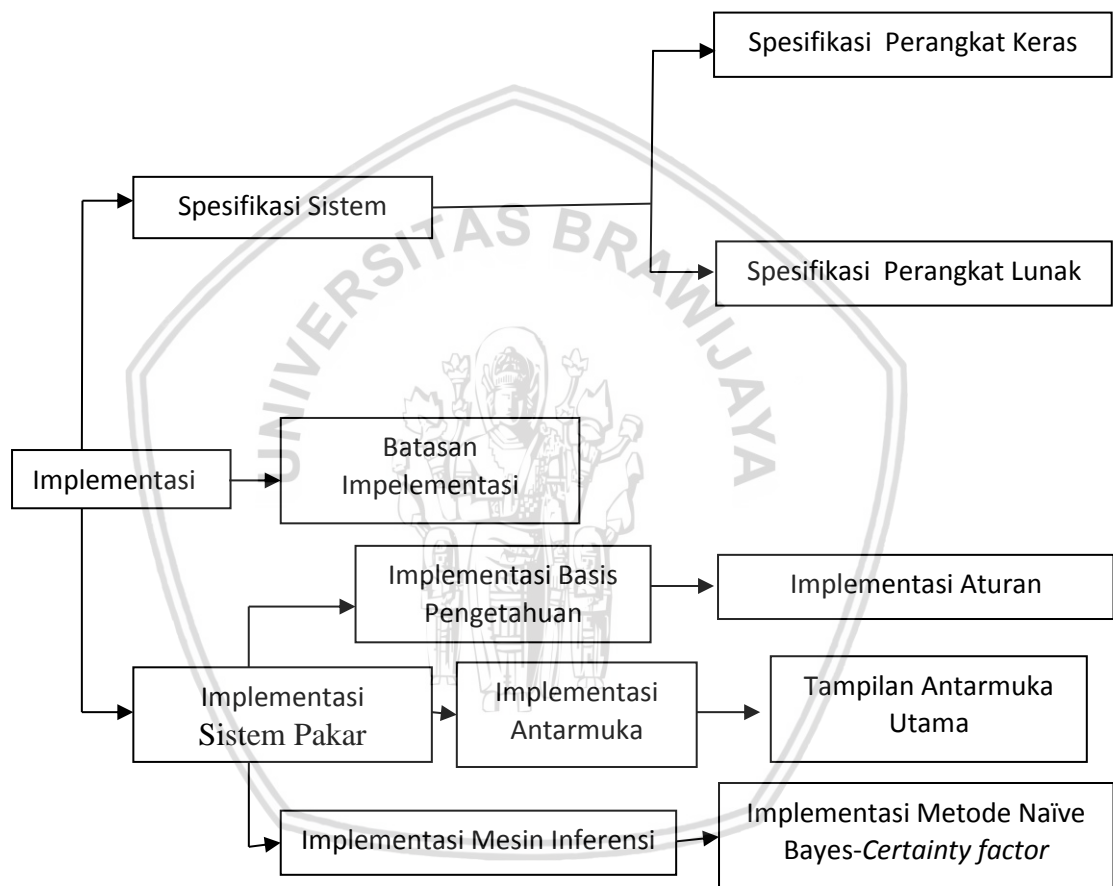
Pada pengujian akurasi ini akan dilakukan sebuah perbandingan hasil yang diperoleh dari sistem dengan yang diperoleh dari pakar. Untuk data yang diperoleh langsung dari pakar di dapatkan sebanyak 25 data uji. Untuk rancangan pengujian akurasi ditunjukkan pada Tabel 4.36.

Tabel 4.36 Rancangan Pengujian Akurasi

No	Faktor Resiko	Deteksi Pakar	Deteksi Sistem	Nilai CF (%)

BAB 5 IMPLEMENTASI

Bab ini membahas mengenai proses implementasi perangkat lunak sesuai dengan hasil yang telah diperoleh dari proses analisis kebutuhan dan perancangan sistem yang telah dibuat. Dalam pembahasan ini terdiri dari penjelasan tentang spesifikasi sistem, Batasan-batasan dalam implementasi, implemementasi algoritme dalam program, impelementasi antarmuka, untuk lebih jelasnya terdapat pohon implementasi sistem pakar yang dapat dilihat pada Gambar 5.1.



Gambar 5.1 Pohon Implementasi

1.1 Spesifikasi Sistem

Hasil dari analisis kebutuhan dan perancangan sistem yang telah dijelaskan sebelumnya pada bab perancangan menjadi patokan dalam melakukan implementasi menjadi sistem yang berfungsi sesuai dengan kebutuhan. Spesifikasi sistem dijelaskan menjadi spesifikasi perangkat keras dan spesifikasi perangkat lunak.



1.1.1 Spesifikasi Perangkat Keras

Tabel 5.1 Spesifikasi Perangkat Keras

Nama Komponen	Spesifikasi
Prosesor	Intel® Core™ i7-3632QM CPU @ 2.20GHz
Memori (RAM)	8,00 GB
Jenis Sistem	Sistem Operasi 64-bit, prosesor berbasis x64
Hardisk	1000 GB

Proses membangun Sistem Pakar Deteksi Dini Penyakit Stroke menggunakan komputer/ laptop dengan spesifikasi perangkat keras yang ditunjukkan pada Tabel 5.1.

1.1.2 Spesifikasi Perangkat Lunak

Proses membangun Sistem Pakar Deteksi Dini Penyakit Stroke menggunakan komputer/ laptop dengan spesifikasi perangkat lunak yang ditunjukkan pada Tabel 5.2.

Tabel 5.2 Spesifikasi Perangkat Lunak

Nama Komponen	Spesifikasi
Sistem Operasi	Microsoft Windows 7 Ultimate
Bahasa Pemrograman	Java
Aplikasi Pemrograman	Android Studio
Emulator	HAXM

1.2 Batasan Implementasi

Batasan dalam implementasi Sistem Pakar Deteksi Dini Penyakit Stroke adalah sebagai berikut:

1. Sistem yang dibangun menggunakan aplikasi android studio dan menggunakan bahasa pemrograman java.
2. Data yang digunakan dalam sistem disimpan dalam bentuk *array*, yakni berupa data-data faktor-faktor yang menyebabkan seseorang terserang penyakit stroke.
3. Masukkan yang dilakukan oleh pengguna ke dalam sistem yakni berupa pilihan resiko-resiko seseorang terserang penyakit stroke
4. Keluaran dari sistem ini adalah salah satu dari 3 tingkat resiko seseorang terserang penyakit stroke.
5. Metode yang digunakan adalah *Naïve Bayes-certainty factor*.

6. Semua pengguna dapat mengakses sistem tanpa harus melakukan login terlebih dahulu dan memiliki hak akses yang sama.
7. Semua pengguna berhak mengakses menu yang ada pada sistem.
8. Data yang digunakan pada sistem adalah permanen, yang artinya data tersebut tidak bias diubah lagi.

1.3 Implementasi Sistem Pakar

Hasil dari perancangan yang sudah dijelaskan sebelumnya pada bab perancangan dijadikan sebagai patokan dalam melakukan proses implementasi Sistem Pakar Deteksi Dini Penyakit Stroke. Beberapa bagian yang akan diimplementasikan adalah implementasi basis pengetahuan, implementasi mesin inferensi dan implementasi antarmuka.

1.3.1 Implementasi Basis Pengetahuan

Pada bagian ini membahas tentang implementasi basis aturan yang akan digunakan dalam sistem pakar ini.

1.3.1.1 Implementasi Basis Aturan

Pada bagian ini akan mengarah ke sub bab basis pengetahuan. Basis aturan ini sesuai dengan data yang diberikan oleh pakar dan digunakan untuk mendapatkan nilai *prior* dan nilai *likelihood*.

