SISTEM PENDUKUNG KEPUTUSAN DETEKSI DINI PENYAKIT STROKE MENGGUNAKAN METODE NAIVE BAYES

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Eka Hastian Astuti
NIM: 135150109111002

TEKNIK INFORMATIKA / ILMU KOMPUTER
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2016

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

PENGESAHAN

SISTEM PENDUKUNG KEPUTUSAN DETEKSI DINI PENYAKIT STROKE MENGGUNAKAN METODE *NAIVE BAYES*

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
EKA HASTIAN ASTUTI
NIM: 135150109111002

Skripsi ini telah diuji dan dinyatakan lulus pada
28 Desember 2015
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I	Dosen Pembimbing II
Arief Andy Soebroto, ST., M.Kom.	Imam Cholissodin, S.Si., M.Kom.
NIP. 19720425 199903 1 002	NIK. 850719 16 1 1 0422

Mengetahui,
Ketua Program Studi Teknik Informatika / Ilmu Komputer

Drs. Marji, M.T.
NIP. 19670801 199203 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, 28 Desember 2015

EKA HASTIAN ASTUTI
NIM: 135150109111002

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT, karena atas segala rahmat dan limpahan hidayah-Nya, penulis dapat menyelesaikan tugas akhir yang berjudul "**SISTEM PENDUKUNG KEPUTUSAN DETEKSI DINI PENYAKIT STROKE MENGGUNAKAN METODE *NAIVE BAYES***".

Dalam pelaksanaan dan penulisan tugas akhir ini penulis mendapatkan banyak bantuan dari berbagai pihak baik secara moril maupun materiil. Dalam kesempatan ini penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada :

1. Arief Andy Soebroto, ST., M.Kom. dan Imam Cholissodin S.Si., M.Kom., selaku dosen pembimbing skripsi yang telah dengan sabar membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini.
2. Ir. Sutrisno, M.T., Ir. Heru Nurwasito, M.Kom, Himawat Aryadita, S.T, M.Sc, dan Bapak Edy Santoso, S.Si., M.Kom. selaku Ketua, Wakil Ketua 1, Wakil Ketua 2 dan Wakil Ketua 3 Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
3. Drs. Marji, M.T. dan Bapak Issa Arwani, S.Kom, M.Sc. selaku Ketua dan Sekretaris Program Studi Teknik Informatika Universitas Brawijaya, yang telah memberikan kesempatan kepada penulis untuk menyelesaikan skripsi ini.
4. Ibunda Nanik Kuntijaningwati, Ayahanda Bambang Ratmoko, Adik Harlian Santa Aji, Adik Nasrullah Rizki Ananda Putra, dan seluruh keluarga besar atas segala nasehat, kasih sayang, perhatian dan kesabarannya di dalam membesarkan dan mendidik penulis, serta yang senantiasa tiada henti-hentinya memberikan doa dan semangat demi terselesaikannya skripsi ini.
5. Ir. Tjutjuk Usmanhadi dan Putra Hermawan selaku Kabag Kemahasiswaan Bidang Kesejahteraan Mahasiswa Universitas Brawijaya dan bagian Kemahasiswaan Fakultas Ilmu Komputer Universitas Brawijaya.
6. Seluruh Dosen Teknik Informatika Universitas Brawijaya atas kesediaan membagi ilmunya kepada penulis.
7. Seluruh Civitas Akademika Teknik Informatika Universitas Brawijaya yang telah banyak memberi bantuan dan dukungan selama penulis menempuh studi di Teknik Informatika Universitas Brawijaya dan selama penyelesaian skripsi ini.
8. Teman–teman Seleksi Alih Program (SAP) Teknik Informatika Angkatan 2013 yang telah memberikan masukan dan inspirasi kepada penulis selama menempuh studi dan menyelesaikan skripsi ini.
9. Anggota tim Penelitian Deteksi Dini Penyakit *Stroke* yaitu Tety Dewi Prastiti, Voni Nurfaizzah, Anis Fitri Nur Masruriyah, dan Deby Putri Indraswari yang selalu bertukar pikiran dan semangat dengan penulis selama menyelesaikan skripsi ini.
10. Rani Anggi Nilam Sari, Fery Fatma Rukmana, Aristiawan, Choirul Rahmadan, dan Guruh Adi Purnomo, yang membantu penulis menyelesaikan permasalahan dalam penyelesaian skripsi ini.

11. Seluruh pihak yang telah membantu kelancaran penulisan tugas akhir yang tidak dapat kami sebutkan satu persatu.

Penulis menyadari bahwa dalam penyusunan tugas akhir ini masih banyak kekurangan baik format penulisan maupun isinya. Oleh karena itu, saran dan kritik yang membangun dari para pembaca senantiasa penulis harapkan guna perbaikan bagi tugas akhir selanjutnya. Semoga tugas akhir ini dapat memberikan manfaat bagi semua pihak, Amin.

Malang, 28 Desember 2015

Penulis
estianti.raihan@gmail.com

ABSTRAK

Stroke merupakan suatu keadaan yang timbul karena berhentinya suplai darah ke bagian otak yang mengakibatkan kematian jaringan otak dengan serangan mendadak yang progresif cepat dan berlangsung 24 jam atau lebih sebagai akibat dari *Cardiovascular Diasease* (CVD). Kesibukan yang luar biasa khususnya di kota besar membuat aktivitas manusia menurun sehingga mengabaikan kesehatan tubuhnya. Seiring dengan menurunnya kualitas dan gaya hidup seperti pola makan tidak teratur, kurang olahraga, stress, merokok, konsumsi alkohol, jam kerja berlebihan, serta konsumsi makanan cepat saji sudah menjadi suatu kebiasaan yang berpotensi menjadi faktor awal yang dapat menimbulkan serangan *stroke*. Penggunaan *Naive Bayes* yang diimplementasikan pada sistem pendukung keputusan diharapkan dapat membantu dokter dalam melakukan deteksi dini risiko penyakit *stroke*. Perancangan yang digunakan yaitu analisa kebutuhan perangkat lunak, perancangan sistem pendukung keputusan, dan perancangan perangkat lunak. Sistem pendukung keputusan ini diimplementasikan dengan bahasa pemrograman PHP yang terintegrasi dengan *database* MySQL. Berdasarkan data yang digunakan, sistem dapat melakukan identifikasi deteksi dini penyakit *stroke* dengan tingkat akurasi sebesar 90.46%. Sehingga dapat disimpulkan, bahwa sistem pendukung keputusan yang sudah dibangun menggunakan metode *Naive Bayes* dapat berfungsi dengan cukup baik.

Kata Kunci : *Stroke*, *Klasifikasi*, *Naive Bayes*, Sistem Pendukung Keputusan

repository.ub.ac.id

ABSTRACT

Stroke is a condition that arises due to the cessation of blood supply to the part of brain, the effection of death brain which can make sudden attack rapidly progressive and 24 hours lasting or more as a result of Cardiovascular Diasease (CVD). Overcrowding especially in large cities make human activities declined, so they are ignored their health. Along with the decline in the quality and lifestyle such as irregular diet, lack of exercise, stress, smoking, alcohol consumption, excessive working hours, as well as the consumption fast food has become a habit that could potentially be the initial factor that can cause a stroke. Implementing Naive Bayes method on decision support system is expected to help physicians in the early detection of stroke risk. The design used is the analysis of software requirements, decision support systems design, and software design. This decision support system is implemented with an integrated programming language PHP with MySQL database. Based on the data, the system can identify early detection for Stroke with an accuracy level of 90.46%. It can be concluded, decision support system that has been built using Naive Bayes methods can work well.

Key word **:** *Stroke, Classification, Naive Bayes, Decision Support System*

UNIVERSITAS BRAWIJAYA

DAFTAR ISI

PENGESAHAN ..... **ii**
PERNYATAAN ORISINALITAS ..... **iii**
KATA PENGANTAR ..... **iv**
ABSTRAK ..... **vi**
ABSTRACT ..... **vii**
DAFTAR ISI ..... **viii**
DAFTAR TABEL ..... **x**
DAFTAR GAMBAR ..... **xii**
DAFTAR PERSAMAAN ..... **xiii**
DAFTAR LAMPIRAN ..... **xiv**
BAB 1 PENDAHULUAN **1**
1.1 Latar Belakang 1
1.2 Rumusan Masalah 3
1.3 Batasan Masalah 3
1.4 Tujuan Penelitian 3
1.5 Manfaat Penelitian 4
1.6 Sistematika Penulisan 4
BAB 2 LANDASAN KEPUSTAKAAN **6**
2.1 Kajian Pustaka Terkait Penelitian Sebelumnya 6
2.2 Sistem 17
2.3 Sistem Pendukung Keputusan (SPK) 17
2.3.1 Perbedaan Sistem Informasi, Sistem Pakar, dan SPK 18
2.3.2 Karakteristik Sistem Pendukung Keputusan 19
2.3.3 Komponen-Komponen Sistem Pendukung Keputusan 20
2.3.4 Keuntungan Sistem Pendukung Keputusan 21
2.3.5 Kemampuan dan Keterbatasan Sistem Pendukung Keputusan 22
2.3.6 Langkah-langkah Pemodelan Sistem Pendukung Keputusan 23
2.4 Stroke 23
2.4.1 Penyebab *Stroke* 24
2.4.2 Faktor Risiko *Stroke* 25
2.5 Ketidakpastian 27

2.6 *Naive Bayes* 27
2.7 *Data Flow* Diagram (DFD) 31
2.8 *Entity Relationship Diagram* (ERD) 33
2.8.1 Notasi 33
2.9 Akurasi 35
BAB 3 METODOLOGI PENELITIAN 36
3.1 Studi Literatur 37
3.2 Analisa Kebutuhan Sistem 37
3.2.1 Kebutuhan Perangkat 37
3.2.2 Kebutuhan Sistem 37
3.3 Pengumpulan Data 38
3.4 Perancangan Sistem 38
3.4.1 Arsitektur Implementasi Sistem Pendukung Keputusan 39
3.4.2 Blok Diagram SPK 40
3.5 Implementasi Sistem 41
3.6 Pengujian SPK 41
3.7 Hasil dan Kesimpulan 42
BAB 4 PENGUJIAN SISTEM 43
4.1 Pengujian Fungsionalitas 43
4.1.1 Skenario Pengujian *Blackbox* 44
4.1.2 Analisis Pengujian *Blackbox* 62
4.2 Pengujian Akurasi 62
4.2.1 Skenario Pengujian Akurasi 62
4.2.2 Analisis Pengujian Akurasi Variasi Jumlah Data Latih 63
BAB 5 PENUTUP 65
5.1 Kesimpulan 65
5.2 Saran 65
DAFTAR PUSTAKA 67
LAMPIRAN A WAWANCARA 69
LAMPIRAN B DATA UJI 80
LAMPIRAN C DATA LATIH 83

DAFTAR TABEL

Tabel 2.1 Kajian Pustaka 7
Tabel 2.2 Karakteristik Metode 13
Tabel 2.3 Konklusi Metode Certainty Factor 15
Tabel 2.4 Diagram Proses Pengambilan Keputusan Dempster-Shafer 16
Tabel 2.5 Diagram Proses Pengambilan Keputusan Metode Naive Bayes 17
Tabel 4.1 Penjelasan Kasus Uji Login 44
Tabel 4.2 Pengujian Blackbox Kasus Uji Login 45
Tabel 4.3 Penjelasan Kasus Uji Mengolah Data User 45
Tabel 4.4 Pengujian Blackbox Kasus Uji Mengolah Data User 45
Tabel 4.4 Pengujian Blackbox Kasus Uji Mengolah Data User (lanjutan) 47
Tabel 4.5 Penjelasan Kasus Uji Create Data User 46
Tabel 4.6 Pengujian Blackbox Kasus Uji Create Data User 46
Tabel 4.7 Penjelasan Kasus Uji Update Data User 47
Tabel 4.8 Pengujian Blackbox Kasus Uji Update Data User 47
Tabel 4.9 Penjelasan Kasus Uji Delete Data User 47
Tabel 4.10 Pengujian Blackbox Kasus Uji Hapus Data User 48
Tabel 4.11 Penjelasan Kasus Uji Menampilkan Detail User 48
Tabel 4.12 Pengujian Blackbox Kasus Uji Menampilkan Detail User 48
Tabel 4.13 Penjelasan Kasus Uji Mengolah Data Faktor Risiko 49
Tabel 4.14 Pengujian Blackbox Kasus Uji Mengolah Data Faktor Risiko 49
Tabel 4.15 Penjelasan Kasus Uji Edit Data Faktor Risiko 49
Tabel 4.15 Penjelasan Kasus Uji Edit Data Faktor Risiko (lanjutan) 51
Tabel 4.16 Pengujian Blackbox Kasus Uji Update Data Faktor Risiko 50
Tabel 4.17 Penjelasan Kasus Uji Delete Data Faktor Risiko 50
Tabel 4.18 Pengujian Fungsionalitas Kasus Uji Delete Data Faktor Risiko 50
Tabel 4.18 Pengujian Fungsionalitas Delete Data Faktor Risiko (lanjutan) 52
Tabel 4.19 Penjelasan Kasus Uji Melihat Data Faktor Risiko 51
Tabel 4.20 Pengujian Fungsionalitas Kasus Uji Melihat Data Faktor Risiko 51
Tabel 4.21 Penjelasan Kasus Uji Mengolah Data Latih 51
Tabel 4.21 Penjelasan Kasus Uji Mengolah Data Latih (lanjutan) 53

Tabel 4.22 Pengujian Blackbox Kasus Uji Mengolah Data Latih 52
Tabel 4.23 Penjelasan Kasus Uji Create Data Latih..... 52
Tabel 4.24 Pengujian Fungsionalitas Kasus Uji Create Data Latih 53
Tabel 4.25 Penjelasan Kasus Uji Update Data Latih..... 53
Tabel 4.26 Pengujian Fungsionalitas Kasus Uji Update Data Latih..... 53
Tabel 4.27 Penjelasan Kasus Uji Delete Data Latih..... 54
Tabel 4.28 Pengujian Fungsionalitas Kasus Uji Delete Data Latih 54
Tabel 4.29 Penjelasan Kasus Uji Melihat Data Latih..... 54
Tabel 4.30 Pengujian Fungsionalitas Kasus Uji Melihat Data Latih 55
Tabel 4.31 Penjelasan Kasus Uji Mengolah Data Deteksi..... 55
Tabel 4.32 Pengujian Blackbox Kasus Uji Mengolah Data Deteksi 55
Tabel 4.33 Penjelasan Kasus Uji Tambah Data Deteksi 56
Tabel 4.34 Pengujian Blackbox Kasus Uji Tambah Data Deteksi 56
Tabel 4.35 Penjelasan Kasus Uji Update Data Deteksi 56
Tabel 4.36 Pengujian Fungsionalitas Kasus Uji Update Data Deteksi..... 57
Tabel 4.37 Penjelasan Kasus Uji Delete Data Deteksi..... 57
Tabel 4.38 Pengujian Fungsionalitas Kasus Uji Delete Data Deteksi 57
Tabel 4.39 Penjelasan Kasus Uji Melihat Data Deteksi..... 58
Tabel 4.40 Pengujian Fungsionalitas Kasus Uji Melihat Data Deteksi 58
Tabel 4.41 Penjelasan Kasus Uji Melihat Hasil Keputusan 58
Tabel 4.42 Pengujian Fungsionalitas Kasus Uji Hasil Keputusan 59
Tabel 4.43 Penjelasan Kasus Uji Perhitungan 59
Tabel 4.44 Pengujian Blackbox Kasus Uji Perhitungan 59
Tabel 4.45 Penjelasan Kasus Uji Logout..... 60
Tabel 4.46 Pengujian Fungsionalitas Kasus Uji Logout 60
Tabel 4.47 Hasil Pengujian Blackbox..... 60
Tabel 4.47 Hasil Pengujian Blackbox (lanjutan) 64
Tabel 4.48 Kombinasi Pengujian Variasi Akurasi Data..... 63

DAFTAR GAMBAR

Gambar 2.1 Diagram Venn Variabel Masukan Stroke 13
Gambar 2.2 Diagram Proses Komputasi Metode Certainty Factor 14
Gambar 2.3 Diagram Proses Metode Dempster-Shafer 15
Gambar 2.4 Diagram Proses Komputasi Metode Naive Bayes 16
Gambar 2.5 Arsitektur Sistem Pendukung Keputusan 21
Gambar 2.6 Stroke Iskemik 24
Gambar 2.7 Stroke Hemoragi 25
Gambar 2.9 Simbol Aktivitas/Proses 31
Gambar 2.10 Simbol Data Flow 32
Gambar 2.11 Simbol Data Store 32
Gambar 2.12 Simbol Entitas 32
Gambar 2.13 Konsep Data 33
Gambar 2.14 Notasi Entitas 33
Gambar 2.15 Notasi Relationship 34
Gambar 2.16 Notasi Entitas 34
Gambar 2.17 Contoh Hubungan Satu ke Satu 34
Gambar 2.18 Contoh Hubungan Satu ke Banyak 35
Gambar 2.19 Contoh Hubungan Banyak ke Banyak 35
Gambar 3.1 Diagram Alir untuk Penentuan Metode Penelitian 36
Gambar 3.2 Arsitektur Implementasi SPK Deteksi Dini Penyakit Stroke 39
Gambar 3.3 Arsitektur Blok Diagram Perancangan Aplikasi 40
Gambar 3.4 Diagram Blok Pengujian Akurasi Sistem 42
Gambar 4.1 Pohon Pengujian dan Analisis 43
Gambar 4.2 Pengujian Variasi Data 64

DAFTAR PERSAMAAN

Persamaan **(2.1)** Indeks Masa Tubuh 26
Persamaan **(2.2)** Probabilitas *Naive Bayes* 27
Persamaan **(2.3)** Probabilitas Akhir *Naive Bayes* 28
Persamaan **(2.4)** Gaussian 29
Persamaan **(2.5)** Nilai *Mean* 29
Persamaan **(2.6)** Standar Deviasi 29
Persamaan **(2.7)** Nilai Akurasi 35

DAFTAR LAMPIRAN

LAMPIRAN A WAWANCARA..... 69
LAMPIRAN B DATA UJI 80
LAMPIRAN C DATA LATIH..... 83

BAB 1 PENDAHULUAN

Bab ini menguraikan mengenai latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah dan sistematika pembahasan Sistem Pendukung Keputusan Deteksi Dini Penyakit Stroke Menggunakan Metode *Naïve Bayes*.

1.1 Latar Belakang

Stroke adalah suatu keadaan yang timbul karena berhentinya suplai darah ke bagian otak yang mengakibatkan kematian jaringan otak sehingga seseorang dapat menderita kelumpuhan bahkan kematian (Batticaca, 2012). *Stroke* merupakan *deficit neurologis* (gangguan saraf) lokal dan atau global yang mempunyai serangan mendadak yang progresif cepat dan berlangsung 24 jam atau lebih sebagai akibat dari *Cardiovascular Diasease* (CVD) (Ariani, 2012). *Stroke* merupakan salah satu penyakit tersering penyebab kesakitan dan kematian nomor dua di Eropa serta nomor tiga di Amerika Serikat atau yang paling ditakuti di negara maju setelah penyakit jantung dan kanker (Batticaca, 2012). *Stroke* juga merupakan penyebab utama kecacatan atau disabilitas serius jangka panjang tertinggi yang harus ditangani secara cepat dan tepat (Muttaqin, 2008). *Stroke* di dunia mengenai 500.000 orang per tahun, dan jumlah ini meningkat setiap tahunnya pada orang dewasa (Mandera, 2006).

Kesibukan yang luar biasa khususnya di kota besar membuat aktivitas manusia menurun sehingga mengabaikan kesehatan tubuhnya. Seiring dengan menurunnya kualitas dan gaya hidup seperti pola makan tidak teratur, kurang olahraga, stress, merokok, konsumsi alkohol, jam kerja berlebihan, serta konsumsi makanan cepat saji sudah menjadi suatu kebiasaan yang berpotensi menjadi faktor awal yang dapat menimbulkan serangan *stroke* (Mukhlis, 2010). Pemeriksaan dini penyakit *stroke* oleh dokter sebaiknya perlu dilakukan untuk mencegah dan mengurangi risiko terserang *stroke*. Dokter mengalami kendala untuk bisa menyimpulkan pasien terserang *stroke* jika data gejala yang ada semi terstruktur atau data yang ada membutuhkan perhitungan-perhitungan serta analisis yang terperinci. Pelayanan kesehatan yang kurang memadai karena terbatasnya tenaga medis dalam memutuskan pasien terserang *stroke* menjadi salah satu alasan seorang dokter membutuhkan Sistem Pendukung Keputusan (SPK).

SPK mempermudah dokter melakukan deteksi dini penyakit *stroke* karena SPK dapat membantu memetakan konsep medis kedalam model komputasi. SPK merupakan sistem informasi yang menyediakan informasi, pemodelan, dan pemanipulasian data secara interaktif. SPK dimaksudkan untuk mendukung para pengambil keputusan dalam situasi yang semi terstruktur dan situasi yang tidak terstruktur, dimana tidak ada seorangpun yang tahu secara pasti bagaimana seharusnya keputusan dibuat (E. Turban, 2005). Dengan adanya SPK ini, diharapkan dapat membantu dokter untuk memaksimalkan deteksi dini penyakit

stroke. Beberapa penelitian terkait penggunaan SPK umumnya dapat membantu pengambilan keputusan dengan baik.

Penelitian terhadap penyakit *Stroke* pernah dilakukan oleh Linda Atika menggunakan metode *Certainty Factor* (CF). Dalam penelitian tersebut terdapat kriteria penilaian seperti jenis kelamin, usia, riwayat, hipertensi, jantung, diabetes, kolesterol, obesitas, merokok, inaktivitas fisik, dan alat kontrasepsi. Hasil dari penelitian ini berupa angka persen kemungkinan penyakit *stroke* yang diperoleh dari dialog interaktif antara sistem pakar dan pasien dimana pasien akan memberikan jawaban atas pertanyaan tersebut. Keterbatasan dari metode ini adalah bila ada *rule-rule* baru yang ditambahkan setelah CF(*Rule*) dihitung, maka perhitungan CF(*Rule*) harus dihitung dari awal (Atika, 2012).

Penelitian selanjunya dilakukan oleh Ahmad Fashel Sholeh, terhadap penyakit *Stroke* menggunakan metode *Fuzzy Mamdani*. Dalam penelitian tersebut terdapat kriteria penilaian seperti tekanan darah, kadar gula darah, kolesterol total, kolesterol-*low density lipoprotein* (LDL), usia, asam urat, jenis kelamin, *blood urea nitrogen* (BUN) dan kreatinin. Hasil dari penelitian yaitu risiko penyakit *stroke* tinggi, sedang, rendah dengan akurasi 82,98%. Keterbatasan metode ini adalah setiap aturannya masih bersifat perkiraan (Ahmad Fashel Sholeh, 2012). Selanjutnya penelitian yang dilakukan oleh M. Khusnul Mukhlis terhadap penyakit *Stroke* menggunakan metode *Naive Bayes* dan metode Jaringan Syaraf Tiruan. Dalam penelitian tersebut terdapat kriteria penilaian seperti Umur, Nadi, Jenis Kelamin, Penyakit Darah Tinggi, Penyakit Diabetes, Jantung, dan Darah. Hasil dari penelitian yaitu diagnosa pasien *suspect stroke* atau tidak *suspect stroke* dengan akurasi 99% pada metode Jaringan Syaraf Tiruan dan 97% pada metode *Naive Bayes* (Mukhlis, 2010).

Penelitian selanjunya dilakukan oleh Sri Kusumadewi terhadap status gizi menggunakan *Naive Bayesian Classification*. Dalam penelitian tersebut terdapat lima (5) kategori penilaian seperti tinggi badan, berat badan, *sex*, lingkar pergelangan, dan lingkar perut, serta hasil status gizi. Hasil akurasi dari penelitian ini yaitu 93,2% dengan *output* kurus, normal, obesitas ringan, obesitas sedang dan obesitas berat. Keterbatasan metode ini adalah jika data latih tidak memiliki data yang merata pada setiap parameternya, akan mengurangi tingkat akurasi (Kusumadewi, Mei 2009).

Berdasarkan paparan penelitian, penulis mencoba membangun SPK berbasis web menggunakan metode *Naive Bayes*. Metode ini dipilih karena teknik prediksi berbasis probabilistik yang sederhana dengan asumsi ketidaktergantungan yang kuat. Dapat dikatakan fitur pada metode *Naive Bayes* dapat bekerja secara independen (Prasetyo, 2012). SPK ini bekerja dengan cara menerima *input* faktor risiko pada pasien. Melalui data pasien tersebut akan dilakukan penalaran yang dikombinasikan dengan algoritma *Naive Bayes*. Pada penelitian ini, untuk menentukan faktor risiko terkena *stroke* dibutuhkan 8 (delapan) parameter yang terdiri dari tekanan darah, diabetes, riwayat keluarga, merokok, kolesterol, aktivitas fisik, diet, dan riwayat fibrilasi atrium. Data uji penentuan faktor yang digunakan berdasarkan data pada web *National Stroke*

Association (Association, 2009). Dari data faktor risiko penyakit *stroke* akan digunakan untuk menentukan risiko pasien terkena *stroke* ke dalam 3 (tiga) kelas *suspect stroke* yaitu risiko tinggi (*High Risk*), risiko sedang (*Caution*), dan risiko rendah (*Low Risk*). Berdasarkan penjelasan tersebut, maka pada penelitian ini penulis mengambil judul "*Sistem Pendukung Keputusan Deteksi Dini Penyakit Stroke Menggunakan Metode Naive Bayes*".

1.2 Rumusan Masalah

Berdasarkan uraian dari latar belakang yang telah dipaparkan, maka dapat diuraikan rumusan masalah yang mendasari penelitian ini, diantaranya adalah sebagai berikut :

1. Bagaimana merancang sistem pendukung keputusan deteksi dini penyakit *stroke* menggunakan metode *Naive Bayes* ?
2. Bagaimana implementasi dari sistem pendukung keputusan deteksi dini penyakit *stroke* menggunakan metode *Naive Bayes* ?.
3. Bagaimana hasil pengujian akurasi pada sistem pendukung keputusan deteksi dini penyakit *stroke* menggunakan metode *Naive Bayes* ?

1.3 Batasan Masalah

Berdasarkan uraian dari latar belakang masalah dan rumusan masalah yang telah dikemukakan, maka dapat disimpulkan bahwa ruang lingkup pembahasan dibatasi pada :

1. SPK ini ditujukan dan dirancang untuk digunakan oleh dokter dalam melakukan deteksi dini risiko penyakit *stroke*.
2. Data yang digunakan sebagai data *training* pada skripsi ini adalah dataset *stroke,* yang diambil berdasarkan hasil tes laboratorium pasien *stroke* dan faktor risiko penyakit *stroke* pada web *National Stroke Association*.
3. *Input* yang digunakan untuk mengidentifikasi faktor risiko *stroke* ada 8 (delapan), yaitu : tekanan darah, diabetes, riwayat keluarga, merokok, kolesterol, aktivitas fisik, diet, dan riwayat fibrilasi atrium. Dengan hasil identifikasi yang dihasilkan dari proses klasifikasi, terdiri dari tiga kelas yaitu menderita gangguan *stroke* dengan risiko tinggi, risiko sedang, dan risiko rendah yang bersumber dari web *National Stroke Association*.
4. Sistem yang dibangun bertujuan untuk mendukung pengambilan keputusan bukan untuk menggantikan pengambil keputusan dalam menentukan keputusan akhir.

1.4 Tujuan Penelitian

Tujuan yang ingin dicapai dalam penelitian ini adalah untuk merancang, mengimplementasikan, dan mengetahui hasil pengujian akurasi hasil klasifikasi pada Sistem Pendukung Keputusan Deteksi Dini Penyakit *Stroke* Menggunakan Metode *Naive Bayes.*

1.5 Manfaat Penelitian

Manfaat yang dapat diperoleh dari penelitian ini adalah :

1. SPK ini dapat digunakan dalam memajukan teknologi untuk memecahkan masalah dan memberikan informasi tentang penanggulangan risiko *stroke*, agar masyarakat dapat meningkatkan kualitas kesehatan secara dini.
2. Memberikan kontribusi dalam bidang pengetahuan tentang penerapan SPK Deteksi Dini Penyakit *Stroke*, karena dengan semakin banyak ilmu yang diterapkan akan didapatkan penelitian yang beragam dengan hasil pengujian yang semakin akurat.
3. Memudahkan dokter karena sistem yang dikembangkan dapat dimanfaatkan sebagai metode alternatif dan juga sebagai alat bantu dalam mendeteksi dini risiko terkena penyakit *stroke* menggunakan model komputasi dengan data yang semi terstruktur.

1.6 Sistematika Penulisan

Sistematika penulisan yang disusun pada skripsi ini terdapat 7 bab dan diuraikan sebagai berikut :

BAB I PENDAHULUAN

Pada Bab ini berisi tentang latar belakang pemilihan judul, rumusan masalah, batasan masalah, tujuan, manfaat, dan sistematika penulisan untuk SPK deteksi dini penyakit *stroke*. Objek yang dipilih adalah untuk faktor risiko penyakit *stroke* sedangkan metode yang digunakan adalah *Naive Bayes*.

BAB II LANDASAN KEPUSTAKAAN

Pada Bab ini akan membahas referensi dan teori yang mendukung dalam perancangan dan pengembangan SPK. Beberapa teori yang dibutuhkan antara lain : teori yang berkaitan dengan SPK, metode *Naive Bayes*, dan faktor risiko penyakit *Stroke*. Sedangkan kajian pustaka ditampilkan dalam bentuk tabel yang berisi beberapa penelitian yang terkait untuk SPK deteksi dini penyakit *stroke*.

