



**PERBANDINGAN JARINGAN SYARAF TIRUAN DAN
NAIVE BAYES DALAM DETEKSI SESEORANG
TERKENA PENYAKIT STROKE**

skripsi

disajikan sebagai salah satu syarat
untuk memperoleh gelar Sarjana Sains
Program Studi Matematika

oleh

Istatik Rohmana

4111410022

JURUSAN MATEMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS NEGERI SEMARANG

2014

PERNYATAAN

Saya menyatakan bahwa skripsi ini bebas plagiat, dan apabila di kemudian hari terbukti terdapat plagiat dalam skripsi ini, maka saya bersedia menerima sanksi sesuai ketentuan peraturan perundang-undangan.

Semarang, Agustus 2014



6000
Istati Rohmana
4111410022

PENGESAHAN

Skripsi yang berjudul:

Perbandingan Jaringan Syaraf Tiruan dan Naive Bayes dalam Deteksi
Seseorang Terkena Penyakit Stroke

disusun oleh

Istatik Rohmana

4111410022

Telah dipertahankan dihadapan sidang Panitia Ujian Skripsi Fakultas Matematika
dan Ilmu Pengetahuan Alam, Universitas Negeri Semarang pada tanggal 26
Agustus 2014.

Panitia:



Ketua
Prof. Dr. Wiyanto, M.Si.
196310121988031001

Penguji I

Much Aziz Muslim, S.Kom., M.Kom.
197404202008121001

Anggota Penguji/
Pembimbing

Riza Arifudin, S.Pd., M.Cs.
198005252005011001

Sekretaris

Drs. Arief Agoestanto, M.Si.
196807221993031005

Penguji II

Endang Sugiharti, S.Si., M.Kom.
197401071999032001

MOTTO DAN PERSEMBAHAN

Motto

- ❏ Katakanlah: Adakah sama orang-orang yang mengetahui dengan orang-orang yang tidak mengetahui? Sesungguhnya orang-orang yang barokallah yang dapat menerima pelajaran (Az-Zumar: 9)
- ❏ Ajari aku menggunakan pena, akan kutulis gemericik air, udara dingin. Kabut senja, sampai daun gugur (Yus R. Ismail)
- ❏ The aim of education should be to teach us rather how to think, than what to think-rather to improve our minds, so as to enable us to think for ourselves, than to load the memory with thoughts of other men (Bill Beattie)

Persembahan

Skripsi ini kupersembahkan untuk:

- ❏ Bapakku Syuba'i As dan Ibuku Sumilah atas kasih sayang, dukungan yang diberikan, dan doa yang tidak ada hentinya dipanjatkan.
- ❏ Kakak aku Isbat Uzzin Nadhori dan Irwan Khoiril Anam yang senantiasa membantu, menyemangati, dan menghiburku.
- ❏ Sahabatku Lollipop dan Proxima atas semangatnya.
- ❏ Teman-teman kos atas dukungan dan canda tawanya.
- ❏ Almamaterku Universitas Negeri Semarang.

KATA PENGANTAR

Syukur Alhamdulillah penulis panjatkan kehadiran Allah SWT atas segala rahmat, taufik dan hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul “**Perbandingan Jaringan Syaraf Tiruan dan Naive Bayes dalam Deteksi Seseorang Terkena Penyakit Stroke**”.

Dalam penulisan skripsi ini penulis mendapat banyak bantuan dari berbagai pihak. Untuk itu perkenankanlah penulis mengucapkan terima kasih kepada:

1. Prof. Dr. Fathur Rokhman, M. Hum., Rektor Universitas Negeri Semarang.
2. Prof. Dr. Wiyanto, M.Si., Dekan FMIPA Universitas Negeri Semarang.
3. Drs. Arief Agoestanto, M.Si., Ketua Jurusan Matematika FMIPA Universitas Negeri Semarang.
4. Dra. Kristina Wijayanti, M.Si., Ketua Prodi Matematika FMIPA Universitas Negeri Semarang.
5. Riza Arifudin, S.Pd., M.Cs., Dosen pembimbing yang telah memberikan bimbingan, motivasi, dan semangat.
6. Ibu, Bapak, dan Kakak-kakakku tercinta yang selalu memberikan doa, semangat serta dukungan baik secara moral maupun spiritual.
7. Seluruh pihak di RS TUGUREJO Semarang yang telah memberikan izin untuk penelitian serta senantiasa membantu dalam hal pengambilan data.
8. Segenap civitas matematika Unnes, khususnya Bapak dan Ibu dosen yang telah memberikan ilmunya dengan tulus.

9. Teman-teman lollipop, proxima, dankkn ngesop yang selalu menghibur dan telah berjuang bersama dalam suka dan duka.

10. Semua pihak yang telah membantu terselesaikannya penulisan skripsi ini.

Akhir penulisan berharap semoga skripsi ini dapat bermanfaat bagi para pembaca.

Semarang, Agustus 2014

Penulis

ABSTRAK

Rohmana, Istatik. 2014. *Perbandingan Jaringan Syaraf Tiruan dan Naive Bayes dalam Deteksi Seseorang Terkena Penyakit Stroke*. Skripsi, Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Negeri Semarang. Pembimbing Riza Arifudin, S.Pd., M.Cs.

Kata kunci : Stroke, Jaringan Syaraf Tiruan, Naive Bayes, MATLAB.

Jumlah penderita stroke menunjukkan peningkatan di setiap tahunnya. Hal ini karena pola kesehatan seseorang yang tidak dijaga. Akan dibuat sebuah aplikasidengan menggunakan *software* MATLAB. Metode yang baik dalam mesin pembelajaran berdasarkan data *training* adalah metode Jaringan Syaraf Tiruan dan Naive Bayes, dengan menggunakan variabel data faktor gejala penyakit strokedigunakan sebagai data *training* dalam proses pembelajaran dari sistem yang akan dibuat untuk menentukan *Suspectstroke* atau tidak.

Permasalahan yang diangkat dalam penelitian ini adalah (1) Bagaimana mengimplementasikan metode Jaringan Syaraf Tiruan dan metode Naive Bayes guna proses mendeteksi seseorang terkena stroke atau tidak? (2) Metode manakah antara Jaringan Syaraf Tiruan dan Naive Bayes yang memiliki tingkat keakuratan terbaik untuk mengolah data *training* pada permasalahan di atas?

Adapun tujuan dari penelitian ini: (1) Membangun sebuah sistem untuk memprediksi kemungkinan seseorang *Suspectstroke* atau tidak berkaitan dengan *inputan* yang dimasukkan pasien. (2) Membandingkan keakuratan dari dua metode yang dipakai dalam aplikasi yang dibuat dengan menghitung *error* dari *output* percobaan dengan data *training* sendiri sebagai *inputan* pada metode.

Metode yang digunakan untuk menganalisis permasalahan dalam penelitian ini adalah dengan studi pustaka, perumusan masalah, mengumpulkan dan mengolah data-data penelitian, merancang sistem, membuat sistem, menguji sistem, dan penarikan kesimpulan.

Peneletian ini menghasilkan simpulan yaitu : (1) Perbandingan persentase dari kedua metode yaitu untuk Jaringan Syaraf Tiruan diperoleh persentase keberhasilan sebesar 71,11 persen, sedangkan dalam Naive Bayes diperoleh hasil persentase sebesar 80,555 persen. (2) Naive Bayes lebih akuratdalam hal pengambilan keputusan data baru, namun Jaringan Syaraf Tiruan memiliki teknik yang lebih bagus dibandingkan dengan Naive Bayes. Jaringan Syaraf Tiruan mempunyai karakteristik yang adaptif yaitu belajar dari data sebelumnya, sedangkan pada Naive Bayes menggunakan probabilitas dari data *input* untuk menentukan *Suspect* atau tidak.

Saran dari penelitian ini adalah (1) penggunaan metode lain perlu dilakukan dalam penanganan masalah yang sama agar dapat menilai metode mana yang paling cocok dengan permasalahan yang ada. (2) Untuk penelitian selanjutnya diharapkan adanya variabel *input* yang lebih rinci sebagai data uji dan data target. (3) Perlu ada penelitian dengan masalah yang sama dengan menggunakan variabel satu jenis data saja agar mengetahui perbedaan tingkat akuratan.

DAFTAR ISI

	Halaman
KATAPENGANTAR.....	v
ABSTRAK.....	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xiv
DAFTAR GAMBAR.....	xvii
DAFTAR LAMPIRAN.....	xix
BAB	
I PENDAHULUAN	
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah.....	5
1.3 Batasan Masalah.....	6
1.4 Tujuan dan Manfaat	
1.4.1 Tujuan Penulisan.....	6
1.4.2 Manfaat Penulisan.....	7
1.5 Sistematika Penulisan	
1.5.1 Bagian Awal.....	8
1.5.2 Bagian Pokok.....	8
1.5.3 Bagian Akhir.....	9
II LANDASAN TEORI	
2.1 Stroke atau <i>Kardioserebrovaskuler</i> (Gangguan Pembuluh Darah Otak)....	10

2.1.1	DefinisiStroke.....	10
2.1.2	Faktor Risiko.....	11
2.1.2.1	FaktorRisiko yang tidak bisa dirubah.....	11
2.1.2.2	FaktorRisiko yang dapat dirubah (diperbaiki).....	12
2.1.2.3	GejalaStroke.....	12
2.2	Kecerdasan Buatan (<i>Artifical Intelligent</i>).....	13
2.2.1	Konsep Penelitian dalam Kecerdasan Buatan.....	14
2.2.2	Domain Penelitian dalam Kecerdasan Buatan.....	15
2.3	Penyiapan Data (<i>Preprosesing</i>).....	15
2.3.1	Penanganan Terhadap Data Yang Hilang (<i>Missing Data</i>).....	15
2.4	Transformasi Data.....	17
2.5	Pengenalan Pola.....	19
2.5.1	Deteksi.....	19
2.5.2	Klasifikasi.....	20
2.5.3	Pengenalan.....	20
2.6	Jaringan Syaraf Tiruan.....	20
2.6.1	Inspirasi Biologi.....	20
2.6.2	Sejarah Jaringan Syaraf Tiruan.....	21
2.6.3	Definisi Jaringan Syaraf Tiruan.....	22
2.6.4	Arsitektur Jaringan.....	26
2.6.4.1	Jaringan Layar Tunggal.....	26
2.6.4.2	Jaringan Layar Jamak.....	27
2.6.4.3	Jaringan <i>Reccurent</i>	28

2.6.5	<i>Backpropagation</i>	28
2.6.6	Fungsi Aktivasi.....	30
2.6.6.1	Fungsi Sigmoid Biner.....	31
2.6.6.2	Fungsi Sigmoid Bipolar.....	31
2.6.6.3	Fungsi Tangen Hiperbolik.....	32
2.6.7	Pelatihan Standart <i>Backpropagation</i>	33
2.6.7.1	Fase 1, yaitu propagasi maju.....	33
2.6.7.2	Fase 1, yaitu propagasi maju.....	33
2.6.7.3	Fase 3, yaitu perubahan bobot.....	33
2.6.8	Algoritma Pelatihan.....	34
2.6.9	Membuat Aplikasi untuk <i>Backpropagation Learning</i>	40
2.7	Metode Naive Bayes.....	44
2.7.1	Teorema Bayes (soebanar, 2008).....	44
2.7.2	Naive Bayes <i>Classifier</i>	46
2.7.3	Keunggulan dan Kelemahan Metode Bayesian.....	49
2.7.4	Pemanfaatan Metode Bayesian.....	50
2.7.5	Penerapan Metode Bayes.....	51
2.8	<i>Matrix Laboratory</i> (Matlab).....	52
2.8.1	Jendela-jendela pada Matlab.....	55
2.8.1.1	<i>Command Window/Editor</i>	55
2.8.1.2	<i>Current Directory</i>	55
2.8.1.3	<i>Command History</i>	55
2.8.1.4	<i>Editor/ Debugger (Editor M-File/ Pencarian Kesalahan)</i> ..	56

2.8.1.5 <i>Workspace</i>	56
2.8.2 GUIDE Matlab.....	56
2.8.2.1 Keunggulan GUIDE Matlab.....	57
METODE PENELITIAN	
3.1 Studi Pustaka.....	58
3.2 Perumusan Masalah.....	58
3.3 Pemecahan Masalah.....	59
3.3.1 Pengumpulan Data dan Pengolahan Data.....	59
3.3.2 Perancangan, Pembuatan, Pengujian, dan Perbaikan Sistem.....	60
3.3.2.1 Perancangan Sistem.....	60
3.3.2.2 Pembuatan Sistem.....	62
3.3.2.2.1 Rancangan <i>Input</i>	63
3.3.2.2.2 Metode Jaringan Syaraf Tiruan.....	63
3.3.2.2.3 Metode Naive Bayes.....	64
3.3.2.2.4 DFD (<i>Data Flow Diagram</i>).....	66
3.3.2.3 Pengujian Sistem.....	69
3.3.2.4 Perbaikan.....	69
3.4 Penarikan Kesimpulan.....	69
HASIL DAN PEMBAHASAN	
4.1 Hasil Penelitian.....	70
4.1.1 Tahap Pengambilan Data.....	70
4.1.2 Pendefisian <i>Input</i> dan Target.....	70
4.1.2.1 Pendefisian <i>Input</i>	71