1.3.2 Implementasi Mesin Inferensi

Dalam proses membangun Sistem Pakar Deteksi Dini Penyakit Stroke ini terdapat tiga tahapan dalam proses perhitungannya, yang pertama adalah proses mencari nilai *prior* yang didapatkan dari peluang suatu tingkat resiko, tahap yang kedua adalah mencari nilai *likelihood* yang didapatkan dari peluang darimasing-masing tingkat resiko sesuai dengan masukkan yang dilakukan pengguna. Yang ketiga adalah proses mencari nilai *posterior* yang didapatkan dari perkalian dari nilai *prior* dan nilai *likelihood* dari masing-masing tingkat resiko. Untuk proses *Certainty factor* sendiri ini yakni untuk mencari nilai kepastian dari hasil yang didapatkan dari proses perhitungan *naïve Bayes*.

1.3.2.1 Implementasi Algoritme *Naïve Bayes-Certainty Factor*

Implementasi Algoritme dalam proses mendeteksi penyakit stroke mengacu pada proses perhitungan *Naive bayes-Certainty Factor*. Proses mendeteksi pertama dilakukan dengan melihat kondisi kesehatan dari pengguna, kemudian pengguna memasukkan kondisi gejala dalam diri pengguna apa saja untuk dilakukan proses pendeteksian sesuai dengan masukkan yang ada.

Implementasi algoritme proses mendeteksi dengan proses perhitungan *Naïve Bayes-Certainty factor* dapat dilihat pada Kode Program 5.1 dan 5.2.

Kode Program 5.1 Algoritme *Naïve Bayes*

1	<pre>public void Hitung_Jumlah_Kemunculan_Status_Gejala (String status_tek_darah, String status_diabetes, String riwayat_keluarga,</pre>
---	--

```
2      String merokok, String status_kolesterol, String
3      aktifitas_fisik,
4      String status_diet, String riwayat_fibrilasi, String
5      data[][]){
6
7      jumlah_status_gejala = new int[8][3];
8      jumlah_status_gejala = isi_data(jumlah_status_gejala);
9      //Hitung jumlah status tekanan_darah
10     for (int i=0; i < data.length; i++){
11         if (data[i][0] == status_tek_darah &&
12         data[i][data[0].length-1] == "Rendah"){
13             jumlah_status_gejala[0][0] += 1;
14         } else
15         if (data[i][0] == status_tek_darah &&
16         data[i][data[0].length-1] == "Sedang"){
17             jumlah_status_gejala[0][1] += 1;
18         } else
19         if (data[i][0] == status_tek_darah &&
20         data[i][data[0].length-1] == "Tinggi"){
21             jumlah_status_gejala[0][2] += 1;
22         }
23     }
24     //Hitung jumlah diabetes
25     for (int i=0; i < data.length; i++){
26         if (data[i][1] == status_diabetes &&
27         data[i][data[0].length-1] == "Rendah"){
28             jumlah_status_gejala[1][0] += 1;
29         } else
30         if (data[i][1] == status_diabetes &&
31         data[i][data[0].length-1] == "Sedang"){
32             jumlah_status_gejala[1][1] += 1;
33         } else
34         if (data[i][1] == status_diabetes &&
35         data[i][data[0].length-1] == "Tinggi"){
36             jumlah_status_gejala[1][2] += 1;
37         }
38     }
39     //Hitung jumlah riwayat keluarga
40     for (int i=0; i < data.length; i++){
41         if (data[i][2].equalsIgnoreCase(riwayat_keluarga) &&
42         data[i][data[0].length-1] == "Rendah"){
43             jumlah_status_gejala[2][0] += 1;
44         } else
45         if (data[i][2].equalsIgnoreCase(riwayat_keluarga) &&
46         data[i][data[0].length-1] == "Sedang"){
47             jumlah_status_gejala[2][1] += 1;
48         } else
49         if (data[i][2].equalsIgnoreCase(riwayat_keluarga) &&
50         data[i][data[0].length-1] == "Tinggi"){
51             jumlah_status_gejala[2][2] += 1;
52         }
53     }
54     //Hitung jumlah merokok
55     for (int i=0; i < data.length; i++){
56         if (data[i][3].equalsIgnoreCase(merokok) &&
57         data[i][data[0].length-1] == "Rendah"){
58             jumlah_status_gejala[3][0] += 1;
59         } else
60         if (data[i][3].equalsIgnoreCase(merokok) &&
61         data[i][data[0].length-1] == "Sedang"){
62             jumlah_status_gejala[3][1] += 1;
63         } else
64         if (data[i][3].equalsIgnoreCase(merokok) &&
65         data[i][data[0].length-1] == "Tinggi"){
66             jumlah_status_gejala[3][2] += 1;
```

```

43     }
44     }
45     //Hitung jumlah kolesterol
46     for (int i=0; i < data.length; i++){
47         if (data[i][4] == status_kolesterol &&
48 data[i][data[0].length-1] == "Rendah"){
49             jumlah_status_gejala[4][0] += 1;
50         } else
51         if (data[i][4] == status_kolesterol &&
52 data[i][data[0].length-1] == "Sedang"){
53             jumlah_status_gejala[4][1] += 1;
54         } else
55         if (data[i][4] == status_kolesterol &&
56 data[i][data[0].length-1] == "Tinggi"){
57             jumlah_status_gejala[4][2] += 1;
58         }
59     }
60     //Hitung jumlah aktifitas fisik
61     for (int i=0; i < data.length; i++){
62         if (data[i][5].equalsIgnoreCase(aktifitas_fisik) &&
63 data[i][data[0].length-1] == "Rendah"){
64             jumlah_status_gejala[5][0] += 1;
65         } else
66         if (data[i][5].equalsIgnoreCase(aktifitas_fisik) &&
67 data[i][data[0].length-1] == "Sedang"){
68             jumlah_status_gejala[5][1] += 1;
69         } else
70         if (data[i][5].equalsIgnoreCase(aktifitas_fisik) &&
71 data[i][data[0].length-1] == "Tinggi"){
72             jumlah_status_gejala[5][2] += 1;
73         }
74     }
75     //Hitung jumlah diet
76     for (int i=0; i < data.length; i++){
77         if (data[i][6] == status_diet && data[i][data[0].length-
78 1] == "Rendah"){
79             jumlah_status_gejala[6][0] += 1;
80         } else
81         if (data[i][6] == status_diet && data[i][data[0].length-
82 1] == "Sedang"){
83             jumlah_status_gejala[6][1] += 1;
84         } else
85         if (data[i][6] == status_diet && data[i][data[0].length-
86 1] == "Tinggi"){
87             jumlah_status_gejala[6][2] += 1;
88         }
89     }
90     //Hitung jumlah riwayat fibrilasi
91     for (int i=0; i < data.length; i++){
92         if (data[i][9].equalsIgnoreCase(riwayat_fibrilasi) &&
93 data[i][data[0].length-1] == "Rendah"){
94             jumlah_status_gejala[7][0] += 1;
95         } else
96         if (data[i][9].equalsIgnoreCase(riwayat_fibrilasi) &&
97 data[i][data[0].length-1] == "Sedang"){
98             jumlah_status_gejala[7][1] += 1;
99         } else
100        if (data[i][9].equalsIgnoreCase(riwayat_fibrilasi) &&
101 data[i][data[0].length-1] == "Tinggi"){
102            jumlah_status_gejala[7][2] += 1;
103        }
104    }
105 }
106 public int[][] getJumlah_status_gejala() {