BAB III METODOLOGI PENELITIAN

Pada Bab ini membahas tentang metodologi dalam penulisan yang terdiri dari studi literatur, analisis kebutuhan, pengambilan data, perancangan sistem, implementasi, pengujian, dan pengambilan kesimpulan.

BAB IV PENGUJIAN

Bab ini memuat hasil pengujian terhadap sistem yang telah diimplementasikan. Metode pengujian yang digunakan adalah metode *black box testing*. Kemudian hasil dari pengujian juga akan dianalisis untuk mengetahui bagaimana kesesuaian fungsionalitas sistem dengan kebutuhan sistem.

BAB V PENUTUP

Bab ini memuat kesimpulan yang diperoleh dari pembuatan dan pengujian perangkat lunak yang dikembangkan dalam SPK deteksi dini penyakit *stroke* menggunakan metode *Naive Bayes* dan saran untuk pengembangan penelitian selanjutnya.

BAB 2 LANDASAN KEPUSTAKAAN

Pada bab ini berisi tinjauan pustaka yang meliputi kajian pustaka dan dasar teori yang diperlukan untuk penelitian mengenai deteksi dini penyakit *stroke* menggunakan metode *Naive Bayes*. Kajian pustaka pada penelitian ini adalah membahas penelitian sebelumnya dan metode yang akan diterapkan serta penelitian yang akan diusulkan. Kajian pustaka yang dicantumkan pada bab ini yaitu dari hasil penelitian yang pernah dilakukan oleh Linda Atika (Atika, 2012), Ahmad Fashel Sholeh (Ahmad Fashel Sholeh, 2012), M. Khusnul Mukhlis (Mukhlis, 2010), Ni Made Ari Lestari, dkk (N. M. Lestari, Januari 2013), Angga Hardika P. (Pratama, 2014), Sri Kusumadewi (Kusumadewi, Mei 2009), dan Elyza Gustri Wahyuni, dkk (Prijodiprojo, 2013).

Dasar teori adalah membahas teori yang diperlukan untuk menyusun penelitian yang diusulkan. Dasar teori yang akan dibahas yaitu Sistem Pendukung Keputusan, Penyakit *Stroke*, Metode *Naive Bayes*, *Data Flow Diagram* (DFD), dan *Entity Relationship Diagram* (ERD).

2.1 Kajian Pustaka Terkait Penelitian Sebelumnya

Berdasarkan judul skripsi yang dibahas, penulis akan menjelaskan beberapa hasil penelitian yang relevan untuk mendukung penelitian dalam skripsi ini. Beberapa penelitian tentang *stroke* dan penelitian lain yang telah dilakukan dengan menggunakan beberapa metode terkait penggunaan metode, dan penelitian terkait metode pembanding. Penelitian pertama yang dilakukan oleh Linda Atika tentang sistem pakar pendeteksi prediksi penyakit *stroke* menggunakan metode *Certainty Factor* dengan sebelas (11) parameter data pasien. Penelitian kedua dari Ahmad Fashel Sholeh tentang permasalahan untuk mendiagnosa pasien terkena *stroke* menggunakan Logika Fuzzy Mamdani dengan sembilan (9) parameter data pasien. Penelitian ketiga dari M. Khusnul Mukhlis tentang permasalahan untuk mendiagnosa pasien terkena *stroke* menggunakan metode *Naïve Bayes* dan Jaringan Syaraf Tiruan dengan tujuh (7) parameter data pasien. Penelitian keempat dari Ni Made Ari Lestari, dkk. tentang permasalahan untuk klasifikasi pencarian pasangan dengan tulisan melalui *website* menggunakan metode *Naïve Bayes* dengan empat (4) parameter data individu. Penelitian kelima dari Angga Hardika P. tentang sistem pakar untuk identifikasi hama dan penyakit tanaman tebu menggunakan metode *Naïve Bayes* dengan tiga (3) parameter data tanaman tebu. Penelitian keenam dari Sri Kusumadewi tentang klasifikasi status gizi menggunakan metode *Naïve Bayes Classifier* dengan lima (5) parameter data pasien. Penelitian ketujuh dari Elyza Gustri Wahyuni tentang sistem pakar untuk mendeteksi jantung koroner menggunakan metode *Dempster-Shafer* dengan delapan (8) parameter data pasien. Untuk mengetahui uraian dari penelitian sebelumnya dapat dilihat pada Tabel 2.1 berikut.

Tabel 2.1 Kajian Pustaka

No.	Judul	Obyek Input dan Parameter	Metode Proses	Hasil Penelitian Output
1.	Sistem Pakar Pendeteksi Prediksi Kemungkinan Penyakit *Stroke* (Atika, 2012)	Deteksi Penyakit *Stroke*	Metode *Certainty Factor*	Deteksi Penyakit *Stroke*
		Data Kondisi Pasien : 1. Jenis Kelamin, 2. Usia, 3. Riwayat, 4. Hipertensi, 5. Jantung, 6. Diabetes, 7. Kolesterol, 8. Obesitas, 9. Merokok, 10. Inaktivitas Fisik, dan 11. Alat Kontrasepsi	Langkah-langkah *Certainty Factor* : 1. Menentukan bobot tiap gejala $CF(E_i)$ 2. Melakukan perhitungan pengaruh kombinasi gejala: $CF(E) = \sum_{i=1}^{n} \omega_i * CF(E_i)$ 3. Melakukan perhitungan pengambilan kesimpulan CF(H) dengan rumus : $CF(H) = CF(H,E)*CF(E)$	Presentase kemungkinan penyakit *Stroke*
2.	Aplikasi Pendukung Keputusan Untuk Deteksi Dini Risiko Penyakit *Stroke* Menggunakan Logika *Fuzzy* Mamdani : Studi Kasus Di RS XYZ (Ahmad Fashel Sholeh, 2012)	Deteksi Dini Risiko Penyakit *Stroke*	Menggunakan Logika *Fuzzy* Mamdani	Diagnosa Status Risiko *Stroke*
		Data Kondisi Pasien : 1. Tekanan Darah, 2. Umur, 3. Jenis Kelamin, 4. Kolesterol Total, 5. Kolesterol LDL, 6. Kadar Gula Darah, 7. Asam Urat, 8. *Blood Urea Nitrogen*, dan 9. Kreatinin	Langkah-langkah *Fuzzy* Mamdani : 1. Menentukan variabel masukan 2. Menentukan fungsi keanggotaan pada tiap variabel 3. Identifikasi aturan logika *fuzzy*	1. Rendah 2. Normal 3. Tinggi

Tabel 2.1 Kajian Pustaka (lanjutan)

No.	Judul	Obyek	Metode	Hasil Penelitian
		Input dan Parameter	**Proses**	**Output**
3.	Diagnosa Kemungkinan Pasien Terkena *Stroke* Dengan Menggunakan Metode *Naive Bayes* Dan Metode Jaringan Syaraf Tiruan Berbasis Web (Mukhlis, 2010)	Diagnosa Kemungkinan Pasien Terkena *Stroke*	Metode *Naive Bayes* Dan Metode Jaringan Syaraf Tiruan	Diagnosa Status Risiko *Stroke*
		1. Umur, 2. Nadi, 3. Jenis Kelamin, 4. Penyakit Darah Tinggi, 5. Penyakit Diabetes, 6. Jantung, dan 7. Darah	*Naive Bayes* : 1. Mencari nilai probabilitas 2. Membuat tabel probablilitas 3. Mencari nilai *mean* dari setiap parameter 4. Mencari nilai deviasi dari masing-masing parameter 5. Membuat tabel *mean* dan deviasi Jaringan Saraf Tiruan : 1. Inisialisasi penimbang 2. Menentukan nilai *learning rate* 3. Hitung keluaran JST	1. Suspect *Stroke* 2. Tidak Suspect *Stroke*
4.	*Personality Types Classification for Indonesian Text in Partners Searching Website Using Naive Bayes Methods* (N. M. Lestari, Januari 2013)	Klasifikasi Tulisan Bahasa Indonesia dalam menentukan kepribadian seseorang	Metode *Naive Bayes*	Pemilihan pasangan sesuai dengan kepribadian berdasarkan tulisan tangan
		1. Sanguine, 2. Choleric, 3. Melancholy, dan 4. Phlegmatic	*Naive Bayes:* 1. *Input* deskripsi kepribadian 2. Menghitung nilai masing-masing kategori: $P(V_j) \Pi P(a_i \mid V_j)$ 3. Menentukan kategori dengan: $P(V_j) \Pi P(a_i \mid V_j)$ max 4. Kategori Dokumen	1. True 2. False

Tabel 2.1 Kajian Pustaka (lanjutan)

No.	Judul	Obyek	Metode	Hasil Penelitian
		Input dan Parameter	**Proses**	**Output**
5.	Aplikasi Sistem Pakar Untuk Identifikasi Hama dan Penyakit Tanaman Tebu Dengan Metode *Naive Bayes* Berbasis Web (Pratama, 2014)	Hama Tanaman Tebu	Metode *Naive Bayes*	Penentuan Hama dan Penyakit Tanaman Tebu
		1. Gejala umum dengan gejala yang ditunjukkan seperti tanaman mati. 2. Keadaan Daun dengan gejala yang ditunjukkan seperti terdapat noda pada daun. 3. Keadaan Batang dengan gejala yang ditunjukkan seperti batang mengering, batang roboh, dan batang bengkok	*Naive Bayes* : 1. Menghitung prior 2. Menghitung *likelihood* 3. Menghitung posterior dan klasifikasi	Solusi dan penanganannya
6.	Klasifikasi Status Gizi Menggunakan *Naive Bayesian Classification* (Kusumadewi, Mei 2009)	Klasifikasi Status Gizi	Metode *Naive Bayesian Classification*	Hasil Klasifikasi Status Gizi
		1. Tinggi Badan, 2. Berat Badan, 3. *Sex*, 4. Lingkar Pergelangan, dan 5. Lingkar Perut	*Naive Bayes* : 1. Menghitung *mean* dan standar deviasi 2. Menghitung probabilitas kategori status gizi variabel diskret 3. Menghitung probabilitas setiap kategori 4. Menghitung *likelihood* 5. Mengestimasi Probabilitas P(a_1 / v_1)	1. Kurus 2. Normal 3. Obesitas Ringan 4. Obesitas Sedang 5. Obesitas Berat

Tabel 2.1 Kajian Pustaka (lanjutan)

No.	Judul	Obyek	Metode	Hasil Penelitian
		Input dan Parameter	**Proses**	**Output**
7.	*Prototype* Sistem Pakar untuk Mendeteksi Tingkat Resiko Penyakit Jantung Koroner dengan Metode *Dempster-Shafer* (Studi Kasus : RS. PKU Muhammadiyah Yogyakarta) (Prijodiprojo, 2013)	Deteksi Resiko Penyakit Jantung Koroner	Metode *Dempster-Shafer*	Deteksi Tingkat Resiko Penyakit Jantung Koroner
		1. Kolesterol, 2. Gula Darah, 3. Tekanan Darah, 4. BMI, 5. Usia, 6. Jenis Kelamin, 7. Batuk, dan 8. Sesak Napas	*Dempster-Shafer :* 1. Mencari data kriteria gejala 2. Menentukan nilai densitas gejala antara 0-1 3. Menentukan *frame of discrement* 4. Menentukan probabilitas densitas (m) 5. Melakukan perhitungan kombinasi m1 dan m2 6. Identifikasi variabel keluaran	1. Jantung Koroner Berat 2. Jantung Koroner Sedang 3. Jantung Koroner Ringan
8.	Sistem Pendukung Keputusan Deteksi Dini Penyakit *Stroke* Menggunakan Metode *Naive Bayes* [usulan]	Deteksi Dini Penyakit *Stroke*	Metode *Naive Bayes*	Diagnosa Status Risiko *Stroke*
		1. Tekanan Darah 2. Diabetes 3. Riwayat Keluarga 4. Merokok 5. Kolesterol 6. Aktifitas Fisik 7. Diet 8. Riwayat Fibrilasi Atrium	*Naive Bayes :* 1. Perhitungan nilai *mean* dan standar deviasi pada kriteria yang bersifat kontinu. 2. Perhitungan nilai probabilitas sebuah fakta yang bersifat diskret. 3. Perhitungan nilai probabilitas akhir 4. Memilih nilai hasil perhitungan probabilitas akhir yang terbesar sebagai hasil deteksi	1. Rendah 2. Sedang 3. Tinggi

Sumber : (Atika, 2012), (Ahmad Fashel Sholeh, 2012), (Mukhlis, 2010), (N. M. Lestari, Januari 2013), (Pratama, 2014), (Kusumadewi, Mei 2009), (Prijodiprojo, 2013), [usulan]

Penelitian terhadap penyakit *stroke* pernah dilakukan oleh Linda Atika tahun 2012 dengan judul "*Sistem Pakar Pendeteksi Prediksi Kemungkinan Penyakit Stroke*", menggunakan metode *Certainty Factor*. Dalam penelitian tersebut terdapat sebelas (11) kriteria penilaian seperti jenis kelamin, usia, riwayat, hipertensi, jantung, diabetes, kolesterol, obesitas, merokok, inaktivitas fisik, dan alat kontrasepsi. Hasil dari penelitian ini berupa angka persen kemungkinan penyakit *stroke* yang diperoleh dari dialog interaktif antara sistem pakar dan pasien dimana pasien akan memberikan jawaban atas pertanyaan yang ditanyakan sistem (Atika, 2012).

Penelitian terhadap penyakit *stroke* selanjunya dilakukan oleh Ahmad Fashel Sholeh tahun 2012 dengan judul "*Aplikasi Pendukung Keputusan Untuk Deteksi Dini Risiko Penyakit Stroke Menggunakan Logika Fuzzy Mamdani : Studi Kasus Di RS XYZ*". Dalam penelitian tersebut menggunakan sembilan (9) kriteria penilaian seperti tekanan darah, kadar gula darah, kolesterol total, *kolesterol-low density* lipoprotein (LDL), usia, asam urat, jenis kelamin, *blood urea nitrogen* (BUN) dan kreatinin. Hasil kalkulasi perhitungan dari *inputan* dengan Logika *Fuzzy* Mamdani digunakan untuk memutuskan kategori tingkat risiko penyakit *stroke* yang diderita pasien. Hasil dari penelitian yaitu risiko penyakit *stroke* tinggi, sedang, rendah dengan akurasi 82,98% (Ahmad Fashel Sholeh, 2012).

Penelitian terhadap penyakit *stroke* selanjunya dilakukan oleh M. Khusnul Mukhlis pada tahun 2011 dengan judul "*Diagnosa Kemungkinan Pasien Terkena Stroke Dengan Menggunakan Metode Naive Bayes dan metode Jaringan Syaraf Tiruan Berbasis Web*". Dalam penelitian tersebut terdapat delapan (8) kriteria penilaian seperti Umur, Nadi, Jenis Kelamin, Penyakit Darah Tinggi, Penyakit Diabetes, Jantung, dan Darah. Hasil inputan dari *user* akan masuk kedalam *database*, kemudian program akan membaca *database* untuk melakukan proses *training*. Dari masing-masing algoritma akan memunculkan nilai W (untuk JST) dan nilai *mean*, standart deviasi, dan probabilitas (untuk *Naive Bayes*) dimana data inputan dihitung kemungkinannya, dan kemungkinan terbesar menjadi kesimpulannya. Hasil dari penelitian yaitu diagnosa pasien *suspect stroke* atau tidak *suspect stroke* dengan akurasi 99% pada metode Jaringan Syaraf Tiruan dan 97% pada metode *Naive Bayes* (Mukhlis, 2010).

Penelitian selanjunya dilakukan oleh Ni Made Ari Lestari, I Ketut Gede Darma Putra, dan AA Ketut Agung Cahyawan pada tahun 2013 dengan judul "*Personality Types Classification for Indonesian Text in Partners Searching Website Using Naive Bayes Methods*". Dalam penelitian tersebut terdapat empat (4) kategori penilaian seperti Sanguine, Choleric, Melancholy, dan Phlegmatic. Penelitian ini melakukan inputan data dari *user* yang dibaca oleh program sebagai *text*-mining dan dihitung dengan metode *Naive Bayes*. Perhitungan dengan kategori tertinggi akan dipilih sebagai *potensial partner*. Hasil akurasi dari penelitian ini yaitu 92,5% dengan *output* benar dan salah (N. M. Lestari, Januari 2013).

Penelitian selanjutnya dilakukan oleh Angga Hardika P. pada tahun 2014 dengan judul "*Aplikasi Sistem Pakar Untuk Identifikasi Hama Dan Penyakit*

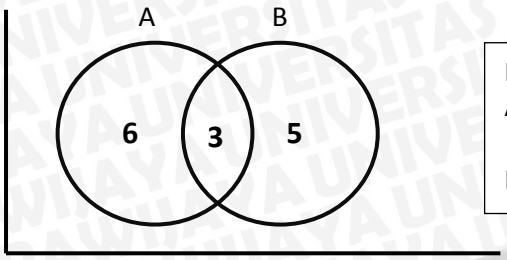
Tanaman Tebu Dengan Metode Naive Bayes Berbasis Web". Objek dari penelitian tersebut adalah tanaman tebu dan metode yang digunakan adalah metode *Naive Bayes*. Dalam penelitian tersebut terdapat delapan (8) gejala seperti Penggerek Pucuk, Penggerek Batang, Boktor, Kumbang Penggerek Tebu, Rayap, Uret, Tonggeret, dan Kepinding Tanah. Proses identifikasi hama dan penyakit tebu dilakukan dengan memasukkan fakta gejala yang nanti akan dihitung dengan metode *Naive Bayes*. Jenis hama/penyakit tebu dengan nilai probabilitas tertinggi akan diambil sebagai hasil identifikasi sistem. Dalam penelitian tersebut keluaran berupa probabilitas jenis penyakit tebu, dan memberikan solusi penanganan yang tepat untuk jenis penyakit tebu tersebut. Hasil akurasi dari penelitian ini yaitu 94,28% (Pratama, 2014).

Penelitian selanjunya dilakukan oleh Sri Kusumadewi pada tahun 2009 dengan judul "Klasifikasi Status Gizi Menggunakan *Naive Bayesian Classification*". Dalam penelitian tersebut terdapat lima (5) kategori penilaian seperti tinggi badan, berat badan, *sex*, lingkar pergelangan, dan lingkar perut, serta hasil status gizi. Pengujian dilakukan dengan menggunakan sampel data mahasiswa sebagai data *training* untuk menghitung data *testing*. Perhitungan dengan masing-masing algoritma akan diulang beberapa kali untuk mendapatkan keluaran parameter terbaik. Hasil akurasi dari penelitian ini yaitu 93,2% dengan *output* kurus, normal, obesitas ringan, obesitas sedang dan obesitas berat (Kusumadewi, Mei 2009).

Penelitian yang dilakukan oleh Elyza Gustri Wahyuni dan Widodo Prijodiprojo tahun 2013 dengan judul "*Prototype Sistem Pakar Untuk Mendeteksi Tingkat Resiko Penyakit Jantung Koroner Dengan Metode Dempster Shafer*". Dalam penelitian tersebut terdapat delapan (8) kriteria penilaian seperti kolesterol, gula darah, tekanan darah, BMI, usia, jenis kelamin, batuk sesak napas. Penelitian ini melakukan perhitungan gejala dan faktor resiko dari kemungkinan awal sampai kemungkinan terakhir sebagai fungsi densitasnya. Hasil densitas terbesar akan menjadi hasil diagnosa. Hasil dari penelitian yaitu diagnosa Jantung Koroner Berat, Jantung Koroner Sedang, Jantung Koroner Ringan, dengan presentase akurasi sebesar 100% *(Prijodiprojo, 2013)*.

Pada penelitian sebelumnya telah dibahas penelitian mengenai "Aplikasi Pendukung Keputusan Untuk Deteksi Dini Risiko Penyakit *Stroke* Menggunakan Metode *Fuzzy* Mamdani" yang dilakukan oleh Ahmad Fashel Sholeh (Ahmad Fashel Sholeh, 2012). Perbedaan pada penelitian sebelumnya dengan penulis yaitu dari kriteria faktor risiko penyakit *stroke* yang dideteksi oleh sistem dan metode yang digunakan pada SPK. Pada kedua penelitian *Stroke* sesuai Gambar 2.1, terdapat 3 (tiga) persamaan variabel masukan antara penulis dan peneliti sebelumnya yaitu tekanan darah, kolesterol, dan diabetes. Sedangkan terdapat lima (5) perbedaan parameter seperti tekanan darah, riwayat keluarga, merokok, aktivitas fisik, dan riwayat fibrilasi atrium sesuai dengan data yang diperoleh dari web *National Stroke Association* (NSA).

Perbedaan kriteria faktor risiko *stroke* dapat dilihat pada diagram venn seperti pada Gambar 2.1 berikut.



Keterangan :
A = Penelitian Sebelumnya tentang *Stroke*
B = Penelitian Penulis

Gambar 2.1 Diagram Venn Variabel Masukan *Stroke*

Sumber : [Perancangan]

Ada 6 (enam) gejala berbeda yang digunakan oleh peneliti sebelumnya yaitu usia, asam urat, jenis kelamin, BUN (*Blood Urea Nitrogen*), dan kreatinin. Parameter yang digunakan pada penelitian ini berdasarkan pada parameter stroke yang sudah diteliti oleh NSA (Association, 2009). Selain itu metode yang digunakan pada penelitian ini adalah *Naive Bayes*, sedangkan penelitian sebelumnya menggunakan metode *Fuzzy* Mamdani. Sehingga, studi kasus penelitian ini akan menggunakan 8 variabel masukan antara lain tekanan darah, diabetes, riwayat keluarga, merokok, kolesterol, aktivitas fisik, diet, dan riwayat *fibrilasi atrium* (detak jantung). Sedangkan *output* dari penelitian yang akan dilakukan dan penelitian sebelumnya sama yaitu risiko tinggi, risiko sedang dan risiko rendah.

Berdasarkan dari beberapa penelitian yang telah dilakukan, maka penulis mengusulkan penelitian dengan judul "Sistem Pendukung Keputusan Deteksi Dini Penyakit *Stroke* Menggunakan Metode *Naive Bayes*" dengan data uji penentuan faktor yang digunakan berdasarkan pada web *National Stroke Association* (Association, 2009). Penelitian ini membahas deteksi dini risiko penyakit *stroke* menggunakan metode *Naive Bayes* untuk menentukan risiko pasien terkena *stroke* ke dalam 3 (tiga) kelas *suspect stroke* yaitu risiko tinggi, risiko sedang, dan risiko rendah. *Naive Bayes* merupakan pengembangan dari teknik probabilitas yang digunakan untuk menangani ketidakpastian. Terdapat beberapa teknik untuk menangani masalah ketidakpastian antara lain teknik probabilitas, faktor kepastian (semi probabilitas), dan logika *fuzzy* (Pratama, 2014). Berikut ini akan dijabarkan karakteristik metode ketidakpastian yang terkait dengan metode yang akan digunakan oleh peneliti yaitu *Certainty Factor*, *Dempster-Shafer*, dan *Naive Bayes*. Karakteristik dari metode-metode tersebut akan disajikan dalam Tabel 2.2 berikut ini.

Tabel 2.2 Karakteristik Metode

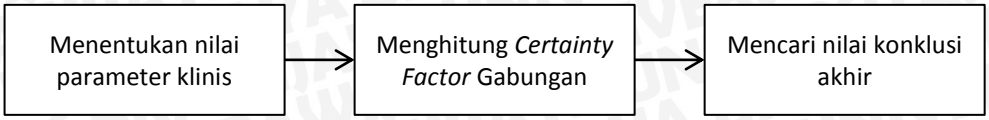
	Certainty Factor	***Dempster-Shafer***	***Naive Bayes***
Penjelasan	• Pada metode *Certainty Factor* terdapat nilai kepercayaan, nilai kepercayaan ini merupakan nilai yang mewakili	• Pada metode *Dempster-Shafer* terdapat *Frame of Discernment* yang merupakan himpunan semesta pembicaraan dari	• Pada metode *Naive Bayes* terdapat fitur independen atau memiliki ketidaktergantungan (independensi) yang kuat (Naif)

Tabel 2.2 Karakteristik Metode (lanjutan)

	Certainty Factor	***Dempster-Shafer***	***Naive Bayes***
	derajat kepastian suatu premis. • Terdapat dua nilai kepercayaan pada metode ini yaitu nilai kepercayaan yang diberikan oleh pakar dan nilai kepercayaan yang diberikan oleh pengguna	sekumpulan hipotesa. • Metode ini menggunakan nilai densitas dalam melakukan proses perhitungan, nilai densitas ini menunjukkan besarnya kepercayaan *evidence*	• Fitur yang dimiliki oleh sebuah data tidak berkaitan dengan ada atau tidaknya fitur yang lain dalam data
Input	Nilai Kepercayaan Parameter	Nilai Densitas Parameter	Parameter
Penalaran	*Rule*	Densitas	Probabilitas
Perhitungan	Menghitung seluruh kemungkinan konklusi	*Frame of Discrement*	Menghitung seluruh probabilitas tiap *class*
Output	• Hanya menghasilkan satu *output* solusi • *Output* dari metode ini adalah nilai kepercayaan tertinggi yang didapatkan dari hasil perhitungan kombinasi setiap *rule* yang terbentuk	• *Output* yang dihasilkan bisa lebih dari satu (alternaif solusi) • *Output* dari metode ini adalah nilai densitas tertinggi, jika nilai densitas tertinggi yang dihasilkan lebih dari satu maka akan digunakan sebagai sebagai alternatif solusi.	• Hanya menghasilkan satu *output* solusi • *Output* dari metode ini adalah nilai probabilitas terakhir (posterior) yang tertinggi

Sumber : (Atika, 2012) (Prijodiprojo, 2013) (Prasetyo, 2012)

Metode *Certainty Factor* merupakan cara dari penggabungan kepercayaan dan ketidakpercayaan dalam bilangan tunggal. Diciptakan untuk mengakomodir ungkapan yang biasa diungkapkan oleh dokter seperti “mungkin”, “kemungkinan besar”, atau “hampir pasti” (T. Sutojo, 2011). Cara kerja *Certainty Factor* yaitu dengan cara menggunakan nilai parameter klinis yang menunjukkan besarnya nilai kepercayaan (Atika, 2012). Semua kemungkinan konklusi yang mungkin akan dihitung sehingga nantinya terbentuk nilai akhir konklusi. Nilai konklusi terbesar akan dianggap sebagai konklusi akhir yang paling cocok sebagai keputusan akhir. Blok Diagram pada proses komputasi metode *Certainty Factor* ditunjukkan pada Gambar 2.2 berikut.



Gambar 2.2 Diagram Proses Komputasi Metode *Certainty Factor*

Sumber : (T. Sutojo, 2011)

Tabel 2.3 menunjukkan bahwa pada metode *Certainty Factor* jika dalam sebuah SPK terdapat beberapa kemungkinan konklusi maka nilai semua kemungkinan konklusi akan dihitung (T. Sutojo, 2011).

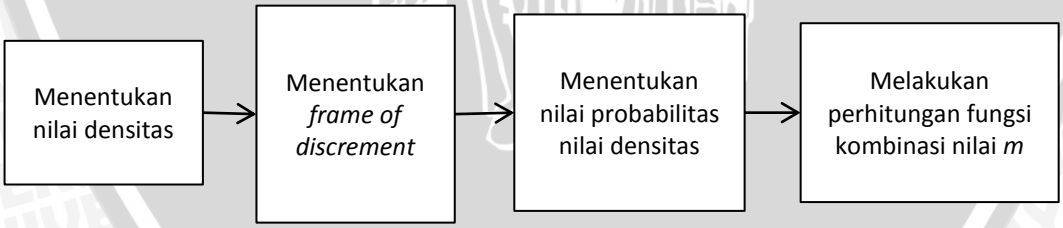
Tabel 2.3 Konklusi Metode *Certainty Factor*

Evidence	Solusi
Parameter Klinis	Penyakit A Penyakit A Penyakit C Kesimpulan A

Sumber : (T. Sutojo, 2011)

Metode *Dempster-Shafer* menunjukkan suatu cara untuk memberikan bobot keyakinan sesuai dengan fakta yang didapatkan. Metode ini dapat membedakan ketidakpastian dan ketidaktahuan. Teori *Dempster-Shafer* merupakan representasi dan kombinasi dari ketidakpastian, dimana teori ini memiliki beberapa karakteristik yang sesuai dengan cara berfikir seorang pakar, namun dengan dasar matematika yang kuat.

Metode *Dempster-Shafer* dikenal dengan adanya *frame of discrement* yang dinotasikan dengan θ. *Frame of discrement* merupakan semesta pembicaraan dari sekumpulan hipotesis sehingga sering disebut dengan *environtment* yang digunakan untuk mengaitkan kepercayaan elemen-elemen θ karena tidak semua *evidence* secara langsung mendukung tiap elemen (Prijodiprojo, 2013). Untuk itu dibutuhkan probabilitas fungsi densitas (*m*), dimana *m* tidak hanya mendefinisikan elemen-elemen θ saja, namun juga semua subsetnya. Hasil keluaran yang memiliki nilai terbesar akan menjadi keputusan sistem. Gambar 2.3 berikut merupakan diagram proses komputasi metode *Dempster-Shafer.*



Gambar 2.3 Diagram Proses Metode *Dempster-Shafer*

Sumber : (Prijodiprojo, 2013)

Tabel 2.4 menunjukkan bahwa pada metode *Dempster-Shafer* hanya akan menghitung kesimpulan yang merupakan superset dari fakta yang ada. Jika diketahui θ adalah *B*, *I*, *A*, dan *N*. Jika *X* adalah subset dari θ dengan *m1* sebagai fungsi densitasnya dan *Y* adalah subset dari θ dengan *m2* sebagai fungsi densitasnya. Maka dapat dibentuk fungsi kombinasi *m1* dan *m2* sebagai *m3*. Nilai densitas yang terbesar yang akan menjadi hasil keputusan akhir dari sistem.