4.1.2.2 Penetapan Target.....	72
4.2 Tahap Pengolahan Data.....	75
4.3 Perancangan Desain Sistem.....	75
4.3.1 Tampilan Halaman Utama.....	75
4.3.2 Tampilan Form <i>Inputan</i> Deteksi.....	77
4.4 Tahap Implementasi Sistem.....	80
4.4.1 Tampilan Form Halaman Utama.....	80
4.4.2 Tampilan Form <i>Inputan</i> Deteksi.....	81
4.5 Analisis Jaringan Syaraf Tiruan.....	86
4.5.1 Penyiapan data untuk Jaringan Syaraf Tiruan.....	86
4.5.2 Tahap Perancangan Sistem.....	87
4.5.2.1 Perancangan Sistem Pelatihan.....	87
4.5.2.2 Perancangan Arsitektur Jaringan Syaraf Tiruan.....	88
4.5.2.2.1 Bobot dan Bias.....	88
4.5.2.2.2 Jumlah Neuron pada Lapisan Tersembunyi.....	88
4.5.2.2.3 <i>Error Goal</i> (Kinerja Tujuan).....	89
4.5.2.2.4 <i>Learning Rate</i> (Laju Pembelajaran).....	89
4.5.2.2.5 Fungsi Aktivasi.....	90
4.5.2.2.6 <i>Maksimum Epoch</i>	92
4.5.3 Hasil Arsitektur Jaringan Syaraf Tiruan <i>Backpropagation</i>	94
4.5.4 Hasil Kerja Sistem Jaringan Syaraf Tiruan.....	94

4.6	Naive	
	Bayes.....	98
4.6.1	Tahap perancangan Sistem.....	98
4.6.1.1	Perancangan Sistem Pelatihan.....	99
4.6.2	Proses Perhitungan Naive Bayes.....	99
4.6.3	Hasil Kerja Sistem Naive Bayes.....	109
4.7	Hasil Perbandingan Jaringan Syaraf Tiruan dan Metode Naive Bayes..	113
4.7.1	Hasil Analisis Faktor Risiko yang Berpengaruh.....	114
4.7.1.1	Hasil Analisis Faktor Risiko dengan Jenis Data Opsional yang Berpengaruh.....	114
4.7.1.2	Hasil Analisis Faktor Risiko dengan Jenis Data Numerik yang Berpengaruh.....	116
PENUTUP		
5.1	Simpulan.....	122
5.2	Saran.....	123
DAFTAR PUSTAKA.....		124
Lampiran.....		126

DAFTAR TABEL

Tabel	Halaman
Tabel 2.1 Klasifikasi Tekanan Darah.....	12
Tabel 2.2 Tabel Perbandingan Nilai.....	17
Tabel 2.3 Tabel Merangking Pilihan tiap faktor.....	18
Tabel 2.4 Tabel Normalisasi.....	19
Tabel 2.5 Tabel Contoh Data Naive Bayes.....	51
Tabel 3.1 Tabel Contoh Data Masukan.....	63
Tabel 4.1 Data Pelatihan Pasien.....	75
Tabel 4.2 Data Pengujian Pasien.....	76
Tabel 4.3 Tabel Contoh Hasil Normalisasi.....	87
Tabel 4.4 Variansi <i>learning Rate</i> dan Banyaknya <i>Neuron Hidden Layer</i>	90
Tabel 4.5 Analisis Hasil Variasi Arsitektur Jaringan.....	93
Tabel 4.6 Hasil Uji Jaringan Syaraf Tiruan 20 Data Pengujian.....	94
Tabel 4.7 Hasil Uji Jaringan Syaraf Tiruan 30 Data Pengujian.....	95
Tabel 4.8 Hasil Uji Jaringan Syaraf Tiruan 40 Data Pengujian.....	96

Tabel 4.9 Jenis Data Pada Variabel gejala.....	100
Tabel 4.10 Perhitungan <i>Mean</i> dan Standart Deviasi pada Data Numerik.....	100
Tabel 4.11 Probabilitas Jenis kelamin.....	101
Tabel 4.12 Probabilitas Hipertensi.....	101
Tabel 4.13 Probabilitas Diabetes Militus.....	101
Tabel 4.14 Probabilitas Kesadaran.....	102
Tabel 4.15 Probabilitas Pusing.....	102
Tabel 4.16 Probabilitas Mual.....	102
Tabel 4.17 Probabilitas Kesemutan.....	104
Tabel 4.18 Probabilitas Pernah Stroke.....	102
Tabel 4.19 Tabel Hasil Uji Naive Bayes 20 Data Pengujian.....	109
Tabel 4.20 Tabel Hasil Uji Naive Bayes 30 Data Pengujian.....	110
Tabel 4.21 Tabel Hasil Uji Naive Bayes 40 Data Pengujian.....	111
Tabel 4.22 Hasil Perbandingan Jaringan Syaraf Tiruan dan Naive Bayes.....	114
Tabel 4.23 <i>Inputan</i> Pengujian dengan 1 Faktor Risiko.....	115
Tabel 4.24 Hasil pengujian Faktor Variabel Opsional yang Paling Berpengaruh	116
Tabel 4.25 <i>Inputan</i> Pengujian pada Umur.....	117

Tabel 4.26 Hasil pengujian Faktor Variabel Numerik Umur yang Paling Berpengaruh.....	117
Tabel 4.27 <i>Inputan</i> Pengujian pada Nadi.....	118
Tabel 4.28 Hasil pengujian Faktor Variabel Numerik Nadi yang Paling Berpengaruh.....	119
Tabel 4.29 <i>Inputan</i> Pengujian pada Kolesterol.....	119
Tabel 4.30 Hasil pengujian Faktor Variabel Numerik Kolesterol yang Paling Berpengaruh.....	120

DAFTAR GAMBAR

Gambar	halaman
Gambar 2.1 Susunan Saraf Manusia.....	21
Gambar 2.2 Sebuah Sel Syaraf Tiruan.....	25
Gambar 2.3 Jaringan <i>Layer</i> Tunggal.....	26
Gambar 2.4 Jaringan <i>Layer</i> Jamak.....	27
Gambar 2.5 Jaringan <i>Recurrent</i>	28
Gambar 2.6 Arsitektur <i>Backpropagation</i>	30
Gambar 2.7 Fungsi Sigmoid Biner dengan <i>range</i> (0,1).....	31
Gambar 2.8 Fungsi Sigmoid Bipolar dengan <i>Range</i> (-1,1).....	32
Gambar 2.9 Struktur Jaringan Syaraf Tiruan untuk aplikasi.....	36
Gambar 2.10 Membuat GUI.....	40
Gambar 2.11 Hasil Rancangan GUI untuk Pelatihan.....	41
Gambar 2.12 Menyimpan GUI.....	42
Gambar 2.13 Proses Pelatihan JST.....	44
Gambar 2.14 Teorema Bayes.....	45
Gambar 3.1 Diagram Rancangan Sistem.....	62
Gambar 3.2 Diagram Alur Jaringan Syaraf Tiruan.....	64
Gambar 3.4 Diagram Alur Naive Bayes.....	66
Gambar 3.5 Diagram Konteks (DFD level 0).....	67
Gambar 3.6 <i>Data Flow Diagram</i> level 1.....	68

Gambar 3.7 <i>Data Flow Diagram</i> level 2 metode Jaringan Syaraf Tiruan.....	68
Gambar 3.8 <i>Data Flow Diagram</i> level 2 metode Niave Bayes.....	69
Gambar 4.1 Rancangan Halaman Utama.....	76
Gambar 4.2 Rancangan Halaman Petunjuk Penggunaan.....	76
Gambar 4.3 Desain Tampilan Halaman Utama.....	77
Gambar 4.4 Desain Tampilan Petunjuk Penggunaan.....	77
Gambar 4.5 Rancangan Tampilan Form <i>Inputan</i> Deteksi.....	78
Gambar 4.6 Desain Tampilan Form Deteksi Penyakit Stroke.....	79
Gambar 4.7 Form Halaman Utama.....	80
Gambar 4.8 Form Petunjuk Penggunaan.....	80
Gambar 4.9 Tampilan <i>Form Inputan</i> Deteksi.....	81
Gambar 4.10 Arsitektur Jaringan dengan 10 <i>Neuron Hidden</i>	91

LAMPIRAN

Lampiran	Halaman
Desain Tampilan <i>Interface</i> untuk Halaman Utama.....	126
Kode Program untuk halaman Depan.....	128
Desain Tampilan Program Deteksi Penyakit Stroke.....	129
Kode Program Deteksi Penyakit Stroke.....	131
Kode Jaringan Syaraf Tiruan.....	153
Kode Naive Bayes.....	155
Matriks Data Pelatihan.....	156
Matriks Data Pengujian.....	161
Tabel Pelatihan Pasien.....	163
Tabel Pengujian Pasien.....	167
Hasil Normalisasi Data Numerik Pada Data Pelatihan.....	169
Perhitungan Probabilitas Naive Bayes.....	175
Hasil Analisis Sistem.....	179
Contoh Program Saat Dijalankan.....	184
Data Pasien <i>Suspect</i> Penyakit Stroke Di RS. Tugurejo Semarang.....	186
Data Pasien Tidak <i>Suspect</i> Penyakit Stroke Di RS. Tugurejo Semarang.....	189
Surat Ketetapan Dosen Pembimbing.....	190
Surat Ijin Penelitian dari Fakultas.....	191

Surat Ijin Penelitian dari RS TUGUREJO Semarang.....	192
--	-----

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kemajuan peradaban manusia sudah semakin berkembang pesat disegala bidang kehidupan. Ilmu pengetahuan dan teknologi dewasa ini menjadi bagian yang tidak terpisahkan dari kehidupan masyarakat modern. Kesibukan yang luar biasa terutama di kota besar membuat manusia terkadang lalai terhadap kesehatan tubuhnya. Pola makan tidak teratur, kurang olahraga, jam kerja berlebihan, dan konsumsi makanan cepat saji sudah menjadi kebiasaan lazim yang berpotensi menimbulkan serangan stroke.

Beberapa dasa warsa yang lalu, stroke identik sebagai penyakit kaum manula, namun kini banyak kaum muda yang mendapat serangan stroke. Tidak heran jika stroke semakin sering menjadi bahan perbincangan di kalangan masyarakat. Menurut Dinas Kesehatan Provinsi Jawa Tengah (2006:42), stroke adalah suatu penyakit menurunnya fungsi syaraf secara akut yang disebabkan oleh gangguan pembuluh darah otak, terjadi secara mendadak dan cepat menimbulkan gejala dan tanda sesuai dengan daerah otak yang terganggu. Stroke disebabkan oleh kurangnya aliran darah yang mengalir ke otak, atau terkadang menyebabkan pendarahan di otak. Jadi stroke merupakan masalah medik yang sering dijumpai, gangguan neurologik ini sering terjadi secara mendadak dan tidak jarang menyebabkan kematian.

Berdasarkan data WHO (2010) setiap tahunnya terdapat 15 juta orang diseluruh dunia menderita stroke. Diantaranya ditemukan jumlah kematian sebanyak 5 juta orang dan 5 juta orang lainnya mengalami kecacatan yang permanen. Penyakit stroke telah menjadi masalah kesehatan yang menjadi penyebab utama kecacatan pada usia dewasa dan merupakan salah satu penyebab terbanyak di dunia. Stroke menduduki urutan ketiga sebagai penyebab utama kematian setelah penyakit jantung koroner dan kanker di negara-negara berkembang. Negara berkembang juga menyumbang 85,5 persen dari total kematian akibat stroke di seluruh dunia. Dua pertiga penderita stroke terjadi di negara-negara yang sedang berkembang. Terdapat sekitar 13 juta korban stroke baru setiap tahun, dimana sekitar 4,4 juta diantaranya meninggal dalam 12 bulan.