```

```

84     return jumlah_status_gejala;
85 }
86 private int[][] isi_data(int[][] jumlah_status_gejala) {
87     int status_gejala[][] = new
88     int[jumlah_status_gejala.length][jumlah_status_gejala[0].length]
89     ;
90     for (int i=0; i < status_gejala.length; i++){
91         for (int j=0; j < status_gejala[0].length; j++){
92             status_gejala[i][j] = 0;
93         }
94     }
95     return status_gejala;
96 }
97 public void hitung_likelihood(int jumlah_status_gejala[][], int
98     jlh_rendah, int jlh_sedang, int jlh_tinggi){
99     likelihood = new
100     double[jumlah_status_gejala.length][jumlah_status_gejala[0].leng
101     th];
102     for (int i=0; i < jumlah_status_gejala.length; i++){
103         for (int j=0; j < jumlah_status_gejala[0].length; j++){
104             if (j == 0){
105                 likelihood[i][j] =
106                 (double)jumlah_status_gejala[i][j]/jlh_rendah;
107             } else
108             if (j == 1){
109                 likelihood[i][j] =
110                 (double)jumlah_status_gejala[i][j]/jlh_sedang;
111             } else
112             if (j == 2){
113                 likelihood[i][j] =
114                 (double)jumlah_status_gejala[i][j]/jlh_tinggi;
115             }
116         }
117     }
118     public double[][] getLikelihood(){
119         return likelihood;
120     }
121     public void hitung_posterior(double likelihood[][], double
122     prior_rendah, double prior_sedang, double prior_tinggi){
123         posterior = new double[likelihood[0].length];
124         posterior = inisialisasi(posterior);
125
126         for (int i=0; i < likelihood.length; i++){
127             for (int j=0; j < likelihood[0].length; j++){
128                 posterior[j] *= likelihood[i][j];
129             }
130         }
131         posterior[0] *= prior_rendah;
132         posterior[1] *= prior_sedang;
133         posterior[2] *= prior_tinggi;
134     }
135     public double[] getPosterior() {
136         return posterior;
137     }
138
139     private double[] inisialisasi(double[] posterior) {
140         double pos[] = new double[posterior.length];
141
142         for (int i=0; i < pos.length; i++){
143             pos[i] = 1;
144         }
145         return pos;
146     }
147     public void status_penyakit(double posterior[]){

```

```

125     int indeks_status = 0;
126     double temp = posterior[0];
127
128     for (int i=0; i < posterior.length; i++){
129         if (posterior[i] > temp){
130             temp = posterior[i];
131             indeks_status = i;
132         }
133     }
134     if (indeks_status == 0){
135         status_penyakit = "Rendah";
136     } else
137     if (indeks_status == 1){
138         status_penyakit = "Sedang";
139     } else
140     if (indeks_status == 2){
141         status_penyakit = "Tinggi";
142     }
143 }
144 public String getStatus_penyakit() {
145     return status_penyakit;
146 }
147 }

```

Penjelasan kode program metode *naïve bayes* Kode Program 5.1, yakni:

1. 1-90 kode program untuk menginisialisasi faktor resiko yang dimasukkan oleh pengguna menjadi kelas rendah, sedang, atau tinggi dan menyimpannya untuk dilakukan proses perhitungan nilai *prior*.
2. 91-107 kode program untuk menghitung nilai *likelihood*
3. 108-125 kode program untuk menghitung nilai *posterior*
4. 126-140 kode program untuk menampilkan status akhir dari mendeteksi penyakit stroke.

Kode Program 5.2 Algoritme *Certainty factor*

```

1     public void Cari_CF_Pakar(String status_penyakit, String
2     CF_Pakar[][]){
3         this.temp_nilai_cf_pakar = new String[CF_Pakar.length/3];
4         int a=0;
5         for (int i=0; i < CF_Pakar.length; i++){
6             if (CF_Pakar[i][1].equalsIgnoreCase(status_penyakit)){
7                 this.temp_nilai_cf_pakar[a] = CF_Pakar[i][2];
8                 a++;
9             }
10        }
11    }
12    public String[] getTemp_nilai_cf_pakar() {
13        return temp_nilai_cf_pakar;
14    }
15    public void Convert_CF_Pakar(String temp_cf_pakar[]){
16        this.nilai_cf_pakar = new double[temp_cf_pakar.length];
17
18        for (int i=0; i < temp_cf_pakar.length; i++){
19            this.nilai_cf_pakar[i] =
20            Double.parseDouble(temp_cf_pakar[i]);
21        }
22    }
23    public double[] getNilai_cf_pakar() {

```



```
15     return nilai_cf_pakar;
16 }
17 public void setCF_User(String status_tek_darah, String
18 status_diabetes, String riwayat_keluarga,
19 String merokok, String status_kolesterol,
20 String aktifitas_fisik,
21 String status_diet, String
22 riwayat_fibrilasi, double CF_Pakar[], String CF_User[]){
23
24     this.nilai_cf_user = new double[CF_Pakar.length];
25     String status1 = "Darah "+status_tek_darah;
26     String status2 = "Diabetes "+status_diabetes;
27     String status3 = "Keluarga "+riwayat_keluarga;
28     String status4 = "Merokok "+merokok;
29     for (int i=0; i < CF_User.length; i++){
30         if (status1.equalsIgnoreCase(CF_User[i])){
31             this.nilai_cf_user[i] = 1;
32         } else
33         if (status2.equalsIgnoreCase(CF_User[i])){
34             this.nilai_cf_user[i] = 1;
35         } else
36         if (status3.equalsIgnoreCase(CF_User[i])){
37             this.nilai_cf_user[i] = 1;
38         } else
39         if (status4.equalsIgnoreCase(CF_User[i])){
40             this.nilai_cf_user[i] = 1;
41         } else
42         if (status_kolesterol.equalsIgnoreCase(CF_User[i])){
43             this.nilai_cf_user[i] = 1;
44         } else
45         if (aktifitas_fisik.equalsIgnoreCase(CF_User[i])){
46             this.nilai_cf_user[i] = 1;
47         } else
48         if (status_diet.equalsIgnoreCase(CF_User[i])){
49             this.nilai_cf_user[i] = 1;
50         } else
51         if (riwayat_fibrilasi.equalsIgnoreCase(CF_User[i])){
52             this.nilai_cf_user[i] = 1;
53         }
54     }
55 }
56 public void Hitung_CF_User(double CF_Pakar[], double CF_User[]){
57     for (int i=0; i < CF_Pakar.length; i++){
58         this.nilai_cf_user[i] = CF_Pakar[i]*CF_User[i];
59     }
60 }
61 public double[] getNilai_cf_user() {
62     return nilai_cf_user;
63 }
64 public void Hitung_CF_Combine(double CF_User[]){
65     this.cf_combine = new double[CF_User.length-1];
66
67     for (int i=0; i < cf_combine.length; i++){
68         if (i == 0){
69             this.cf_combine[i] = CF_User[i]+CF_User[i+1]*(1-
70 CF_User[i]);
71         } else {
72             this.cf_combine[i] = CF_User[i+1]+this.cf_combine[i-
73 1]*(1-CF_User[i+1]);
74         }
75     }
76 }
77 public double[] getCf_combine() { return cf_combine;
78 }
79 }
```

Penjelasan kode program metode *certainty factor* Kode Program 5.2, yakni:

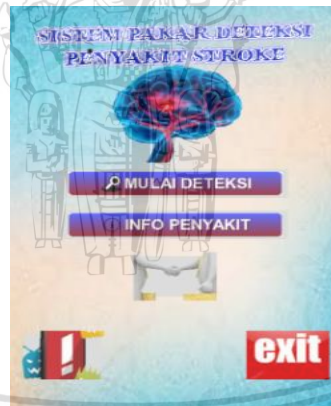
5. 1-11 kode program untuk mencari nilai CF pakar.
6. 14-41 kode program untuk mencari nilai CF *user*
7. 42-48 kode program untuk menghitung perkalian nilai CF pakar dengan CF *user*
8. 49-59 kode program untuk menghitung kombinasi nilai CF pakar dengan nilai CF *user*

1.3.3 Implementasi Antarmuka

Antarmuka Sistem Pakar Deteksi Penyakit Stroke ini digunakan oleh pengguna untuk berinteraksi dengan sistem. Antarmuka untuk tampilan dari sistem sendiri terbagi menjadi enam halaman yakni halaman utama, halaman mulai deteksi, halaman informasi penyakit, halaman informasi perangkat, halaman informasi pengembang, dan halaman hasil deteksi penyakit stroke nya.