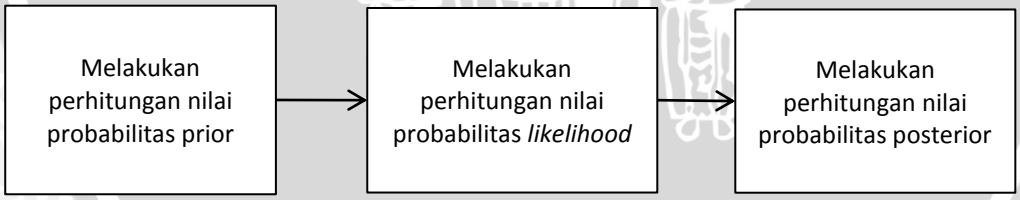
Tabel 2.4 Diagram Proses Pengambilan Keputusan *Dempster-Shafer*

Evidence	Solusi
Gejala	Nilai Kombinasi *m* → Penyakit B, Penyakit I, Penyakit A, Penyakit N

Sumber : (Prijodiprojo, 2013)

Metode yang ketiga yaitu *Naive Bayes,* merupakan metode atau teknik yang berbasis probabilistik sederhana yang berdasarkan pada penerapan teorema bayes dengan asumsi independensi yang kuat. Independensi merupakan sebuah fitur data yang tidak berkaitan dengan ada atau tidak fitur lain dalam data yang sama (Darmanto, 2014). Contohnya, pada kasus klasifikasi habitat hewan dengan fitur darat dan laut. Dalam dunia nyata, hewan yang memiliki lendir dan licin biasanya hidup di laut atau tempat yang berair, dan hewan yang memiliki kulit kasar atau berbulu biasanya hidup di darat atau udara. Di sini ada hubungan antara hewan dan habitatnya. Dalam Bayes, hal tersebut tidak dipandang sehingga pada masing-masing fitur seolah tidak memiliki hubungan apapun (Prasetyo, 2012).

Pada Gambar 2.4 menjelaskan bahwa metode *Naive Bayes* menghitung seluruh probabilitas *class* yang ada. Metode *Naive Bayes* bekerja dengan cara mengklasifikasikan secara statistik yang biasanya digunakan untuk memprediksi probabilitas keanggotaan suatu *class*. Tabel 2.5 merupakan diagram proses pengambilan keputusan metode *Naive Bayes*.



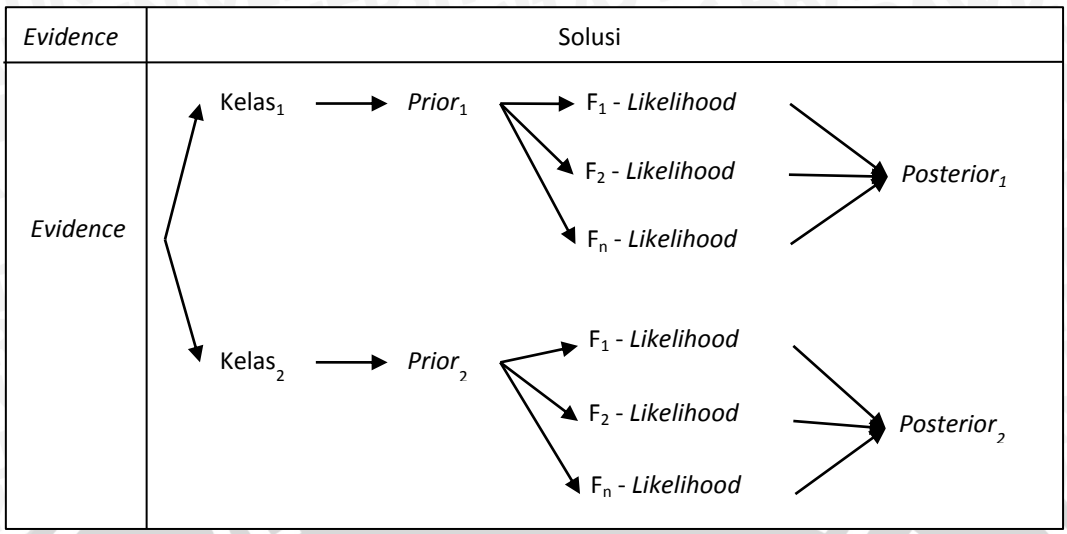
Gambar 2.4 Diagram Proses Komputasi Metode *Naive Bayes*

Sumber : (Prasetyo, 2012)

Adapun langkah-langkah pengerjaan metode *Naive Bayes* adalah :

1. Melakukan perhitungan nilai probabilitas prior pada setiap parameter
2. Melakukan perhitungan nilai probabilitas *likelihood* sebuah fakta
3. Melakukan perhitungan nilai probabilitas posterior masing-masing kelas
4. Melakukan sorting nilai probabilitas posterior. Nilai probabilitas posterior terbesar merupakan hasil keputusan sistem.

Tabel 2.5 Diagram Proses Pengambilan Keputusan Metode *Naive Bayes*



Sumber : (Prasetyo, 2012)

Metode *Naive Bayes* akan digunakan untuk menentukan nilai probabilitas yang nantinya digunakan untuk melakukan identifikasi tingkat kategori risiko penyakit *stroke*. Hasil pengolahan sistem adalah diagnosa penyakit *stroke*.

2.2 Sistem

Suatu sistem pada dasarnya adalah kumpulan dari elemen-elemen yang erat hubungannya satu dengan yang lain, yang saling berinteraksi untuk mencapai tujuan tertentu. Sistem merupakan bagian dari sistem buatan manusia yang dikontrol oleh satu atau lebih komputer yang dapat membantu aktivitas masyarakat modern. Secara sederhana suatu sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau variabel yang terorganisir, saling berinteraksi, saling tergantung satu sama lain, dan terpadu. Dari definisi ini dapat dirinci lebih lanjut pengertian sistem secara umum, yaitu (Jogiyanto, 1990):

1. Setiap sistem terdiri dari unsur-unsur.
2. Unsur-unsur tersebut merupakan bagian terpadu sistem yang terkontrol dan saling berinteraksi.
3. Unsur sistem tersebut bekerja sama untuk mencapai tujuan sistem.
4. Suatu sistem merupakan bagian dari sistem lain yang lebih besar.

2.3 Sistem Pendukung Keputusan (SPK)

SPK merupakan sistem informasi interaktif yang menyediakan informasi, pemodelan, dan pemanipulasian data. Sistem itu digunakan untuk membantu pengambilan keputusan dalam situasi yang semi terstruktur dan situasi tidak terstruktur, di mana tak seorang pun tahu secara pasti bagaimana keputusan seharusnya dibuat (E. Turban, 2005). SPK dimaksudkan digunakan sebagai alat bantu bagi para pengambil keputusan untuk memperluas kapabilitas mereka, namun tidak untuk menggantikan peran penilaian mereka. SPK ditujukan untuk

keputusan yang memerlukan penilaian yang sama sekali tidak dapat didukung oleh algoritma (E. Turban, 2005). Efraim Turban mengemukakan bahwa SPK merupakan sebuah sistem yang dibuat untuk mendukung para pengambil keputusan manajerial dalam situasi keputusan semi terstruktur (E. Turban, 2005). SPK merupakan sistem yang dapat diperluas untuk mampu mendukung pemodelan keputusan, berorientasi terhadap perencanaan masa depan, dan digunakan pada interval yang tidak regular dan tidak terencana (E. Turban, 2005).

Dari beberapa definisi di atas dapat dikatakan bahwa SPK adalah suatu sistem informasi spesifik yang ditujukan untuk membantu manajemen dalam mengambil keputusan yang berkaitan dengan persoalan yang bersifat semi terstruktur dan tidak terstruktur. Sistem ini memiliki fasilitas untuk menghasilkan berbagai alternatif yang secara interaktif dapat digunakan oleh pemakai. SPK memungkinkan untuk membantu pengambilan keputusan dalam melakukan analisa data yang dihasilkan dari pemrosesan transaksi dan sumber informasi internal dengan mudah. Kata berbasis komputer merupakan kata kunci, karena hampir tidak mungkin membangun SPK tanpa memanfaatkan komputer sebagai alat bantu, terutama untuk menyimpan data serta mengelola model.

2.3.1 Perbedaan Sistem Informasi, Sistem Pakar, dan SPK

Pada subbab ini akan membahas mengenai perbedaan Sistem Pendukung Keputusan (SPK), Sistem Informasi (SI), dan Sistem Pakar (SP). Berikut penjelasan untuk pengertian dari SI, SP, dan SPK.

1 Sistem Informasi
Sistem Informasi merupakan sekumpulan komponen untuk membentuk sistem yang saling berkaitan antara satu komponen dengan komponen lainnya, dengan tujuan untuk menghasilkan suatu informasi pada bidang tertentu (Putri, 2013).

2. Sistem Pakar
Sistem pakar merupakan suatu sistem yang menggunakan pengetahuan dari seorang pakar yang kemudian dimasukkan kedalam komputer. Seseorang yang bukan pakar menggunakan sistem pakar untuk bisa meningkatkan kemampuannya dalam memecahkan suatu masalah, sedangkan seorang pakar menggunakan sistem pakar untuk *knowledge assistant* (Prihatini, Juni 2012).

3. Sistem Pendukung Keputusan
Sistem pendukung keputusan adalah sistem berbasis komputer yang dirancang untuk mempertinggi efektifitas pengambilan keputusan dari masalah semi terstruktur (E. Turban, 2005).

Perbedaan antara SPK dengan SI yaitu SPK memiliki basis model dan basis pengetahuan sebelum proses pengambilan keputusan. SPK menyediakan informasi khusus untuk pendukung keputusan, menganalisis masalah dan melihat peluang. Sedangkan SI tidak memiliki basis model dan pengetahuan, hanya sifatnya memberikan informasi kepada *user* tanpa melibatkan proses basis

pengetahuan sebelum ditampilkan kepada *user*. SI menyediakan dan menghasilkan informasi yang bersifat rutin dan terprogram. Informasi ini diperoleh dari hasil ekstraksi dan manipulasi data.

SP dan SPK sama-sama menggunakan metode sebagai prosesnya. Perbedaannya, SPK merefleksikan gaya kemampuan manajer, sebaliknya SP memberikan peluang untuk mendapatkan kemampuan dalam membuat keputusan melebihi kemampuan yang dimiliki manajer. SPK merupakan suatu sistem yang interaktif, yang membantu pengambil keputusan melalui penggunaan data dan model-model keputusan untuk memecahkan masalah yang sifatnya semi terstruktur maupun yang tidak terstruktur. Semi terstruktur yaitu jika ada sebagian dari masalahnya mempunyai jawaban yang jelas. Sedangkan tidak terstruktur yaitu jika keputusan yang memerlukan proses pengambilan keputusannya harus memberikan penilaian, evaluasi, dan pengertian untuk memecahkan masalahnya.

SP mempunyai kemampuan untuk menjelaskan jalur penalaran yang diikuti pencapaian pemecahan tertentu, penjelasan mengenai bagaimana pemecahan dicapai akan lebih berguna daripada pemecahan itu sendiri. SP mirip dengan SPK dalam menyediakan dukungan pemecahan masalah tingkat tinggi untuk pemakai. SP memberikan *output* berupa informasi yang dapat meng*update* pengetahuannya untuk menambahkan keakuratan data atau informasi (pasif, karena sistem hanya mampu berpikir sesuai dengan pengetahuan yang didapatkan dari pakar, yang digunakan untuk menggantikan keberadaan seorang pakar tetapi tidak untuk menggantikan pengetahuan pakar) yang telah dihasilkan. Pengetahuan yang didapat dari pakar digunakan sebagai dasar oleh sistem pakar untuk melakukan konsultasi. SPK lebih menggunakan *database*, sedangkan SP menggunakan *knowledge base.*

2.3.2 Karakteristik Sistem Pendukung Keputusan

Karakteristik dan kapabilitas dari Sistem Pendukung Keputusan adalah sebagai berikut (E. Turban, 2005):

a. Dukungan untuk para pengambil keputusan, terutama pada situasi semi terstruktur dan tak terstruktur.
b. Dukungan untuk seluruh level manajerial, dari ekskutif puncak sampai manajer lini.
c. Dukungan untuk masing-masing individu dan kelompok.
d. Dukungan untuk seluruh keputusan independen dan atau sekuensial.
e. Dukungan di seluruh fase proses pengambilan keputusan: inteligensi, desain, pilihan, dan implementasi.
f. Dukungan pada berbagai proses dan gaya pengambilan keputusan.
g. Kemampuan sistem beradaptasi dengan cepat dimana pengambil keputusan dapat menghadapi masalah-masalah baru dan pada saat yang sama dapat menanganinya dengan cara mengadaptasikan sistem terhadap kondisi-kondisi perubahan yang terjadi.

h. Pengguna merasa nyaman. *User-friendly*, kapabilitas grafis yang kuat, dan sebuah bahasa interaktif yang alami.
i. Peningkatan terhadap keefektifan pengambilan keputusan (akurasi, *timelines,* kualitas) dari pada efisiensi (biaya).
j. Pengambil keputusan mengontrol penuh semua langkah proses pengambilan keputusan dalam memecahkan masalah.
k. Pengguna akhir dapat mengembangkan dan memodifikasi sistem sederhana.
l. Menggunakan model-model dalam penganalisisan situasi pengambilan keputusan.
m. Disediakannya akses untuk berbagai sumber data, format, dan tipe, mulai dari sistem informasi geografi (GIS) sampai sistem berorientasi objek.

Dapat dilakukan sebagai alat *stand alone* yang digunakan oleh seorang pengambil keputusan pada satu lokasi atau didistribusikan di satu organisasi keseluruhan dan di beberapa organisasi sepanjang rantai persediaan.

2.3.3 Komponen-Komponen Sistem Pendukung Keputusan

Ada empat (4) komponen Sistem Pendukung Keputusan, diantaranya (E. Turban, 2005) :

1. *Data Component* (Manajemen Data)

Database Management System adalah sistem yang menyediakan akses data serta semua program kontrol yang diperlukan untuk mendapatkan data tersebut sesuai dengan yang dibutuhkan oleh program sebagai bahan analisis dan pertimbangan. Data mencakup fakta tentang operasi internal, tren, kecerdasan dan atau riset pasar, dan informasi yang tersedia secara umum. DBMS harus cukup canggih sebagai sarana untuk membantu *user* dalam memberikan hak akses untuk mengambil data ketika mereka tidak tahu dimana data berada secara fisik. DBMS memfasilitasi penggabungan data dari sumber yang berbeda. Selain itu, DBMS dapat menggabungkan data tanpa adanya instruksi dari pengguna dalam menyelesaikan tugas.

2. *Model Management* (Manajemen Model)

Model Base Management System (MBMS) memungkinkan untuk melacak semua model dalam SPK yang mungkin dijalankan selama analisis serta kontrol untuk mejalankan model. MBMS dapat terdiri dari sintak yang diperlukan untuk menjalankan pekerjaan, format dimana data harus dimasukkan sebelum menjalankan model, dan format data setelah menjalankan pekerjaan. MBMS juga menjadi penghubung terhadap model-model lain, sehingga dimungkinkan *output* dari satu model dapat menjadi *input* bagi model-model lainnya. MBMS dapat menganalisis sensitivitas dari model-model yang sedang berjalan untuk membantu memberikan asumsi model pertanyaan pengguna apakah sudah sesuai dalam menentukan pengambilan keputusan.

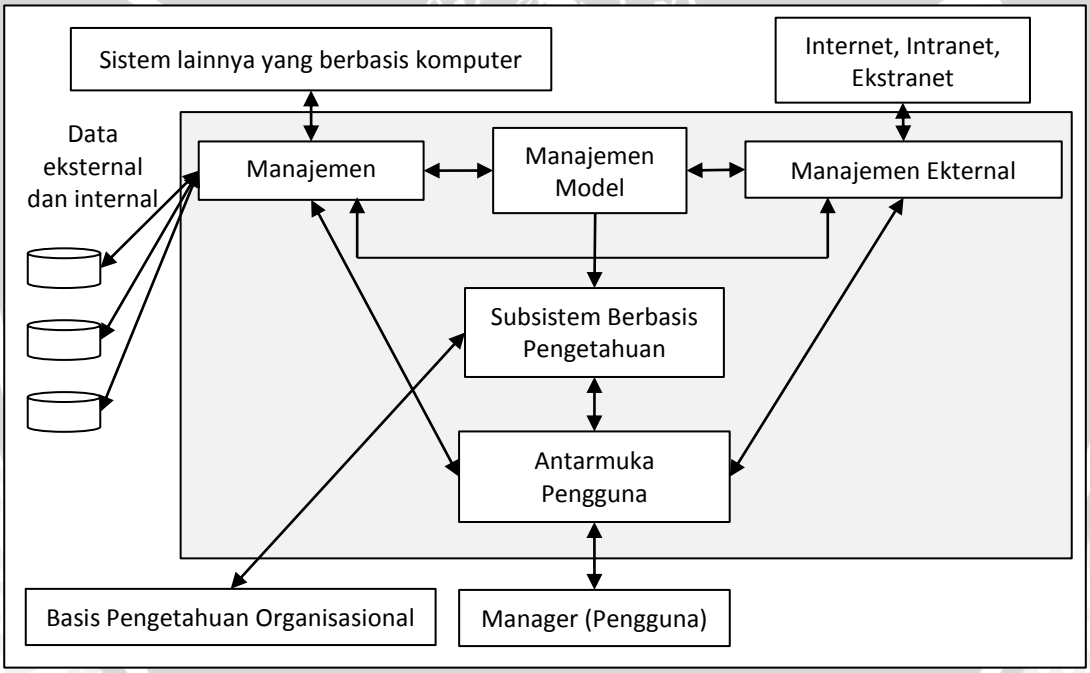
3. ***User Interface* (Antarmuka Pengguna)**

User Interface atau antarmuka pengguna merupakan semua mekanisme informasi dimana informasi merupakan hasil *output* dan *input* dari sistem. *User Interface* menampilkan semua layar *input* data dan model yang diminta oleh pengguna juga menampilkan semua layar hasil *output* yang diminta *user*. Banyak *user* beranggapan bahwa antarmuka *user* adalah SPK yang nyata, karena itu adalah bagian dari sistem yang mereka lihat.

4. **Subsistem Manajemen Berbasis Pengetahuan**

Subsistem tersebut mendukung semua subsistem lain atau bertindak langsung sebagai suatu komponen independen dan bersifat opsional. Selain memberikan inteligensi untuk memperbesar pengetahuan pengambil keputusan, subsistem tersebut bisa diinterkoneksikan dengan repositori pengetahuan (bagian dari sistem manajemen pengetahuan), yang kadang-kadang disebut basis pengetahuan organisasional.

Adapun penjelasan skematik dari penjabaran SPK dapat dilihat pada Gambar 2.5 berikut.



Gambar 2.5 Arsitektur Sistem Pendukung Keputusan

Sumber : (E. Turban, 2005)

2.3.4 Keuntungan Sistem Pendukung Keputusan

Beberapa keuntungan penggunaan SPK antara lain adalah sebagai berikut (E. Turban, 2005):

1. Mampu mendukung pencarian solusi dari berbagai permasalahan yang kompleks.

2. Dapat merespon dengan cepat pada situasi yang tidak diharapkan dalam konsisi yang berubah-ubah.
3. Mampu untuk menerapkan berbagai strategi yang berbeda pada konfigurasi berbeda secara cepat dan tepat.
4. Pandangan dan pembelajaran baru.
5. Sebagai fasilitator dalam komunikasi.
6. Meningkatkan kontrol manajemen dan kinerja.
7. Menghemat biaya dan sumber daya manusia (SDM).
8. Menghemat waktu karena keputusan dapat diambil dengan cepat.
9. Meningkatkan efektivitas manajerial, menjadikan manajer dapat bekerja lebih singkat dan dengan sedikit usaha.
10. Meningkatkan produktivitas analisis

2.3.5 Kemampuan dan Keterbatasan Sistem Pendukung Keputusan

Menurut Turban (E. Turban, 2005), ada beberapa kemampuan yang harus dimiliki oleh sebuah sistem pendukung keputusan, di antaranya adalah sebagai berikut :

1. Menunjang pembuatan keputusan manajemen dalam menangani masalah semi terstruktur dan tidak terstruktur.
2. Membantu manajer pada berbagai tingkatan manajemen, mulai dari manajemen tingkat atas sampai manajemen tingkat bawah.
3. Menunjang pembuatan keputusan secara kelompok dan perorangan.
4. Menunjang pembuatan keputusan yang saling bergantungan dan berurutan.
5. Menunjang tahap-tahap pembuatan keputusan antara lain *intelligence*, *design*, *choice*, dan *implementation*.
6. Menunjang berbagai bentuk proses pembuatan keputusan dan jenis keputusan.
7. Kemampuan untuk melakukan adaptasi setiap saat dan bersifat fleksibel.
8. Kemudahan melakukan interaksi sistem.
9. Meningkatkan efektivitas dalam pembuatan keputusan daripada efisiensi.
10. Mudah dikembangkan oleh pemakai akhir.
11. Kemampuan pemodelan dan analisis dalam pembuatan keputusan.
12. Kemudahan melakukan pengaksesan berbagai sumber dan format data.

Disamping berbagai kemampuan dan karakteristik seperti dikemukakan di atas, sistem pendukung keputusan memiliki juga keterbatasan, antara lain (E. Turban, 2005) :

1. Ada beberapa kemampuan manajemen dan bakat manusia yang tidak dapat dimodelkan, sehingga model yang ada dalam sistem tidak semuanya mencerminkan persoalan yang sebenarnya.
2. Kemampuan suatu sistem pendukung keputusan terbatas pada pengetahuan dasar serta model dasar yang dimilikinya.

3. Proses-proses yang dapat dilakukan oleh sistem pendukung keputusan biasanya tergantung juga pada kemampuan perangkat lunak yang digunakannya.
4. Sistem pendukung keputusan tidak memiliki intuisi seperti yang dimiliki oleh manusia. Karena sistem pendukung keputusan hanya suatu kumpulan perangkat keras, perangkat lunak dan sistem operasi yang tidak dilengkapi oleh kemampuan berpikir.

Secara implisit, sistem pendukung keputusan berlandaskan pada kemampuan dari sebuah sistem berbasis komputer dan dapat melayani penyelesaian masalah.

2.3.6 Langkah-langkah Pemodelan Dalam Sistem Pendukung Keputusan

Saat melakukan pemodelan dalam pembangunan DSS dilakukan langkah-langkah (E. Turban, 2005), yaitu sebagai berikut:

a. Studi Kelayakan (*Intelegence*)
Pada langkah ini, sasaran ditentukan dan dilakukan pencarian prosedur, pengumpulan data, identifikasi masalah, identifikasi kepemilikan masalah, klasifikasi maslah, hingga akhirnya terbentuk sebuah pernyataan masalah.
b. Perancangan (*Design*)
Pada tahapan ini akan di formulasikan model yang akan digunakan dan kriteria-kriteria yang ditentukan. Setelah itu, di cari alternatif model yang bisa menyelesaikan permasalahan tersebut. Langkah selanjutnya adalah memprediksi keluaran yang mungkin. Kemudian di tentukan variabel-variabel model.
c. Pemilihan (*Choice*)
Setelah pada tahap perancangan di tentukan berbagai alternatif model beserta variabel-variabelnya. Pada tahapan ini akan dilakukan pemilihan modelnya, termasuk solusi dari model tersebut. Selanjutnya, dilakukan analisis sensitivitas, yakni dengan mengganti variabel.
d. Membuat DSS
Setelah menentukan model, berikutnya adalah mengimplementasikannya dalam aplikasi DSS.

2.4 Stroke

Stroke adalah suatu keadaan yang timbul karena berhentinya suplai darah ke bagian otak yang mengakibatkan kematian jaringan otak sehingga seseorang dapat menderita kelumpuhan bahkan kematian (Batticaca, 2012). *Stroke* merupakan *deficit neurologis* (gangguan saraf) lokal dan atau global yang mempunyai serangan mendadak yang progresif cepat dan berlangsung 24 jam atau lebih sebagai akibat dari *Cardiovascular Diasease* (CVD) (Ariani, 2012). *Stroke* atau gangguan peredaran darah otak (GPDO) merupakan kelainan fungsi otak yang timbul mendadak dan sering dijumpai pada siapa saja dan kapan saja (Muttaqin, 2008). *Stroke* harus ditangani secara cepat dan tepat. *Stroke* merupakan penyakit yang paling sering menyebabkan cacat berupa kelumpuhan

anggota gerak, seperti gangguan bicara, proses berpikir daya ingat, dan bentuk-bentuk kecacatan yang lain sebagai akibat gangguan fungsi otak (Muttaqin, 2008).

2.4.1 Penyebab *Stroke*

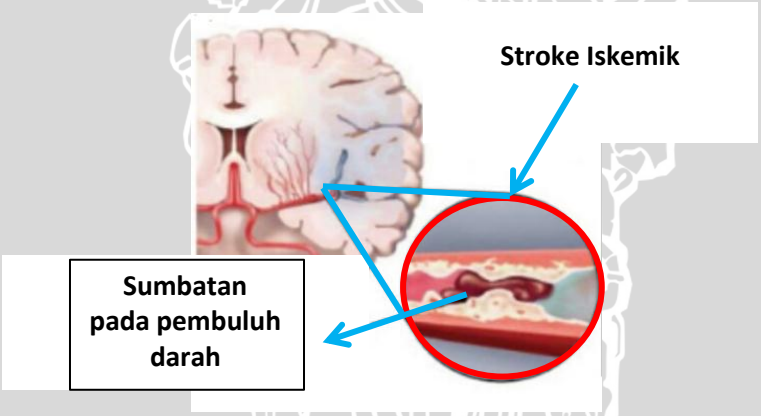
Beberapa faktor penyebab *Stroke* antara lain (Muttaqin, 2008):

a. Trombosis Serebral

Trombosis ini terjadi pada pembuluh darah yang mengalami oklusi sehingga menyebabkan iskemi jaringan otak yang dapat menimbulkan *oedema* dan *kongesti* di sekitarnya. Trombosis biasanya terjadi pada orang tua yang sedang tidur atau bangun tidur. Hal ini dapat terjadi karena penurunan aktivitas simpatis dan penurunan tekanan darah yang dapat menyebabkan iskemi serebral. Tanda dan gejala neuroogis sering kali memburuk pada 48 jam setelah thrombosis (Muttaqin, 2008). Seperti pada Gambar 2.6 dibawah ini.

Beberapa keadaan di bawah ini dapat menyebabkan trombosis otak :

- Aterosklerosis
- Hiperkoagulasi pada polisitemia
- Arteritis (radang pada arteri)
- Emboli

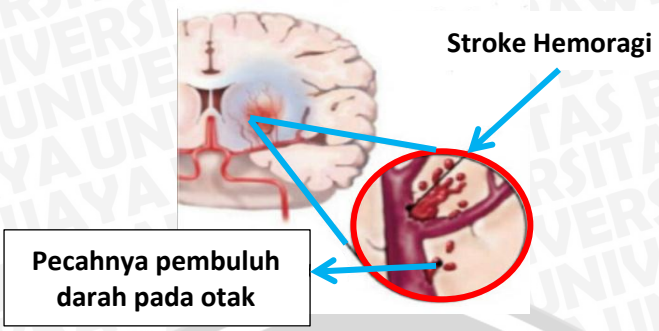


Gambar 2.6 *Stroke* Iskemik

Sumber : (Irfan, 2012)

b. Hemoragi

Pendarahan intrakranial atau *intrascrebral* termasuk pendarahan dalam ruang subaraknoid atau ke dalam jaringan otak sendiri. Pendarahan ini dapat terjadi karena aterosklerosis dan hipertensi (Muttaqin, 2008). Akibat pecahnya pembuluh darah otak menyebabkan perembesan darah ke dalam parenkim otak yang dapat mengakibatkan penekanan, pergeseran dan pemisahan jaringan otak yang berdekatan, sehingga otak akan membengkak, jaringan otak tertekan, sehingga terjadi infark otak, edema, dan mungkin herniasi otak, sesuai Gambar 2.7.



Gambar 2.7 *Stroke* Hemoragi
Sumber : (Irfan, 2012)

c. Hipoksia Umum
Beberapa penyebab yang berhubungan dengan hipoksia umum adalah (Muttaqin, 2008):
- Hipertensi yang parah
- Henti jantung-paru
- Curah jantung turun akibat aritmia

d. Hipoksia Setempat
Beberapa penyebab yang berhubungan dengan hipoksia setempat adalah (Muttaqin, 2008):
- Spasme arteri serebral, yang disertai pendarahan *subaraknoid*
- *Vasokontriksi* arteri otak disertai sakit kepala migrain.