Kasus stroke di provinsi Jawa Tengah tahun 2006 sebesar 12,41 per 1000 penduduk, terdiri atas stroke hemoragik sebesar 3,05 per 1.000 penduduk dan stroke non-hemoragik 9,36 per 1.000 penduduk (Dinkes Provinsi Jateng, 2006:42). Pada tahun 2009 di kota Semarang kasus stroke 3.304. Prevalensi stroke hemoragik di Jawa Tengah tahun 2009 adalah 0,05 persen lebih tinggi dibandingkan angka kejadian tahun 2008 sebesar 0,03 persen. Prevalensi tertinggi adalah di Kabupaten Kebumen sebesar 0,29 persen. Sedangkan prevalensi stroke non-hemoragik pada tahun 2009 sebesar 0,09 persen, mengalami penurunan bila dibandingkan prevalensi tahun 2008 sebesar 0,11 persen. Prevalensi tertinggi adalah kota Surakarta sebesar 0,75 persen (Dinkes Provinsi Jateng, 2009). Prevalensi stroke hemoragik di Jawa Tengah tahun 2011 adalah 0,03 persen sama dengan angka tahun 2010. Prevalensi tertinggi tahun 2011 adalah di Kota

Magelang sebesar 1,34 persen. Sedangkan prevalensi stroke *non-hemoragik* pada tahun 2011 sebesar 0,09 persen, sama dengan prevalensi tahun 2010. Prevalensi tertinggi adalah di Kota Magelang sebesar 3,45 persen (Dinkes Provinsi Jateng, 2011:37).

Penelitian tentang hubungan diantara fenomena-fenomena real merupakan dasar dari tujuan sains dan memainkan peranan penting dalam pengambilan keputusan di dalam kehidupan sehari-hari. Perkembangan teknologi yang sangat pesat seiring dengan kebutuhan manusia yang semakin banyak dan kompleks memungkinkannya untuk digunakan secara luas di berbagai bidang seperti pada dunia bisnis, kesehatan, pendidikan, dan sebagainya. Peranan komputer sangat penting untuk membantu pekerjaan manusia sehari-hari dalam segala aspek bidang. Pemakai mulanya menggunakan komputer sebagai mesin ketik yang dapat bekerja lebih cepat, tepat maupun otomatis. Sejalan dengan perkembangan saat ini, para ahli mencoba menggantikan komputer menjadi suatu alat bantu yang dapat meniru cara kerja otak manusia, sehingga diharapkan suatu saat akan tercipta komputer yang dapat menimbang dan mengambil keputusan sendiri.

Hal inilah yang mendorong lahirnya teknologi AI (*Artificial Intelligence*). Salah satu teknik komputasi yang dikelompokkan dalam AI adalah Jaringan Syaraf Tiruan (*Artificial Neural Network*). Jaringan Syaraf Tiruan merupakan sistem pemrosesan informasi yang mempunyai penampilan karakteristik menyerupai jaringan syaraf biologi. Jaringan Syaraf Tiruan adalah salah satu alternatif pemecahan masalah dan banyak diminati oleh para peneliti pada saat ini. Hal ini karena keluwesan yang dimiliki oleh Jaringan Syaraf Tiruan,

baik perancangan maupun dalam penggunaannya. Aplikasi Jaringan Syaraf Tiruan telah banyak dimanfaatkan dalam berbagai kepentingan, beberapa aplikasinya antara lain di dalam perbankan, telekomunikasi, dan kesehatan. Di bidang perbankan, Jaringan Syaraf Tiruan dapat digunakan memprediksi pemberian kredit, prediksi Kurs valuta asing, dan prediksi pasar saham. Di bidang kesehatan, Jaringan Syaraf Tiruan dalam mendiagnosa jenis penyakit menyimpan sejumlah data meliputi informasi pada gejala, deteksi, dan informasi lainnya.

Metode Naive Bayes merupakan pendekatan statistik untuk melakukan inferensi induksi pada persoalan klasifikasi. Metode ini merupakan metode yang baik di dalam mesin pembelajaran berdasarkan data *training*, dengan menggunakan probabilitas bersyarat sebagai dasar. Selain itu metode Naive Bayes juga dapat menangani data berupa numerik. Namun metode ini hanya dapat digunakan dalam persoalan klasifikasi dengan *supervised learning* dan data-data kategorial (Basuki, 2006). Metode Naive Bayes dapat dimanfaatkan dalam berbagai bidang untuk menyelesaikan masalah. Misalnya, mengenali buah berdasarkan fitur-fitur buah, menentukan keputusan aksi, dan dapat digunakan menentukan deteksi suatu penyakit berdasarkan data-data gejala. Maka metode ini dapat dipakai untuk mendukung pengambilan keputusan dalam deteksi penyakit seperti pada penyakit stroke.

Akan sangat membantu jika ada suatu alat untuk memprediksi penyakit stroke pada seseorang. Untuk itu, pada penelitian ini dipelajari dan dicoba membuat aplikasi Jaringan Syaraf Tiruan dan Naive Bayes memprediksi penyakit stroke. Dalam memprediksi jenis penyakit menyimpan sejumlah data, yaitu

informasi gejala-gejala yang terdapat pada seseorang. Pelatihan dapat dipresentasikan dengan *input* yang terdiri atas serangkaian gejala yang diderita oleh seseorang. Setelah itu, akan melatih *input* gejala tersebut sehingga ditemukan suatu akibat dari gejala tersebut yaitu *Suspect* atau tidaknya seseorang terkena stroke.

Dalam penelitian ini peneliti ingin membandingkan Jaringan Syaraf Tiruan dengan metode Naive Bayes. Maka akan dibangun sebuah aplikasi yang dapat memprediksi seseorang *Suspect* stroke atau tidak melalui pembelajaran dari kumpulan data riwayat kesehatan yang termasuk dalam kategori faktor risiko dan gejala dari penyakit stroke. Supaya memberikan referensi salah satu metode yang dapat digunakan untuk memprediksi seseorang terkena stroke.

1.2 Perumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka rumusan masalah dari penelitian ini sebagai berikut.

- a. Bagaimana mengimplementasikan metode Jaringan Syaraf Tiruan dan metode Naive Bayes guna proses mendeteksi seseorang terkena stroke atau tidak?
- b. Metode manakah antara Jaringan Syaraf Tiruan atau Naive Bayes yang memiliki tingkat keakuratan terbaik untuk mengolah data *training* pada permasalahan di atas?

1.3 Batasan Masalah

Agar dalam pembahasan skripsi ini tidak terlalu meluas, maka penulis mencantumkan pembatasan masalah sebagai berikut.

- a. Penyakit stroke yang dimaksudkan dalam penelitian ini adalah stroke *non-haemoragik*.
- b. Kategori data yang akan digunakan untuk acuan proses analisa merupakan faktor risiko dan gejala stroke yang akan diuraikan pada landasan teori.
- c. Data *training* yang digunakan pada aplikasi yang akan dibuat sejumlah 120 data yang diambil dari Rekam Medis RSUD Tugurejo Semarang.
- d. *Output* dari program ini berupa perkiraan *Suspectstroke* atau tidak yang akan digunakan menghitung tingkat keakuratan metode yang dipakai.
- e. Ada 2 metode yang digunakan dan dibandingkan dalam program yang akan dibuat yaitu metode Jaringan Syarat Tiruan dan metode Naive Bayes.
- f. Data numerik yang akan diinputkan pada pengujian dalam penelitian adalah batas nilai *minimal* sampai batas *maksimal* pada data pelatihan.

1.4 Tujuan dan Manfaat

1.4.1 Tujuan Penelitian

Adapun tujuan dari penelitian ini, sebagai berikut.:

- a. Membangun sebuah sistem untuk memprediksi kemungkinan seseorang *Suspectstroke* atau tidak berkaitan dengan *inputan* yang dimasukkan pasien.
- b. Membandingkan keakuratan dari dua metode yang dipakai dalam aplikasi yang akan dibuat dengan menghitung *error* dari *output* percobaan dengan

menggunakan data *training* sendiri sebagai *inputan* pada masing-masing metode.

1.4.2 Manfaat Penelitian

Manfaat dari penelitian ini sebagai berikut.

a. Bagi Peneliti

- 1) Dapat memberikan referensi tentang salah satu metode untuk memprediksi seseorang terkena stroke.
- 2) Peneliti dapat mengetahui tingkat keakuratan metodologi yang digunakan untuk memprediksi seseorang terkena stroke.

b. Bagi Mahasiswa

Semoga penelitian ini dapat memberikan gagasan baru bagi mahasiswa untuk mengembangkan dan merancang aplikasi menggunakan Jaringan Syaraf Tiruan dan metode Naive Bayes yang nantinya dapat dimanfaatkan dalam kehidupan sehari-hari.

c. Bagi Masyarakat

- 1) Dapat mengetahui faktor risiko dalam penentuan *Suspectstroke* atau tidak.
- 2) Dapat menguji kemungkinan seseorang termasuk *Suspectstroke* atau tidak berdasarkan riwayat kesehatannya dengan metode yang dibuat oleh peneliti.

1.5 Sistematika Penulisan

Secara garis besar penulisan skripsi ini terdiri atas tiga bagian yaitu bagian awal, bagian pokok, dan bagian akhir yang masing-masing diuraikan sebagai berikut.

1.5.1 Bagian Awal

Dalam penulisan skripsi ini bagian awal berisi halaman judul, halaman pengesahan, motto dan persembahan, kata pengantar, abstrak, daftar isi, daftar tabel, daftar gambar, dan daftar lampiran.

1.5.2 Bagian Pokok

Bagian pokok dari penulisan skripsi ini adalah isi skripsi yang terdiri atas lima bab, sebagai berikut.

1) BAB I PENDAHULUAN

Bab ini membahas tentang latar belakang, perumusan masalah, batasan masalah, tujuan dan manfaat, serta sistematika penulisan skripsi.

2) BAB II LANDASAN TEORI

Bab ini membahas tentang penyakit stroke serta teori-teori kecerdasan buatan sehubungan dengan Jaringan Syaraf Tiruan dan metode Naive Bayes yang berkaitan dengan penyelesaian penelitian ini, yang didapatkan dari berbagai

macam buku serta sumber-sumber terkait lainnya yang berhubungan dengan pembuatan penelitian ini.

3) BAB III METODE PENELITIAN

Bab ini berisi tentang prosedur atau langkah-langkah yang dilakukan dalam penelitian ini meliputi studi pustaka, perumusan masalah, pemecahan masalah, analisis pemecahan masalah, dan penarikan simpulan.

4) BAB IV HASIL DAN PEMBAHASAN

Bab ini menjelaskan tentang pengujian sistem secara umum maupun terperinci. Pengujian sistem secara umum akan membahas mengenai lingkungan uji coba untuk menggunakan sistem ini. Selanjutnya secara lebih terperinci dijelaskan dalam pengujian sistem meliputi skenario pengujian, beserta langkah-langkah dalam uji coba sistem.

5) BAB V PENUTUP

Bab ini berisi simpulan hasil penelitian dan saran-saran peneliti.

1.5.3 Bagian Akhir

Barisi daftar pustaka sebagai acuan penulisan yang mendukung kelengkapan skripsi ini dan lampiran-lampiran yang melengkapi uraian pada bagian isi.

BAB II

LANDASAN TEORI

Dalam bab ini akan dibahas mengenai teori–teori penting yang dapat menunjang dan menjadi acuan dalam pembuatan skripsi. Bagian tersebut meliputi Penyakit stroke, Penyiapan Data (*Preprosesing*), Kecerdasan buatan, Metode Jaringan Syaraf Tiruan, Metode Naive Bayes, Matlab, dan teori penunjang lainnya. Bagian-bagian tersebut akan diuraikan seperti di bawah ini.

2.1 Stroke atau *Kardioserebrovaskuler* (Gangguan Pembuluh Darah Otak)

2.1.1 DefinisiStroke

Beberapa definisi tentang stroke secara teoritis dari beberapa literatur dapat digambarkan sebagai berikut.

Stroke adalah gangguan fungsi saraf yang disebabkan oleh gangguan aliran darah dalam otak yang dapat timbul secara mendadak dalam beberapa detik atau secara cepat dalam beberapa jam dengan gejala atau tanda-tanda sesuai dengan daerah yang terganggu.

Definisi menurut WHO (2004): stroke adalah terjadinya gangguanfungsional otak fokal maupun global secara mendadak dan akut yang berlangsung lebih dari 24 jam akibat gangguan aliran darah otak. Menurut Neil F Gordon, sebagaimana dikutip oleh Lingga (2013: 3), stroke adalah gangguan potensial yang fatal pada suplai darah bagian otak. Tidak adasatupun bagian tubuh manumur yang dapat bertahan bila terdapat gangguan suplai darah dalam waktu

relatif lama sebab darah sangat dibutuhkan dalam kehidupan terutama oksigen pengangkut bahan makanan yang dibutuhkan pada otak dan otak adalah pusat sistem kontrol tubuh termasuk perintah dari semua gerakan fisik.