5.3.3.1 Tampilan Antarmuka Halaman Utama

Tampilan halaman utama ditunjukkan pada Gambar 5.2 merupakan halaman awal yang akan muncul ketika sistem dijalankan, ketika membuka aplikasi tersebut pengguna dapat mengakses empat menu yakni mulai deteksi, info penyakit, info perangkat, dan info pengembang.



Gambar 5.2 Tampilan Antarmuka Halaman Utama

5.3.3.3 Tampilan Antarmuka Info Penyakit



Gambar 5.3 Tampilan Antarmuka Halaman Info Penyakit



Tampilan antarmuka info penyakit ditunjukkan pada Gambar 5.3 dimana tampilan ini akan muncul ketika pengguna memiliki menu info penyakit pada halaman utama. Halaman info penyakit ini berisi tentang informasi penyakit dari penyakit stroke itu sendiri dan juga informasi dari beberapa penyebab seseorang terserang penyakit stroke.

5.3.3.2 Tampilan Antarmuka Mulai Deteksi

Tampilan antarmuka mulai deteksi ini ditunjukkan pada Gambar 5.4 dimana tampilan ini akan muncul ketika pengguna memilih menu mulai deteksi dan akan menampilkan daftar pilihan berupa pertanyaan yang bersifat *radiobutton* dan isian yang harus diisi oleh pengguna.

Isi Semua Pertanyaan Dengan Benar

1. Apakah Anda seorang perokok?

Merokok

Kadang

Tidak

2. Berapakah Tekanan Darah yang anda derita?

Sistolik:

Diastolik:

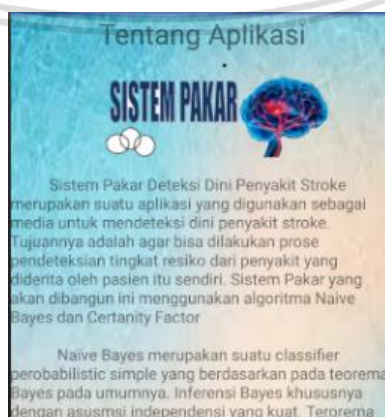
3. Berapakah gula darah yang anda derita?

Gula Darah:

Gambar 5.4 Tampilan Antarmuka Halaman Mulai Deteksi

5.3.3.3 Tampilan Antarmuka Info Perangkat

Tampilan antarmuka Info perangkat ditunjukkan pada Gambar 5.5 dimana tampilan ini akan muncul ketika pengguna memiliki menu info perangkat pada halaman utama. Halaman info perangkat ini berisi tentang informasi tentang penjelasan dari sistem pakar itu sendiri dan juga informasi dari metode yang digunakan dalam proses pengembangan sistem tersebut.



Gambar 5.5 Tampilan Antarmuka Halaman Info Perangkat

5.3.3.4 Tampilan antarmuka Info Pengembang

Tampilan antarmuka Info perangkat ditunjukkan pada Gambar 5.6 dimana tampilan ini akan muncul ketika pengguna memiliki menu info pengembang pada halaman utama. Halaman info pengembang ini berisi tentang siapa saja yang mengembangkan Sistem Pakar Deteksi Penyakit Stroke ini.



Gambar 5.6 Tampilan Antarmuka Halaman Info Pengembang

5.3.3.4 Tampilan antarmuka Hasil Deteksi

Tampilan antarmuka Hasil Deteksi ditunjukkan pada Gambar 5.7 dimana tampilan ini akan muncul ketika pengguna telah selesai mengisi daftar pertanyaan yang ada di dalam menu mulai deteksi dimana di dalam halaman ini berisi hasil dari proses mendeteksi penyakit stroke.

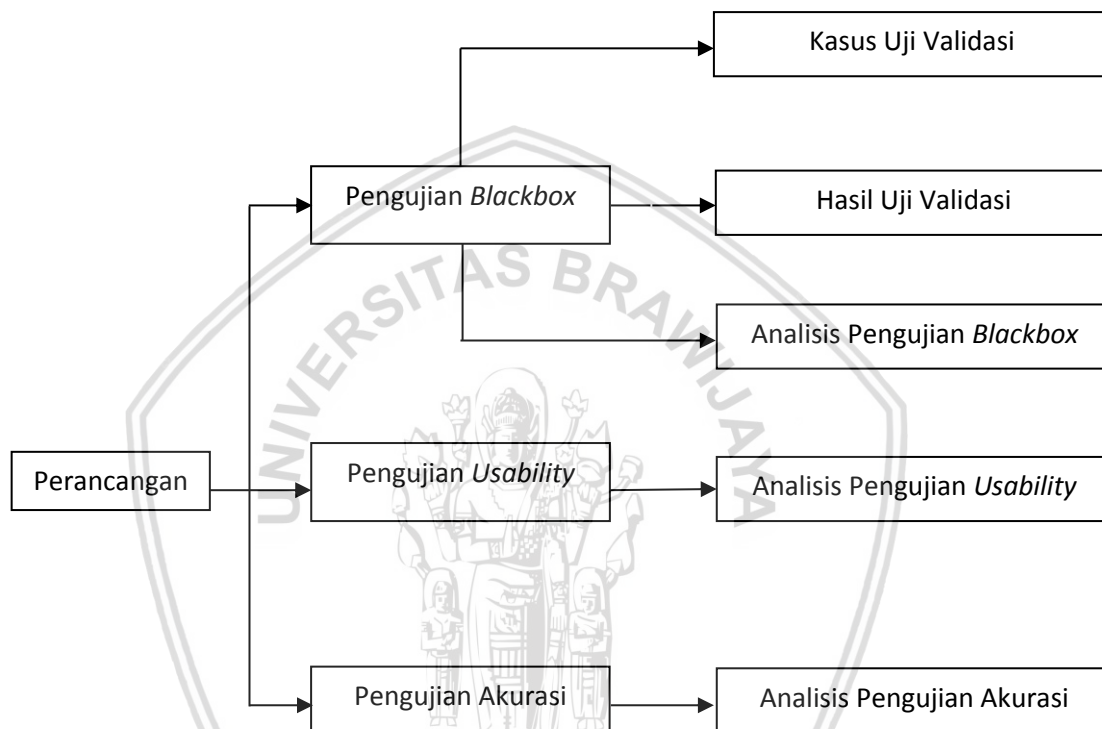


Gambar 5.7 Tampilan Antarmuka Halaman Hasil Deteksi



BAB 6 PENGUJIAN DAN ANALISIS

Pada bab ini akan dijelaskan mengenai pengujian algoritme yang terdiri dari skenario pengujian dan analisis hasil pengujian sistem pakar deteksi dini penyakit stroke. Pengujian yang dilakukan adalah *blackbox testing*, uji angket dan pengujian akurasi dengan metode membandingkan perhitungan manual dengan perhitungan sistem. Pohon Pengujian dan Analisis ditunjukkan pada Gambar 6.1.