2.4.2 Faktor Risiko *Stroke*

Beberapa faktor risiko penyebab *stroke* antara lain (Association, 2009):

1. Tekanan Darah
Tekanan darah tinggi penyebab utama seseorang terkena *stroke*. Tekanan darah memiliki kemampuan untuk mendorong darah menuju dinding arteri. Tekanan darah tinggi memicu jantung memompa lebih keras untuk menyuplai darah ke seluruh tubuh yang dapat melemahkan pembuluh darah dan merusak organ-organ utama, seperti otak. Orang yang memiliki tekanan darah tinggi memiliki satu setengah kali risiko mengalami *stroke* dibandingkan dengan mereka yang memiliki tekanan darah yang normal dari 120/80. Hal penting yang dapat dilakukan adalah mengontrolnya. Hal ini dapat dilakukan melalui kebiasaan makan yang sehat, aktivitas fisik, atau obat-obatan.
2. Denyut Jantung
Afib (*Atrial Fibrillation*) adalah jenis denyut jantung tidak teratur. Meskipun dapat terjadi pada semua usia, hal ini lebih sering terjadi pada orang 65 tahun dan lebih tua. Afib lebih sering terjadi pada orang dengan tekanan darah tinggi, penyakit jantung, atau diabetes. Afib menimbulkan risiko *stroke* karena darah yang menuju hati cenderung membentuk gumpalan yang kemudian dapat dibawa ke otak, menyebabkan *stroke*. Jangka panjang yang tidak diobati Afib juga dapat melemahkan jantung,

yang menyebabkan gagal jantung. Tujuan untuk mengobati Afib adalah untuk mengembalikan ritme jantung secara teratur, yang dapat dilakukan dengan pemberian obat atau melalui penggunaan stimulasi listrik.

3. Merokok

Merokok dapat meningkatkan risiko *stroke* bila dibandingkan dengan bukan perokok. Merokok meningkatkan pembentukan gumpalan, pengentalan darah, dan meningkatkan jumlah penumpukan plak di arteri.

4. Kolesterol

Kolesterol adalah zat lemak dalam darah. Hal ini dapat diproduksi dengan baik oleh tubuh atau ditemukan dalam makanan. Kolesterol yang tinggi dalam arteri dapat memblokir aliran normal ke otak dan menyebabkan *stroke*. Dengan kolesterol tinggi risiko penyakit jantung dan *atherosclerosis* juga meningkat. Kadar kolesterol dapat dikontrol melalui kebiasaan makan yang sehat, aktivitas fisik, atau obat-obatan.

5. Diabetes

Pada penderita diabetes, tubuh tidak memproduksi insulin yang cukup (Diabetes Tipe I) atau sel mengabaikan insulin (Diabetes Tipe II). Tanpa insulin, tubuh tidak dapat memproses gula, yang merupakan bahan bakar dasar bagi sel-sel dalam tubuh. Orang dengan diabetes lebih mungkin untuk mengalami *stroke* hingga empat kali daripada orang yang tidak, terutama karena orang-orang dengan diabetes sering memiliki faktor risiko *Stroke* lainnya, seperti tekanan darah tinggi, fibrilasi atrium, dan kolesterol tinggi. Berat badan, olahraga, perubahan kebiasaan makan, obat-obatan oral, dan insulin tembakan adalah cara-cara untuk mengendalikan diabetes.

6. Aktifitas Fisik

Aktivitas fisik meliputi apa saja yang membuat tubuh menjadi bergerak. Sebuah penelitian baru menunjukkan bahwa orang yang berolahraga lima kali atau lebih per minggu memiliki risiko *stroke* berkurang.

7. Diet

Diet sehat dapat membantu dalam mengurangi risiko penyakit kronis, meningkatkan kesehatan secara keseluruhan, dan membantu untuk mencapai atau mempertahankan berat badan yang sehat. Makan sehat termasuk membuat keputusan tentang pilihan makanan dan menyeimbangkan kalori. Diet dapat ditentukan dengan mengetahui IMT (Indeks Massa Tubuh). Pada rumus IMT, status berat badan dihitung dengan membandingkan berat badan/BB (kg) dengan kuadrat tinggi badan/TB (m). Persamaan IMT dapat ditunjukkan melalui Persamaan 2.1 (Kesehatan.co, 2015):

$$IMT = \frac{BB}{(TB \; x \; TB)} \qquad \textbf{(2.1)}$$

8. Riwayat Keturunan

Risiko *stroke* dapat meningkat jika anggota keluarga (orang tua, kakek-nenek, atau saudara) telah mengalami *stroke* atau serangan jantung pada usia dini.

2.5 Ketidakpastian

Sistem kecerdasan buatan yang dikembangkan memiliki pengetahuan yang lengkap terhadap suatu permasalahan, sehingga sistem tersebut dapat dengan mudah memberikan solusi dengan menggunakan pendekatan logika. Apabila sistem yang dikembangkan tidak dapat mengakses seluruh fakta terhadap suatu permasalahan, maka sistem harus bekerja dalam ketidakpastian dan kesamaran. Berdasarkan permasalahan tersebut sistem harus menggunakan beberapa teknik khusus yang dapat menangani ketidakpastian dan kesamaran dalam menyelesaikan suatu permasalahan. Berikut ini akan dijelaskan mengenai teknik yang dapat digunakan untuk menangani ketidakpastian dan kesamaran pengetahuan (Pratama, 2014):

1. Teknik Probabilitas
   Teknik ini memanfaatkan Teorema Bayes yang menunjukkan hubungan sebab akibat yang terjadi diantara *evidence-evidence* yang ada. Teori *Naive Bayes* merupakan salah satu pendekatan yang digunakan oleh teknik probabilitas.
2. Faktor Kepastian, teknik penalaran tertua yang digunakan pada sistem MYCIN. Teknik faktor kepastian bersifat semi probabilitas karena notasi probabilitas tidak sepenuhnya digunakan.
3. Logika *Fuzzy*, teknik penalaran baru yang diperkenalkan oleh Zadeh. Dalam setiap variabel dalam logika *fuzzy* memiliki rentang nilai tertentu yang akan digunakan untuk menghitung nilai fungsi keanggotaannya.

2.6 *Naive Bayes*

Bayes merupakan suatu teknik prediksi berbasis probabilistik sederhana yang berdasar pada penerapan teorema bayes. Fitur independen merupakan model yang digunakan oleh metode Bayes. Bayes memiliki ketidaktergantungan (independensi) yang kuat (Naif). Independensi yang kuat dalam Bayes (terutama *Naive Bayes*) adalah ketidakterkaitannya suatu fitur dengan fitur lain dalam suatu data. Formula Bayes dinyatakan dalam Persamaan 2.2 berikut (Prasetyo, 2012):

$$P(H \mid E) = \frac{P(E|H) \,.\, P(H)}{P(E)} \qquad \textbf{(2.2)}$$

Dimana :

P(H|E) = Probabilitas posterior bersyarat (*conditional probability*) suatu kejadian *H* terjadi jika diberikan *evidence*/bukti *E* terjadi

P(H) = Probabilitas awal (*prior*) kejadian *H* terjadi tanpa memandang *evidence* apapun

P(E) = Probabilitas awal (*prior*) *evidence E* terjadi tanpa memandang kejadian/*evidence* yang lain

P(E|H) = Probabilitas sebuah *evidence E* terjadi akan mempengaruhi hipotesis *H*

Dasar dari aturan Bayes yaitu hasil dari hipotesis atau kejadian (*H*) dapat diperkirakan berdasarkan pada beberapa *evidence* (*E*) yang diamati. Berikut merupakan beberapa hal penting dari aturan Bayes (Prasetyo, 2012) :

1. Nilai dari sebuah probabillitas awal/priori *H* atau *P(H)* adalah probabilitas dari suatu hipotesis sebelum bukti diamati;
2. Nilai dari sebuah probabilitas akhir *H* atau *P(H|E)* adalah probabilitas dari suatu hipotesis setelah bukti.

Hipotesis dalam teorema Bayes merupakan label kelas yang menjadi target dalam sebuah klasifikasi, sedangkan bukti atau *evidence* merupakan fitur yang menjadi masukan dalam klasifikasi. *Naive Bayes* dilambangkan dengan *P(X|Y)*, *X* merupakan masukan yang berupa fitur-fitur dan *Y* merupakan kelas dalam sebuah klasifikasi. Notasi *P(X|Y)* berarti probabilitas label kelas *Y* didapatkan setelah fitur-fitur *X* diamati, notasi ini merupakan probabilitas akhir (*Posterior Probability*) untuk *Y* dan *P(Y)* merupakan notasi dari probablitas awal (*Prior Probability*). Berikut merupakan Persamaan 2.3, formulasi *Naive Bayes* untuk klasifikasi (Prasetyo, 2012) :

$$P(Y \mid X) = \frac{P(Y)\Pi_{i=1}^{q} . P(X_i \mid Y)}{P(X)} \quad \textbf{(2.3)}$$

Dimana :

P(Y|X)	=	Probabilitas akhir (*Posterior Probability*), *probability* label kelas *Y* didapatkan setelah fitur-fitur *X* diamati
P(Y)	=	Probabilitas awal (*Prior Probability*) kelas *Y* tanpa memandang fitur apapun
$\Pi_{i=1}^{q} . P(X_i \mid Y)$	=	Probabilitas independen kelas *Y* dari semua fitur *X*
P(X)	=	Probabilitas awal (*Prior Probability*) *evidence X* tanpa memandang kejadian/*evidence* yang lain
Set fitur *X*	=	$\{X_1, X_2, X_3, \ldots, X_q\}$
q	=	Atribut (*q* dimensi)

Nilai *P(X)* selalu tetap sehingga dalam perhitungan prediksi nantinya hanya tinggal menghitung bagian $P(Y)\Pi_{i=1}^{q} . P(X_i \mid Y)$ kemudian mencari nilai *P(Y|X)* terbesar sebagai kelas yang dipilih sebagai hasil prediksi. Probabilitas independen $\Pi_{i=1}^{q} . P(X_i \mid Y)$ merupakan pengaruh semua fitur dari data terhadap setiap kelas *Y* (Prasetyo, 2012).

Secara umum, untuk kasus dengan menggunakan metode *Naive Bayes* yang mudah dihitung untuk fitur kategoris seperti kalsifikasi hewan dengan fitur "melahirkan" dengan nilai {bertelur, beranak}. Namun, jika fitur yang akan dihitung bersifat numerik (kontinu), maka akan dilakukan perhitungan khusus. Caranya adalah :

1. Melakukan diskretisasi pada setiap atribut di fitur kontinu yang akan diestimasi.

2. Mengasumsikan setiap fitur sesuai dengan data pelatihan menggunakan fungsi distribusi Gaussian. Distribusi Gaussian dikarakteristikan ke dalam dua parameter yaitu *mean* (μ) dan varian (σ^2).

Adapun fungsi densitas Gauss dapat dilihat pada Persamaan 2.4 berikut.

$$P=N(X_i = x_i \mid Y = y_i) = \frac{1}{\sqrt{2\pi}\sigma_{ij}} exp^{-\frac{(x_i-\mu_{ij})^2}{2\sigma_{ij}^2}} \quad \textbf{(2.4)}$$

Parameter μ_{ij} bisa didapat dari *mean* pada sampel X_i ($\bar{x}$) dari semua data latih yang menjadi milik kelas y_i, sedangkan σ_{ij}^2 dapat diperkirakan dari varian sampel (s^2) dari data latih.

Adapun fungsi untuk mencari nilai *Mean* dapat dilihat pada Persamaan 2.5 berikut.

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n} \quad \textbf{(2.5)}$$

Perhitungan *mean* dilakukan dengan menjumlahkan seluruh nilai data suatu kelompok sampel, kemudian dibagi dengan jumlah sampel tersebut. Dimana $\bar{x}$ merupakan rata-rata hitung, x_i merupakan nilai sampel ke-*i*, dan *n* adalah jumlah sampel.

Adapun fungsi untuk mencari nilai standar deviasi dapat dilihat pada Persamaan 2.6 berikut.

$$s = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}} \quad \textbf{(2.6)}$$

Perhitungan standar deviasi adalah dengan mengurangi setiap nilai data dengan rata-rata kelompok data tersebut, selanjutnya semua hasilnya dijumlahkan. Dimana *s* merupakan standar deviasi (simpangan baku), x_i merupakan nilai x ke *i*, $\bar{x}$ merupakan rata-rata, *n* adalah ukuran sampel.

Contoh Kasus 1 :

Jika diketahui terjadi gejala pada tanaman tebu berupa batang mengering dan batang berlubang (Pratama, 2014).

1. Melakukan perhitungan nilai probabilitas prior pada setiap jenis kelas penyakit atau hama tanaman tebu *P(H)*
   Contoh :
   P(Penggerek Pucuk) = 4/41 = 0,09756
   P(Penggerek Batang) = 4/41 = 0,09756
   P(Boktor) = 3/41 = 0,07317
   dst…
2. Melakukan perhitungan nilai probabilitas prior likelihood sebuah fakta pada suatu kelas hama atau penyakit tanaman tebu *P(E|H)*.
   Contoh :
   Jika diketahui *input* gejala *User :*

Batang mengering (G17) dan batang berlubang (G20).
P(G17 | Penggerek Pucuk) = 0/4 = 0
P(G20 | Penggerek Pucuk) = 0/4 = 0
dst…

3. Melakukan perhitungan nilai probabilitas posterior masing-masing kelas hama atau penyakit tanaman tebu.
   Contoh :
   $P(H|E) = P(H)$ x $P(E_1, E_2|H)$
   P(Penggerek Pucuk | *E*) = P(Penggerek Pucuk) x P(G17 | Penggerek Pucuk) x P(G20 | Penggerek Pucuk)
   P(Penggerek Pucuk | *E*) = 0,09756 x 0 x 0 = 0
   dst…
4. Nilai probabilitas posterior *class* terbesar merupakan hasil keputusan diagnosa sistem.

Contoh Kasus 2:

Misalkan seperti pada penelitian yang dilakukan oeh Diana Laily Fithri, dkk (Darmanto, 2014), untuk memprediksi kelulusan mahasiswa dengan 8 parameter.

- Hasil Prediksi
  a. Perhitungan tepat waktu :
     1) P (mahasiswa tepat waktu/total mahasisa data training) * P (Laki-Laki | Tepat waktu) * P (Dalam kota | Tepat waktu) * P (18-20 | Tepat waktu) * P (Mahasiswa | Tepat waktu) * P (Belum menikah | Tepat waktu) * P (2.80 | Tepat waktu) * P (116 | Tepat waktu) * P (Aktif | Tepat waktu)
     2) Jadi, 0,43 * 0,35 * 0,44 * 0,13 * 0,49 * 0,44 * 0,37 * 0,51 * 0,49 = 0,00017
  b. Perhitungan terlambat :
     1) P (mahasiswa terlambat dalam data training/total mahasisa data training) * P (Laki-Laki | Terlambat) * P (Dalam kota | Terlambat) * P (18-20 | Terlambat) * P (Mahasiswa | Terlambat) * P (Belum menikah | Terlambat) * P (2.80 | Terlambat) * P (116 | Terlambat) * P (Aktif | Terlambat)
     2) Jadi, 0,57 * 0,60 * 0,52 * 0,87 * 0,14 * 0,15 * 0,11 * 0,47 * 0,14 = 0,0003
  c. Prediksi tepat waktu = 0,00017
     Prediksi terlambat = 0,0003
  d. Penentuan prediksi 0.00017 > 0,0003

  Jadi, prediksi mahasiswa tersebut Tepat Waktu.

Alasan penulis menggunakan metode *Naive Bayes* pada sistem pendukung keputusan deteksi dini penyakit *stroke*, adalah :

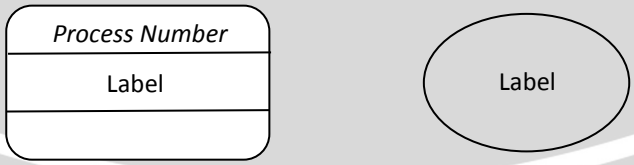
1. Fitur-fitur pada metode *Naive Bayes* dapat bekerja secara independen (Prasetyo, 2012).
2. Metode *Naive Bayes* merupakan metode yang klasifikasi yang sangat efektif (mendapatkan hasil yang tepat) dan efisien (penalaran yang cepat) (Kusumadewi, Mei 2009).

2.7 *Data Flow* Diagram (DFD)

DFD adalah suatu metode yang digunakan dalam pemodelan sistem yang berguna untuk menggambarkan sistem secara terstruktur. Model ini menggambarkan sistem sebagai jaringan kerja antar fungsi yang berhubungan satu sama lain dengan aliran dan penyimpanan data. Proses aliran data dapat dijelaskan dengan menggunakan simbol maupun notasi tertentu. DFD juga dapat digunakan untuk menjelaskan proses aliran data yang terperinci atau proses dekomposisi, namun DFD tidak dapat menyajikan urutan operasi. DFD memiliki beberapa karakteristik seperti mendukung tahap analisis dan kebutuhan desain sistem, sebuah teknik diagram dengan penjelasan, menggambarkan jaringan kegiatan/proses target sistem, memungkinkan untuk perilaku paralel dan *asynchronous*, perbaikan bertahap melalui dekomposisi hirarki proses. DFD memiliki beberapa elemen, yaitu (Parno, 2010):

1. Aktivitas/proses

   Aktivitas atau proses merupakan transformasi data yang memiliki *data flow diagram* sebagai *input* dan *output.* Sebuah aktivitas dapat didekomposisi untuk proses yang lebih terperinci. Label dari aktivitas/proses harus berupa *verb*/kata kerja. Aktivitas terhubung dengan spesifikasi proses. Dalam memodelkan sebuah aktivitas/proses terdapat beberapa aturan, seperti aktivitas/proses selalu di dalam sistem, perlakuan eksternal entitas atau sistem terhadap data tidak akan dimodelkan. Proses harus memiliki minimal satu *input* dan *output* dan harus memiliki *input* yang cukup untuk menciptakan *output*. Simbol untuk elemen aktivitas/proses ditunjukkan pada Gambar 2.9 (Parno, 2010).

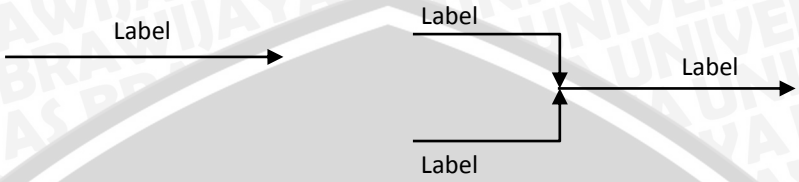


Gambar 2.8 Simbol Aktivitas/Proses
Sumber : (Parno, 2010)

2. *Data Flow*

   Data flow menunjukkan arus pergerakan suatu informasi dalam sebuah sistem. *Data flow* merupakan elemen konektor yang menghubungkan dua ujung aktivitas/proses, *data store*, entitas eksternal, dan sebagainya.

Panah yang terdapat dalam simbol *data flow* menunjukkan arah pergerakan suatu data. Dalam memodelkan sebuah *data flow* terdapat beberapa aturan seperti *data flow* bergerak dari entitas eksternal ke sistem atau sebaliknya dan mengalir dari simbol internal untuk simbol internal yang lain, namun selalu dimulai atau berakhir pada suatu proses. Simbol *data flow* ditunjukkan pada Gambar 2.10 (Parno, 2010).



Gambar 2.9 Simbol *Data Flow*
Sumber : (Parno, 2010)

3. *Data Store*

Data store atau penyimpanan data adalah sebuah elemen DFD untuk menyimpan data secara permanen dan menyajikan sebuah *placeholder* untuk *database. Data store* bersifat pasif dan hanya dilayani oleh suatu proses. Beberapa aturan dalam memodelkan *data store* seperti *data store* tidak dapat mengubah status data dengan sendirinya, *data store* harus dimasukkan ke dalam sistem untuk proses transformasi (tambah, hapus, dan *update* data). Setiap menyimpan data pada DFD harus sesuai entitas pada ERD. Simbol untuk *data store* ditunjukkan pada Gambar 2.11 (Parno, 2010).

Gambar 2.10 Simbol *Data Store*
Sumber : (Parno, 2010)

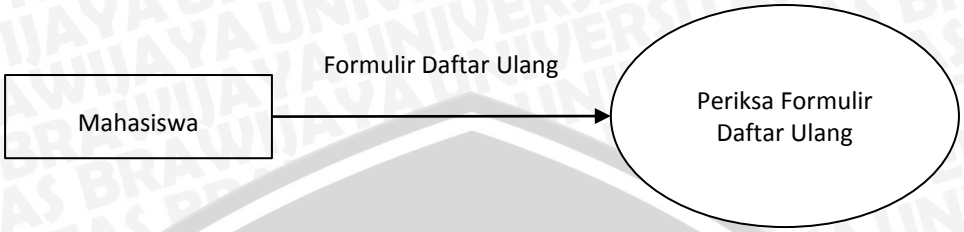
4. Entitas

Entitas merupakan asal dan tujuan dari aliran data eksternal yang menyediakan koneksi ke konteks sistem. Entitas bersifat pasif dimana hanya mengirimkan atau menerima data. Dalam memodelkan entitas terdapat beberapa aturan seperti, sistem dan *data store* yang berdiri di luar sistem namun berinteraksi dengan sistem. Entitas menerima informasi dari sistem, memicu sistem untuk suatu pergerakan data, dan memberikan informasi baru ke sistem. Simbol untuk entitas ditunjukkan pada Gambar 2.12 (Parno, 2010).

Gambar 2.11 Simbol Entitas

Sumber : (Parno, 2010)

Konsep data dari suatu sumber yang sama menuju ke tujuan yang sama dan mempunyai hubungan ditunjukkan pada Gambar 2.13 (Parno, 2010).



Gambar 2.12 Konsep Data

Sumber : (Parno, 2010)

Gambar 2.13 menjelaskan bahwa Mahasiswa adalah proses, panah dari Formulir Daftar Ulang adalah konektor, sedangkan Periksa Formulir daftar ulang adalah label.

2.8 *Entity Relationship Diagram* (ERD)

ERD adalah model konseptual yang mendeskripsikan hubungan antar penyimpanan (dalam DFD). Karena itu, ERD berbeda dengan DFD (DFD memodelkan fungsi sistem). ERD digunakan untuk memodelkan struktur data dan hubungan antar data, karena hal ini relatif kompleks. Dengan ERD kita dapat menguji model dengan mengabaikan proses yang harus dilakukan (Parno, 2010). *Entity* dapat berhubungan satu sama-lain. Hubungan ini dinamakan *relationship* (relasi). Dalam ERD hubungan ini dapat terdiri dari sejumlah *entity* yang disebut sebagai derajat hubungan. Tetapi pada umumnya hampir semua model hanya menggunakan hubungan dengan derajat dua (*binary-relationship*) (Parno, 2010).

2.8.1 Notasi

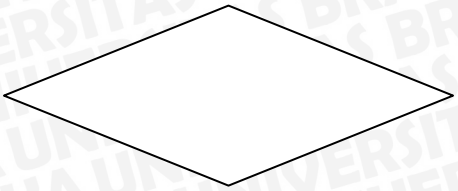
Sebelum membuat ERD terdapat beberapa notasi yang harus dipahami. Terdapat tiga notasi dasar dalam ERD yaitu (Parno, 2010).

1. Entitas adalah sebuah benda atau objek di dunia nyata yang dapat dibedakan dengan benda atau objek yang lain. Entitas dilambangkan dengan bentuk persegi panjang seperti pada Gambar 2.14 (Parno, 2010).

Gambar 2.13 Notasi Entitas

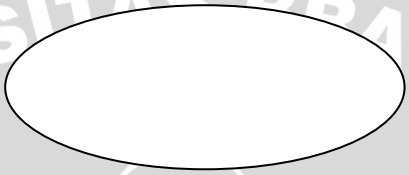
Sumber : (Parno, 2010)

2. *Relationship* adalah hubungan antara entitas. *Relationship* dilambangkan dengan bentuk *diamond* seperti pada Gambar 2.15 (Parno, 2010).



Gambar 2.14 Notasi *Relationship*
Sumber : (Parno, 2010)

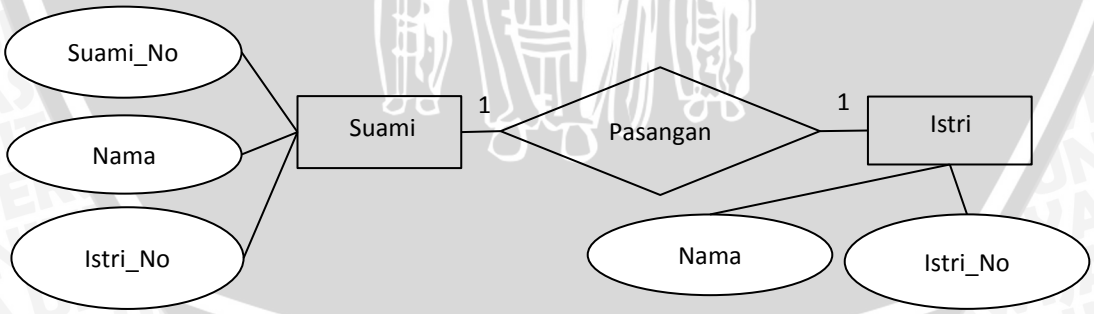
3. Atribut adalah sebutan untuk mewakili suatu entitas. Lambang atribut adalah bentuk *ellips* seperti pada Gambar 2.16 (Parno, 2010).



Gambar 2.15 Notasi Entitas
Sumber : (Parno, 2010)

Pada suatu hubungan dan tidak jadi masalah berapapun derajat hubungannya, antar *entity* selalu ada 3 jenis hubungan biner yaitu (Parno, 2010):

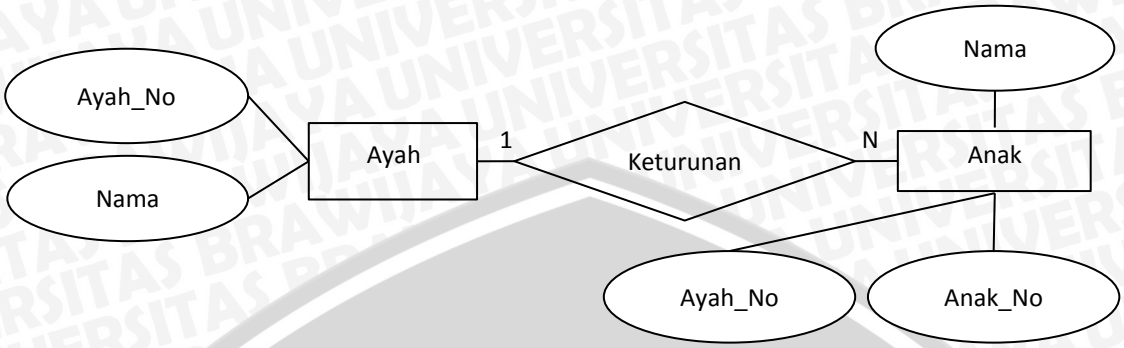
1. Hubungan satu ke satu (1:1), yaitu jika dalam suatu hubungan antara data pertama dan data kedua adalah satu berbanding satu. Sebagai contoh pada Gambar 2.17 dijelaskan bahwa seorang Suami memiliki seorang Istri dan tidak ada Istri tertentu yang memiliki Suami lebih dari satu. Suami adalah entitas dengan Suami_No sebagai *primary key* dan Nama sebagai atribut. Istri juga sebuah entitas dengan Istri_No sebagai *primary key* dan Nama sebagai atribut (T. Sutojo, 2011).



Gambar 2.16 Contoh Hubungan Satu ke Satu
Sumber : (Parno, 2010)

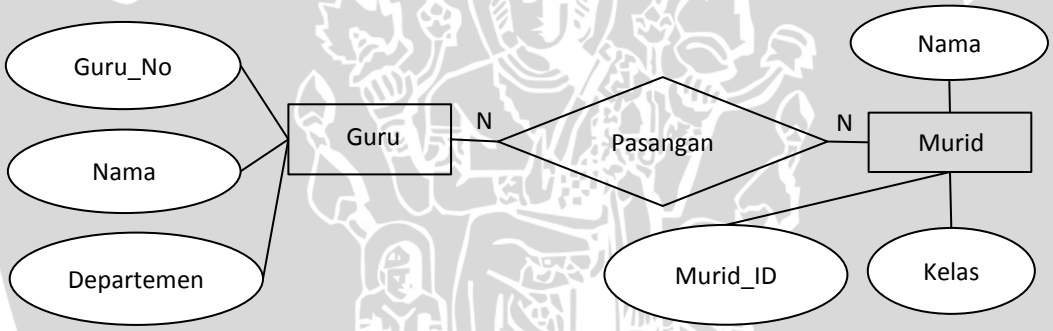
2. Hubungan satu ke banyak atau banyak ke satu (1:M atau M:1), yaitu hubungan antara data pertama dan data kedua adalah satu berbanding banyak atau dapat pula dibalik menjadi banyak lawan satu. Hubungan ini dijelaskan pada Gamber 2.18 untuk suatu hubungan keturunan dimana satu Anak hanya dapat memiliki satu Ayah dan satu Ayah dapat memiliki Anak

lebih dari satu. Ayah merupakan entitas dan Ayah_No adalah atribut sedangkan Anak merupakan entitas dan Anak_No adalah atribut.



Gambar 2.17 Contoh Hubungan Satu ke Banyak
Sumber : (Parno, 2010)

4. Hubungan banyak ke banyak (N:N), yaitu hubungan antara data pertama dan data kedua adalah banyak berbanding banyak. Hubungan ini dijelaskan pada Gambar 2.19. Pada Gambar 2.19 menggambarkan suatu hubungan antara Guru dan Murid. Satu Guru dapat mengajar beberapa Murid dan beberapa Murid dapat diajar oleh beberapa Guru (Parno, 2010).