Stroke diartikan oleh awam dengan istilah penyakit lumpuh, padahal stroke tidak selalu disertakan dengan kelumpuhan. Stroke juga disebut serangan otak. Sebutan yang terakhir ini barangkali lebih tepat karena stroke adalah suatu kondisi yang ditandai dengan serangan otak akibat pukulan telak yang terjadi secara mendadak. Dalam bahasa medis, stroke disebut CVA (*Cerebro Vascular Accident*). Merujuk pada istilah medis, stroke didefinisikan sebagai gangguan syaraf permanen akibat terganggunya peredaran darah ke otak, yang terjadi sekitar 24 jam atau lebih. Sidrom klinis ini terjadi secara mendadak serta bersifat progresif sehingga menimbulkan kerusakan otak secara akut dengan tanda klinis yang terjadi secara fokal dan atau global (Lingga, 2013 : 1).

2.1.2 Faktor Risiko

Menurut Lingga (2013:19) secara garis besar, faktor risiko stroke dibagi menjadi dua, yaitu faktor tidak terkendali atau faktor yang bersifat menetap dan faktor yang bersifat dapat dikendalikan atau faktor tidak tetap. Adapun variabel-variabel yang akan dijadikan acuan untuk data *training* yaitu :

2.1.2.1 Faktor Risiko yang tidak bisa dirubah

- 1) Umur.
- 2) Jenis kelamin.

2.1.2.2 *Faktor Risiko yang dapat dirubah (diperbaiki)*

1) Tekanan darah

Klasifikasi tekanan darah dapat dilihat pada Tabel 2.1 :

Tabel 2.1 Klasifikasi Tekanan Darah

Sistolik	Diastolik	Keterangan
< 120	< 80	Optimal
<130	<65	Normal
130-139	85-89	Prehipertensi (Normal)
140-159	90-99	Hipertensi 1 (Tinggi)
160-179	100-99	Hipertensi 2 (Tinggi)
>180	>90	Hipertensi 3 (Tinggi)

(Sumber: Lingga, 2013:29)

2) Diabetes militus

Deteksi Diabetes militus apabila :

- 1) Terdapat dua hasil dari GDP (Glukosa Darah Puasa) >120 mg/dl, 2jPP (2 jam Pasca Puasa) >200 mg/dl, atau Random >200 mg/dl, atau
- 2) Memberikan pernyataan apabila sudah mempunyai riwayat menderita Diabetes Militus.

3) Riwayat menderita penyakit stroke

2.1.2.3 *Gejala Stroke*

- 1) Sering pusing disertai mual dan pening yang berlangsung terus-menerus meskipun telah minum obat penahan rasa sakit,

- 2) Muka terasa tebal, telapak kaki dan tangan kebas atau mati rasa,
- 3) Koordinasi anggota gerak (tangan dan kaki) tidak seperti biasanya, misalnya sulit digerakkan,
- 4) Tulisan menjadi jelek atau bahkan tidak bias dibaca,
- 5) Mendadak mengalami kebingungan,
- 6) Sering kejang, pingsan, dan bahkan koma, dll.

2.2 Kecerdasan Buatan (*Artifical Intelligent*)

Menurut Minskiy, sebagaimana dikutip oleh Kusriani (2006: 3-6), kecerdasan buatan adalah suatu ilmu yang mempelajari cara membuat komputer melakukan sesuatu seperti yang dilakukan oleh manusia. Definisi lain diungkapkan oleh H.A. Simon, sebagaimana dikutip oleh Kusriani (2006: 3-6) kecerdasan buatan (*Artifical Intelligent*) merupakan kawasan penelitian, aplikasi dan instruksi yang terkait dengan pemrograman komputer untuk melakukan sesuatu hal yang dalam pandangan manusia itu adalah - cerdas.

Ada tiga tujuan kecerdasan buatan, yaitu: membuat komputer lebih cerdas, mengerti tentang kecerdasan, dan membuat mesin lebih berguna. Yang dimaksud kecerdasan adalah kemampuan untuk belajar atau mengerti dari pengalaman, memahami pesan yang kontradiktif dan ambigu, menanggapi dengan cepat dan baik atas situasi yang baru, menggunakan penalaran dalam memecahkan masalah serta menyelesaikannya dengan efektif.

2.2.1 Konsep Penelitian dalam Kecerdasan Buatan

Ada beberapa konsep yang harus dipahami dalam kecerdasan buatan, diantaranya :

1) *Turing Test*-Metode Pengujian Kecerdasan

Turing Test merupakan sebuah metode pengujian kecerdasan yang dibuat oleh Alan Turing. Proses uji ini melibatkan seorang penanya (manusia) dan dua obyek yang ditanyai. Yang satu adalah seorang manusia dan satunya adalah sebuah mesin yang akan diuji. Penanya tidak dapat melihat langsung kepada obyek yang ditanyai. Penanya diminta untuk membedakan mana jawaban komputer dan mana jawaban manusia berdasarkan kedua obyek tersebut. Jika penanya tidak dapat membedakan mana jawaban mesin dan mana jawaban manusia maka Turing berpendapat bahwa mesin yang diuji tersebut diasumsikan “Cerdas”.

2) Pemrosesan Simbolik

Komputer semula didesain untuk memproses bilangan/angka-angka (pemrosesan numerik). Sementara manusia dalam berpikir dan menyelesaikan masalah lebih bersifat simbolik, tidak didasarkan pada sejumlah rumus atau melakukan komputasi matematis. Sifat penting dari AI adalah bahwa AI merupakan bagian dari ilmu komputer yang melakukan proses secara simbolik dan *non-algoritmik* dalam penyelesaian masalah.

3) *Heuristic*

Istilah *heuristic* diambil dari bahasa Yunani yang berarti menemukan. *Heuristic* merupakan suatu strategi untuk melakukan proses

pencarian (*search*) ruang *problem* secara selektif, yang memandu proses pencarian yang kita lakukan disepanjang jalur yang memiliki kemungkinan sukses paling besar.

4) Penarikan Kesimpulan (*Inferencing*)

AI mencoba membuat mesin memiliki kemampuan berpikir atau mempertimbangkan (*reasoning*). Kemampuan berpikir (*reasoning*) termasuk di dalamnya proses penarikan kesimpulan (*inferencing*) berdasarkan fakta-fakta dan aturan dengan menggunakan metode *heuristic* atau pencarian lainnya.

5) Pencocokan Pola (*Pattern Matching*)

2.2.2 Domain Penelitian dalam Kecerdasan Buatan

Beberapa bidang penelitian dalam kecerdasan buatan di antaranya:

- (1) *Formal tasks* (matematika, *games*)
- (2) *Mundane task* (*perception, robotics, natural language, common sense, reasoning*)
- (3) *Expert tasks* (*financial analysis, medical diagnosis, engineering, scientific analysis, dll*)

2.3 Penyiapan Data (*Preprocessing*)

2.3.1 Penanganan Terhadap Data Yang Hilang (*Missing Data*)

Metode *Data Mining* seringkali mensyaratkan semua nilai data lengkap dan tidak ada yang hilang. Padahal dalam kenyataannya banyak terdapat atribut atau *field* dari beberapa *record* yang tidak diketahui nilainya. Solusi paling sederhana untuk mengatasi hal ini adalah mereduksi data set dengan cara menghapus semua *record* yang berisi nilai yang kosong atau tidak diketahui. Untuk data berukuran

besar, mungkin cara ini tidak terlalu berpengaruh terhadap model data *mining* yang dihasilkan. Akan tetapi lain halnya jika data-data yang dihapus ini memiliki potensi yang sangat besar dan ukuran data set yang digunakan untuk *generate* model tidak terlalu besar. Maka solusi yang paling ideal adalah dengan cara melakukan teknik tersendiri untuk menangani data yang kosong ini sehingga dapat diisi dengan suatu nilai yang tepat agar data tersebut dapat turut andil dalam proses pembuatan model (Prabowo, 2013: 4).

Solusi praktis untuk menangani data yang hilang adalah data *miner* bersama-sama dengan pakar domain secara manual menguji data-data yang kosong kemudian memperkirakan nilai yang tepat untuk data tersebut. Akan tetapi metode ini akan membutuhkan waktu yang lama apalagi jika data yang ditangani berukuran besar dan berdimensi banyak.

Pendekatan kedua dilakukan dengan beberapa cara penggantian suatu nilai konstanta terhadap nilai yang hilang tersebut, antara lain :

- 1) Mengganti semua nilai yang hilang dengan konstanta yang sifatnya global (untuk menentukan konstanta yang bersifat global sangat tergantung pada aplikasi yang ditangani).
- 2) Mengganti nilai yang hilang dengan fitur *mean* (nilai rata-rata).

Selain kedua cara diatas, cara lain yang dilakukan adalah dengan menginterpretasikan nilai yang hilang sebagai nilai “*don't care*”. Dengan cara ini, suatu sampel data dengan nilai yang kosong akan digantikan oleh beberapa data dari himpunan sampel buatan yang berisi semua kemungkinan yang ada dari domain nilai tersebut. Sebagai contoh, jika sampel X merupakan sampel tiga

dimensi dan dinyatakan $X = \{1,?,3\}$, dimana nilai fitur kedua adalah nilai yang hilang (dinyatakan dengan tanda ?), maka proses akan *generate* lima sampel buatan jika diketahui domain fiturnya adalah $= [0,1,2,3,4]$.

$X1=\{1,0,3\}$, $X2=\{1,1,3\}$, $X3=\{1,2,3\}$, $X4=\{1,3,3\}$, dan $X5=\{1,4,3\}$.

2.4 Transformasi Data

Data mentah perlu dilakukan proses transformasi untuk meningkatkan performansinya. Transformasi yang umum digunakan adalah dengan melakukan normalisasi terhadap data tersebut. Contoh pada Tabel 2.2 berikut ini tiap 17actor memiliki skala yang tidak sama, ada yang besar 17actor17 yang kecil (Ratna, 2011:12).

Pada Tabel 2.2 dapat dilihat bahwa tidak adil jika nilai dari semua 17actor dijumlahkan untuk setiap pilihan yang ada kemudian membandingkannya dikarenakan skala dari tiap 17 faktor tidak sama.

Tabel 2.2 Tabel Perbandingan Nilai

Kriteria	Alternatif	Pilihan X	Pilihan Y	Pilihan Z	Skala
Faktor A		1	4	5	0-5
Faktor B		20	70	50	1-100
Faktor C		-2	0	1	-2 sampai +2
Faktor D		0-4	0.75	0.4	0 sampai 1
Jumlah		19.4	74.75	56.4	
Nilai normalisasi		12.9 %	49.7 %	37.5 %	

(Sumber: Ratna, 2011:12)

Untuk mendapatkan hasil yang lebih adil, kita menggunakan 2 cara, yaitu :

(1) Meranking pilihan dari tiap faktor.

Dari Tabel 2.3 dapat dilihat bahwa pilihan Y dan Z memiliki nilai normalisasi yang sama.

Tabel 2.3 Tabel Merangking Pilihan Tiap Faktor

Kriteria	Alternatif	Pilihan X	Pilihan Y	Pilihan Z
Faktor A		3	2	1
Faktor B		3	1	2
Faktor C		3	2	1
Faktor D		2	1	2
Jumlah		11	6	6
Nilai normalisasi		12.9 %	49.7 %	37.5 %

(Sumber: Ratna, 2011:13)

Nilai tiap baris berada diantara 1,2, dan 3 menunjukkan *ranking* tiap pilihan (berdasarkan nilai yang ditunjukkan Tabel 2.1). Karena nilai dengan *ranking* terendah yang lebih dipilih maka kita perlu untuk menormalkan jumlah dengan cara:

$$\text{nilai normalitas} = \frac{1}{2} \left(1 - \frac{\text{jumlah}}{\text{total jumlah}} \right)$$

(2) Mengubah nilai sehingga tiap faktor mempunyai *range* yang sama

Sebagai contoh, kita akan mengubah nilai pada Tabel 2.3 sehingga mempunyai *range* 0-1. Untuk mengubahnya, kita menggunakan cara geometri sederhana pada garis lurus

$$\text{nilai baru} = \frac{nab - nbb}{naa - nba} (\text{nilai lama} - nba) + nbb$$

Keterangan :

nab : batas atas nilai baru

nbb : batas bawah nilai baru

n_{aa} : batas atas nilai asli

n_{ba} : batas bawah nilai asli

Tabel 2.4 menunjukkan nilai setiap faktor dengan *range* yang sama yaitu 0-1. Dengan memiliki skala yang sama, maka total penjumlahan setiap faktor untuk semua pilihan yang ada bersifat adil dan akurat.