Gambar 6.1 Pohon Pengujian dan Analisis

6.1 Blackbox Testing

Pengujian *Blackbox* ini merupakan pengujian yang digunakan untuk mengetahui apakah sistem yang dibangun sudah sesuai dengan kebutuhan atau tidak. Bahan-bahan yang telah dirancang di daftar kebutuhan akan digunakan untuk melakukan pengujian validasi. Pengujian *blackbox* ini difokuskan kepada alur kerja dari sistem itu sendiri. Proses pengujian ini dilakukan dengan menggunakan kasus uji untuk mengetahui apakah kinerja sistem sesuai dengan daftar kebutuhan. Berikut merupakan kasus uji yang digunakan:

6.1.1 Kasus Uji Validasi

Pada kasus uji ini akan menjelaskan bagaimana prosedur pengujian kebutuhan dengan hasil implemementasi sistem pakar deteksi dini penyakit stroke. Berikut ini adalah kasus uji validasi untuk melakukan proses deteksi yang ditunjukkan pada Tabel 6.1.

Tabel 6.1 Kasus Uji Validasi Halaman Mulai Deteksi

Nomor Kasus Uji	U01
Nama Kasus Uji	Menampilkan halaman pertanyaan proses pendeteksian
Objek Kasus Uji	Kebutuhan Fungsional (KF_01)
Tujuan Pengujian	Untuk memastikan sistem dapat menampilkan pertanyaan dalam proses pendeteksian penyakit stroke
prosedur Pengujian	<ol style="list-style-type: none"> 1. Sistem dijalankan 2. Memilih menu Mulai Deteksi
Hasil yang diharapkan	Sistem dapat menampilkan tampilan pertanyaan yang telah disediakan oleh sistem dalam proses mendeteksi penyakit stroke.

Berikut ini adalah kasus uji validasi untuk hasil proses deteksi penyakit stroke dengan memasukkan inputan yang telah disediakan yang ditunjukkan pada Tabel 6.2.

Tabel 6.2 Kasus Uji Validasi Halaman Hasil Deteksi

Nomor Kasus Uji	U02
Nama Kasus Uji	Menampilkan hasil proses deteksi penyakit stroke
Objek Kasus Uji	Kebutuhan Fungsional (KF_02)
Tujuan Pengujian	Untuk memastikan sistem dapat menampilkan hasil deteksi dalam proses pendeteksian penyakit stroke
prosedur Pengujian	<ol style="list-style-type: none"> 1. Sistem dijalankan 2. Memilih menu Mulai Deteksi 3. Mengisi semua pertanyaan yang telah disediakan 4. Memilih tombol mulai deteksi 5. Sistem menampilkan hasil deteksi
Hasil yang diharapkan	Sistem dapat menampilkan tampilan hasil deteksi sesuai dengan masukkan dari pengguna dalam proses mendeteksi penyakit stroke.

Berikut ini adalah kasus uji validasi untuk menampilkan halaman informasi penyakit stroke yang ditunjukkan pada Tabel 6.3.

Tabel 6.3 Kasus Uji Validasi Halaman info penyakit

Nomor Kasus Uji	U03
Nama Kasus Uji	Menampilkan halaman info penyakit
Objek Kasus Uji	Kebutuhan Fungsional (KF_03)
Tujuan Pengujian	Untuk memastikan sistem dapat menampilkan halaman informasi tentang penyakit stroke.
prosedur Pengujian	<ol style="list-style-type: none"> 1. Sistem dijalankan 2. Memilih menu info penyakit 3. Sistem menampilkan halaman info penyakit
Hasil yang diharapkan	Sistem dapat menampilkan tampilan halaman informasi penyakit yang telah disediakan oleh sistem.

Berikut ini adalah kasus uji validasi untuk menampilkan halaman informasi penyakit stroke yang ditunjukkan pada Tabel 6.4.

Tabel 6.4 Kasus Uji Validasi Halaman info perangkat

Nomor Kasus Uji	U04
Nama Kasus Uji	Menampilkan halaman info perangkat
Objek Kasus Uji	Kebutuhan Fungsional (KF_04)
Tujuan Pengujian	Untuk memastikan sistem dapat menampilkan halaman informasi tentang perangkat sistem deteksi penyakit stroke.
prosedur Pengujian	<ol style="list-style-type: none"> 1. Sistem dijalankan 2. Memilih menu info perangkat 3. Sistem menampilkan halaman info perangkat
Hasil yang diharapkan	Sistem dapat menampilkan tampilan halaman informasi perangkat yang telah disediakan oleh sistem.

Berikut ini adalah kasus uji validasi untuk menampilkan halaman informasi pengembang dari sistem pakar deteksi penyakit stroke yang ditunjukkan pada Tabel 6.5.

Tabel 6.5 Kasus Uji Validasi Halaman info pengembang

Nomor Kasus Uji	U05
Nama Kasus Uji	Menampilkan halaman info pengembang

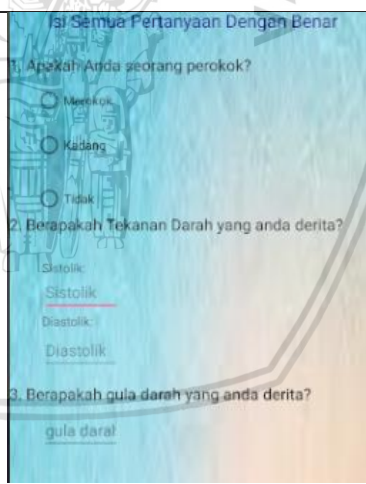
Objek Kasus Uji	Kebutuhan Fungsional (KF_05)
Tujuan Pengujian	Untuk memastikan sistem dapat menampilkan halaman informasi pengembang.
prosedur Pengujian	<ol style="list-style-type: none"> 1. Sistem dijalankan 2. Memilih menu info pengembang 3. Sistem menampilkan info pengembang
Hasil yang diharapkan	Sistem dapat menampilkan tampilan halaman informasi pengembang yang telah disediakan oleh sistem.

6.1.2 Hasil Uji Validasi

Pada hasil uji validasi ini akan menampilkan hasil dari uji validasi secara keseluruhan yang sudah diuji. Berikut adalah hasil pengujian yang ditunjukkan pada Tabel 6.6.

Tabel 6.6 Hasil Uji Validasi

Nomor Kasus Uji	Hasil	Status Validasi
U01	Sistem dapat menampilkan tampilan pertanyaan yang telah disediakan oleh sistem dalam proses mendeteksi penyakit stroke.	Valid
U02	Sistem dapat menampilkan tampilan hasil deteksi sesuai dengan masukan dari pengguna dalam proses mendeteksi penyakit stroke.	Valid



<p>U03</p>	<p>Sistem dapat menampilkan tampilan halaman informasi penyakit yang telah disediakan oleh sistem.</p>		<p>Valid</p>
<p>U04</p>	<p>Sistem dapat menampilkan tampilan halaman informasi perangkat yang telah disediakan oleh sistem.</p>		<p>Valid</p>
<p>U05</p>	<p>Sistem dapat menampilkan tampilan halaman informasi pengembang yang telah disediakan oleh sistem.</p>		<p>Valid</p>

1.1.3 Analisis Pengujian *Blackbox*

Pada hasil pengujian *blackbox* uji validasi Tabel 6.6 yang menghasilkan status valid pada semua pengujian. Dengan hasil tersebut maka presentase pengujiannya sebesar 100% sehingga dapat dihasilkan kesimpulan bahwa fungsionalitas sistem telah berjalan dengan apa yang diharapkan.