Gambar 2.18 Contoh Hubungan Banyak ke Banyak
Sumber : (T. Sutojo, 2011)

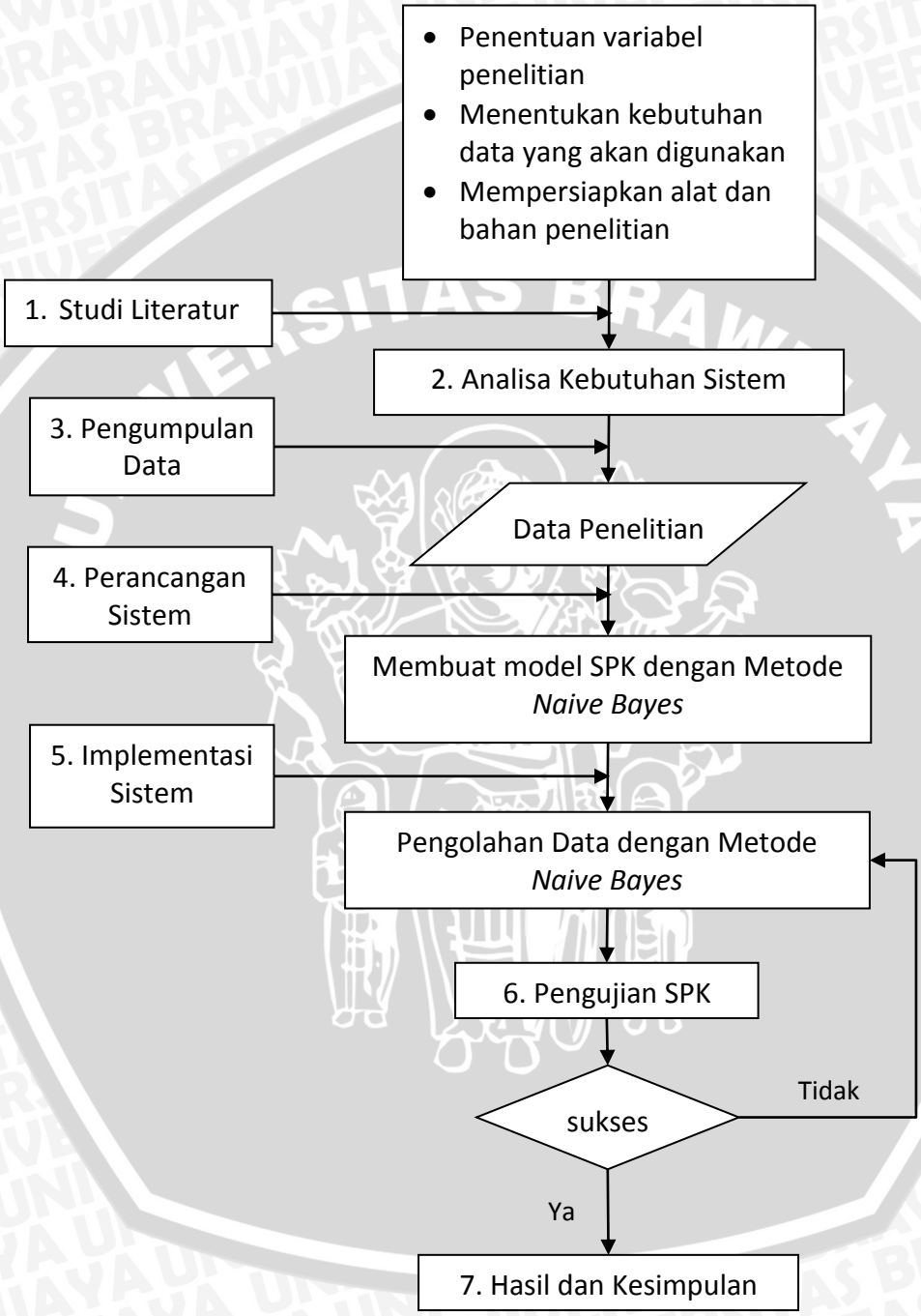
2.9 Akurasi

Akurasi memiliki pengertian seberapa dekat hasil pengukuran terhadap angka sebenarnya (*true value* atau *reference value*). Dalam penelitian ini akurasi dihitung dari jumlah hasil yang tepat dibagi dengan jumlah data. Perhitungan akurasi dapat menggunakan rumus seperti pada persamaan 2.16 (Pratama, 2014).

$$Akurasi = \frac{\sum data\ uji\ benar}{\sum jumlah\ total\ data\ uji}\ x\ 100\% \qquad \textbf{(2.7)}$$

BAB 3 METODOLOGI PENELITIAN

Pada metode penelitian akan dilakukan penelitian dengan langkah-langkah pada Gambar 3.1 sebagai berikut :



Gambar 3.1 Diagram Alir untuk Penentuan Metode Penelitian

Sumber : [Perancangan]

Langkah-langkah pada Gambar 3.1, dalam merancang sistem pendukung keputusan dengan urutan sebagai berikut :

3.1 Studi Literatur

Studi literatur menjelaskan dasar teori yang digunakan untuk menunjang penulisan skripsi. Setelah melakukan objek penelitian yang akan dikerjakan langkah selanjutnya adalah mencari dan memahami studi pustaka/literatur untuk memahami konsep-konsep rancangan aplikasi yang harus dipelajari agar dalam perancangan aplikasi tidak mengalami kendala yang berarti. Dalam penelitian ini literatur yang dicari adalah mengenai :

a. Penyakit *Stroke*;
b. Sistem Pendukung Keputusan;
c. Metode *Naive Bayes*;
d. *Data Flow Diagram* (DFD);
e. *Entity Relationship Diagram* (ERD);

3.2 Analisa Kebutuhan Sistem

Analisa kebutuhan bertujuan untuk menganalisis dan mendapatkan semua kebutuhan yang diperlukan dalam pembuatan sistem pendukung keputusan deteksi dini penyakit *Stroke* menggunakan metode *Naive Bayes*. Analisis kebutuhan disesuaikan dengan lokasi dan variabel penelitian, menentukan kebutuhan data yang akan digunakan, dan mempersiapkan alat dan bahan penelitian.

3.2.1 Kebutuhan Perangkat

- Kebutuhan *Hardware*, meliputi
  - Laptop dengan *Processsor* Intel® Core™ i3-2330M, CPU 2.20GHz
  - Memory 2 GB disarankan 4 GB DDR3
  - *Harddisk* dengan kapasitas 500 Gb
  - Monitor dengan ukuran 14”
- Kebutuhan *Software*, meliputi
  - Sistem operasi *Microsoft Windows* 7 *Ultimate* 32-bit
  - Notepad++ atau Adobe Dreamweaver CC
  - Adobe Photoshop
  - XAMPP sebagai *server* yang berdiri sendiri (*localhost*) yang terdiri yaitu *Apache HTTP Server*, *MySQL* database
  - MySQL sebagai server *Database Management System*

3.2.2 Kebutuhan Sistem

- Data yang dibutuhkan, meliputi :
  - Literatur/data Penyakit *Stroke*
  - Data faktor penyebab penyakit *Stroke* untuk menentukan hasil deteksi dini penyakit *Stroke*, meliputi tekanan darah, diabetes, riwayat keluarga, merokok, kolesterol, aktivitas fisik, diet, riwayat fibrilasi atrium.

- Data hasil uji *suspect Stroke* yang diperoleh data penelitian dengan metode *Naive Bayes*.

3.3 Pengumpulan Data

Pada tahap ini dilakukan pengumpulan data dan pembelajaran data-data yang berhubungan dengan sistem pendukung keputusan deteksi dini penyakit *stroke* menggunakan metode *Naive Bayes*. Pengumpulan data dilakukan dengan mencari beberapa referensi melalui media internet dan buku untuk mendapatkan data sampel sebagai bahan acuan untuk pengembangan perangkat lunak. Data sampel yang dimaksud adalah data faktor risiko *stroke* sebagai atribut beserta kriteria yang dijadikan sebagai acuan dalam pengambilan keputusan.

3.4 Perancangan Sistem

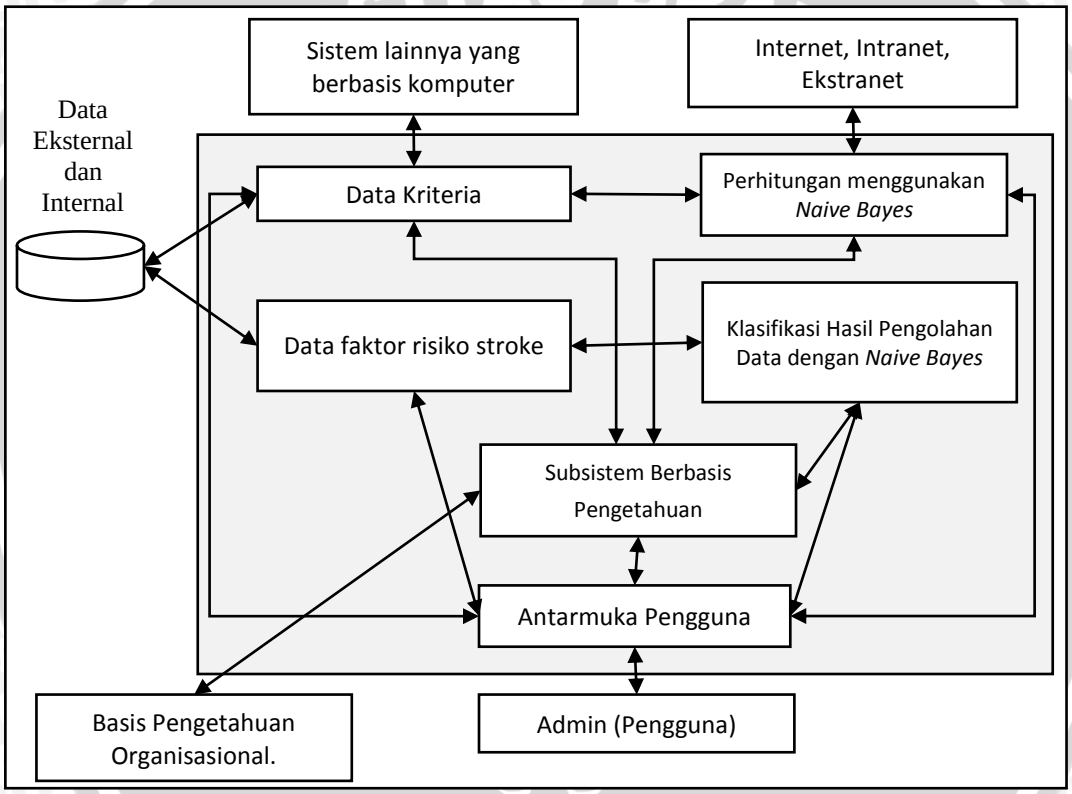
Perancangan sistem dibangun berdasarkan hasil pengambilan data dan analisis kebutuhan yang telah dilakukan. Perancangan sistem menjelaskan desain dari model *Naive Bayes* sebagai metode untuk perancangan sistem pendukung keputusan deteksi dini penyakit *stroke* bagi pasien.

a. Perancangan diagram blok SPK dan DFD
   Data *input*an yang digunakan adalah nilai kritera serta data deteksi dini penyakit *stroke*. Sedangkan hasil keluaran atau *output* adalah data hasil perhitungan deteksi dini penyakit *stroke* menggunakan metode *Naive Bayes.* DFD digunakan sebagai perangkat yang memodelkan sistem untuk mempermudah memahami sistem secara jelas dan terstruktur.
b. Perancangan subsistem manajemen data
   Subsistem manajemen data mengatur penyimpanan data dalam *database*.
c. Perancangan subsistem manajemen berbasis pengetahuan
   Subsistem manajemen berbasis pengetahuan berisi pengetahuan terkait penentuan faktor risiko *stroke* untuk mendukung kebutuhan subsistem manajemen lainnya.
d. Perancangan subsistem manajemen model
   Pada subsistem manajemen model menggunakan *Naive Bayes* sebagai model untuk melakukan klasifikasi risiko *stroke*. Subsistem manajemen model menjelaskan penggunaan metode perhitungan *Naive Bayes* sebagai penentuan *suspect stroke* untuk mendukung pengujian SPK.
e. Perancangan subsistem antarmuka pengguna
   Perancangan subsistem memudahkan pengguna untuk menggunakan sistem yang dibangun.
f. Perancangan algoritma
   Perancangan algoritma *Naive Bayes* meliputi proses pengolahan data probabilitas, data dari basis pengetahuan, proses perhitungan *Naive Bayes*, dan data laporan.

3.4.1 Arsitektur Implementasi Sistem Pendukung Keputusan

Implementasi sistem pendukung keputusan deteksi dini penyakit *stroke* terbagi menjadi bagian yang saling terintegrasi dan membentuk arsitektur implementasi SPK ini. Arsitektur dari sistem pendukung keputusan deteksi dini penyakit *stroke* berisi tentang data pengetahuan (berisi teori-teori pendukung untuk bisa menjelaskan tentang informasi yang dibutuhkan), subsistem manajemen model (berisi pemodelan untuk perhitungan algoritma sistem yang dijelaskan dengan *flowchart* dan *pseudocode*), dan subsistem manajemen data (berisi penjelasan struktur atau cara kerja sistem yang dijelaskan dengan DFD, ERD, dan PDM).

Arsitektur blok diagram perancangan aplikasi sistem pendukung keputusan deteksi dini penyakit *stroke* dapat ditunjukan dalam Gambar 3.2.



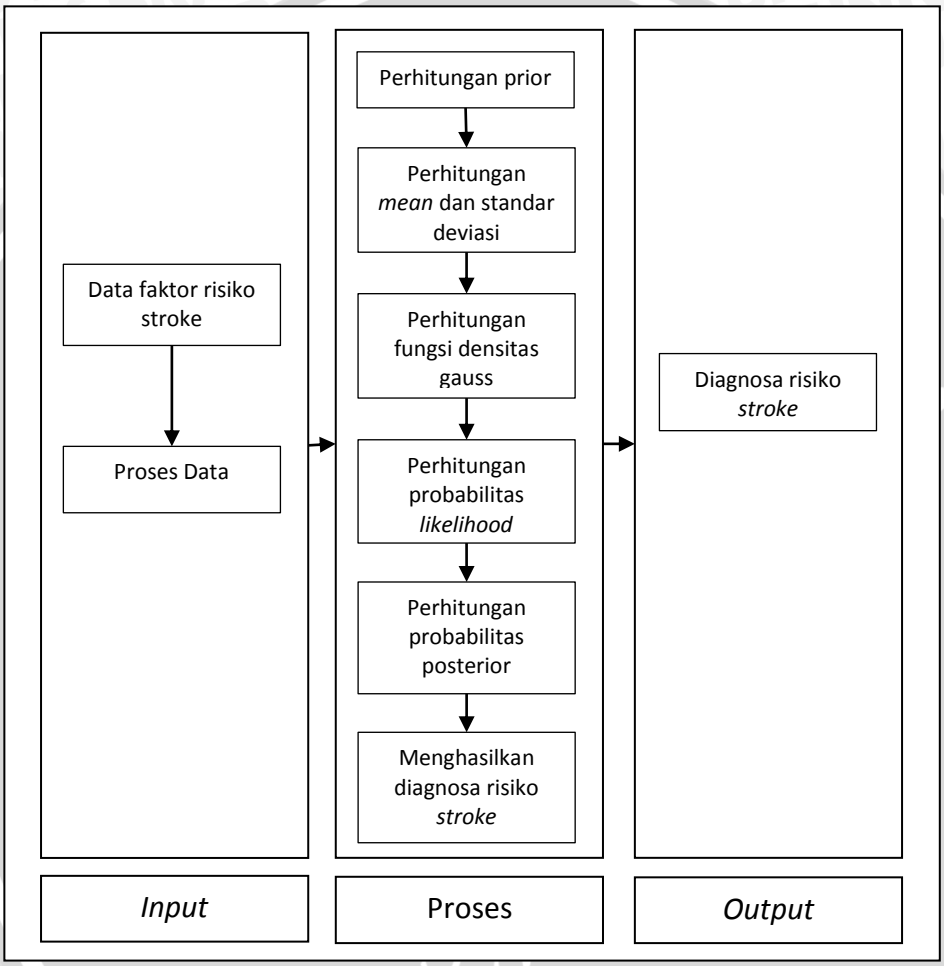
Gambar 3.2 Arsitektur Implementasi SPK Deteksi Dini Penyakit *Stroke*
Sumber : [Perancangan]

Subsistem *data component* pada SPK deteksi dini penyakit *Stroke* yaitu data-data faktor risiko *stroke* yang diperoleh dari penelitian sebelumnya yang dilakukan oleh Mukhlis [3]. Pada SPK *database eksternal* dan *internal* mewakili subsistem manajemen data, subsistem antarmuka pengguna yaitu *interface* yang disediakan untuk pengguna. Subsistem *model management* yaitu *Naive Bayes*. Subsistem berbasis pengetahuan menjelaskan dalam membentuk alternatif dari basis pengetahuan organisasional yang sesuai kriteria dan data faktor risiko *stroke*. Internet, intranet, dan eksternet merupakan model eksternal SPK. *User*

merupakan pengguna SPK. Digunakan garis penghubung panah bolak-balik untuk menggambarkan adanya proses keluar-masuk data.

3.4.2 Blok Diagram SPK

Perancangan blok diagram SPK menjelaskan tentang diagram alir dari sistem yang akan dibangun. Secara umum sistem ini terdiri dari proses *input*, proses *Naive Bayes*, dan proses *output*. Perancangan sistem dapat dilihat lebih jelas pada arsitektur perancangan blok diagramnya pada Gambar 3.3.



Gambar 3.3 Arsitektur Blok Diagram Perancangan Aplikasi
Sumber : [Perancangan]

Pada Gambar 3.3 terdiri dari beberapa blok diagram proses, yaitu:

a. *Input*
   Input pada sistem ini yaitu berupa parameter faktor risiko *stroke* yang terdiri dari tekanan darah, diabetes, riwayat keluarga, merokok, kolesterol, aktivitas fisik, diet, dan riwayat fibrilasi atrium.
b. Proses
   Proses perhitungan pada penelitian ini menggunakan metode *Naive Bayes* untuk menentukan keputusan deteksi dini penyakit *stroke* berdasarkan data

training. Adapun langkah-langkah untuk melakukan perhitungan dengan menggunakan metode *Naive Bayes* :

1. Menghitung nilai probabilitas prior
2. Menghitung *mean* dan standar deviasi
3. Menghitung fungsi densitas gauss
4. Menghitung probabilitas *likelihood*
5. Menghitung probabilitas posterior
6. Pengambilan Keputusan

c. *Output*

Keputusan sistem pendukung keputusan deteksi dini penyakit *stroke* akan diambil berdasarkan hasil perhitungan probabilitas posterior terbesar. *Output* dari sistem ini adalah diagnosa risiko *stroke* dengan kategori risiko tinggi, risiko sedang, dan risiko rendah.

3.5 Implementasi Sistem

Sistem dibuat sesuai desain yang telah dibuat. Proses pengimplementasian dalam pembuatan aplikasi ini menggunakan bahasa pemrograman web dengan menggunakan MySQL sebagai *database* manajemen sistem, serta *tools* pendukung lainnya. Tahapan-tahapan yang ada dalam implementasi antara lain :

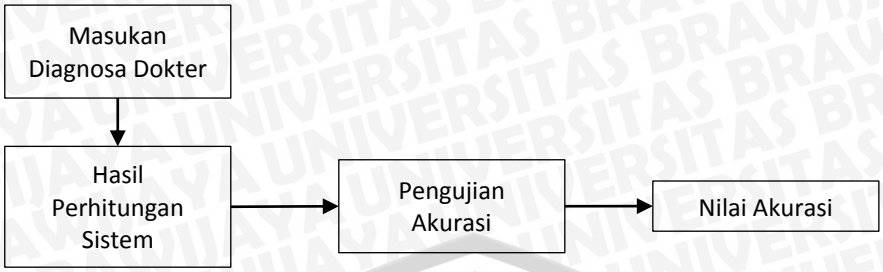
1. Implementasi antarmuka SPK
2. Implementasi basis data dengan menggunakan DBMS MySQL pada server localhost (XAMPP) yang bertujuan untuk memudahkan melakukan manipulasi dan penyimpanan data.
3. Implementasi algoritma menggunakan metode *Naive Bayes* ke dalam bahasa pemrograman PHP.
4. *Output* berupa diagnosa *suspect* penyakit *stroke*.

3.6 Pengujian SPK

Pengujian ini dapat dilakukan untuk tampilan dari sistem perangkat lunak. Tujuan *test case* ini menunjukkan fungsi perangkat lunak ketika dijalankan. Selain itu, *testing* juga berguna untuk mencari kesalahan-kesalahan sistem pendukung keputusan deteksi dini penyakit *stroke* menggunakan metode *Naive Bayes.* Pengujian sistem dilakukan melalui dua cara, meliputi :

1. Pengujian Validasi, dilakukan dengan menggunakan tabel yang menjelaskan hubungan kesesuaian antara fungsi hasil kerja sistem dengan daftar kebutuhan sistem
2. Pengujian Akurasi, dilakukan dengan membandingkan data *output* sistem perhitungan manual dengan data *output* perhitungan sistem.

Pengujian akurasi dilakukan dengan membandingkan hasil diagnosa sistem dengan hasil diagnosa dokter. Skenario pengujian akurasi ditunjukkan pada Gambar 3.4 berikut.



Gambar 3.4 Diagram Blok Pengujian Akurasi Sistem
Sumber : [Perancangan]

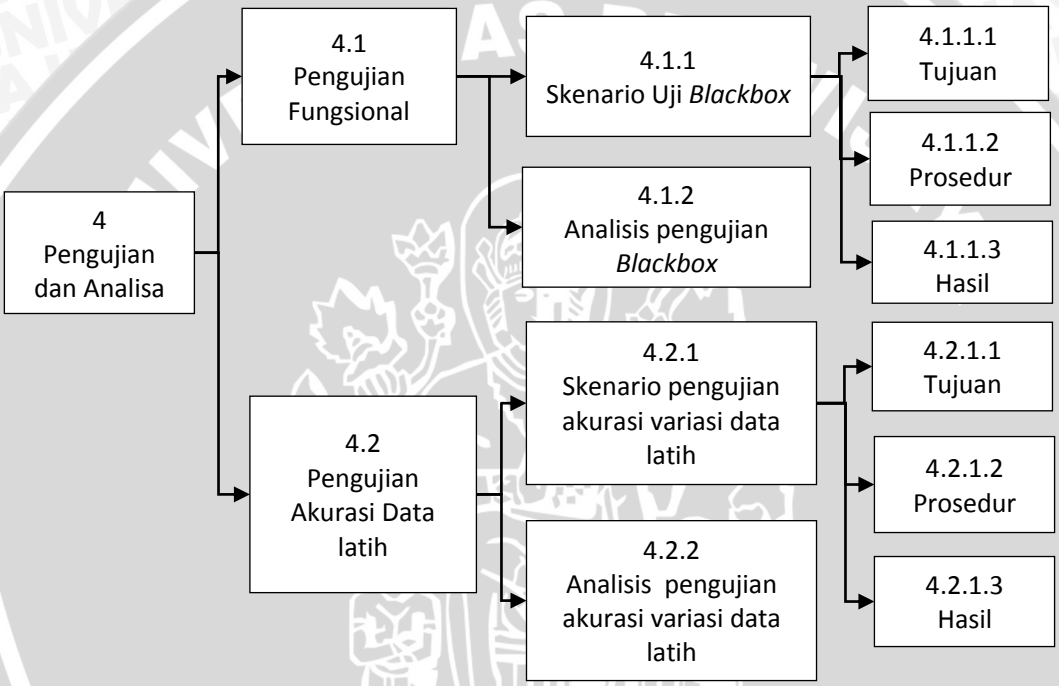
Pada pengujian akurasi ini akan disediakan beberapa kasus uji yang akan didiagnosa oleh seorang dokter dan sistem pendukung keputusan deteksi dini penyakit *Stroke* menggunakan metode *Naive Bayes*. Hasil diagnosa sistem akan dicocokkan dengan hasil diagnosa dokter untuk mendapatkan nilai akurasi sistem. Rumus untuk mencari nilai akurasi menggunakan Persamaan 2-16. Pengujian akurasi dilakukan terhadap variasi data dengan cara mengubah data *training*. Pengujian variasi data terbagi menjadi lima skenario yaitu jumlah data *training* 80%:20%, 70%:30%, 60%:40%, 50%:50%, dan 40%:60%.

3.7 Hasil dan Kesimpulan

Pengambilan kesimpulan dilakukan setelah semua tahapan perancangan sistem aplikasi telah selesai dilakukan. Kesimpulan diambil dari hasil pengolahan data terhadap sistem yang dibangun. Tahap terakhir dari penulisan adalah saran yang dimaksudkan untuk memperbaiki kesalahan-kesalahan yang terjadi dan menyempurnakan penulisan serta untuk memberikan pertimbangan atas pengembangan aplikasi selanjutnya.

BAB 4 PENGUJIAN SISTEM

Pada bab ini membahas tentang proses pengujian dan analisa sistem pendukung keputusan deteksi dini penyakit *stroke* menggunakan metode *Naive Bayes*. Proses pengujian pada sistem ini dilakukan dengan 2 tahap pengujian yaitu pengujian fungsionalitas dan pengujian akurasi. Pengujian fungsionalitas digunakan untuk menguji sistem, apakah sistem yang dibangun telah sesuai dengan daftar kebutuhan sistem. Pengujian akurasi digunakan untuk mengukur tingkat akurasi dari hasil keputusan sistem dengan hasil keputusan pakar. Analisa hasil pengujian dilakukan untuk menganalisa hasil pengujian yang telah dilakukan. Pohon perancangan pengujian dan analisa ditunjukkan pada Gambar 4.1.



Gambar 4.1 Pohon Pengujian dan Analisis
Sumber : [Pengujian]

4.1 Pengujian Fungsionalitas

Pada pengujian fungsionalitas ini akan dijelaskan tentang skenario pengujian pertama yaitu pengujian fungsionalitas sistem berdasarkan daftar kebutuhan sistem menggunakan *blackbox testing*. Pengujian *blackbox* merupakan pengujian yang dilakukan terhadap sistem untuk mengetahui apakah sistem yang dibangun telah sesuai dengan daftar kebutuhan sistem yang telah ditentukan sebelumnya. Daftar kebutuhan yang digunakan dalam proses pengujian fungsionalitas ditunjukkan pada daftar kebutuhan. Pada tabel tersebut terdapat kebutuhan 23 yang harus terdapat pada sistem yang dibangun, 23 daftar kebutuhan yang telah diimplementasikan pada sistem tersebut nantinya akan diuji dengan pengujian fungsionalitas untuk mengetahui tingkat kesesuaian antara kinerja sistem dengan daftar kerja sistem yang telah disebutkan. Pengujian *blackbox* tidak

menekankan pada jalannya algoritma sistem melainkan untuk menemukan kesesuaian antara hasil kinerja dari sistem yang telah dibangun dengan daftar kebutuhan *user*.

4.1.1 Skenario Pengujian *Blackbox*

Pada pengujian *blackbox* ini akan dilakukan pengamatan tentang tujuan, prosedur, dan hasil akhir terhadap cara kerja sistem pendukung keputusan deteksi dini penyakit *stroke* menggunakan metode *Naive Bayes* secara *eksternal*. Berikut ini akan dijelaskan mengenai tujuan, prosedur, dan hasil analisis dari pengujian *blackbox*.

4.1.1.1 Tujuan

Tujuan dari pengujian *blackbox* adalah untuk mengetahui apakah sistem dapat berjalan sesuai dengan rancangan dan kebutuhan pengguna dan mengetahui apakah fitur-fitur yang terdapat pada sistem sudah dapat digunakan dengan benar dan tidak terdapat *error*.

4.1.1.2 Prosedur

Prosedur pada pengujian *blackbox* ini dilakukan dengan cara membuat kasus uji pada setiap antarmuka dan pada fitur-fitur yang ada dan telah ditentukan pada daftar kebutuhan. Daftar kasus uji terdiri dari nama kasus uji yang akan dilakukan, tujuan pengujian, prosedur pengujian, dan hasil yang diharapkan. Berikut ini merupakan kasus uji dalam pengujian *blackbox*.

1. Kasus Uji *Login*

Pengujian *blackbox* pada kasus uji *login* menjelaskan tentang pengujian dari proses *login*, seperti yang ditunjukkan pada Tabel 4.1.

Tabel 4.1 Penjelasan Kasus Uji *Login*

Nama Kasus Uji	Kasus Uji *Login*
Tujuan Pengujian	Memastikan bahwa sistem dapat menerima data *login* berupa *email* dan *password*. Dengan pengujian akan diketahui bahwa sistem dapat memenuhi kebutuhan fungsional untuk kebutuhan masuk ke dalam sistem dengan tujuan agar *user* dapat masuk atau menggunakan SPK deteksi dini penyakit *stroke* sesuai dengan hak akses.
Prosedur Pengujian	1. *User* masuk ke halaman utama 2. *User* memilih menu *login* 3. *User* mengisi data *login* 4. *User* menekan tombol *login*
Hasil yang Diharapkan	1. Sistem dapat masuk ke halaman utama sesuai hak akses jika *login* sukses 2. Sistem akan menolak jika data *login* yang diinputkan tidak sesuai dengan *database*.

Sumber : [Pengujian]

Skenario pengujian *blackbox* pada kasus uji *login* ditunjukkan pada Tabel 4.2 berikut.