Tabel 2.4 Tabel Normalisasi

Kriteria	Alternatif	Pilihan X	Pilihan Y	Pilihan Z
Faktor A		0.2	0.8	1
Faktor B		0.192	0.697	0.495
Faktor C		0	0.5	0.75
Faktor D		0.4	0.75	0.4
Jumlah		0.792	2.747	2.645
Nilai normalisasi		12.8 %	44.4 %	42.8 %

(Sumber: Ratna, 2011:14)

2.5 Pengenalan Pola

Menurut Gonzalez dan Woods, sebagaimana dikutip oleh Abidin (2010), pola adalah suatu kuantitatif atau gambar struktural dari suatu objek atau beberapa entitas yang menarik dari suatu citra. Secara umum, Pola terbentuk oleh satu atau lebih ciri citra. Dengan kata lain, pola adalah susunan dari ciri-ciri citra. Ciri-ciri yang dapat diperoleh dari suatu citra, misalnya histogram, arah, dan magnitudo tepi, warna, luas daerah dan sebagainya. Ilmu yang mempelajari klasifikasi atau penggambaran dari suatu objek berdasarkan ciri-cirinya adalah pengenalan pola. Pengenalan pola dibedakan menjadi tiga yaitu.

2.5.1 Deteksi

Menurut Poerwadarminta sebagaimana yang dikutip Purnamasari (2013:31), deteksi adalah usaha untuk menemukan keberadaan, anggapan, atau kenyataan. Deteksi umumnya berkaitan dengan segmentasi dan proses *thresholding*, misalnya dalam mendeteksi daun pada suatu gambar, maka benda yang berwarna hijau akan terdeteksi sebagai daun.

2.5.2 Klasifikasi

Klasifikasi adalah proses menemukan sekumpulan model atau fungsi yang menjelaskan dan membedakan data ke dalam kelas-kelas tertentu dengan tujuan menggunakan model tersebut dalam menentukan kelas dari suatu objek yang belum diketahui kelasnya (Maharani, 2009). Misalnya dalam pengklasifikasian mobil.

2.5.3 Pengenalan

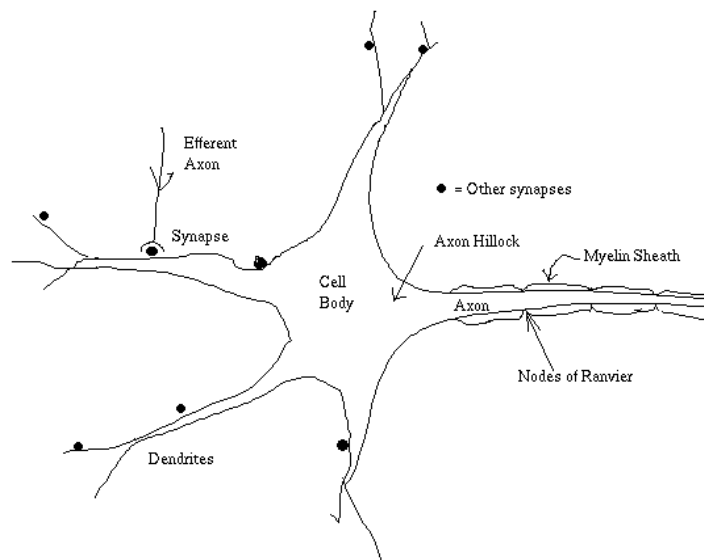
Pengenalan pola bertujuan menentukan kelompok atau kategori pola berdasarkan ciri-ciri yang dimiliki oleh pola tersebut. Tujuan pengelompokan adalah untuk mengenali suatu objek dalam citra. Misalnya, dalam mengenali suatu wajah pada gambar, maka wajah akan dideteksi dan diproses untuk dibandingkan dengan database wajah dikenal sebelumnya untuk menentukan siapa orang tersebut.

2.6 Jaringan Syaraf Tiruan

2.6.1 Inspirasi Biologi

Jaringan Syaraf Tiruan keluar dari penelitian kecerdasan buatan, terutamapercobaan untuk menirukan *fault-tolerance* dan kemampuan untuk belajar dari sistem syaraf biologi dengan model struktur *low-level* dari otak.

Otak terdiri dari sekitar (10.000.000.000) sel syaraf yang saling berhubungan. Sel syaraf mempunyai cabang struktur *input* (dendrites), sebuah inti sel dan percabangan struktur *output* (axon). Axon dari sebuah sel terhubung dengan dendrites yang lain melalui sebuah *synapses*. Ketika sebuah sel syaraf aktif, kemudian menimbulkan suatu *signal electrochemical* pada axon. Signal ini melewati *synapses* menuju ke sel syaraf yang lain. Sebuah sel syaraf lain akan mendapatkan signal jika memenuhi batasan tertentu yang sering disebut dengan nilai ambang atau (*threshold*) (Eliyani,2005:1).



Gambar 2.1 Susunan Saraf Manusia (Eliyani, 2005:2)

2.6.2 Sejarah Jaringan Syaraf Tiruan

Menurut Siang(2005: 4), sejarah Jaringan Syaraf Tiruan adalah sebagai berikut.

- 1) Sejarah Jaringan Syaraf Tiruan pertama kali secara sederhana diperkenalkan oleh McCulloch dan Pitts pada tahun 1943. McCulloch dan Pitts menyimpulkan bahwa kombinasi beberapa neuron sederhana menjadi sebuah sistem neural akan meningkatkan kemampuan komputasinya. Bobot dalam jaringan yang diusulkan oleh McCulloch dan Pitts diatur untuk melakukan fungsi logika sederhana. Fungsi aktivasi yang dipakai adalah fungsi *threshold*.
- 2) Pada tahun 1958, Rosenblatt memperkenalkan dan mulai mengembangkan model jaringan baru yang terdiri dari beberapa lapisan yang disebut Perceptron. Metode pelatihan diperkenalkan untuk mengoptimalkan hasil iterasinya.
- 3) Widrow dan Hoff (1960) mengembangkan perceptron dengan memperkenalkan aturan pelatihan jaringan, yang dikenal sebagai aturan delta (atau sering disebut kuadrat rata-rata terkecil). Aturan ini akan mengubah bobot perceptron apabila keluaran yang dihasilkan tidak sesuai dengan target yang diinginkan.
- 4) Rumelhart (1986) mengembangkan perceptron menjadi *Backpropagation*, yang memungkinkan jaringan diproses melalui beberapa *layer* karena apa yang dilakukan peneliti terdahulu hanya menggunakan jaringan dengan *layer* tunggal (*single layer*).
- 5) Selain itu, beberapa model Jaringan Syaraf Tiruan lain juga dikembangkan oleh Kohonen (1972), Hopfield (1982), dan lain-lain. Pengembangan yang ramai dibicarakan sejak tahun 1990an adalah aplikasi model-model Jaringan Syaraf Tiruan untuk menyelesaikan berbagai masalah di dunia nyata.

2.6.3 Definisi Jaringan Syaraf Tiruan

Jaringan Syaraf Tiruan (JST) didefinisikan sebagai suatu sistem pemrosesan informasi yang mempunyai karakteristik menyerupai jaringan syaraf manusia (Hermawan, 2006: 3). Beberapa istilah dalam Jaringan Syaraf Tiruan yang sering ditemui adalah sebagai berikut (Puspitaningrum, 2006: 13-15).

- 1) Neuron atau Node atau Unit: sel syaraf tiruan yang merupakan elemen pengolahan Jaringan Syaraf Tiruan. Setiap neuron menerima data *input*, memproses *input* tersebut kemudian mengirimkan hasilnya berupa sebuah *output*.
- 2) Jaringan: kumpulan neuron yang saling terhubung dan membentuk lapisan.
- 3) Lapisan tersembunyi (*hidden layer*): lapisan yang tidak secara langsung berinteraksi dengan dunia luar. Lapisan ini memperluas kemampuan Jaringan Syaraf Tiruan dalam menghadapi masalah-masalah yang kompleks.
- 4) *Input*: sebuah nilai *input* yang akan diproses menjadi nilai *output*.
- 5) *Output*: solusi dari nilai *input*.
- 6) Bobot: nilai matematis dari sebuah koneksi antar neuron.
- 7) Fungsi aktivasi: fungsi yang digunakan untuk *mengupdate* nilai-nilai bobot per-iterasi dari semua nilai *input*.
- 8) Fungsi aktivasi sederhana adalah mengkalikan *input* dengan bobotnya dan kemudian menjumlahkannya (disebut penjumlahan sigma). Berbentuk linier atau tidak linier, dan sigmoid.
- 9) Paradigma pembelajaran: bentuk pembelajaran, *supervised learning*, atau *unsupervised learning*.

Menurut Siang (2005: 2-3), Jaringan Syaraf Tiruan (JST) adalah sistem pemrosesan informasi yang memiliki karakteristik mirip dengan jaringan syaraf biologi. Jaringan Syaraf Tiruan dibentuk sebagai generalisasi model matematika dari jaringan syaraf biologi, dengan asumsi bahwa :

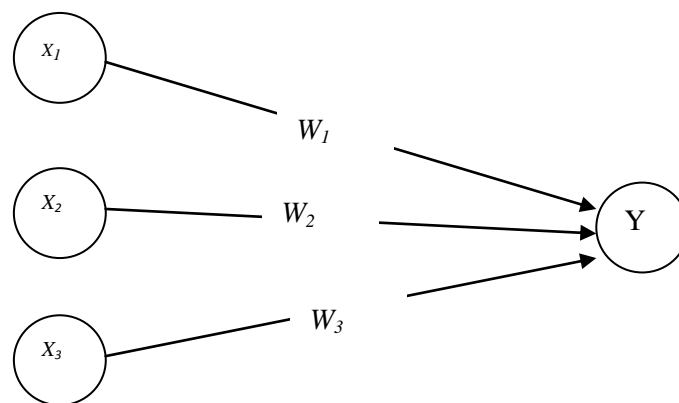
- 1) Pemrosesan informasi terjadi pada banyak elemen sederhana (neuron).
- 2) Sinyal dikirimkan diantara neuron-neuron melalui penghubung-penghubung.
- 3) Penghubung antar neuron memiliki bobot yang akan memperkuat atau memperlemah sinyal.
- 4) Untuk menentukan *output*, setiap neuron menggunakan fungsi aktivasi (biasanya bukan fungsi linier) yang dikenakan pada jumlahan *input* yang diterima. Besarnya *output* ini selanjutnya dibandingkan dengan suatu batasambang.

Jaringan Syaraf Tiruan ditentukan oleh 3 hal :

- 1) Pola hubungan antar neuron (disebut arsitektur jaringan).
- 2) Metode untuk menentukan bobot penghubung (disebut metode *training/learning/algoritma*).
- 3) Fungsi aktivasi (fungsi transfer).

Neuron dalam Jaringan Syaraf Tiruan sering diganti dengan istilah simpul. Setiap simpul tersebut berfungsi untuk menerima atau mengirim sinyal dari atau ke simpul-simpul lainnya. Pengiriman sinyal disampaikan melalui penghubung. Kekuatan hubungan yang terjadi antara setiap simpul yang saling terhubung dikenal dengan nama bobot.

Arsitektur jaringan dan algoritma pelatihan sangat menentukan model-model Jaringan Syaraf Tiruan. Arsitektur tersebut gunanya untuk menjelaskan arah perjalanansinyal atau data di dalam jaringan. Sedangkan algoritma belajar menjelaskan bagaimana bobot koneksi harus diubah agar pasangan masukan-keluaran yang diinginkan dapat tercapai. Dalam setiap perubahan harga bobot koneksi dapat dilakukan dengan berbagai cara, tergantung pada jenis algoritma pelatihan yang digunakan. Dengan mengatur besarnya nilai bobot ini diharapkan bahwa kinerja jaringan dalam mempelajari berbagai macam pola yang dinyatakan oleh setiap pasangan masukan-keluaran akan meningkat. Sebagai contoh, perhatikan neuron Y .



Gambar 2.2 Sebuah Sel Syaraf Tiruan (Siang, 2005: 3)

Y menerima *input* dari neuron x_1 , x_2 , dan x_3 dengan bobot hubungan masing-masing adalah w_1 , w_2 dan w_3 . Ketiga *impuls* neuron yang ada dijumlahkan $net = x_1w_1 + x_2w_2 + x_3w_3$. Besarnya *impuls* yang diterima oleh Y mengikuti fungsi aktivasi $y = f(net)$. Apabila nilai fungsi aktivasi cukup

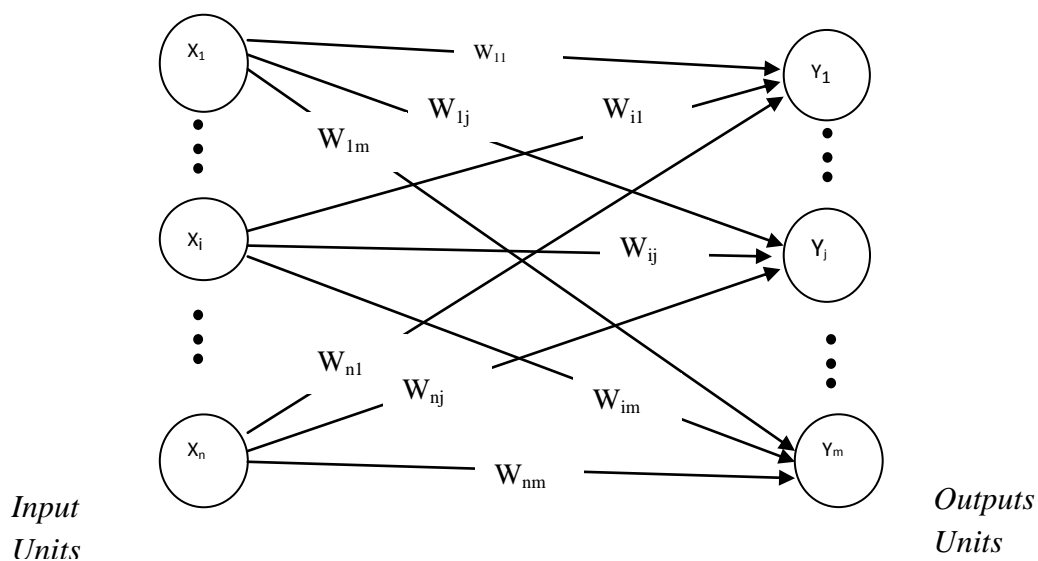
kuat, maka sinyal akan diteruskan. Nilai fungsi aktivasi (keluaran model jaringan) juga dapat dipakai sebagai dasar untuk merubah bobot (Siang, 2011: 4).