6.2 Pengujian *Usability*

Pengujian *usability* dilakukan dengan cara membuat sebuah kuisisioner untuk diisi oleh pengguna yang telah menguji atau menggunakan sistem yang telah dibuat. Setiap pertanyaan yang ada akan diberikan sebuah nilai, yakni satu sampai dengan lima dan totalnya nanti akan dibagi dengan jumlah pertanyaan yang ada. Hasil dari perhitungan tersebut akan menjadi acuan apakah sistem layak atau



tidak, semakin tinggi nilai yang didapatkan maka semakin baik sistem tersebut. Untuk pertanyaan kuisisioner dapat dilihat pada Tabel 6.7.

Tabel 6.7 Kuisisioner

No	Pertanyaan	Pilihan				
		SS	S	KS	TS	STS
1	Sistem ini sangat mudah digunakan					
2	Saya suka dengan tampilan sistem ini					
3	Sistem ini mudah dioperasikan					
4	Bahasa yang digunakan didalam sistem mudah dimenegerti					
5	Menu yang ada didalam sistem mudah dimengerti					

Keterangan:

SS (Sangat Setuju): 5

S (Setuju): 4

KS (Kurang Setuju): 3

TS (Tidak Setuju): 2

STS (Sangat Tidak Setuju): 1

Kuisisioner yang dibuat sebanyak 20 lembar yang berisi 5 pertanyaan, masing-masing dari pertanyaan memiliki nilai satu sampai lima. Berikut merupakan hasil kuisisioner yang telah diisi oleh pengguna dan ditunjukkan pada Tabel 6.8.

Tabel 6.8 Hasil Kuisisioner

Kuisisioner	Pertanyaan				
	Ke-1	Ke-2	Ke-3	Ke-4	Ke-5
Ke-1	5	4	5	5	4
Ke-2	5	4	5	5	5
Ke-3	5	5	4	5	5
Ke-4	5	5	5	5	5
Ke-5	4	5	5	4	5
Ke-6	5	5	5	5	5
Ke-7	5	5	5	5	5

Ke-8	4	5	5	3	4
Ke-9	5	5	5	5	5
Ke-10	5	5	5	5	5
Ke-11	5	5	5	5	5
Ke-12	5	4	3	3	4
Ke-13	5	5	5	5	5
Ke-14	5	5	5	5	5
Ke-15	4	5	5	4	4
Ke-16	5	4	5	5	5
Ke-17	5	5	5	5	5
Ke-18	5	5	5	5	4
Ke-19	4	5	5	5	5
Ke-20	5	5	5	5	5
Total	96	96	97	94	95

Sesuai dengan hasil kuisisioner yang ada diatas maka selanjutnya adalah dilakukan proses perhitungan mean untuk mendapatkan rata-rata. Berikut adalah hasil perhitungan hasil kuisisioner pengujian usability:

$$\text{Mean} = \frac{\text{Total Nilai}}{\text{Jumlah Pertanyaan}} = \frac{96+96+97+94+95}{100} = 4,78$$

Keterangan:

Jika Mean = 1 Maka Sistem dianggap sangat buruk

Jika Mean = 2 Maka Sistem dianggap buruk

Jika Mean = 3 Maka Sistem dianggap biasa saja

Jika Mean = 4 Maka Sistem dianggap baik

Jika Mean = 5 Maka Sistem dianggap sangat baik

6.2.1 Analisis Pengujian Usability

Pada hasil pengujian *usability* hasil perhitungan kuisisioner yang ditunjukkan pada Tabel 6.7 dan telah dilakukan proses perhitungan mean untuk menghasilkan rata-rata diperoleh nilai sebesar 4,78. Dari nilai rata-rata tersebut dapat disimpulkan bahwa sistem yang dibangun sudah dianggap baik.

6.3 Pengujian Akurasi

Pengujian akurasi ini digunakan untuk mengetahui besarnya nilai akurasi sistem yang dibuat. Pada pengujian ini melakukan sebuah perbandingan hasil

deteksi dari pakar dengan hasil deteksi yang dilakukan oleh sistem, apakah hasilnya sama atau tidak. Untuk data yang didapatkan sebanyak 25 data uji dan setelah dilakukan proses pengujian sebanyak 25 kali tersebut terdapat 4 dari 25 data uji yang dilakukan oleh sistem hasilnya berbeda dari hasil yang diperoleh dari pakar, maka hanya terdapat 21 data uji yang hasilnya sama antara sistem dan yang diperoleh dari pakar. Untuk rancangan pengujian akurasi ditunjukkan pada Tabel 6.9.

Tabel 6.9 Pengujian Akurasi

No	Faktor Resiko	Deteksi Pakar	Deteksi Sistem	Nilai CF (%)
1	R1: 80/60 (Rendah) R2: 80 (sedang) R3: Iya R4: Tidak merokok R5: 238 (rendah) R6: Rutin R7: 22 (Normal) R8: Tidak Tahu	Tinggi	Tinggi	87,1%
2	R1: 80/60 (Rendah) R2: 180 (sedang) R3: Iya R4: Tidak merokok R5: 100 (rendah) R6: Rutin R7: 20,3 (Normal) R8: Beraturan	Rendah	Tinggi	88,7%
3	R1: 120/80 (Rendah) R2: 206 (tinggi) R3: Tidak Tahu R4: Merokok R5: 210 (sedang) R6: Kadang-kadang R7: 24,6 (Normal) R8: Tidak Tahu	Tinggi	Tinggi	96,7%



4	<p>R1: 100/60 (Rendah)</p> <p>R2: 140 (rendah)</p> <p>R3: Tidak Ada</p> <p>R4: Kadang-kadang</p> <p>R5: 116 (rendah)</p> <p>R6: Rutin</p> <p>R7: 21,8 (Normal)</p> <p>R8: Beraturan</p>	Rendah	rendah	98,1%
5	<p>R1: 10/70 (Rendah)</p> <p>R2: 130 (rendah)</p> <p>R3: Tidak Ada</p> <p>R4: Tidak</p> <p>R5: 207 (sedang)</p> <p>R6: Rutin</p> <p>R7: 20,4 (Normal)</p> <p>R8: Beraturan</p>	Rendah	Rendah	97,7%
6	<p>R1: 160/90 (tinggi)</p> <p>R2: 240 (tinggi)</p> <p>R3: Tidak Ada</p> <p>R4: Merokok</p> <p>R5: 150 (rendah)</p> <p>R6: Rutin</p> <p>R7: 32,3 (Tinggi)</p> <p>R8: Tidak Beraturan</p>	Tinggi	Tinggi	98,5%
7	<p>R1: 120/80 (Rendah)</p> <p>R2: 180 (tinggi)</p> <p>R3: Tidak Tahu</p> <p>R4: Merokok</p> <p>R5: 260 (tinggi)</p> <p>R6: Tidak Pernah</p> <p>R7: 31,1 (Tinggi)</p>	Tinggi	Tinggi	99%

	R8: Beraturan			
8	R1: 150/90 (tinggi) R2: 250 (tinggi) R3: Tidak Ada R4: Merokok R5: 300 (tinggi) R6: Tidak Pernah R7: 32,7 (tinggi) R8: Tidak Beraturan	Tinggi	Tinggi	99,2%
9	R1: 160/100 (Rendah) R2: 330 (tinggi) R3: Tidak Tahu R4: Merokok R5: 310 (tinggi) R6: Tidak Pernah R7: 32,9 (tinggi) R8: Tidak Beraturan	Tinggi	Tinggi	99,3%
10	R1: 80/60 (Rendah) R2: 120 (rendah) R3: Tidak Ada R4: Tidak merokok R5: 136 (rendah) R6: Rutin R7: 24,2 (Normal) R8: Beraturan	Rendah	Rendah	98,1%
11	R1: 110/90 (Rendah) R2: 180 (sedang) R3: Tidak Tahu R4: Kadang-kadang R5: 220 (sedang) R6: Kadang-kadang R7: 33,5 (Tinggi) R8: Tidak Tahu	Sedang	Sedang	99,9%