Tabel 4.2 Pengujian *Blackbox* Kasus Uji *Login*

No.	Skenario Pengujian	*Test Case*	Hasil yang Diharapkan	Hasil Pengujian	Status
1.	Memasukkan data *login* yang salah	*Username*: (salah) Password: (salah)	Sistem akan menolak dan memberikan peringatan	Sistem akan menolak dan memberikan peringatan	Valid
2.	Memasukkan data *login* lengkap	*Username*: (benar) Password: (benar)	Sistem akan menerima dan menampilkan halaman awal sesuai hak akses	Sistem akan menerima dan menampilkan halaman awal sesuai hak akses	Valid
3.	Mengosongkan semua *field*, atau salah satu *field* lalu menekan tombol *login*	*Username*: - Password: -	Sistem akan menolak akses *login* dan memberikan peringatan	Sistem akan menolak akses *login* dan memberikan peringatan	Valid

Sumber : [Pengujian]

2. Kasus Uji Mengolah Data *User*

Pengujian *blackbox* pada kasus uji mengolah data *user* ditunjukkan pada Tabel 4.3.

Tabel 4.3 Penjelasan Kasus Uji Mengolah Data *User*

Nama Kasus Uji	Kasus Uji Mengolah *User*
Tujuan Pengujian	Memastikan bahwa sistem dapat menampilkan data *user* berupa *create user, update user,* dan *delete user.*
Prosedur Pengujian	1. admin melakukan *login* 2. admin memilih data master *user*
Hasil yang Diharapkan	Sistem dapat membuka masing-masing data *user* dalam *database*

Sumber : [Pengujian]

Skenario pengujian *blackbox* pada kasus uji mengolah data *user* ditunjukkan pada Tabel 4.4.

Tabel 4.4 Pengujian *Blackbox* Kasus Uji Mengolah Data *User*

No.	Skenario Pengujian	*Test Case*	Hasil yang Diharapkan	Hasil Pengujian	Status
1	Mengosongkan semua *field* atau mengisi salah satu *field* lalu menekan	Nama: - *Username*: - *Password*: - Alamat : - Hak akses: -	Sistem akan menolak dan memberikan peringatan	Sistem akan menolak dan memberikan peringatan	Valid

Tabel 4.4 Pengujian *Blackbox* Kasus Uji Mengolah Data *User* (lanjutan)

No.	Skenario Pengujian	*Test Case*	Hasil yang Diharapkan	Hasil Pengujian	Status
	tombol simpan				
2	Menampilkan halaman data *user* secara lengkap	Nama: - *Username*: - *Password*: - Alamat : - Hak akses: -	Sistem akan menampilkan halaman data *user* secara lengkap	Sistem akan menampilkan halaman data *user* secara lengkap	Valid

Sumber : [Pengujian]

3. Kasus Uji *Create* Data *User*

Pengujian *blackbox* pada kasus uji *create* data *user* ditunjukkan pada Tabel 4.5.

Tabel 4.5 Penjelasan Kasus Uji *Create* Data *User*

Nama Kasus Uji	Kasus Uji Create *User*
Tujuan Pengujian	Memastikan bahwa sistem dapat menambah data *user* berupa nama, *username*, password, alamat, dan hak akses.
Prosedur Pengujian	1. admin melakukan *login* 2. admin memilih data master *user* 3. admin mengisi data *user* 4. admin menekan tombol simpan
Hasil yang Diharapkan	Sistem dapat menyimpan data *user* ke dalam *database*

Sumber : [Pengujian]

Skenario pengujian *blackbox* pada kasus uji *create* data *user* ditunjukkan pada Tabel 4.6.

Tabel 4.6 Pengujian *Blackbox* Kasus Uji *Create* Data *User*

No.	Skenario Pengujian	*Test Case*	Hasil yang Diharapkan	Hasil Pengujian	Status
1	Mengosongkan semua *field* atau mengisi salah satu *field* lalu menekan tombol simpan	Nama: - *Username*: - *Password*: - Alamat : - Hak akses: -	Sistem akan menolak dan memberikan peringatan	Sistem akan menolak dan memberikan peringatan	Valid
2	Memasukkan data *user* secara lengkap	Nama: - *Username*: - *Password*: - Alamat : - Hak akses: -	Sistem akan menerima dan menyimpan di dalam *database*	Sistem akan menerima dan menyimpan dalam *database*	Valid

Sumber : [Pengujian]

4. Kasus Uji *Update* Data *User*

Pengujian *blackbox* pada kasus uji *update* data *user* ditunjukkan pada Tabel 4.7.

Tabel 4.7 Penjelasan Kasus Uji *Update* Data *User*

Nama Kasus Uji	Kasus Uji *Update* Data *User*
Tujuan Pengujian	Memastikan bahwa sistem dapat meng*update* data *user* dari *database*
Prosedur Pengujian	1. admin melakukan *login* 2. admin memilih data master *user* 3. admin menekan tombol edit 4. admin mengisi data *user* 5. admin menekan tombol *update*
Hasil yang Diharapkan	1. Sistem dapat menyimpan data *user* ke dalam database 2. Sistem dapat menampilkan perubahan data *user* ke dalam *database*

Sumber : [Pengujian]

Skenario pengujian *blackbox* pada kasus uji *update* data *user* ditunjukkan pada Tabel 4.8.

Tabel 4.8 Pengujian *Blackbox* Kasus Uji *Update* Data *User*

No.	Skenario Pengujian	*Test Case*	Hasil yang Diharapkan	Hasil Pengujian	Status
1	Melakukan *login* sebagai admin	Menekan tombol *update* data *user* yang akan di edit	Sistem akan menampilkan data *user* yang telah di edit	Sistem akan menampilkan data *user* yang telah di edit	Valid
2	Admin mengedit data *user*	Menekan tombol *update*	Sistem akan menyimpan di dalam *database*	Sistem akan menyimpan dalam *database*	Valid

Sumber : [Pengujian]

5. Kasus Uji *Delete* Data *User*

Pengujian *blackbox* pada kasus uji *delete* data *user* ditunjukkan pada Tabel 4.9.

Tabel 4.9 Penjelasan Kasus Uji *Delete* Data *User*

Nama Kasus Uji	Kasus Uji *Delete* Data *User*
Tujuan Pengujian	Memastikan bahwa sistem dapat menghapus data *user* dari *database*
Prosedur Pengujian	1. admin melakukan *login* 2. admin memilih data master *user* 3. admin menekan tombol hapus data *user* yang akan dihapus
Hasil yang Diharapkan	Sistem dapat menghapus data *user* ke dalam *database*

Sumber : [Pengujian]

Skenario pengujian *blackbox* pada kasus uji hapus data *user* ditunjukkan pada Tabel 4.10.

Tabel 4.10 Pengujian *Blackbox* Kasus Uji Hapus Data *User*

No.	Skenario Pengujian	*Test Case*	Hasil yang Diharapkan	Hasil Pengujian	Status
1	Melakukan *login* sebagai admin	Menekan tombol *delete* data *user*	Sistem akan menghapus data *user* dan terdapat peringatan untuk menghapus data atau batal, jika yakin menghapus data, pengguna menekan tombol OK	Sistem akan menghapus data *user* dari *database* jika yakin menghapus data, *user* menekan tombol OK	Valid

Sumber : [Pengujian]

6. Kasus Uji Menampilkan Detail *User*

Pengujian *blackbox* pada kasus uji menampilkan detail *user* ditunjukkan pada Tabel 4.11.

Tabel 4.11 Penjelasan Kasus Uji Menampilkan Detail *User*

Nama Kasus Uji	Kasus Uji Menampilkan Detail *User*
Tujuan Pengujian	Memastikan bahwa sistem dapat menampilkan detail *user* secara lengkap
Prosedur Pengujian	1. admin melakukan *login* 2. admin memilih data master *user*
Hasil yang Diharapkan	Sistem dapat membuka masing-masing data *user* dalam *database*

Sumber : [Pengujian]

Skenario pengujian *blackbox* pada kasus uji menampilkan detail *user* ditunjukkan pada Tabel 4.12.

Tabel 4.12 Pengujian *Blackbox* Kasus Uji Menampilkan Detail *User*

No.	Skenario Pengujian	*Test Case*	Hasil yang Diharapkan	Hasil Pengujian	Status
1	Mengosongkan semua *field* atau mengisi salah satu *field* lalu menekan tombol simpan	Nama: - *Username*: - *Password*: - Alamat : - Hak akses: -	Sistem akan menolak dan memberikan peringatan	Sistem akan menolak dan memberikan peringatan	Valid
2	Menampilkan halaman data *user* secara lengkap	Nama: - *Username*: - *Password*: - Alamat : - Hak akses: -	Sistem akan menampilkan halaman detail *user* secara lengkap	Sistem akan menampilkan halaman detail *user* secara lengkap	Valid

Sumber : [Pengujian]

7. Kasus Uji Kelola Data Faktor Risiko

Pengujian *blackbox* pada kasus uji kelola data faktor risiko ditunjukkan pada Tabel 4.13.

Tabel 4.13 Penjelasan Kasus Uji Mengolah Data Faktor Risiko

Nama Kasus Uji	Kasus Uji Mengolah Data Faktor Risiko
Tujuan Pengujian	Memastikan bahwa sistem dapat menampilkan data faktor risiko seperti melihat data faktor risiko, *update* data faktor risiko, dan *delete* data faktor risiko
Prosedur Pengujian	1. admin dan ds melakukan *login* 2. admin dan ds memilih data faktor risiko
Hasil yang Diharapkan	Sistem dapat membuka masing-masing data faktor risiko dalam *database*

Sumber : [Pengujian]

Skenario pengujian *blackbox* pada kasus uji mengolah data faktor risiko ditunjukkan pada Tabel 4.14.

Tabel 4.14 Pengujian *Blackbox* Kasus Uji Mengolah Data Faktor Risiko

No.	Skenario Pengujian	*Test Case*	Hasil yang Diharapkan	Hasil Pengujian	Status
1	Menampilkan halaman data faktor risiko secara lengkap	Tekanan Darah: - Diabetes: - Riwayat Keluarga: - Merokok: - Kolesterol:- Aktivitas Fisik: - Diet: - Riwayat Afib: -	Sistem akan menampilkan halaman data faktor risiko secara lengkap	Sistem akan menampilkan halaman data faktor risiko secara lengkap	Valid

Sumber : [Pengujian]

8. Kasus Uji *Edit* Data Faktor Risiko

Pengujian *blackbox* pada kasus uji *edit* data faktor risiko ditunjukkan pada Tabel 4.15.

Tabel 4.15 Penjelasan Kasus Uji *Edit* Data Faktor Risiko

Nama Kasus Uji	Kasus Uji *Edit* Data Faktor Risiko
Tujuan Pengujian	Memastikan bahwa sistem dapat meng*update* data faktor risiko dari *database*
Prosedur Pengujian	1. *User* melakukan *login* 2. *User* memilih data master faktor risiko 3. *User* menekan tombol edit 4. *User* mengisi data faktor risiko 5. *User* menekan tombol *update*
Hasil yang Diharapkan	1. Sistem dapat menyimpan data faktor risiko ke

Tabel 4.15 Penjelasan Kasus Uji *Edit* Data Faktor Risiko (lanjutan)

Hasil yang Diharapkan	dalam *database* 2. Sistem dapat menampilkan perubahan data faktor risiko ke dalam *database*

Sumber : [Pengujian]

Skenario pengujian *blackbox* pada kasus uji *update* data faktor risiko ditunjukkan pada Tabel 4.16.

Tabel 4.16 Pengujian *Blackbox* Kasus Uji *Update* Data Faktor Risiko

No.	Skenario Pengujian	*Test Case*	Hasil yang Diharapkan	Hasil Pengujian	Status
1	Melakukan *login* sebagai admin dan ds	Menekan tombol *update* data faktor risiko yang akan di edit	Sistem akan menampilkan data faktor risiko yang telah di edit	Sistem akan menampilkan data faktor risiko yang telah di edit	Valid
2	Admin dan ds mengedit data faktor risiko	Menekan tombol *update*	Sistem akan menyimpan di dalam *database*	Sistem akan menyimpan dalam *database*	Valid

Sumber : [Pengujian]

9. Kasus Uji *Delete* Data Faktor Risiko

Kasus uji *delete* data faktor risiko menjelaskan tentang pengujian fungsionalitas proses menghapus data faktor risiko pada sistem yang dapat dilakukan oleh *user*. Pengujian fungsionalitas ditunjukkan pada Tabel 4.17.

Tabel 4.17 Penjelasan Kasus Uji *Delete* Data Faktor Risiko

Nama Kasus Uji	Kasus Uji *Delete* Data Faktor Risiko
Tujuan Pengujian	Memastikan bahwa sistem dapat menghapus data faktor risiko dari *database*
Prosedur Pengujian	1. *User* memilih data faktor risiko 2. *User* menekan tombol hapus data faktor risiko yang akan dihapus
Hasil yang Diharapkan	Sistem dapat menghapus data faktor risiko dari *database*

Sumber : [Pengujian]

Skenario pengujian *blackbox* pada kasus uji *delete* data faktor risiko ditunjukkan pada Tabel 4.18.

Tabel 4.18 Pengujian Fungsionalitas Kasus Uji *Delete* Data Faktor Risiko

No	Skenario Pengujian	*Test Case*	Hasil yang Diharapkan	Hasil Pengujian	Status
1	Memilih data faktor risiko yang akan di	Memilih data faktor risiko	Sistem akan menghapus data faktor risiko dan	Sistem akan menghapus data faktor risiko dari	Valid

Tabel 4.18 Pengujian Fungsionalitas Kasus Uji *Delete* Data Faktor Risiko (lanjutan)

No	Skenario Pengujian	*Test Case*	Hasil yang Diharapkan	Hasil Pengujian	Status
	hapus		terdapat peringatan untuk menghapus data atau batal, jika yakin menghapus data, *user* menekan tombol OK	*database* jika yakin menghapus data, *user* menekan tombol OK	

Sumber : [Pengujian]

10. Kasus Uji Melihat Data Faktor Risiko

Kasus uji melihat data faktor risiko menjelaskan tentang fungsionalitas proses data faktor risiko pada sistem yang hanya dapat dilakukan oleh *user* dengan hak akses admin dan dokter spesialis yang ditunjukkan pada Tabel 4.19.

Tabel 4.19 Penjelasan Kasus Uji Melihat Data Faktor Risiko

Nama Kasus Uji	Kasus Uji Melihat Data Faktor Risiko
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat memenuhi kebutuhan fungsional untuk menampilkan data faktor risiko
Prosedur Pengujian	1. Admin dan dokter spesialis masuk ke dalam sistem melalui proses *login* 2. *User* memilih menu pengetahuan untuk melihat data faktor risiko
Hasil yang Diharapkan	Sistem dapat menampilkan data faktor risiko yang terdapat dalam *database*

Sumber : [Pengujian]

Skenario pengujian fungsionalitas pada kasus uji melihat data faktor risiko ditunjukkan pada Tabel 4.20.

Tabel 4.20 Pengujian Fungsionalitas Kasus Uji Melihat Data Faktor Risiko

No.	Skenario Pengujian	*Test Case*	Hasil yang Diharapkan	Hasil Pengujian	Status
1	Memilih menu pengetahuan	Memilih menu pengetahuan	Sistem akan menampilkan data faktor risiko penyakit *stroke*	Sistem akan menampilkan data faktor risiko penyakit *stroke*	Valid

Sumber : [Pengujian]

11. Kasus Uji Mengelola Data Latih

Pengujian *blackbox* kasus uji kelola data latih ditunjukkan pada Tabel 4.21.

Tabel 4.21 Penjelasan Kasus Uji Mengolah Data Latih

Nama Kasus Uji	Kasus Uji Mengolah Data Latih

Tabel 4.21 Penjelasan Kasus Uji Mengolah Data Latih (lanjutan)

Tujuan Pengujian	Memastikan bahwa sistem dapat menampilkan data latih seperti menambah data latih, *update* data latih, dan *delete* data latih
Prosedur Pengujian	1. admin dan ds melakukan *login* 2. admin dan ds memilih data latih
Hasil yang Diharapkan	Sistem dapat membuka data latih dalam *database*

Sumber : [Pengujian]

Skenario pengujian *blackbox* pada kasus uji mengolah data latih ditunjukkan pada Tabel 4.22.

Tabel 4.22 Pengujian *Blackbox* Kasus Uji Mengolah Data Latih

No.	Skenario Pengujian	*Test Case*	Hasil yang Diharapkan	Hasil Pengujian	Status
1	Menampilkan halaman data latih secara lengkap	Tekanan Darah: - Diabetes: - Riwayat Keluarga: - Merokok: - Kolesterol:- Aktivitas Fisik: - Diet: - Riwayat Afib: -	Sistem akan menampilkan halaman data latih secara lengkap	Sistem akan menampilkan halaman data latih secara lengkap	Valid

Sumber : [Pengujian]

12. Kasus Uji *Create* Data Latih

Kasus uji *create* data latih menjelaskan tentang fungsionalitas proses menambah data latih pada sistem yang hanya dapat dilakukan oleh *user* dengan hak akses admin dan dokter spesialis yang ditunjukkan pada Tabel 4.23.

Tabel 4.23 Penjelasan Kasus Uji *Create* Data Latih

Nama Kasus Uji	Kasus Uji *Create* Data Latih
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat memenuhi kebutuhan fungsional untuk menambah data latih
Prosedur Pengujian	1. admin dan dokter spesialis masuk ke dalam sistem melalui proses *login* 2. *User* memilih menu data latih 3. *User* mengisi data latih 4. *User* menekan tombol simpan
Hasil yang Diharapkan	1. Sistem dapat menyimpan data ke dalam *database* 2. Sistem dapat menampilkan data latih yang telah disimpan

Sumber : [Pengujian]

Skenario pengujian fungsionalitas pada kasus uji *create* data latih ditunjukkan pada Tabel 4.24.

Tabel 4.24 Pengujian Fungsionalitas Kasus Uji *Create* Data Latih

No.	Skenario Pengujian	*Test Case*	Hasil yang Diharapkan	Hasil Pengujian	Status
1	Menginputkan data pada semua *field*	Mengisi Nilai faktor :- Tingkat resiko :-	Sistem akan menerima proses tambah data latih dan menampilkan data telah disimpan setelah menekan tombol simpan	Sistem akan menerima proses tambah data latih dan menampilkan data telah disimpan setelah menekan tombol simpan	Valid

Sumber : [Pengujian]

1. Kasus Uji *Update* Data Latih

Kasus uji *update* data latih menjelaskan tentang fungsionalitas proses mengedit data latih pada sistem yang hanya dapat dilakukan oleh *user* dengan hak akses admin dan dokter spesialis yang ditunjukkan pada Tabel 4.25.

Tabel 4.25 Penjelasan Kasus Uji *Update* Data Latih

Nama Kasus Uji	Kasus Uji *Update* Data Latih
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat memenuhi kebutuhan fungsional untuk mengedit data latih
Prosedur Pengujian	1. *User* memilih data latih 2. *User* menekan tombol edit 3. *User* mengisi data latih yang akan diedit 4. *User* menekan tombol *update*
Hasil yang Diharapkan	Sistem dapat menampilkan perubahan data latih yang telah diedit

Sumber : [Pengujian]

Skenario pengujian fungsionalitas pada kasus uji *update* data latih ditunjukkan pada Tabel 4.26.

Tabel 4.26 Pengujian Fungsionalitas Kasus Uji *Update* Data Latih

No.	Skenario Pengujian	*Test Case*	Hasil yang Diharapkan	Hasil Pengujian	Status
1	Melakukan *login user*	Menekan tombol *update* data latih yang akan di edit	Sistem akan menampilkan data latih yang telah di edit	Sistem akan menampilkan data latih yang telah di edit	Valid
2	Admin, dan dokter spesialis mengedit data *user*	Menekan tombol *update*	Sistem akan menyimpan di dalam *database*	Sistem akan menyimpan dalam *database*	Valid

Sumber : [Pengujian]

14. Kasus Uji *Delete* Data Latih

Kasus uji *delete* data latih menjelaskan tentang pengujian fungsionalitas proses menghapus data latih pada sistem yang dapat dilakukan oleh *user*. Pengujian fungsionalitas ditunjukkan pada Tabel 4.27.

Tabel 4.27 Penjelasan Kasus Uji *Delete* Data Latih

Nama Kasus Uji	Kasus Uji *Delete* Data Latih
Tujuan Pengujian	Memastikan bahwa sistem dapat menghapus data latih dari *database*
Prosedur Pengujian	1. *User* memilih data latih 2. *User* menekan tombol hapus data latih yang akan dihapus
Hasil yang Diharapkan	Sistem dapat menghapus data latih dari *database*

Sumber : [Pengujian]

Skenario pengujian *blackbox* pada kasus uji *delete* data *user* ditunjukkan pada Tabel 4.28.

Tabel 4.28 Pengujian Fungsionalitas Kasus Uji *Delete* Data Latih

No	Skenario Pengujian	*Test Case*	Hasil yang Diharapkan	Hasil Pengujian	Status
1	Memilih data latih yang akan di hapus	Memilih data latih	Sistem akan menghapus data latih dan terdapat peringatan untuk menghapus data atau batal, jika yakin menghapus data, *user* menekan tombol OK	Sistem akan menghapus data latih dari *database* jika yakin menghapus data, *user* menekan tombol OK	Valid

Sumber : [Pengujian]

15. Kasus Uji Melihat Data Latih

Kasus uji melihat data latih menjelaskan tentang fungsionalitas proses data latih pada sistem yang hanya dapat dilakukan oleh *user* dengan hak akses admin dan dokter spesialis yang ditunjukkan pada Tabel 4.29.

Tabel 4.29 Penjelasan Kasus Uji Melihat Data Latih

Nama Kasus Uji	Kasus Uji Melihat Data Latih
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat memenuhi kebutuhan fungsional untuk menampilkan data latih
Prosedur Pengujian	1. Admin dan dokter spesialis masuk ke dalam sistem melalui proses *login* 2. *User* memilih menu pengetahuan untuk melihat data latih
Hasil yang Diharapkan	Sistem dapat menampilkan data latih yang terdapat dalam *database*

Sumber : [Pengujian]

Skenario pengujian fungsionalitas pada kasus uji melihat data latih ditunjukkan pada Tabel 4.30.

Tabel 4.30 Pengujian Fungsionalitas Kasus Uji Melihat Data Latih

No.	Skenario Pengujian	*Test Case*	Hasil yang Diharapkan	Hasil Pengujian	Status
1	Memilih menu data *training*	Memilih menu data *training*	Sistem akan menampilkan data *training* penyakit *stroke*	Sistem akan menampilkan data *training* penyakit *stroke*	Valid

Sumber : [Pengujian]

16. Kasus Uji Mengelola Data Deteksi

Pengujian *blackbox* pada kasus uji kelola data deteksi ditunjukkan pada Tabel 4.31.

Tabel 4.31 Penjelasan Kasus Uji Mengolah Data Deteksi

Nama Kasus Uji	Kasus Uji Mengolah Data Deteksi
Tujuan Pengujian	Memastikan bahwa sistem dapat menampilkan data deteksi seperti menambah data deteksi, *update* data deteksi, dan *delete* data deteksi
Prosedur Pengujian	1. *User* melakukan *login* 2. *User* memilih data pengujian
Hasil yang Diharapkan	Sistem dapat membuka data pengujian dalam *database*

Sumber : [Pengujian]

Skenario pengujian *blackbox* pada kasus uji mengolah data deteksi ditunjukkan pada Tabel 4.32.

Tabel 4.32 Pengujian *Blackbox* Kasus Uji Mengolah Data Deteksi

No.	Skenario Pengujian	*Test Case*	Hasil yang Diharapkan	Hasil Pengujian	Status
1	Menampilkan halaman data deteksi secara lengkap	Tekanan Darah: - Diabetes: - Riwayat Keluarga: - Merokok: - Kolesterol:- Aktivitas Fisik: - Diet: - Riwayat Afib: -	Sistem akan menampilkan halaman data dateksi secara lengkap	Sistem akan menampilkan halaman data deteksi secara lengkap	Valid

Sumber : [Pengujian]

17. Kasus Uji Menambah Data Deteksi

Pengujian *blackbox* pada kasus uji tambah data deteksi ditunjukkan pada Tabel 4.33.

Tabel 4.33 Penjelasan Kasus Uji Tambah Data Deteksi

Nama Kasus Uji	Kasus Uji Tambah Data Deteksi
Tujuan Pengujian	Memastikan bahwa sistem dapat menambah data deteksi
Prosedur Pengujian	1. *User* melakukan *login* 2. *User* memilih data pengujian
Hasil yang Diharapkan	Sistem dapat membuka data pengujian dalam *database*

Sumber : [Pengujian]

Skenario pengujian *blackbox* pada kasus uji tambah data deteksi ditunjukkan pada Tabel 4.34.

Tabel 4.34 Pengujian *Blackbox* Kasus Uji Tambah Data Deteksi

No.	Skenario Pengujian	*Test Case*	Hasil yang Diharapkan	Hasil Pengujian	Status
1	Menampilkan halaman pengujian secara lengkap	Tekanan Darah: - Diabetes: - Riwayat Keluarga: - Merokok: - Kolesterol:- Aktivitas Fisik: - Diet: - Riwayat Afib: -	Sistem akan menampilkan halaman diagnosa secara lengkap	Sistem akan menampilkan halaman diagnosa secara lengkap	Valid

Sumber : [Pengujian]

18. Kasus Uji *Update* Data Deteksi

Kasus uji *update* data deteksi menjelaskan tentang fungsionalitas proses mengedit data deteksi pada sistem yang hanya dapat dilakukan oleh *user* yang ditunjukkan pada Tabel 4.35.

Tabel 4.35 Penjelasan Kasus Uji *Update* Data Deteksi

Nama Kasus Uji	Kasus Uji *Update* Data Deteksi
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat memenuhi kebutuhan fungsional untuk mengedit data deteksi
Prosedur Pengujian	1. *User* memilih data deteksi 2. *User* menekan tombol edit 3. *User* mengisi data deteksi yang akan diedit 4. *User* menekan tombol *update*
Hasil yang Diharapkan	Sistem dapat menampilkan perubahan data deteksi yang telah diedit

Sumber : [Pengujian]

Skenario pengujian fungsionalitas pada kasus uji *update* data deteksi ditunjukkan pada Tabel 4.36.

Tabel 4.36 Pengujian Fungsionalitas Kasus Uji *Update* Data Deteksi

No.	Skenario Pengujian	*Test Case*	Hasil yang Diharapkan	Hasil Pengujian	Status
1	Melakukan *login user*	Menekan tombol *update* data deteksi yang akan di edit	Sistem akan menampilkan data deteksi yang telah di edit	Sistem akan menampilkan data deteksi yang telah di edit	Valid
2	*User* mengedit data deteksi	Menekan tombol *update*	Sistem akan menyimpan di dalam *database*	Sistem akan menyimpan dalam *database*	Valid

Sumber : [Pengujian]

19. Kasus Uji *Delete* Data Deteksi

Kasus uji *delete* data deteksi menjelaskan tentang pengujian fungsionalitas proses menghapus data deteksi pada sistem yang dapat dilakukan oleh *user*. Pengujian fungsionalitas ditunjukkan pada Tabel 4.37.

Tabel 4.37 Penjelasan Kasus Uji *Delete* Data Deteksi

Nama Kasus Uji	Kasus Uji *Delete* Data Deteksi
Tujuan Pengujian	Memastikan bahwa sistem dapat menghapus data deteksi dari *database*
Prosedur Pengujian	1. *User* memilih data deteksi 2. *User* menekan tombol hapus data deteksi yang akan dihapus
Hasil yang Diharapkan	Sistem dapat menghapus data deteksi dari *database*

Sumber : [Pengujian]

Skenario pengujian *blackbox* pada kasus uji *delete* data deteksi ditunjukkan pada Tabel 4.38.

Tabel 4.38 Pengujian Fungsionalitas Kasus Uji *Delete* Data Deteksi

No	Skenario Pengujian	*Test Case*	Hasil yang Diharapkan	Hasil Pengujian	Status
1	Memilih data deteksi yang akan di hapus	Memilih data deteksi	Sistem akan menghapus data deteksi dan terdapat peringatan untuk menghapus data atau batal, jika yakin menghapus data, *user* menekan tombol OK	Sistem akan menghapus data deteksi dari *database* jika yakin menghapus data, *user* menekan tombol OK	Valid

Sumber : [Pengujian]

20. Kasus Uji Melihat Data Deteksi

Kasus uji melihat data deteksi menjelaskan tentang fungsionalitas proses data deteksi pada sistem yang hanya dapat dilakukan oleh *user* yang ditunjukkan pada Tabel 4.39.

Tabel 4.39 Penjelasan Kasus Uji Melihat Data Deteksi

Nama Kasus Uji	Kasus Uji Melihat Data Deteksi
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat memenuhi kebutuhan fungsional untuk menampilkan data deteksi
Prosedur Pengujian	1. *User* masuk ke dalam sistem melalui proses *login* 2. *User* memilih menu pengujian untuk melihat data deteksi
Hasil yang Diharapkan	Sistem dapat menampilkan data deteksi yang terdapat dalam *database*

Sumber : [Pengujian]

Skenario pengujian fungsionalitas pada kasus uji melihat data deteksi ditunjukkan pada Tabel 4.40.

Tabel 4.40 Pengujian Fungsionalitas Kasus Uji Melihat Data Deteksi

No.	Skenario Pengujian	*Test Case*	Hasil yang Diharapkan	Hasil Pengujian	Status
1	Memilih menu data pengujian	Memilih menu data pengujian	Sistem akan menampilkan data pengujian penyakit *stroke*	Sistem akan menampilkan data pengujian penyakit *stroke*	Valid

Sumber : [Pengujian]

21. Kasus Uji Menampilkan Hasil Keputusan

Kasus uji hasil deteksi dini penyakit *stroke* menampilkan hasil riwayat deteksi pasien yang beresiko *stroke* yang dapat dilakukan oleh *user*. Pengujian *blackbox* pada kasus uji hasil riwayat deteksi pasien ditunjukkan pada Tabel 4.41.