2.6.4 Arsitektur Jaringan

Menurut Puspitaningrum (2006:11) Arsitektur Jaringan Syaraf Tiruan digolongkan menjadi 3 model:

2.6.4.1 Jaringan Layer Tunggal

Dalam jaringan ini, sekumpulan *input* neuron dihubungkan langsung dengan sekumpulan *output*nya, seperti Gambar 2.3 berikut ini :



Gambar 2.3 Jaringan *Layer* Tunggal (Puspitaningrum, 2006:12)

Keterangan :

x_1, x_i, x_n : Nilai *input* ke 1 sampai n

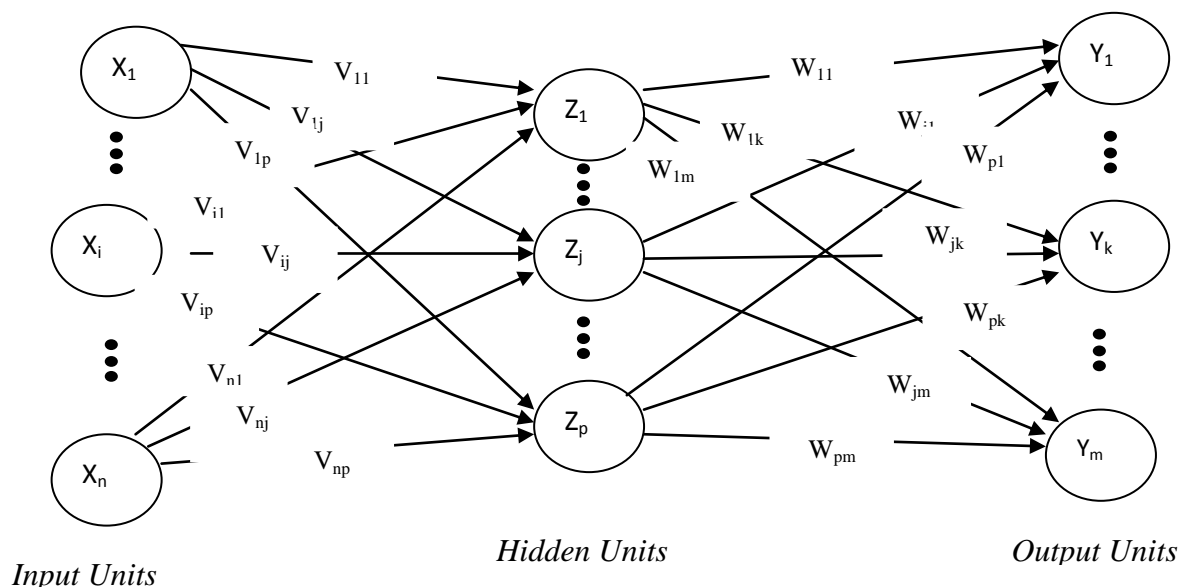
y_1, y_j, y_m : Nilai *Output* hasil pembangkitan nilai *input* oleh suatu fungsi aktivasi

w_i : Bobot atau nilai

Pada Gambar 2.3 diperlihatkan bahwa arsitektur jaringan layar tunggal dengan n buah masukan (x_1, x_2, \dots, x_n) dan m buah keluaran (Y_1, Y_2, \dots, Y_m) . Dalam jaringan ini semua unit *input* dihubungkan dengan semua unit *output*. Tidak ada unit *input* yang dihubungkan dengan unit *input* lainnya dan unit *output* pun demikian.

2.6.4.2 Jaringan Layar Jamak

Jaringan Layar Jamak dapat dilihat dalam Gambar 2.4 sebagai berikut ini:



Gambar 2.4 JaringanLayer Jamak (Puspitaningrum,2006:12)

Keterangan :

x_1, x_i, x_n : Nilai *input* ke 1 sampai n

z_1, z_j, z_p : Nilai dari neuron hidden *layer*

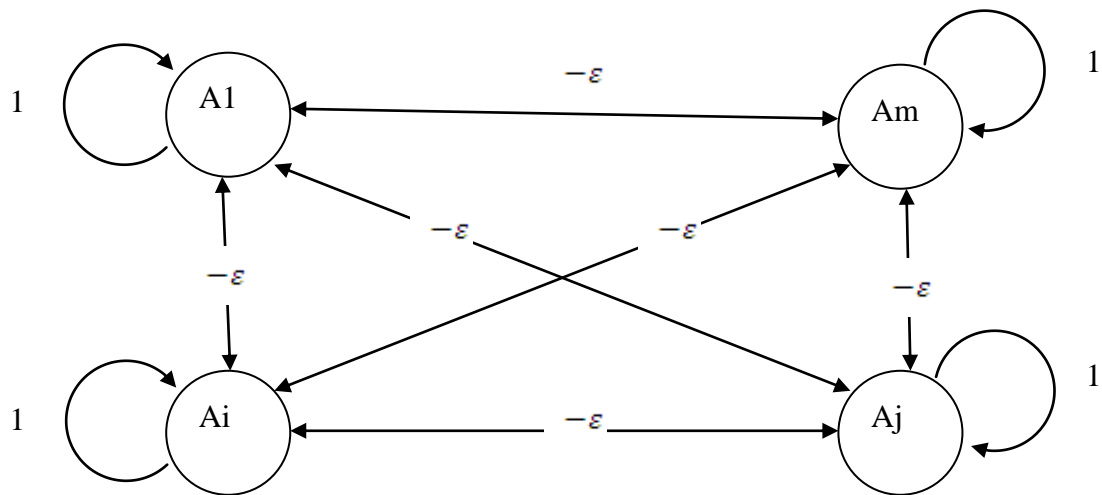
x_1, x_i, x_n : Nilai *output* hasil pembangkitan nilai *input* oleh suatu fungsi aktivasi

v_1, w_j : Bobot atau nilai

Pada Gambar 2.4 diperlihatkan jaringan dengan n buah unit masukan (x_1, x_2, \dots, x_n) , sebuah layer tersembunyi yang terdiri dari m buah unit (z_1, z_2, \dots, z_p) dan m buah unit keluaran (y_1, y_2, \dots, y_m) . Jaringan layer jamak dapat menyelesaikan masalah yang lebih kompleks dibandingkan dengan layer tunggal, meskipun kadangkala proses pelatihan lebih kompleks dan lama.

2.6.4.3 Jaringan Recurrent

Model jaringan *recurrent* mirip dengan jaringan layer tunggal ataupun ganda. Hanya saja, ada neuron *output* yang memberikan sinyal pada unit *input* (sering disebut *feedback loop*). Dengan kata lain sinyal mengalir dua arah, yaitu maju dan mundur. Untuk lebih jelas dapat dilihat pada Gambar 2.5.



Gambar 2.5 Jaringan Recurrent

2.6.5 Backpropagation

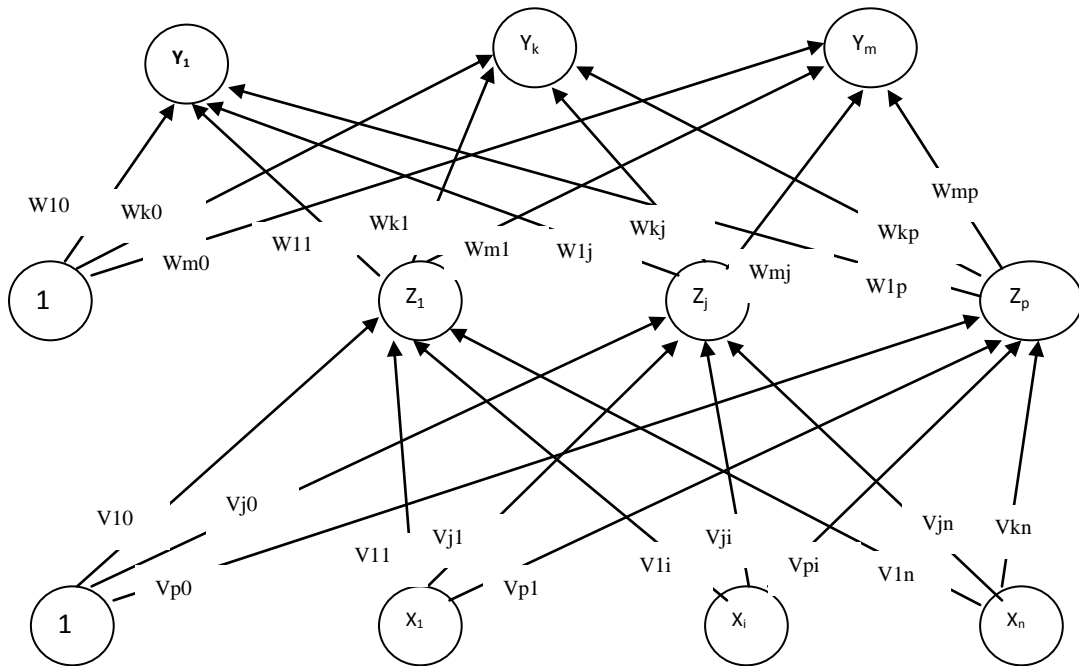
Backpropagation adalah salah satu metode dari Jaringan Syaraf Tiruan yang dapat diaplikasikan dengan baik dalam bidang peramalan

(forecasting). *Backpropagation* melatih jaringan untuk mendapatkan keseimbangan antara kemampuan jaringan mengenali pola yang digunakan selama *training* serta kemampuan jaringan untuk memberikan respon yang benar terhadap pola masukan yang serupa namun tidak sama dengan pola yang dipakai selama pelatihan (Siang, 2005: 97).

Menurut Kusumadewi (2004:116) dalam pelatihan dengan *backpropagation* sama halnya seperti pelatihan pada jaringan syaraf yang lain. Pada jaringan *feedforward* (umpan maju), pelatihan dilakukan dalam rangka perhitungan bobot sehingga pada akhir pelatihan akan diperoleh bobot-bobot yang baik. Selama proses pelatihan, bobot-bobot diatur secara iteratif untuk meminimumkan *error* (kesalahan) yang terjadi. Kesalahan dihitung berdasarkan rata-rata kuadrat kesalahan (MSE). Rata-rata kuadrat kesalahan juga dijadikan dasar perhitungan untuk kerja fungsi aktivasi.

Sebagian besar pelatihan untuk jaringan *feedforward* (umpan maju) menggunakan gradien dari fungsi aktivasi untuk menentukan bagaimana mengatur bobot-bobot dalam rangka meminimumkan kinerja. Gradien ini ditentukan dengan menggunakan suatu teknik yang disebut *backpropagation*. Dan pada dasarnya, algoritma pelatihan standar *backpropagation* akan menggerakkan bobot dengan arah gradien *negatif*. Prinsip dasar dari algoritma *backpropagation* adalah memperbaiki bobot-bobot jaringan dengan arah yang membuat fungsi aktivasi menjadi turun dengan cepat.

Backpropagation memiliki beberapa unit yang ada dalam satu atau lebih layer tersembunyi. Gambar 2.6 adalah arsitektur *backpropagation* dengan n buah



Gambar 2.6 Arsitektur *Backpropagation* (Siang, 2005:98)

masukan (ditambah sebuah bias), sebuah layer tersembunyi yang terdiri dari p unit (ditambah sebuah bias), serta m buah keluaran. v_{ji} merupakan bobot garis dari unit masukan x_i ke unit layer tersembunyi z_j (v_{j0} merupakan bobot garis yang menghubungkan bias di unit masukan ke unit layer tersembunyi z_j). w_{kj} merupakan bobot dari layer tersembunyi z_j ke unit keluaran y_k (w_{k0} merupakan bobot dari bias di layer tersembunyi ke unit keluaran y_k) (Siang, 2005:98).