12	R1: 110/90 (Rendah) R2: 190 (sedang) R3: Tidak Tahu R4: Kadang-kadang R5: 360 (tinggi) R6: tidak Pernah R7: 33,6 (tinggi) R8: Tidak Beraturan	Sedang	Sedang	99,7%
13	R1: 80/60 (Rendah) R2: 210 (tinggi) R3: Tidak Ada R4: Tidak merokok R5: 145 (rendah) R6: Tidak Pernah R7: 32,7 (tinggi) R8: Beraturan	Tinggi	Tinggi	97,2%
14	R1: 120/80 (Rendah) R2: 206 (tinggi) R3: Tidak Tahu R4: Merokok R5: 210 (sedang) R6: Kadang-Kadang R7: 24,6 (Normal) R8: Tidak Tahu	Sedang	Sedang	99,7%
15	R1: 100/90 (Rendah) R2: 180 (sedang) R3: Tidak Tahu R4: Kadang-kadang R5: 330 (tinggi) R6: Tidak Pernah R7: 24,7 (Normal) R8: Tidak Tahu	Sedang	Sedang	99,7%
	R1: 100/60 (Rendah) R2: 190 (sedang)			



16	R3: Tidak Tahu R4: Kadang-kadang R5: 216 (sedang) R6: Tidak Pernah R7: 30,1 (tinggi) R8: Beraturan	Sedang	Sedang	99,5%
17	R1: 60/80 (Rendah) R2: 190 (sedang) R3: Tidak Tahu R4: Kadang-kadang R5: 218 (sedang) R6: Rutin R7: 30,1 (tinggi) R8: Beraturan	Sedang	Sedang	99,7%
18	R1: 140/80 (Rendah) R2: 170 (sedang) R3: Tidak Tahu R4: Tidak merokok R5: 220 (sedang) R6: Kadang-kadang R7: 32,2 (tinggi) R8: Beraturan	Sedang	Sedang	99,8%
19	R1: 120/80 (Rendah) R2: 180 (sedang) R3: Tidak Tahu R4: Merokok R5: 220 (sedang) R6: Tidak R7: 26,7 (sedang) R8: Tidak Beraturan	Sedang	Sedang	98,8%
20	R1: 80/100 (Rendah) R2: 178 (sedang) R3: Tidak Tahu R4: Kadang-kadang	Sedang	Sedang	99,4%

	R5: 120 (rendah) R6: Rutin R7: 20,0 (Normal) R8: Tidak Beraturan			
21	R1: 90/60 (Rendah) R2: 109 (rendah) R3: Tidak Ada R4: Tidak merokok R5: 109 (rendah) R6: Rutin R7: 23,7 (Normal) R8: Tidak Beraturan	Rendah	Tinggi	91,5%
22	R1: 110/70 (Rendah) R2: 300 (sedang) R3: Tidak Ada R4: Tidak merokok R5: 100 (rendah) R6: Rutin R7: 33,9 (tinggi) R8: Tidak Beraturan	Tinggi	Tinggi	96,7%
23	R1: 90/60 (Rendah) R2: 135 (rendah) R3: Tidak Ada R4: Tidak merokok R5: 100 (rendah) R6: Tidak Pernah R7: 23,1 (Normal) R8: Beraturan	Rendah	Tinggi	92,9%
	R1: 90/70 (Rendah) R2: 100 (sedang) R3: Tidak Ada			

24	R4: Tidak merokok R5: 108 (rendah) R6: Rutin R7: 20,0 (Normal) R8: Tidak Tahu	Rendah	Tinggi	90,1%
25	R1: 80/60 (Rendah) R2: 120 (rendah) R3: Tidak Ada R4: Tidak merokok R5: 120 (rendah) R6: Kadang R7: 23,1 (Normal) R8: Tidak Tahu	Rendah	Tinggi	93,8%

Yang kemudian nilai akurasi akan didapatkan dengan rumus sebagai berikut:

$$akurasi = \frac{\text{Output yang sesuai}}{\text{jumlah percobaan}} \times 100\%$$

$$akurasi = \frac{21}{25} \times 100\% = 84\%$$

Maka Akurasi yang didapatkan adalah sebesar 84%.

6.3.1 Analisis Pengujian Akurasi

Pada hasil pengujian akurasi sistem pakar deteksi dini penyakit stroke menggunakan metode *Naïve Bayes-Certainty factor* yang telah dilakukan dengan menghasilkan akurasi sebesar 84%. Dari hasil akurasi tersebut dapat disimpulkan bahwa hasil yang didapatkan sudah cukup akurat.

BAB 7 PENUTUP

1.1 Kesimpulan

Berdasarkan hasil pengujian dan analisis hasil penelitian yang sudah dilakukan pada sistem pakar deteksi dini penyakit stroke menggunakan metode *Naïve Bayes-Certainty factor*, maka didapatkan beberapa kesimpulan sebagai berikut:

1. Sistem yang dihasilkan sesuai dengan perancangan sistem secara keseluruhan sebagai solusi dari permasalahan dalam mendeteksi dini penyakit stroke yang dapat membantu masyarakat dalam mendeteksi secara dan tepat.
2. Berdasarkan hasil pengujian yang telah dilakukan dan memperoleh hasil sebagai berikut:
 - a. Pengujian *blackbox* mendapatkan *presentase* pengujiannya sebesar 100% sehingga dapat diberikan kesimpulan bahwa fungsionalitas sistem telah berjalan dengan apa yang diharapkan.
 - b. Pengujian *usability* menghasilkan rata-rata diperoleh nilai sebesar 4,78. Dari nilai rata-rata tersebut dapat disimpulkan bahwa sistem yang dibangun sudah dianggap baik.
 - c. Pengujian akurasi memiliki nilai *presentase* akurasi sebesar 84% karena terdapat 4 data uji dari 25 data uji yang tidak sesuai dengan hasil yang diperoleh dari pakar yang disebabkan adanya kesamaan diantara beberapa resiko yang ada.

1.2 Saran

Sistem Pakar deteksi dini penyakit stroke menggunakan metode *naïve bayes-certainty factor* ini masih memiliki beberapa kekurangan. Saran yang dapat diberikan untuk pengembangan sistem agar menjadi lebih baik adalah sebagai berikut:

1. Mengembangkan algoritme lain agar akurasi bias lebih optimal, untuk pengembangan metode *naïve bayes-certainty factor* bisa menggunakan data lain selain parameter data tingkat resiko stroke.
2. Pengembangan sistem ini dibangun dengan berbasis android, diharapkan dengan penelitian selanjutnya bisa diubah menjadi tipe mobile lainnya, contohnya ios.
3. Pengembangan dapat dilakukan dengan cara menambah parameter sistem pakar dalam proses mendeteksi dini penyakit stroke.