Tabel 4.41 Penjelasan Kasus Uji Melihat Hasil Keputusan

Nama Kasus Uji	Kasus Uji Hasil Keputusan
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat memenuhi kebutuhan fungsional untuk menampilkan hasil keputusan
Prosedur Pengujian	1. admin, dokter spesialis, dan dokter umum masuk ke dalam sistem melalui proses *login* 2. Pengguna memilih menu laporan hasil deteksi
Hasil yang Diharapkan	Sistem dapat menampilkan hasil keputusan pasien yang beresiko terkena penyakit *stroke*

Sumber : [Pengujian]

Skenario pengujian fungsionalitas pada kasus uji hasil keputusan pasien ditunjukkan pada Tabel 4.42.

Tabel 4.42 Pengujian Fungsionalitas Kasus Uji Hasil Keputusan

No.	Skenario Pengujian	*Test Case*	Hasil yang Diharapkan	Hasil Pengujian	Status
1	Memilih menu laporan	Memilih menu laporan	Sistem akan menampilkan laporan hasil riwayat deteksi penyakit *stroke*	Sistem akan menampilkan laporan hasil riwayat deteksi penyakit *stroke*	Valid

Sumber : [Pengujian]

22. Kasus Uji Menampilkan Perhitungan

Pengujian *blackbox* pada kasus uji perhitungan data pengujian ditunjukkan pada Tabel 4.43.

Tabel 4.43 Penjelasan Kasus Uji Perhitungan

Nama Kasus Uji	Kasus Uji Mengolah Perhitungan
Tujuan Pengujian	Memastikan bahwa sistem dapat menampilkan perhitungan seperti nilai *mean* dan standar deviasi, nilai probabilitas, serta nilai gaussian
Prosedur Pengujian	1. *User* melakukan *login* 2. *User* memilih data diagnosa
Hasil yang Diharapkan	Sistem dapat membuka data pengujian dalam diagnosa

Sumber : [Pengujian]

Skenario pengujian *blackbox* pada kasus uji perhitungan data pengujian ditunjukkan pada Tabel 4.44.

Tabel 4.44 Pengujian *Blackbox* Kasus Uji Perhitungan

No.	Skenario Pengujian	*Test Case*	Hasil yang Diharapkan	Hasil Pengujian	Status
1	Menampilkan halaman diagnosa secara lengkap	Menampilkan hasil perhitungan pada diagnosa	Sistem akan menampilkan halaman perhitungan secara lengkap	Sistem akan menampilkan halaman perhitungan secara lengkap	Valid

Sumber : [Pengujian]

23. Kasus Uji *Logout*

Kasus uji *logout* menjelaskan proses fungsionalitas proses *logout* seperti pada tabel 4.45 berikut.

Tabel 4.45 Penjelasan Kasus Uji *Logout*

Nama Kasus Uji	Kasus Uji *Logout*
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat memenuhi kebutuhan fungsional untuk *logout*
Prosedur Uji	1. *User* sudah masuk ke dalam sistem sesuai dengan hak akses 2. *User* memilih menu *logout*
Hasil yang diharapkan	Sistem dapat menampilkan halaman utama

Sumber : [Pengujian]

Tabel 4.46 merupakan tabel kasus pengujian fungsionalitas untuk kasus uji *Logout*.

Tabel 4.46 Pengujian Fungsionalitas Kasus Uji *Logout*

No.	Skenario Pengujian	*Test Case*	Hasil yang Diharapkan	Hasil Pengujian	Status
1.	Memilih menu *logout* dan *user* dapat keluar dari sistem	Memilih menu *logout*	Sistem akan keluar dan menampilkan halaman utama	Sistem akan keluar dan menampilkan halaman utama	Valid

Sumber : [Pengujian]

4.1.1.3 Hasil

Dari prosedur pada pengujian fungsionalitas menggunakan *blackbox* yang terdiri dari tabel-tabel, menunjukkan bahwa sistem dapat berjalan sesuai dengan kebutuhan fungsional yang ada. Hal ini dapat dilihat dari sistem yang dapat berjalan dengan baik sesuai dengan kebutuhan fungsional yang ada dan hasil yang diharapkan. Hasil pengujian *blackbox* berdasarkan kasus uji masing-masing kebutuhan akan ditunjukkan pada Tabel 4.47.

Tabel 4.47 Hasil Pengujian *Blackbox*

No.	Nama Kasus Uji	Hasil yang Diharapkan	Hasil yang Didapatkan	Status
1	*Login*	Sistem berhasil menerima data *login* berupa email dan *password* sehingga *user* dapat mengakses menu pada sistem sesuai dengan hak akses masing-masing *user*	Sistem berhasil menerima data *login* berupa email dan *password* sehingga *user* dapat mengakses menu pada sistem sesuai dengan hak akses masing-masing *user*	Valid
2	Mengolah data *user*	Sistem dapat mengolah data *user*	Sistem dapat mengolah data *user*	Valid
3	*Create* data *user*	Sistem dapat menambah data *user*	Sistem dapat menambah data *user*	Valid
4	*Update* data *user*	Sistem dapat meng*update* data *user*	Sistem dapat meng*update* data *user*	Valid
5	*Delete* data *user*	Sistem dapat menghapus data *user*	Sistem dapat menghapus data *user*	Valid
6	Menampilkan detail data *user*	Sistem dapat menampilkan data *user*	Sistem dapat menampilkan data *user*	Valid
7	Mengelola data faktor risiko	Sistem dapat mengelola data faktor risiko	Sistem dapat mengelola data faktor risiko	Valid

Tabel 4.47 Hasil Pengujian *Blackbox* (lanjutan)

No.	Nama Kasus Uji	Hasil yang Diharapkan	Hasil yang Didapatkan	Status
8	*Edit* data faktor risiko	Sistem dapat menge*update* data faktor risiko	Sistem dapat meng*update* data faktor risiko	Valid
8	*Edit* data faktor risiko	Sistem dapat menge*update* data faktor risiko	Sistem dapat meng*update* data faktor risiko	Valid
9	D*elete* data data faktor risiko	Sistem dapat menghapus data faktor risiko	Sistem dapat menghapus data faktor risiko	Valid
10	Melihat data data faktor risiko	Sistem dapat melihat data faktor risiko	Sistem dapat melihat data faktor risiko	Valid
11	Mengelola data latih	Sistem dapat mengolah data latih	Sistem dapat mengolah data latih	Valid
12	*Create* data latih	Sistem dapat menambah data latih	Sistem dapat menambah data latih	Valid
13	*Edit* data latih	Sistem dapat meng*update* data latih	Sistem dapat meng*update* data latih	Valid
14	*Delete* data latih	Sistem dapat menghapus data latih	Sistem dapat menghapus data latih	Valid
15	Melihat data latih	Sistem dapat melihat data latih	Sistem dapat melihat data latih	Valid
16	Mengelola data deteksi dini penyakit *stroke*	Sistem dapat mengolah deteksi dini penyakit *stroke*	Sistem dapat mengolah data deteksi dini penyakit *stroke*	Valid
17	*Create* data deteksi	Sistem dapat menambah data deteksi dini penyakit *stroke*	Sistem dapat menambah data deteksi dini penyakit *stroke*	Valid
18	*Edit* data deteksi	Sistem dapat meng*update* data deteksi dini penyakit *stroke*	Sistem dapat men*update* data deteksi dini penyakit *stroke*	Valid
19	*Delete* data deteksi	Sistem dapat menghapus data deteksi dini penyakit *stroke*	Sistem dapat menghapus data deteksi dini penyakit *stroke*	Valid
20	Melihat data deteksi	Sistem dapat melihat data deteksi dini penyakit *stroke*	Sistem dapat melihat data deteksi dini penyakit *stroke*	Valid
21	Melihat hasil keputusan deteksi dini penyakit *stroke*	Sistem dapat melihat hasil keputusan deteksi dini penyakit *stroke*	Sistem dapat melihat hasil keputusan deteksi dini penyakit *stroke*	Valid
22	Menampilkan proses perhitungan	Sistem dapat menampilkan proses perhitungan	Sistem dapat menampilkan proses perhitungan	Valid
23	*Logout*	Sistem dapat menerima proses *logout* yang dilakukan oleh ds dan du sehingga *user* dapat keluar dari sistem dan sistem menampilkan halaman utama	Sistem dapat menerima proses *logout* yang dilakukan oleh ds dan du sehingga *user* dapat keluar dari sistem dan sistem menampilkan halaman utama	Valid

Sumber : [Pengujian]

Berdasarkan Tabel diatas, pengujian *blackbox* dilakukan dengan menguji 23 kebutuhan fungsional. Hasil dari pengujian *blackbox* sistem pendukung keputusan deteksi dini penyakit *stroke* memiliki nilai validasi dengan perhitungan sebagai berikut.

$$\text{Validasi} = \frac{jumlah\ tindakan\ yang\ dilakukan}{jumlah\ tindakan\ dalam\ daftar\ kebutuhan} \text{ x 100%}$$
$$= \frac{23}{23} \text{ x 100%}$$
$$= 100\%$$

4.1.2 Analisis Pengujian *Blackbox*

Berdasarkan pengujian *blackbox* kesesuaian yang didapatkan dengan hasil yang diharapkan mencapai akurasi 100%. Hal ini membuktikan bahwa sistem pendukung keputusan deteksi dini penyakit *stroke* menggunakan metode *Naive Bayes* dapat dijalankan sesuai dengan kebutuhan *user*.

4.2 Pengujian Akurasi

Pengujian akurasi dilakukan untuk menghitung tingkat akurasi antara hasil deteksi sistem dengan hasil deteksi yang dilakukan oleh pakar terhadap sejumlah data latih. Pengujian akurasi dilakukan untuk mencari nilai akurasi tertinggi. Sehingga hasil pengujian akurasi akan digunakan sebagai nilai ketetapan pada proses perhitungan metode *Naive Bayes*.

4.2.1 Skenario Pengujian Akurasi

Pengujian akurasi dilakukan dengan cara menyesuaikan hasil diagnosa yang dilakukan oleh sistem dengan hasil diagnosa pakar sejumlah data latih. Pengujian data latih dilakukan terhadap beberapa data yang dipilih secara acak. Pada pengujian ini akan dilakukan pengujian variasi data yang digunakan yaitu 80%:20%, 70%:30%, 60%:40%, 50%:50%, dan 40:60%, dari total keseluruhan data yang berjumlah 150 data.

Pengujian terhadap data latih dilakukan dengan mengambil sejumlah 40%, 50%, 60%, 70% dan 80% dari keseluruhan data latih yang digunakan (120 data). Masing-masing n% data latih yang akan diuji memiliki 5 variasi komposisi. Variasi komposisi tersebut didapatkan dengan cara pembulatan ke bawah, pembulatan ke atas, dan random.

Pengujian pertama dilakukan dengan mengambil data sejumlah 80% dari keseluruhan data latih yang digunakan (120 data) dengan keseluruhan data uji yang digunakan (30 data). Pengujian kedua dilakukan dengan mengambil data sejumlah 70% dari keseluruhan data latih yang digunakan (105 data) dengan keseluruhan data uji yang digunakan (45 data). Pengujian ketiga dilakukan dengan mengambil data sejumlah 60% dari keseluruhan data latih yang digunakan (90 data) dengan keseluruhan data uji yang digunakan (60 data). Pengujian keempat dilakukan dengan mengambil data sejumlah 50% dari keseluruhan data latih yang digunakan (75 data) dengan keseluruhan data uji yang digunakan (75 data). Pengujian kelima dilakukan dengan mengambil data

sejumlah 40% dari keseluruhan data latih yang digunakan (60 data) dengan keseluruhan data uji yang digunakan (90 data).

4.2.1.1 Tujuan

Tujuan dari pengujian akurasi dan variasi jumlah data adalah untuk mencari nilai akurasi tertinggi sebagai komposisi terbaik pada data yang ada. Akurasi tertinggi akan digunakan sebagai ketetapan nilai dalam metode *Naive Bayes* pada proses perhitungan sistem pendukung keputusan deteksi dini penyakit *stroke*.

4.2.1.2 Prosedur

Prosedur pada pengujian akurasi terhadap variasi data dilakukan dengan cara menguji sistem yaitu melakukan deteksi pada sejumlah data latih yang berbeda-beda. Variasi data latih yang digunakan dalam pengujian ini yaitu 80%, 70%, 60%, 50% dan 40%. Pengujian dilakukan terhadap 20%, 30%, 40%, 50%, dan 60% data uji yang dipilih secara acak.

4.3.1.3 Hasil

Pengujian terhadap variasi data latih telah dilakukan maka akan diperoleh nilai akurasi pada masing-masing jumlah data latih. Nilai akurasi dari masing masing jumlah data latih digunakan untuk mencari nilai variasi data latih yang tertinggi. Hasil akurasi ditujukan untuk mengetahui hasil perhitungan sistem sama dengan hasil deteksi pakar jika bernilai 1, sebaliknya jika hasil akurasi bernilai 0 artinya keluaran dari deteksi sistem tidak sama dengan deteksi pakar.

Pengujian 5 skenario dengan beberapa data latih menggunakan beberapa data uji yang dipilih secara acak. Data uji yang digunakan tersebut akan dilihat kesesuaiannya antara hasil diagnosa pakar dengan hasil diagnosa sistem kemudian akan dihitung akurasi dari masing-masing komposisi dan akan diambil nilai rata-rata akurasi dari masing-masing komposisi untuk dijadikan sebagai akurasi sistem. Hasil pengujian variasi data latih akan ditunjukkan pada Tabel 4.48.

Tabel 4.48 Kombinasi Pengujian Variasi Akurasi Data

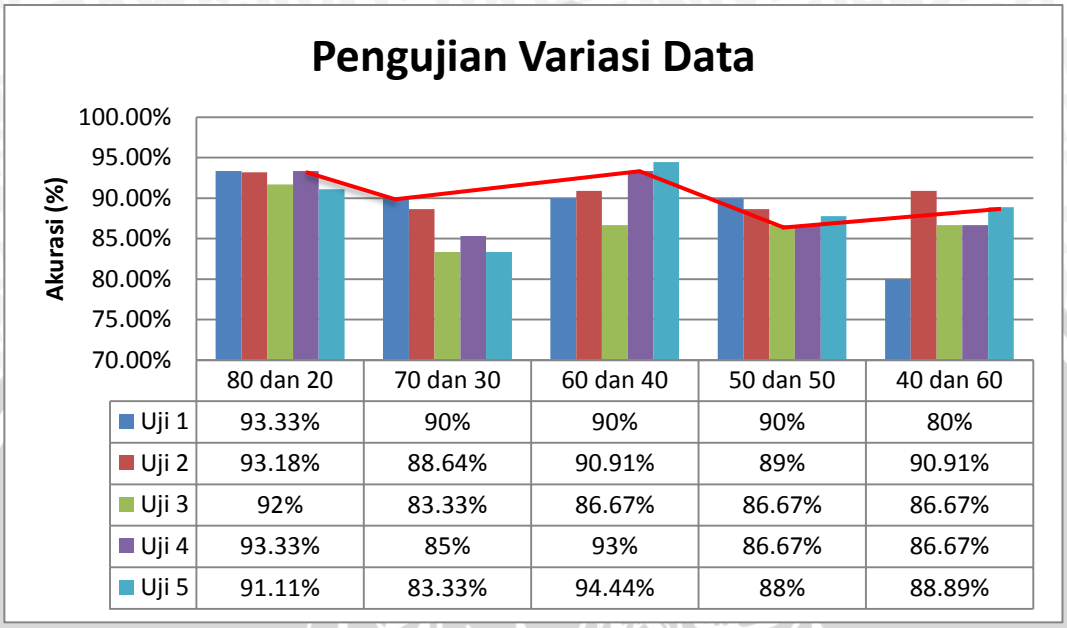
Uji	Rasio				
	80% : 20%	70% : 30%	60% : 40%	50% : 50%	40% : 60%
1	93.33%	93.18%	91.67%	93.33%	91.11%
2	90%	88.64%	83.33%	85.33%	83.33%
3	90%	90.91%	86.67%	93.33%	94.44%
4	90%	88.64%	86.67%	86.67%	87.78%
5	80%	90.91%	86.67%	86.67%	88.89%
Rerata	88.67%	90.46%	87%	89.1%	89.11%

Sumber : [Pengujian]

4.2.2 Analisis Pengujian Akurasi Variasi Jumlah Data Latih

Berdasarkan pengujian akurasi variasi data latih maka diperoleh nilai akurasi pada masing-masing data latih. Pada akurasi tersebut akan dilihat nilai yang memiliki akurasi yang tertinggi seperti yang terdapat pada Gambar 4.2. Nilai

akurasi mengalami kenaikan dan penurunan dikarenakan jumlah antar variasi data latih tidak sama. Pada komposisi data latih 80% menunjukkan akurasi sebesar 88.67% dan pada komposisi data latih 70% sebesar 90.46%. Pada komposisi data latih 60% menunjukkan akurasi sebesar 87% dan pada komposisi data latih 50% sebesar 89.1%. Pada komposisi data latih 40% menunjukkan akurasi sebesar 89.11%.



	80 dan 20	70 dan 30	60 dan 40	50 dan 50	40 dan 60
Uji 1	93.33%	90%	90%	90%	80%
Uji 2	93.18%	88.64%	90.91%	89%	90.91%
Uji 3	92%	83.33%	86.67%	86.67%	86.67%
Uji 4	93.33%	85%	93%	86.67%	86.67%
Uji 5	91.11%	83.33%	94.44%	88%	88.89%

Gambar 4.2 Pengujian Variasi Data
Sumber : [Pengujian]

Berdasarkan kelima skenario pengujian akurasi terhadap variasi data menghasilkan nilai rata-rata akurasi masing-masing skenario sebesar 88.67%, 90.46%, 87%, 891%, dan 89.11%. Dengan pemaparan hasil rata-rata pengujian variasi data dapat diambil kesimpulan bahwa tingkat akurasi tertinggi didapat ketika variasi berjumlah 70% dan 30% dari keseluruhan data. Hal ini membuktikan bahwa komposisi jumlah data kasus berpengaruh dalam hasil akurasi sistem. Semakin banyak data latih belum tentu dapat menjamin akurasi sistem pendukung keputusan yang dihasilkan akan semakin baik. Sehingga, dalam menentukan data latih harus memperhatikan komposisi jumlah variasi data kasus pada masing-masing *class* untuk menghasilkan sistem pendukung keputusan yang baik.

BAB 5 PENUTUP

Bab ini membahas tentang kesimpulan dan saran dari penelitian yang telah dilakukan berdasarkan hasil perancangan, implementasi dan pengujian sistem.

5.1 Kesimpulan

Berdasarkan hasil perancangan dan pengujian yang dilakukan pada sistem pendukung keputusan deteksi dini penyakit *stroke* menggunakan metode *Naive Bayes*, maka dapat diambil kesimpulan sebagai berikut :

1. Proses deteksi dini penyakit *stroke* dilakukan dengan cara memasukkan faktor risiko dari inputan yang diberikan oleh DS dan DU. Melalui faktor risiko tersebut akan dilakukan perhitungan dengan metode *Naive Bayes* untuk mendapatkan nilai probabilitas posterior pada setiap tingkat risiko *stroke* yang menjadi studi kasus pada penelitian ini. Tingkat risiko *stroke* dengan nilai probabilitas tertinggi akan di ambil sebagai hasil identifikasi sistem.
2. Sistem pendukung keputusan deteksi dini penyakit *stroke* menggunakan metode *Naive Bayes* ini memiliki kinerja sistem yang mampu berjalan dengan baik sesuai kebutuhan fungsional. Hal ini berdasarkan hasil pengujian *blackbox* yang telah membuktikan bahwa seluruh fungsi dapat bekerja sesuai dengan hasil yang diharapkan.
3. Berdasarkan pengujian akurasi yang sudah dilakukan, sistem pendukung keputusan deteksi dini penyakit *stroke* menggunakan metode *Naive Bayes* mempunyai tingkat akurasi sebesar 90.46% dan nilai *error* sebesar 9.54%. Hal ini disebabkan karena komposisi data kasus berpengaruh dalam hasil akurasi sistem. Semakin banyak data *training* belum tentu dapat menjamin akurasi sistem pakar yang dihasilkan akan semakin baik. Komposisi data pada masing-masing *class* menentukan akurasi sistem pendukung keputusan.

5.2 Saran

Sistem pendukung keputusan deteksi dini penyakit *stroke* menggunakan metode *Naive Bayes* ini masih memiliki beberapa kekurangan. Saran yang dapat diberikan untuk pengembangan sistem agar menjadi lebih baik antara lain :

1. Dapat dilakukan penambahan data latih atau menambah faktor risiko ke dalam basis pengetahuan yang nantinya dapat digunakan dalam proses identifikasi terhadap diagnosa penyakit *stroke*.
2. Dapat dilakukan pembobotan faktor risiko ke dalam basis pengetahuan untuk memudahkan perhitungan dengan data kualitatif sederhana yang nantinya dapat digunakan dalam proses identifikasi terhadap diagnosa penyakit *stroke*.
3. Untuk pengembangan lebih lanjut, sistem dapat dikembangkan dengan menggunakan metode yang berbeda atau mengkombinasikan metode *Naive*

Bayes dengan metode lain guna memperoleh sistem yang lebih akurat, efektif, dan efisien.

DAFTAR PUSTAKA

A. Ambica, S. G. (Anakapalle 2013). An Efficient Expert System For Diabetes By Naive Bayesian Classifier. *IJETT Volume 4*, 4634-4639.

Ahmad Fashel Sholeh, A. D. (2012). Aplikasi Pendukung Keputusan Untuk Deteksi Dini Risiko Penyakit Stroke Menggunakan Logika Fuzzy Mamdani : Studi Kasus Di RS XYZ . *Teknik POMITS Volume 1*, 1-6.

Ariani, T. A. (2012). *Sistem Neurobehavior.* Jakarta: Salemba Medika.

Association, N. S. (2009). Retrieved from www.stroke.org [Diakses 12 Maret 2015].

Atika, L. (2012). Sistem Pakar Pendeteksi Kemungkinan Penyakit Stroke. *Teknik Informatika Universitas Bina Darma*, 247-253.

Batticaca, F. B. (2012). *Asuhan Keperawatan pada Klien dengan Gangguan Sistem Persarafan.* Jakarta: Salemba Medika.

Darmanto, D. L. (2014). Sistem Pendukung Keputusan Untuk Memprediksi Kelulusan Mahasiswa Menggunakan Metode Naive Bayes. *Universitas Muria Kudus*.

E. Turban, J. E. (2005). *Desicion Support System and Intelligent System.* Yogyakarta: Andi.

Harati, P. (2013, Juli). Retrieved from pintarharati.blogspot.com/2013/07/certainty-factor.html?m=1, [Diakses 19 April 2015].

IMT, I. M. (2013). Retrieved from www.posyandu.com, [Diakses 23 Maret 2015].

Irfan, M. (2012). *Fisioterapi Bagi Insan Stroke.* Yogyakarta: Graha Ilmu.

Jogiyanto. (1990). *Analisis dan Disain. p.2.* Yogyakarta: Andi.

Kesehatan.co. (2015). *Info Kesehatan, Kecantikan, dan Pola Hidup Sehat*. Retrieved April 19, 2015, from Kesehatan.co: http://www.kesehatan.co/menghitung-berat-badan-ideal-dengan-rumus-indeks-massa-tubuh/, [Diakses 19 April 2015].

Kusrini, T. L. (2009). *Algoritma Data Mining.* Yogyakarta: Andi.

Kusumadewi, S. (Mei 2009). Klasifikasi Status Gizi Menggunakan Naive Bayes Classification. *Teknik Informatika Universitas Islam Indonesia Volume 3*, 6-11.

Mandera, L. I. (2006). *Buku Saku Dokter Keluarga Edisi 3.* Jakarta: Penerbit Buku Kedokteran EGC.

Md. F. Kabir, d. (August 2011). Enhanched Classification Accuracy on Naive Bayes Data Mining Models. *International Journal of Computer Applications Volume 28*, 9-16.

Mukhlis, M. K. (2010). Diagnosa Kemungkinan Pasien Terkena Stroke Dengan Menggunakan Metode Naive Bayes Dan Metode Jaringan Syaraf Tiruan Berbasis Web.

Muttaqin, A. (2008). *Pengantar Asuhan Keperawatan Klien Dengan Gangguan Sistem Pernapasan.* Jakarta: Salemba Medika.

N. M. Lestari, I. K. (Januari 2013). Personality Types Classification for Indonesian Text in Partners Searching Website Using Naive Bayes Methods. *IJCSI Volume 10*, 1-8.

Parno, S. M. (2010). *Lecture Notes : Sistem Informasi Data Flow Diagram.*

Prasetyo, E. (2012). *Data Mining - Konsep dan Aplikasi Menggunakan MATLAB.* Yogyakarta: Andi.

Pratama, A. H. (2014). Aplikasi Sistem Pakar Untuk Identifikasi Hama Dan Penyakit Tanaman Tebu Dengan Metode Naive Bayes Berbasis Web. *Universitas Brawijaya*, 1-10.

Prihatini, P. M. (Juni 2012). Metode Ketidakpastian dan Kesamaran Dalam Sistem Pakar. *Lontar Komputer Volume 2*, 29-42.

Prijodiprojo, E. G. (2013). Prototype Sistem Pakar Untuk Mendeteksi Tingkat Resiko Penyakit Jantung Koroner Dengan Metode Dempster-Shafer. *IJCCS Voume 7*, 133-144.

Putri, T. S. (2013). Retrieved from Beda Sistem Pakar dan Sistem Pendukung Keputusan, [Diakses 23 Maret 2015].

Ratnasari, W. S. (2013). Sistem Pakar Diagnosa Penyakit Mata Menggunakan Naive Bayes Classifier. *Universitas Trunojoyo Bangkalan*.

Stroke. (2012). Retrieved from www.terapistroke.com, [Diakses 23 Maret 2015].

T. Sutojo, d. (2011). *Kecerdasan Buatan.* Yogyakarta: Andi.

LAMPIRAN A WAWANCARA

DOKUMEN WAWANCARA

SISTEM PENDUKUNG KEPUTUSAN DETEKSI DINI PENYAKIT *STROKE*

Diajukan untuk memenuhi sebagian persyaratan Skripsi

KEMENTERIAN RISET TEKNOLOGI DAN PEDIDIKAN TINGGI

UNIVERSITAS BRAWIJAYA

PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER

MALANG

2015

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

KATA PENGANTAR

Puji syukur kami panjatkan atas kehadirat Allah SWT, yang telah melimpahkan rahmat dan hidayah-Nya sehingga kami dapat menyelesaikan dokumen wawancara ini dengan lancar. Tak lupa dalam kesempatan ini kami mengucapkan terima kasih kepada pimpinan dan staff Rumah Sakit Saiful Anwar Malang yang telah membantu dan memberikan kesempatan untuk melakukan wawancara demi kelengkapan data-data yang dibutuhkan. Pada kesempatan ini kami ingin melakukan wawancara untuk melengkapi data-data sebagai pendukung skripsi yang sedang kami kerjakan. Topik skripsi yang sedang kami susun adalah sistem pendukung keputusan deteksi dini penyakit *Stroke*. Adapun tujuan dari wawancara ini adalah untuk memperoleh pengetahuan dari seorang pakar sebagai pendukung dalam membangun sebuah sistem pakar. Dokumen ini disusun untuk memudahkan wawancara yang akan kami laksanakan kepada pakar. Pakar yang kami maksudkan disini adalah dokter spesialis saraf yang bertugas sebagai tenaga medis di Rumah Sakit Saiful Anwar Malang.

Penulis menyadari bahwa dokumen wawancara ini masih jauh dari sempurna, oleh karena itu kritik dan saran yang sifatnya membangun sangat diperlukan untuk memperbaiki mutu penulisan selanjutnya dan juga kebaikan penulis secara pribadi. Besar harapan kami semoga wawancara ini dapat bermanfaat untuk semua pihak.

Malang, Maret 2015

Penulis

DAFTAR WAWANCARA

DOKUMEN WAWANCARA **70**
KATA PENGANTAR **71**
DAFTAR WAWANCARA **72**
MATERI WAWANCARA **73**
A. Umum 73
B. Penyakit *Stroke* 74
C. Cara Diagnosa 75
TABEL NILAI KEPERCAYAAN **77**
TABEL ISTILAH GEJALA **77**
NILAI SKOR GEJALA **78**

MATERI WAWANCARA

Tujuan Wawancara	:	1. Untuk mendapatkan informasi tentang tingkat resiko pada penyakit *Stroke* di Kota Malang. 2. Untuk mendapatkan data pasien yang beresiko terkena penyakit *Stroke* di Kota Malang. 3. Untuk mengetahui prosedur pemeriksaan pasien saraf di RS Saiful Anwar Malang. 4. Untuk mengetahui cara diagnose resiko penyakit *Stroke* yang dilakukan oleh Dokter.
Objek Wawancara	:	Konsultan *Stroke* (Dokter Spesialis Saraf)
Target Person Wawancara	:	dr. Eko Ari Setijono Marhendraputro, SpS
Tempat Wawancara	:	Rumah Sakit Saiful Anwar Malang
Pembimbing Utama I	:	Arief Andy Soebroto, ST., M.Kom

A. Umum

Tujuan : Materi wawancara ini bertujuan untuk mengetahui gambaran umum pencegahan penyakit *Stroke* di RS Saiful Anwar Malang

No.	Pertanyaan	Jawaban	Checklist
A1	Bagaimana pengertian penyakit saraf? (tujuan : untuk mengetahui pengertian dari penyakit saraf)	Adanya gangguan pada sistem saraf. Sistem saraf dibagi menjadi 2 yaitu susuan saraf pusat dan susunan saraf tepi. Gangguan bisa berupa gangguan karena proses degenaratif, gangguan karena neoplasma, gangguan karena infeks, gangguan pada pembuluh darah, gangguan karena benturan dan penyebab yang lain.	
A2	Apa saja penyakit saraf itu? (tujuan : untuk mengetahui penyakit apa yang termasuk dari penyakit saraf)	Penyebab degeneratif misalnya pada kasus parkinson, gangguan karena benturan contoh kecelakaan lalu lintas, gangguan karena neoplasma contoh tumor otak, gangguan karena infeksi contoh meningitis dan gangguan pembuluh darah contoh stroke	

B. Penyakit *Stroke*

Tujuan : Materi wawancara ini bertujuan untuk mengetahui penyebab dan tingkat resiko pada penyebab penyakit *Stroke*.