2.6.6 Fungsi Aktivasi

Dalam *backpropagation*, fungsi aktivasi yang dipakai harus memenuhi beberapa syarat yaitu: kontinu, terdiferensial dengan mudah, dan merupakan fungsi yang tidak turun (Siang, 2005: 99). Fungsi aktivasi diharapkan jenuh

(mendekati nilai-nilai *maksimum* dan *minimum* secara asimtot) (Puspitaningrum, 2006:133).

Beberapa fungsi yang digunakan menurut Puspitaningrum (2006: 133) adalah sebagai berikut.

2.6.6.1 Fungsi Sigmoid Biner

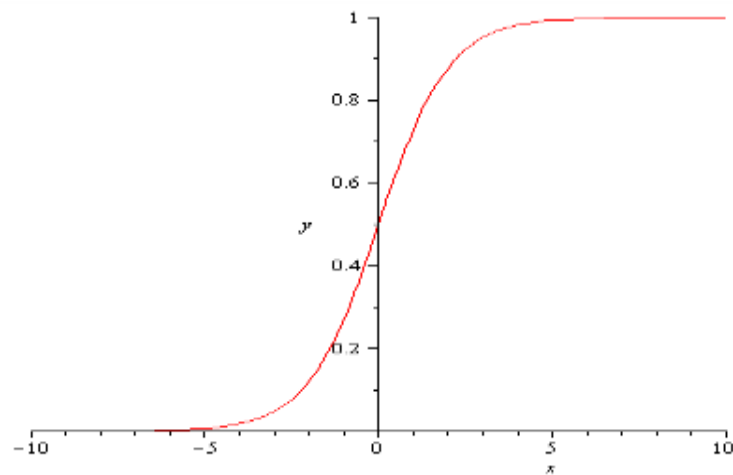
Fungsi ini merupakan fungsi yang umum digunakan. *Range*-nya adalah $(0,1)$ dan didefinisikan sebagai berikut:

$$f_1(x) = \frac{1}{1 + e^{-x}}$$

dengan turunan:

$$f_1'(x) = f_1(x)(1 - f_1(x))$$

Fungsi sigmoid biner diilustrasikan pada Gambar 2.7 sebagai berikut:



Gambar 2.7 Fungsi Sigmoid Biner dengan *range* $(0,1)$

2.6.6.2 Fungsi Sigmoid Bipolar

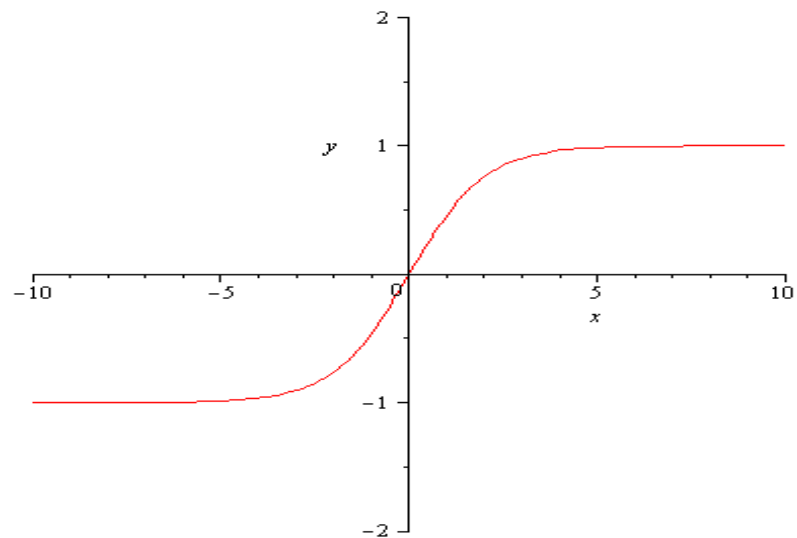
Fungsi sigmoid bipolar merupakan fungsi yang umum digunakan dan memiliki *range* $(-1,1)$ yang didefinisikan sebagai:

$$f_2(x) = 2f_1(x) - 1$$

dengan turunan:

$$f_2'(x) = \frac{1}{2}(1 + f_2(x))(1 - f_2(x))$$

Fungsi sigmoid bipolar diilustrasikan pada Gambar 2.8 sebagai berikut



Gambar 2.8 Fungsi Sigmoid Bipolar dengan *Range* (-1,1)

2.6.6.3 Fungsi Tangen Hiperbolik

Fungsi tangen hiperbolik didefinisikan sebagai:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

dengan turunan:

$$\tanh'(x) = (1 + \tanh(x))(1 - \tanh(x))$$

2.6.7 Pelatihan Standart *Backpropagation*

Ada tiga fase Pelatihan *backpropagation* menurut Siang (2005:100-101) antaralain :

2.6.7.1 Fase 1, yaitu *propagasi maju*

Dalam propagasi maju, setiap sinyal masukan dipropagasi (dihitung maju) ke layar tersembunyi hingga layar keluaran dengan menggunakan fungsi aktivasi yang ditentukan. Selama tahap ini, setiap unit masukan (X_i) menerima sinyal masuk dan mengirim sinyal ini ke setiap unit tersembunyi Z_1, \dots, Z_p . Setiap unit tersembunyi kemudian menghitung aktifasinya dan mengirim sinyalnya (z_j) ke setiap unit keluaran. Setiap unit keluaran (Y_k) menghitung aktifasinya (y_k) untuk menunjukkan respon jaringan terhadap pola masukan yang diberikan.

2.6.7.2 Fase 2, yaitu *propagasi mundur*

Kesalahan (selisih antara keluaran jaringan dengan target yang diinginkan) yang terjadi dipropagasi mundur mulai dari garis yang berhubungan langsung dengan unit-unit di layar keluaran. Selama pelatihan, setiap unit keluaran membandingkan aktifasi y_k dengan targetnya t_k untuk menentukan galat antara pola masukan dengan unit keluaran tersebut. Setelah didapat galat, faktor $\delta_k (k = 1, \dots, m)$ dihitung. δ_k digunakan untuk mendistribusikan galat pada unit keluaran Y_k kembali ke seluruh unit pada lapis sebelumnya (unit tersembunyi yang terhubung dengan Y_k).

2.6.7.3 Fase 3, yaitu *perubahan bobot*

Pada fase ini dilakukan modifikasi bobot untuk menurunkan kesalahan yang terjadi. Galat yang diperoleh pada langkah 2 dipakai untuk mengubah bobot antara keluaran dengan lapisan tersembunyi. Dengan cara yang sama, faktor δ_j ($j = 1, \dots, p$) dihitung untuk setiap unit Z_j . Faktor δ_j digunakan untuk mengubah bobot antara lapisan tersembunyi dengan lapisan masukan. Setelah seluruh faktor δ ditentukan, bobot untuk seluruh lapisan langsung disesuaikan. Penyesuaian bobot w_{jk} (dari unit tersembunyi Z_j ke unit keluaran Y_k) didasarkan pada faktor δ_k dan aktivasi dari unit Z_j , yaitu z_j . Penyesuaian bobot v_{ij} (dari unit masukan X_i ke unit tersembunyi Z_j) adalah didasarkan pada faktor δ_j dan aktivasi unit masukan x_i . Ketiga fase tersebut diulang-ulang terus hingga kondisi penghentian dipenuhi.

2.6.8 Algoritma Pelatihan

Menurut Siang (2005:102-104) Algoritma pelatihan untuk jaringan dengan satu *hidden layer* (dengan fungsi aktivasi sigmoid biner) adalah sebagai berikut.

Langkah 0 : Inisialisasi semua bobot dengan bilangan acak kecil

Langkah 1 : Jika kondisi penghentian belum terpenuhi, lakukan langkah 2-9

Langkah 2 : Untuk setiap pasang data pelatihan, lakukan langkah 3-8

Fase I : Propagasi Maju

Langkah 3 : Tiap unit masukan menerima sinyal dan meneruskannya ke unit tersembunyi di atasnya

Langkah 4 : Hitung semua keluaran di unit tersembunyi z_j ($j = 1, 2, \dots, p$)

$$z_{netj} = v_{j0} + \sum_{i=1}^n x_i v_{ji}$$

$$z_j = f(z_{netj}) = \frac{1}{1 + e^{-z_{netj}}}$$

Langkah 5 : Hitung semua keluaran jaringan di unit y_k ($k = 1, 2, \dots, m$)

$$y_{netk} = w_{k0} + \sum_{j=1}^p z_j w_{kj}$$

$$y_k = f(y_{netk}) = \frac{1}{1 + e^{-y_{netk}}}$$

Fase II: Propagasi Mundur

Langkah 6 : Hitung faktor δ unit keluaran berdasarkan kesalahan di setiap unit keluaran y_k ($k = 1, 2, \dots, m$)

$$\delta_k = (t_k - y_k) f'(y_{netk}) = (t_k - y_k) y_k (1 - y_k)$$

δ_k merupakan unit kesalahan yang akan dipakai dalam perubahan bobot layar dibawahnya (langkah 7)

Hitung suku perubahan bobot w_{kj} (yang akan dipakai nanti untuk merubah bobot w_{kj}) dengan laju percepatan α

$$\delta w_{kj} = \alpha \delta_k z_j \quad ; k = 1, 2, \dots, m \quad ; j = 0, 1, \dots, p$$

Langkah 7 : Hitung faktor δ unit tersembunyi berdasarkan kesalahan di setiap unit tersembunyi z_j ($j = 1, 2, \dots, p$)

$$\delta_{netj} = \sum_{k=1}^m \delta_k w_{kj}$$

Faktor δ unit tersembunyi:

$$\delta_j = \delta_{netj} f'(z_{netj}) = \delta_{netj} z_j (1 - z_j)$$

Hitung suku perubahan bobot v_{ij} (yang dipakai nanti untuk merubah bobot v_{ij})

$$\delta v_{ij} = \alpha \delta_j x_i \quad ; j = 1, 2, \dots, p \quad ; i = 0, 1, \dots, n$$

Fase III: Perubahan Bobot

Langkah 8 : Hitung semua perubahan bobot

Perubahan bobot garis yang menuju ke unit keluaran:

$$w_{jk}(\text{baru}) = w_{jk}(\text{Lama}) + \delta w_{jk}$$

Dari unit ke-1 sampai unit ke-p dilapisan tersembunyi juga dilakukan peng-update-an pada bias dan bobotnya ($i=1,2,\dots,n$; $j=1,\dots,p$):

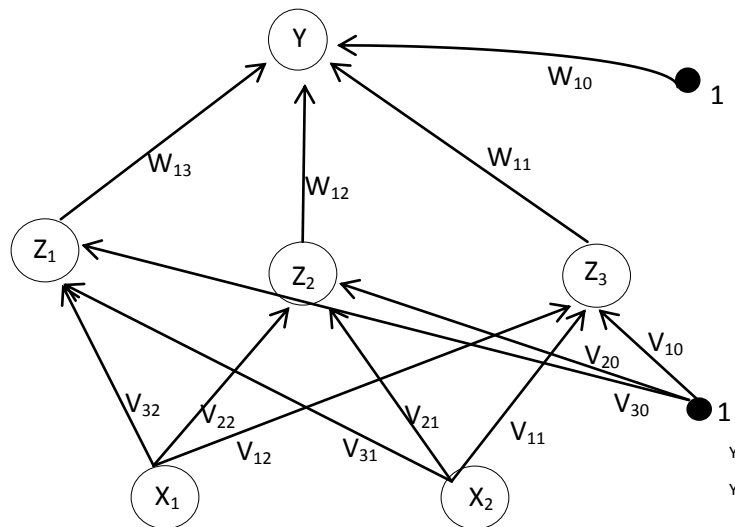
$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \delta v_{ij}$$

Langkah 9 : Test kondisi berhenti.

Setelah pelatihan selesai dilakukan, jaringan dapat dipakai untuk pengenalan pola. Dalam hal ini, hanya propagasi maju (langkah 4 dan 5) saja yang dipakai untuk menentukan keluaran jaringan.

Apabila fungsi aktivasi yang dipakai bukan sigmoid biner, maka langkah 4 dan 5 harus disesuaikan. Demikian juga turunannya pada langkah 6 dan 7.

Di sini contoh dari algoritma pelatihan dalam Jaringan Syaraf Tiruan dapat menyelesaikan masalah seperti yang ditunjukkan pada Gambar 2.9 memiliki fungsi aktivasi sigmoid dan dilatih dengan algoritma *backpropagation*.