DAFTAR PUSTAKA

- Abdullah, Putri Puja. 2014. Sistem Pakar untuk Mendiagnosa Penyakit pada Ayam dengan Metode *Certainty factor* Berbasis Android Skripsi. Tidak diterbitkan. Departemen Matematika Fakultas MIPA Universitas Lampung, Lampung.
- Affan, Achmad. 2017. Sistem Pakar Diagnosis Penyakit Kucing Menggunakan Metode *Naïve Bayes-Certainty factor* Berbasis Android. Program Studi Informatika Ilmu Komputer FILKOM Universitas Brawijaya Malang.
- Al Ihsan S. H. 2012. Pengembangan Sistem Pakar Agribisnis Cabai (*Capsicum Annuum*.l) Berbasis Android. Program Studi Ilmu Komputer Institut Pertanian Bogor.
- Amalia, Nila. 2009. Membaca Gejala dan Diagnosa Stroke. Buku Biru. Jogjakarta.
- Antonios, Silliman. 2005. Diabetes Mellitus And Stroke. Northeast Florida Medicine. 17-22.
- Ardhitama, Ivan. 2014. Aplikasi Sistem Pakar Untuk Mendiagnosa Dan Penanganan Dini Gangguan Autisme Pada Anak Dengan Metode *Certainty factor* Berbasis Web. Program Studi Informatika Ilmu Komputer PTIIK Universitas Brawijaya Malang.
- Ardianto, et al. 2012. Expert Sistem Development for Early Detection and Treatment on Cow Disease Based on Android Mobile with Review of Knowledge Representation Technique. Jurnal teknik ITS, 1: 310 – 315.
- Association, N. S., 2009. Stroke Risk Scorecard, s.l: National Stroke Association
- Astuti, Puji Indah. 2016. Pengembangan Sistem Pakar Identifikasi Awal Penyakit Kedelai dengan Pendekatan *Naïve Bayes* Berbasis android. Skripsi. Tidak diterbitkan. Institut Pertanian Bogor, Bogor.
- Dinas Kesehatan Provinsi Jawa Tengah. 2006. *Profil Kesehatan Provinsi Jawa Tengah Tahun 2006*. Dinas Kesehatan Provinsi Jawa Tengah
- Farida, Ida. 2009. Mengantisipasi Stroke. Yogyakarta : Buku Biru.
- Hardika, Angga. 2014. Aplikasi Sistem Pakar Untuk Indentifikasi Hama Dan Penyakit Tanaman Tebu Dengan Metode *Naive bayes* Berbasis Web. Program Studi Informatika/Ilmu Komputer PTIIK Universitas Brawijaya Malang.
- Hidayat, Syaeful. 2010. Aplikasi untuk mendeteksi jenis penyakit pada tanaman tebu dan cara penanganannya berbasis web. Skirpsi. Tidak diterbitkan. Jurusan Teknik Informatika Fakultas Teknik dan Ilmu Komputer Universitas Indonesia Bandung.
- Hidayati. Nur., et al. 2013. Sistem pakar berbasis web untuk identifikasi hama penyakit pada budidaya tanaman jamur menggunakan metode *Certainty factor*. Teknik Informatika PTIIK Universitas Brawijaya Malang.

- Huda, Arif Akbarul. 2013. "LiveCoding! 9 Aplikasi Android Buatan Sendiri". Yogyakarta: penerbit Andi.
- Immamura, et al. 2008. LDL Cholesterol and the Development of Stroke Subtypes and Coronary Heart Disease.
- Indraswari, Deby Putri. 2015. Sistem Pendukung Keputusan Deteksi Dini pada Penyakit Stroke menggunakan algoritme Dempster Shafer. Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya Malang.
- Iwai S., Markowitz S. M., Mittal S., Stein K. M., Lerman B. B., 2005. *Electrophysiology of Cardiac Arrhythmias*. In Rosendorff, C. *Essential Cardiology: Principles and Practice*. Second Edition. New Jersey: Humana Press pp 285-302
- Josephson M. E., Zimetbaum P., 2005. The Tachyarrhythmias. In Kasper, D. L., Braunwald, E., Fauci, A. S., Hauser, S. L., Longo, D. L., Jameson, J. L. *Harrison's Principles Internal Medicine*. 16th Edition. United State of America: McGraw-Hill pp 1342-46
- Kementerian Kesehatan RI. 2007. Riset Kesehatan Dasar, Riskesdas 2007. Kementerian Kesehatan RI, Jakarta.
- Kementerian Kesehatan RI. 2013. Riset Kesehatan Dasar, Riskesdas 2013. Kementerian Kesehatan RI, Jakarta.
- Prasetyo, Eko. 2012. DATA MINING – Konsep dan Aplikasi menggunakan MATLAB. Yogyakarta. Andi.
- Prihatini, Putu Manik. 2011. Metode Ketidakpastian Dan Kesamaran Dalam Sistem Pakar. Staf Pengajar Jurusan Teknik Elektro Politeknik negeri Bali.
- Roger, V.L. et al., 2011. AHA Heart Disease and Stroke Statistics 2011 update: a report from the American Heart Association. *Circulation* 2011;123:e18-e209.)
- Rosenthal L., Borczuk P., Chandrakantan A., Greenberg M. L., 2012. *Atrial Fibrillation*. Available from:<http://emedicine.medscape.com> Diakses tanggal 10 Maret 2017
- Rusdi, Nurlaela Isnawati. 2009. *Awas! Anda Bisa Mati Cepat Akibat Hipertensi & Diabetes*, Yogyakarta: Power Books (IHDINA).
- Saraswati, Sylvia. 2009. *Diet Sehat untuk Penyakit Asam Urat, Diabetes, Hipertensi dan Stroke*. Jogjakarta: A plus Book.
- Setyowati, Hermin. 2003. *Beberapa Faktor Resiko Kejadian Stroke di Perjan RSUP Dr. Wahidin Sudirohusodo Makassar Tahun 2004*. Skripsi Tidak Diterbitkan. Fakultas Kesehatan Masyarakat Universitas Hasanuddin : Makassar
- Silalahi, Rumondang M.S. 2011. *Perancangan Aplikasi Sistem Pakar Untuk Diagnosis Penyakit Ginjal Dengan Kombinasi Metode Certainty factor Dan Metode Forward Chaining*. Skripsi. Tidak Diterbitkan. Departemen Ilmu

- Komputer Fakultas Matematika Dan Ilmu Pengetahuan Alam Universitas Sumatera Utara, Medan.
- Sitorus, Rico Januar. 2008. Faktor-Faktor Resiko Yang Memengaruhi Kejadian Stroke Pada Usia Muda Kurang Dari 40 Tahun (Studi Kasus Di Rumah Sakit Di Kota Semarang). Jurnal Epidemiologi Universitas Diponegoro, Institutional Repository.
- Soeharto, I. 2004. Serangan Jantung dan Stroke Hubungannya Dengan Lemak dan Kolesterol. Jakarta: PT. Gramedia Pustaka Utama.
- Suprayogi, Achmad Affan. 2017. Sistem Pakar Diagnosis Penyakit Kucing Menggunakan Metode Naïve Bayes-*Certainty factor* Berbasis Android. Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya.
- T. Sutojo, S. S. M. K. E. M. S. S. M. K. d. D. V. S., 2011. Sistem Pakar. In; B.R. W, ed. Kecerdasan Buatan. Yogyakarta: CV. Andi Offset, pp. 160-178.
- Tambunan, Imelda Sari. 2010. Sistem Pakar untuk mengidentifikasi jenis jagung berdasarkan cirinya. Skripsi. Tidak diterbitkan. Departemen Matematika Fakultas MIPA Universitas Sumatera Utara, Medan.
- Vitahealth, 2003. Stroke. Jakarta, Gramedia Pustaka Utama.
- Viveca, M, Bhat, MD. et.al. 2008. Dose-Response Relationship Between Cigarette Smoking and Risk of Ischemic Stroke in Young Women. Journal Of The American Stroke Association (<http://stroke.ahajournals.org/.Stroke-2008-Bhat-2439-43.pdf>) Diakses tanggal 10 Maret 2017
- Voni, Elpha Armi. 2010. Penggunaan *Certainty factor* dalam perancangan sistem pakar untuk mendiagnosa penyakit Atherosklerosis. Skripsi. Tidak diterbitkan. Departemen Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Sumatera Utara, Medan.
- Wahana Komputer, 2013, "Step by step menjadi pemrogramer Android", Yogyakarta: ANDI.
- World Health Organization. Diabetes. http://www.who.int/topics/diabetes_mellitus/en/. Diakses tanggal 15 Maret 2017.