No.	Pertanyaan	Jawaban	Checklist
B1	Bagaimana pengertian penyakit *Stroke*? (tujuan : untuk mengetahui pengertian dari penyakit *Stroke*)	Stroke adalah deficit neurologi yang timbul akibat adanya gangguan fungsi otak yang terjadinya secara mendadak yang terjadi lebih dari 24 jam yang disebabkan oleh gangguan pembuluh darah otak.	
B2	Faktor apa saja yang bisa menyebabkan seseorang terserang *Stroke*? (tujuan : untuk mendapat informasi mengenai faktor apa saja yang dapat menyebabkan seseorang terdeteksi *Stroke*)	Ada 2 faktor utama : yaitu faktor yang dapat diubah dan faktor yang tidak dapat diubah.	
B3	Apa saja faktor utama yang dapat memicu adanya penyakit *Stroke*? (tujuan : untuk mendapat informasi mengenai faktor utama yang dapat menyebabkan seseorang terdeteksi *Stroke*)	Dua faktor yang tersebut diatas adalah dua faktor utama, hanya saja yang lebih sering terdapat di masyarakat adalah hipertensi, DM, Merokok, dan dislipidemia.	
B4	Apa saja faktor pendukung yang dapat memicu adanya penyakit *Stroke*? (tujuan : untuk mendapat informasi mengenai faktor pendukung yang dapat menyebabkan seseorang terdeteksi *Stroke*)	Riwayat Keluarga, Aktivitas Fisik, Diet, dan Riwayat Fibrilasi Atrium.	
B5	Apa saja faktor pelengkap yang dapat memicu adanya penyakit *Stroke*? (tujuan : untuk mendapat informasi mengenai faktor pelengkap yang dapat menyebabkan seseorang terdeteksi *Stroke*)	yang dimaksud faktor pelengkap adalah faktor yang tidak terlalu sering menyebabkan stroke.	
B6	Bagaimana penggolongan tingkat resiko pada deteksi dini penyakit *Stroke*?	ada resiko rendah ,resiko sedang dan resiko tinggi	

B7	Terdapat faktor yang spesifik yang dapat menentukan tingkat resiko penyakit stroke?	Ada, pada tingkat resiko tinggi	
B8	Faktor apa yang menyebabkan tingkat resiko secara spesifik?	Pada faktor risiko riwayat penyakit stroke jika mempunyai riwayat maka pasien memiliki tingkat resiko tinggi terhadap penyakit stroke.	

C. Cara Diagnosa

Tujuan : Materi wawancara ini bertujuan untuk mengetahui cara diagnosa yang dilakukan oleh dokter untuk mendeteksi tingkat resiko seseorang terkena penyakit *Stroke*.

No.	Pertanyaan	Jawaban	Checklist
C1	Bagaimana prosedur mendeteksi tingkat resiko pada penyakit *Stroke*? (tujuan : untuk mengetahui tata cara pemeriksaan terhadap pasien)	1. Dilakukan pemeriksaan terhadap tekanan darah dan berat badan, serta tinggi badan. 2. Dilakukan eavaluasi klinis berdasarkan Framingham score atau *stroke risk card* 3. Bila masuk resiko sedang atau tinggi dilanjutkan dengan pemeriksaan *Carotis doppler* atau TCD dan bila ada kelainan dilanjutkan dengan pemeriksaan CT angiography atau DSA	
C2	Apakah ada prosedur pemeriksaan selain pemeriksaan melalui gejala klinis untuk mengetahui tingkat resiko terserang *Stroke*? (tujuan : untuk mengetahui apakah ada prosedur lain dalam mendeteksi tingkat resiko pada pasien selain melalui pemeriksaan gejala klinis.)	1. Tidak Ada	

C3	Apa saja gejala klinis yang mungkin timbul jika seseorang terdeteksi penyakit *Stroke* tingkat rendah? (tujuan : untuk mengetahui gejala klinis yang mungkin timbul pada penyakit *Stroke* tingkat rendah.)	Tanpa Gejala	
C4	Apa saja gejala klinis yang mungkin timbul jika seseorang terdeteksi penyakit *Stroke* tingkat sedang? (tujuan : untuk mengetahui gejala klinis yang mungkin timbul pada penyakit *Stroke* tingkat sedang.)	Tanpa Gejala	
C5	Apa saja gejala klinis yang mungkin timbul jika seseorang terdeteksi penyakit *Stroke* tingkat tinggi? (tujuan : untuk mengetahui gejala klinis yang mungkin timbul pada penyakit *Stroke* tingkat tinggi.)	Bisa mulai muncul gejala tergantung lokasi otak yang mulai terganggu, misalnya nggliyeng, merot yang hilang timbul, rasa kesemutan pada sisi tubuh yang juga hilang timbul	

TABEL NILAI KEPERCAYAAN

No.	Gejala	Nilai Kepercayaan (0-1)	Checklist
1.	Tekanan darah		
2.	Fibrilasi atrium		
3.	Merokok		
4.	Kolesterol		
5.	Diabetes		
6.	Aktifitas Fisik		
7.	Diet		
8.	Riwayat Keluarga		

TABEL ISTILAH GEJALA

No.	Jenis Gejala	Deskripsi Gejala	Checklist
1.	Tekanan darah	Tanpa Gejala.	
2.	Fibrilasi atrium	Gejala rasa tidak enak di dada,atau rasa seperti bedebar debar.	
3.	Merokok	Menghisap rokok baik pasif maupun aktif.	
4.	Kolesterol	Pemeriksaan Lab.	
5.	Diabetes	Pemeriksaan lab dan gejala klinis seperti lemas, cepat haus, cepat lapar, berat badan semakin menurun.	
6.	Aktifitas Fisik	Tidak pernah melakukan aktifitas fisik yang cukup.	
7.	Diet	Mengkonsumsi kebutuhan kalori lebih dari ketentuan umur dan berat badan.	
8.	Riwayat Keluarga	Ada riwayat keluarga pernah *Stroke* atau jantung koroner	

NILAI SKOR GEJALA

Gejala Tekanan Darah			Checklist
No.	Tingkat Gejala	Skor	
1	<140/90 mmHg atau lebih	3	
2	120-139/80/89 mmHg	2	
3	<120/80 mmHg	1	

Gejala Fibrilasi Atrium			Checklist
No.	Tingkat Gejala	Skor	
1	Ya	3	
2	Kadang-kadang	2	
3	Tidak	1	

Gejala Merokok			Checklist
No.	Tingkat Gejala	Skor	
1	Ya	3	
2	Kadang-kadang	2	
3	Tidak	1	

Gejala Kolesterol			Checklist
No.	Tingkat Gejala	Skor	
1	<240 mg/dl		
2	200-239 mg/dl		
3	<200 mg/dl		

Gejala Diabetes			Checklist
No.	Tingkat Gejala	Skor	
1	Ya		
2	Borderline		
3	Tidak		

Gejala Aktivitas Fisik			Checklist
No.	Tingkat Gejala	Skor	
1	Tidak Pernah Olahraga	3	
2	Kadang-kadang	2	
3	Olahraga Teratur	1	

Gejala Diet			Checklist
No.	Tingkat Gejala	Skor	
1	Obesitas		
2	Sedikit Obesitas		
3	Normal		

Gejala Riwayat Keluarga			Checklist
No.	Tingkat Gejala	Skor	
1	Ada	3	
2	Tidak Tahu	2	
3	Tidak Ada	1	

Malang, 9 Maret 2015
Dokter Spesialis Saraf RS Saiful Anwar
Malang

dr. Eko Ari Setijono Marhendraputro, SpS (K)

LAMPIRAN B DATA UJI

Data Uji

No.	Tekanan Darah	Diabetes	Riwayat Keluarga	Merokok	Kolesterol	Aktifitas Fisik	Diet	BB	TB	Riwayat Fibrilasi Atrium	Tingkat Resiko
1	80/60	180	Iya	Tidak	100	Rutin Olahraga	20.3	54	1.63	Beraturan	Tinggi
2	150/90	250	Tidak	Merokok	300	Tidak Pernah	32.7	89	1.65	Tidak Beraturan	Tinggi
3	160/100	300	Tidak Tahu	Merokok	310	Tidak Pernah	32.9	80	1.56	Tidak Beraturan	Tinggi
4	80/100	250	Tidak Tahu	Merokok	250	Tidak Pernah	23.3	59	1.59	Tidak Beraturan	Tinggi
5	90/100	205	Tidak Tahu	Merokok	280	Kadang	26.1	71	6.65	Tidak Tahu	Tinggi
6	130/100	210	Tidak	Merokok	246	Rutin Olahraga	25.2	59	1.53	Tidak Tahu	Tinggi
7	130/100	280	Tidak Tahu	Merokok	210	Kadang	27.9	74	1.63	Tidak Tahu	Tinggi
8	160/90	240	Tidak	Merokok	150	Rutin Olahraga	30.1	80	1.63	Tidak Beraturan	Tinggi
9	120/80	220	Tidak Tahu	Merokok	260	Tidak Pernah	31.1	90	1.7	Tidak Beraturan	Tinggi
10	110/90	180	Tidak Tahu	Kadang	220	Kadang	33.5	90	1.64	Tidak Tahu	Sedang
11	100/90	190	Tidak Tahu	Kadang	360	Tidak Pernah	33.6	86	1.6	Tidak Beraturan	Sedang
12	80/60	210	Tidak	Tidak	145	Tidak Pernah	32.7	88	1.64	Beraturan	Sedang
13	120/80	206	Tidak Tahu	Merokok	210	Kadang	24.6	65.3	1.63	Tidak Tahu	Sedang
14	100/90	180	Tidak Tahu	Kadang	330	Tidak Pernah	24.7	62.4	1.59	Tidak Tahu	Sedang
15	100/60	190	Tidak Tahu	Kadang	216	Tidak Pernah	30.1	79	1.62	Beraturan	Sedang
16	150/100	100	Tidak Tahu	Kadang	140	Tidak	25.2	64.5	1.6	Tidak Tahu	Sedang

						Pernah					
17	60/80	190	Tidak Tahu	Kadang	218	Rutin Olahraga	30.1	80	1.63	Beraturan	Sedang
18	130/80	120	Tidak	Kadang	120	Kadang	23.1	64.5	1.7	Tidak Tahu	Sedang
19	140/80	170	Tidak Tahu	Kadang	22	Kadang	32.2	87.6	1.65	Tidak Beraturan	Sedang
20	90/60	109	Tidak	Tidak	109	Rutin Olahraga	23.7	68.5	1.7	Tidak Beraturan	Rendah
21	100/60	140	Tidak	Merokok	116	Rutin Olahraga	21.8	57.9	1.63	Beraturan	Rendah
22	110/70	130	Tidak	Tidak	300	Rutin Olahraga	20.4	53	1.61	Beraturan	Rendah
23	110/70	300	Tidak	Tidak	100	Rutin Olahraga	33.9	101.5	1.73	Tidak Beraturan	Rendah
24	90/60	135	Tidak	Tidak	138	Tidak Pernah	23.1	60	1.61	Beraturan	Rendah
25	80/60	120	Tidak	Tidak	136	Kadang	24.2	60.5	1.58	Beraturan	Rendah
26	100/60	90	Tidak	Tidak	210	Tidak Pernah	22.0	55.6	1.59	Beraturan	Rendah
27	90/70	100	Tidak	Tidak	108	Rutin Olahraga	20.0	51.2	1.6	Tidak Tahu	Rendah
28	80/60	80	Tidak	Tidak	238	Rutin Olahraga	22.0	55	1.58	Tidak Tahu	Rendah
29	120/80	180	Tidak Tahu	Merokok	220	Tidak Pernah	25.7	70	1.65	Tidak Beraturan	Sedang
30	80/100	178	Tidak Tahu	Kadang	120	Tidak Pernah	20.0	51.2	1.6	Tidak Beraturan	Sedang

Malang, 9 Maret 2015
Dokter Spesialis Saraf RS Saiful Anwar
Malang

dr. Eko Ari Setijono Marhendraputro, SpS (K)

LAMPIRAN C DATA LATIH

Data Latih

No.	Tekanan Darah	Diabetes	Riwayat Keluarga	Merokok	Kolesterol	Aktifitas Fisik	Diet	BB	TB	Riwayat Fibrilasi Atrium	Tingkat Resiko
1	Rendah	120	Iya	Tidak	183	Rutin Olahraga	20.57	56	1.65	Beraturan	Tinggi
2	Tinggi	220	Tidak Tahu	Merokok	243	Tidak Pernah Olahraga	33.20	85	1.6	Tidak Beraturan	Tinggi
3	Tinggi	230	Tidak Tahu	Merokok	251	Kadang	32.44	73	1.5	Tidak Beraturan	Tinggi
4	Tinggi	210	Tidak Tahu	Merokok	241	Tidak Pernah Olahraga	32.18	85.5	1.63	Tidak Tahu	Tinggi
5	Tinggi	250	Tidak Tahu	Merokok	249	Kadang	26.13	73.5	1.64	Tidak Tahu	Tinggi
6	Tinggi	280	Tidak Ada	Merokok	205	Kadang	30.99	73.5	1.54	Tidak Beraturan	Tinggi
7	Tinggi	265	Tidak Tahu	Merokok	239	Tidak Pernah Olahraga	31.02	75.5	1.56	Tidak Beraturan	Tinggi
8	Tinggi	350	Tidak Ada	Merokok	247	Tidak Pernah Olahraga	34.06	90.5	1.63	Tidak Beraturan	Tinggi
9	Tinggi	335	Tidak Tahu	Merokok	196	Tidak Pernah Olahraga	31.40	85.5	1.65	Tidak Beraturan	Tinggi
10	Tinggi	256	Tidak Ada	Merokok	186	Tidak Pernah Olahraga	30.48	85.0	1.67	Tidak Beraturan	Tinggi
11	Tinggi	270	Tidak Tahu	Merokok	347	Tidak Pernah Olahraga	30.57	72.5	1.54	Tidak Tahu	Tinggi
12	Tinggi	298	Tidak Tahu	Merokok	248	Tidak Pernah Olahraga	30.44	76.0	1.58	Tidak Tahu	Tinggi
13	Tinggi	310	Tidak Ada	Merokok	278	Tidak Pernah Olahraga	30.86	83.0	1.64	Tidak Tahu	Tinggi
14	Tinggi	331	Tidak Ada	Merokok	280	Tidak Pernah Olahraga	34.16	93.0	1.65	Tidak Beraturan	Tinggi
15	Tinggi	296	Tidak Tahu	Merokok	271	Tidak Pernah Olahraga	31.46	92.0	1.71	Tidak Beraturan	Tinggi
16	Tinggi	305	Tidak Tahu	Merokok	326	Tidak Pernah	21.30	58.0	1.65	Tidak Beraturan	Tinggi

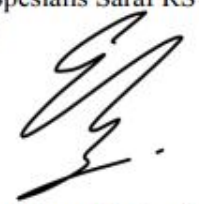
						Olahraga					
17	Tinggi		240	Tidak Tahu	Kadang	248	Kadang	33.40	85.5	1.6	Tidak Tahu	Sedang
18	Sedang		250	Tidak Tahu	Kadang	259	Kadang	33.24	90.5	1.65	Tidak Tahu	Sedang
19	Rendah		210	Tidak Tahu	Kadang	260	Kadang	34.25	91	1.63	Tidak Tahu	Sedang
20	Tinggi		180	Tidak Tahu	Kadang	274	Kadang	37.05	92.5	1.58	Tidak Tahu	Sedang
21	Sedang		190	Tidak Tahu	Kadang	275	Kadang	31.25	80	1.6	Tidak Tahu	Sedang
22	Rendah		165	Tidak Tahu	Kadang	287	Kadang	31.24	83	1.63	Tidak Tahu	Sedang
23	Tinggi		231	Tidak Tahu	Kadang	204	Kadang	33.35	93	1.67	Tidak Tahu	Sedang
24	Sedang		234	Tidak Tahu	Kadang	205	Kadang	31.89	90	1.68	Tidak Tahu	Sedang
25	Rendah		225	Tidak Tahu	Kadang	210	Kadang	38.14	94	1.57	Tidak Tahu	Sedang
26	Tinggi		182	Tidak Tahu	Kadang	208	Kadang	34.25	85.5	1.58	Tidak Tahu	Sedang
27	Sedang		186	Tidak Tahu	Kadang	219	Kadang	30.67	80.5	1.62	Tidak Tahu	Sedang
28	Rendah		196	Tidak Tahu	Kadang	228	Kadang	34.95	94	1.64	Tidak Tahu	Sedang
29	Tinggi		209	Tidak Tahu	Kadang	194	Kadang	35.26	96	1.65	Tidak Tahu	Sedang
30	Sedang		210	Tidak Tahu	Kadang	158	Kadang	30.12	81	1.64	Tidak Tahu	Sedang
31	Rendah		230	Tidak Tahu	Kadang	169	Kadang	31.45	80.5	1.6	Tidak Tahu	Sedang
32	Tinggi		180	Tidak Tahu	Kadang	184	Kadang	31.84	81.5	1.6	Tidak Tahu	Sedang
33	Sedang		189	Tidak Tahu	Kadang	191	Kadang	31.44	82.5	1.62	Tidak Tahu	Sedang
34	Rendah		176	Tidak Tahu	Kadang	149	Kadang	31.00	87.5	1.68	Tidak Tahu	Sedang
35	Tinggi		210	Tidak Tahu	Kadang	289	Kadang	26.53	74	1.67	Tidak Tahu	Sedang
36	Sedang		220	Tidak Tahu	Kadang	276	Kadang	26.40	74.5	1.68	Tidak Tahu	Sedang
37	Rendah		216	Tidak Tahu	Kadang	260	Kadang	29.41	72.5	1.57	Tidak Tahu	Sedang
38	Tinggi		189	Tidak Tahu	Kadang	278	Kadang	25.44	63.5	1.58	Tidak Tahu	Sedang
39	Sedang		198	Tidak Tahu	Kadang	283	Kadang	27.43	72	1.62	Tidak Tahu	Sedang
40	Rendah		179	Tidak Tahu	Kadang	279	Kadang	27.14	73	1.64	Tidak Tahu	Sedang
41	Tinggi		210	Tidak Tahu	Kadang	224	Kadang	25.34	69	1.65	Tidak Tahu	Sedang
42	Sedang		211	Tidak Tahu	Kadang	230	Kadang	25.78	74.5	1.7	Tidak Tahu	Sedang
43	Rendah		210	Tidak Tahu	Kadang	237	Kadang	30.27	77.5	1.6	Tidak Tahu	Sedang
44	Tinggi		188	Tidak Tahu	Kadang	231	Kadang	28.83	78.5	1.65	Tidak Tahu	Sedang
45	Sedang		187	Tidak Tahu	Kadang	232	Kadang	29.92	79.5	1.63	Tidak Tahu	Sedang

46	Rendah		169	Tidak Tahu	Kadang	212	Kadang	29.08	72.6	1.58	Tidak Tahu	Sedang
47	Tinggi		221	Tidak Tahu	Kadang	167	Kadang	29.10	74.5	1.6	Tidak Tahu	Sedang
48	Sedang		221	Tidak Tahu	Kadang	179	Kadang	28.13	72	1.6	Tidak Tahu	Sedang
49	Rendah		210	Tidak Tahu	Kadang	183	Kadang	26.08	71	1.65	Tidak Tahu	Sedang
50	Tinggi		179	Tidak Tahu	Kadang	191	Kadang	25.97	69	1.63	Tidak Tahu	Sedang
51	Sedang		188	Tidak Tahu	Kadang	198	Kadang	27.24	68	1.58	Tidak Tahu	Sedang
52	Rendah		180	Tidak Tahu	Kadang	193	Kadang	27.73	71	1.6	Tidak Tahu	Sedang
53	Tinggi		232	Tidak Tahu	Kadang	247	Kadang	30.49	81	1.63	Beraturan	Sedang
54	Sedang		213	Tidak Tahu	Kadang	251	Kadang	30.12	84	1.67	Beraturan	Sedang
55	Rendah		215	Tidak Tahu	Kadang	259	Kadang	31.36	88.5	1.68	Beraturan	Sedang
56	Tinggi		182	Tidak Tahu	Kadang	276	Kadang	35.09	86.5	1.57	Beraturan	Sedang
57	Sedang		193	Tidak Tahu	Kadang	285	Kadang	31.05	79.5	1.6	Beraturan	Sedang
58	Rendah		195	Tidak Tahu	Kadang	260	Kadang	31.43	83.5	1.63	Beraturan	Sedang
59	Tinggi		230	Tidak Tahu	Kadang	210	Kadang	33.05	82.5	1.58	Beraturan	Sedang
60	Sedang		236	Tidak Tahu	Kadang	211	Kadang	31.64	81	1.6	Beraturan	Sedang
61	Rendah		241	Tidak Tahu	Kadang	213	Kadang	32.81	84	1.6	Beraturan	Sedang
62	Tinggi		172	Tidak Tahu	Kadang	237	Kadang	30.67	83.5	1.65	Beraturan	Sedang
63	Sedang		183	Tidak Tahu	Kadang	231	Kadang	34.18	87.5	1.6	Beraturan	Sedang
64	Rendah		192	Tidak Tahu	Kadang	236	Kadang	30.29	79.5	1.62	Beraturan	Sedang
65	Tinggi		211	Tidak Tahu	Kadang	190	Kadang	30.30	81.5	1.64	Beraturan	Sedang
66	Sedang		228	Tidak Tahu	Kadang	168	Kadang	31.45	80.5	1.6	Beraturan	Sedang
67	Rendah		224	Tidak Tahu	Kadang	170	Kadang	32.69	89	1.65	Beraturan	Sedang
68	Tinggi		168	Tidak Tahu	Kadang	174	Kadang	34.06	90.5	1.63	Beraturan	Sedang
69	Sedang		176	Tidak Tahu	Kadang	190	Kadang	37.45	93.5	1.58	Beraturan	Sedang
70	Rendah		183	Tidak Tahu	Kadang	150	Kadang	35.74	91.5	1.6	Beraturan	Sedang
71	Tinggi		233	Tidak Tahu	Kadang	248	Kadang	26.53	70.5	1.63	Beraturan	Sedang
72	Sedang		245	Tidak Tahu	Kadang	242	Kadang	25.64	71.5	1.67	Beraturan	Sedang
73	Rendah		223	Tidak Tahu	Kadang	289	Kadang	27.81	78.5	1.68	Beraturan	Sedang
74	Tinggi		164	Tidak Tahu	Kadang	260	Kadang	27.18	67	1.57	Beraturan	Sedang
75	Sedang		175	Tidak Tahu	Kadang	264	Kadang	29.64	74	1.58	Beraturan	Sedang

76	Rendah		180	Tidak Tahu	Kadang	273	Kadang	26.96	72.5	1.64	Beraturan	Sedang
77	Tinggi		247	Tidak Tahu	Kadang	231	Kadang	29.10	74.5	1.6	Beraturan	Sedang
78	Sedang		256	Tidak Tahu	Kadang	238	Kadang	28.83	78.5	1.65	Beraturan	Sedang
79	Rendah		217	Tidak Tahu	Kadang	230	Kadang	29.92	79.5	1.63	Beraturan	Sedang
80	Tinggi		180	Tidak Tahu	Kadang	219	Kadang	29.44	73.5	1.58	Beraturan	Sedang
81	Sedang		186	Tidak Tahu	Kadang	211	Kadang	28.13	72	1.6	Beraturan	Sedang
82	Rendah		169	Tidak Tahu	Kadang	214	Kadang	25.97	69	1.63	Beraturan	Sedang
83	Tinggi		211	Tidak Tahu	Kadang	153	Kadang	26.64	77	1.7	Beraturan	Sedang
84	Sedang		222	Tidak Tahu	Kadang	169	Kadang	28.83	78.5	1.65	Beraturan	Sedang
85	Rendah		226	Tidak Tahu	Kadang	189	Kadang	25.53	69.5	1.65	Beraturan	Sedang
86	Tinggi		196	Tidak Tahu	Kadang	193	Kadang	30.30	80.5	1.63	Beraturan	Sedang
87	Sedang		183	Tidak Tahu	Kadang	192	Kadang	30.44	76	1.58	Beraturan	Sedang
88	Rendah		163	Tidak Tahu	Kadang	190	Kadang	27.15	69.5	1.6	Beraturan	Sedang
89	Sedang		161	Tidak Ada	Kadang	232	Rutin Olahraga	28.23	75	1.63	Beraturan	Rendah
90	Rendah		179	Tidak Ada	Kadang	222	Rutin Olahraga	28.51	79.5	1.67	Beraturan	Rendah
91	Sedang		124	Tidak Ada	Kadang	213	Rutin Olahraga	25.43	68.4	1.64	Beraturan	Rendah
92	Rendah		146	Tidak Ada	Kadang	209	Rutin Olahraga	26.91	63	1.53	Beraturan	Rendah
93	Sedang		181	Tidak Ada	Kadang	231	Rutin Olahraga	25.15	62	1.57	Beraturan	Rendah
94	Rendah		192	Tidak Ada	Kadang	238	Rutin Olahraga	26.90	68	1.59	Beraturan	Rendah
95	Sedang		129	Tidak Ada	Kadang	209	Rutin Olahraga	25.41	67.5	1.63	Beraturan	Rendah
96	Rendah		137	Tidak Ada	Kadang	217	Rutin Olahraga	27.64	69	1.58	Beraturan	Rendah
97	Sedang		168	Tidak Ada	Kadang	189	Rutin Olahraga	28.20	69.5	1.57	Beraturan	Rendah
98	Rendah		172	Tidak Ada	Kadang	192	Rutin Olahraga	28.84	67.5	1.53	Beraturan	Rendah
99	Sedang		149	Tidak Ada	Kadang	194	Rutin Olahraga	25.36	62.5	1.57	Beraturan	Rendah
100	Rendah		138	Tidak Ada	Kadang	186	Rutin Olahraga	27.99	79	1.68	Beraturan	Rendah
101	Sedang		172	Tidak Ada	Tidak	164	Rutin Olahraga	25.22	67	1.63	Beraturan	Rendah
102	Rendah		180	Tidak Ada	Tidak	180	Rutin Olahraga	27.27	63	1.52	Beraturan	Rendah
103	Sedang		139	Tidak Ada	Tidak	158	Rutin Olahraga	28.41	66.5	1.53	Beraturan	Rendah
104	Rendah		149	Tidak Ada	Tidak	167	Rutin Olahraga	29.21	72	1.57	Beraturan	Rendah
105	Sedang		169	Tidak Ada	Kadang	210	Rutin Olahraga	28.08	71	1.59	Beraturan	Rendah

106	Rendah		170	Tidak Ada	Kadang	205	Rutin Olahraga	21.08	56	1.63	Beraturan	Rendah
107	Sedang		151	Tidak Ada	Kadang	217	Rutin Olahraga	24.22	62	1.6	Beraturan	Rendah
108	Rendah		159	Tidak Ada	Kadang	226	Rutin Olahraga	20.20	55	1.65	Beraturan	Rendah
109	Sedang		173	Tidak Ada	Tidak	213	Rutin Olahraga	19.95	53	1.63	Beraturan	Rendah
110	Rendah		183	Tidak Ada	Tidak	217	Rutin Olahraga	19.23	48	1.58	Beraturan	Rendah
111	Sedang		152	Tidak Ada	Tidak	236	Rutin Olahraga	21.88	56	1.6	Beraturan	Rendah
112	Rendah		158	Tidak Ada	Tidak	228	Rutin Olahraga	21.48	55	1.6	Beraturan	Rendah
113	Sedang		170	Tidak Ada	Kadang	180	Rutin Olahraga	19.83	54	1.65	Beraturan	Rendah
114	Rendah		189	Tidak Ada	Kadang	176	Rutin Olahraga	20.51	54.5	1.63	Beraturan	Rendah
115	Sedang		130	Tidak Ada	Kadang	179	Rutin Olahraga	21.53	56.5	1.62	Beraturan	Rendah
116	Rendah		142	Tidak Ada	Kadang	180	Rutin Olahraga	21.76	55	1.59	Beraturan	Rendah
117	Sedang		186	Tidak Ada	Tidak	193	Rutin Olahraga	18.86	54.5	1.7	Beraturan	Rendah
118	Rendah		193	Tidak Ada	Tidak	192	Rutin Olahraga	17.22	43	1.58	Beraturan	Rendah
119	Sedang		123	Tidak Ada	Tidak	190	Rutin Olahraga	15.57	45	1.7	Beraturan	Rendah
120	Rendah		143	Tidak Ada	Tidak	195	Rutin Olahraga	18.93	56	1.72	Beraturan	Rendah

Malang, 9 Maret 2015
Dokter Spesialis Saraf RS Saiful Anwar
Malang



dr. Eko Ari Setijono Marhendraputro, SpS (K)