Gambar 2.9 Struktur Jaringan Syaraf Tiruan untuk aplikasi (Siang, 2005:105)

Input $x_1 = 0.2$ dan $x_2 = 0.6$ dan target $d_1 = 1$. Bobotnya lapisan tersembunyi adalah

$W_j = \begin{bmatrix} 1 & 1.5 \\ 0.5 & 2 \\ 2.5 & 3 \end{bmatrix}$ dan bias $b_j = \begin{bmatrix} 1 \\ -0.5 \\ 1.5 \end{bmatrix}$, sedangkan bobot lapisan *output* adalah $W_j =$

$\begin{bmatrix} 3 & 2 & 1 \end{bmatrix}$ dan biasnya $b_j = [-4]$. Menghitung nilai *output* y_j dan nilai bobot bias

baru dengan menggunakan koefisien laju pembelajaran 0.5 dapat dilakukan dengan cara sebagai berikut.

1) Perhitungan arah maju:

Perhitungan net *input* neuron-neuron pada lapisan tersembunyi $z_j = f(z_{in_j})$:

$$\begin{bmatrix} z_{1_in} \\ z_{2_in} \\ z_{3_in} \end{bmatrix} = \begin{bmatrix} 1 & 1.5 \\ 0.5 & 2 \\ 2.5 & 3 \end{bmatrix} \begin{bmatrix} 0.2 \\ 0.6 \end{bmatrix} + \begin{bmatrix} 1 \\ -0.5 \\ 1.5 \end{bmatrix},$$

$$\begin{bmatrix} z_{1_in} \\ z_{2_in} \\ z_{3_in} \end{bmatrix} = \begin{bmatrix} 2.1 \\ 0.8 \\ 3.8 \end{bmatrix}$$

Perhitungan *output* neuron-neuron pada lapisan tersembunyi:

$$y_{netk} = w_{k0} + \sum_{j=1}^p z_j w_{kj}$$

$$y_k = f(y_{netk}) = \frac{1}{1 + e^{-y_{netk}}}$$

$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} 0.891 \\ 0.690 \\ 0.978 \end{bmatrix}$$

Perhitungan arah net *input* neuron lapisan *output* $y_k = f(y_{in_j})$

$$y_{in} = [3 \quad 2 \quad 1] \begin{bmatrix} 0.891 \\ 0.690 \\ 0.978 \end{bmatrix} + (-4)$$

Perhitungan *output* neuron lapisan *output* $\delta_k = (t_k - y_k)f'(y_{netk})$

$$f(y_{netk}) = \frac{1}{1 + e^{-y_{netk}}}$$

$$y = 0.737$$

2) Perhitungan arah balik :

Perhitungan nilai *error* sesuai dengan persamaan :

$$\begin{aligned} \delta_k &= (t_k - y_k)y_k(1 - y_k) \\ &= (1 - 0.737)0.737(1 - 0.737) \\ &= 0.051 \end{aligned}$$

Perhitungan perbaikan bobot dan bias lapisan tersembunyi sesuai persamaan

$$\delta_{netj} = \sum_{k=1}^m \delta_k w_{kj} \text{ dan } \delta_j = \delta_{netj} f'(z_{netj}) = \delta_{netj} z_j (1 - z_j):$$

$$\begin{bmatrix} w_{11} \\ w_{21} \\ w_{31} \\ w_{01} \end{bmatrix} = 0.5 \times 0.051 \begin{bmatrix} 0.891 \\ 0.690 \\ 0.978 \end{bmatrix} = \begin{bmatrix} 0.0227 \\ 0.0176 \\ 0.0249 \\ 0.0255 \end{bmatrix}$$

Perhitungan bobot dan bias baru pada lapisan tersembunyi :

$$\begin{bmatrix} w_{11} \\ w_{21} \\ w_{31} \\ w_{01} \end{bmatrix} = \begin{bmatrix} 3 \\ 2 \\ 1 \\ -4 \end{bmatrix} + \begin{bmatrix} 0.0227 \\ 0.0176 \\ 0.0249 \\ 0.0255 \end{bmatrix} = \begin{bmatrix} 3.0227 \\ 2.0176 \\ 1.0249 \\ -3.9745 \end{bmatrix}$$

Perhitungan perambatan *error* ke lapisan tersembunyi :

$$\delta_1 = 0.051 \times 3.0227 \times 0.8910 \times (1 - 0.8910) = 0.015$$

$$\delta_1 = 0.051 \times 2.0176 \times 0.690 \times (1 - 0.690) = 0.022$$

$$\delta_1 = 0.051 \times 1.0249 \times 0.978 \times (1 - 0.978) = 0.001$$

Perhitungan perbaikan bobot dan bias pada lapisan *input* :

$$\Delta v_{11} = 0.5 \times 0.015 \times 0.2 = 0.0015$$

$$\Delta v_{12} = 0.5 \times 0.022 \times 0.2 = 0.0022$$

$$\Delta v_{13} = 0.5 \times 0.001 \times 0.2 = 0.0001$$

$$\Delta v_{21} = 0.5 \times 0.015 \times 0.6 = 0.0045$$

$$\Delta v_{22} = 0.5 \times 0.022 \times 0.6 = 0.0066$$

$$\Delta v_{23} = 0.5 \times 0.001 \times 0.6 = 0.0003$$

$$\Delta v_{03} = 0.5 \times 0.001 \times 1 = 0.0005$$

Perhitungan bobot dan bias baru pada lapisan *input* sesuai

$$w_{jk}(\text{baru}) = w_{jk}(\text{Lama}) + \delta w_{jk} :$$

$$w_j = \begin{bmatrix} 1 & 1.5 \\ 0.5 & 2 \\ 2.5 & 3 \end{bmatrix} + \begin{bmatrix} 0.0015 & 0.0045 \\ 0.0022 & 0.0066 \\ 0.0001 & 0.0003 \end{bmatrix} = \begin{bmatrix} 1.0015 & 1.5045 \\ 0.5022 & 2.0066 \\ 2.5001 & 3.0003 \end{bmatrix}$$

$$b_j = \begin{bmatrix} 1 \\ -0.5 \\ 1.5 \end{bmatrix} + \begin{bmatrix} 0.0075 \\ 0.011 \\ 0.0005 \end{bmatrix} = \begin{bmatrix} 1.0075 \\ -0.489 \\ 1.5005 \end{bmatrix}$$

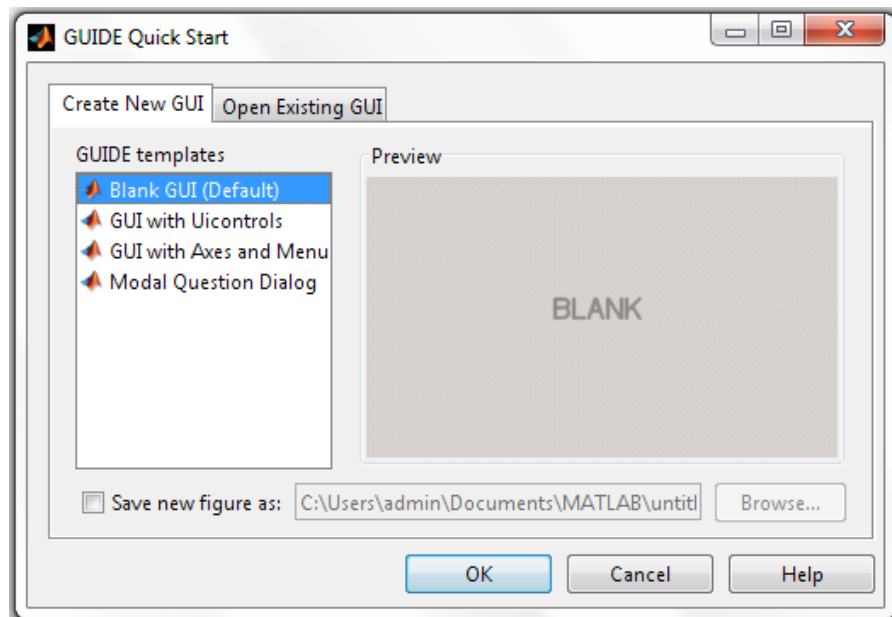
Dalam beberapa kasus pelatihan yang dilakukan memerlukan iterasi yang banyak sehingga membuat proses pelatihan menjadi lama. Untuk mempercepat iterasi dapat dilakukan dengan Parameter α atau laju pemahaman. Nilai α terletak antara 0 dan 1 ($0 \leq \alpha \leq 1$). Jika harga α Semakin besar, maka iterasi yang dipakai semakin sedikit. Akan tetapi jika harga α terlalu besar, maka akan merusak pola yang sudah benar sehingga pemahaman menjadi lambat.

Proses pelatihan yang baik dipengaruhi pada pemilihan bobot awal, karena bobot awal sangat mempengaruhi apakah jaringan mencapai titik minimum lokal atau global, dan seberapa cepat konvergensinya. Oleh karena itu dalam standar *backpropagation*, bobot dan bias diisi dengan bilangan acak kecil dan biasanya bobot awal diinisialisasi secara random dengan nilai antara -0,5 sampai 0,5 (atau -1 sampai 1 atau interval yang lainnya) (Amin, 2012).

2.6.9 Membuat Aplikasi untuk *Backpropagation Learning*

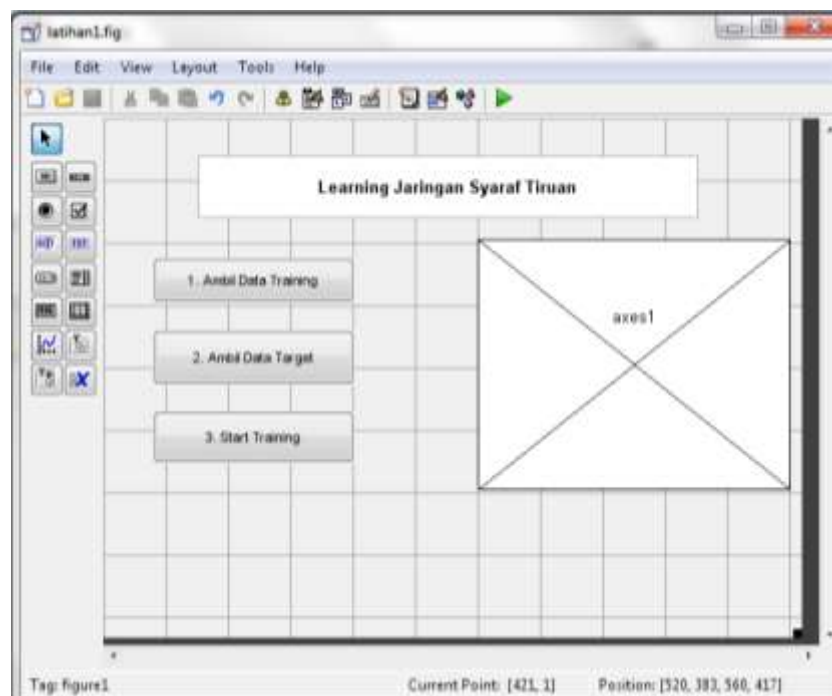
Akan dicoba membuat aplikasi yang fungsinya menghasilkan Jaringan Syaraf Tiruan yang siap digunakan untuk pendukung keputusan, maka akan dilakukan langkah seperti berikut ini.

- 1) Untuk membuat GUI dapat dilakukan dengan mengetik “*guide*” di *command window* atau mengklik “*file-new-GUI*”. Pilih Bank GUI (default) karena kita akan merancang GUI dari awal. Dapat dilihat dalam Gambar 2.10



Gambar 2.10 Membuat GUI

- 2) Rancang Form GUI seperti pada Gambar 2.11 di bawah ini:



Gambar 2.11 Hasil Rancangan GUI untuk Pelatihan

Tuliskan “**Learning Jaringan Syaraf Tiruan**” dibuat dengan cara *men-dragtool* ”*Txt*”, sedangkan untuk tombol tekan *Button* dengan *men-drag tool*


“OK”. Di sini akan dimunculkan gambar JST yang akan dilakukan proses pembelajaran (*learning*), maka kita perlu menge-*drag* simbol Axis.

- 3) Ambil data untuk menampilkan gambar Jaringan Syaraf Tiruan, gambar bisa dicari di internet. Atau buat sendiri seperti berikut ini, simpan di *folder* yang sama dengan GUI Anda.
- 4) Sesaat setelah kita simpan dan beri nama, Matlab akan memberikan fungsi *m-file* dengan nama yang sama dengan nama GUI hanya saja berekstensi *.m yang dibuka bersama dengan *m-file* Editor. Dapat dilihat pada Gambar 2.12



Gambar 2.12 Menyimpan GUI

- 5) Buka *editor Matlab*, masuk ke fungsi “1. Ambil Data Training”. Klik

icon  untuk mencari fungsi tersebut, misalnya *pushbutton 1*. Anda bisa mengganti nama pada isian “Tag” di jendela *property*.

```
datatrain=uigetfile('*.dat','ambil data pelatihan');
datatrain=load(datatrain);
handles.datatrain=datatrain;
guidata(hObject,handles
```

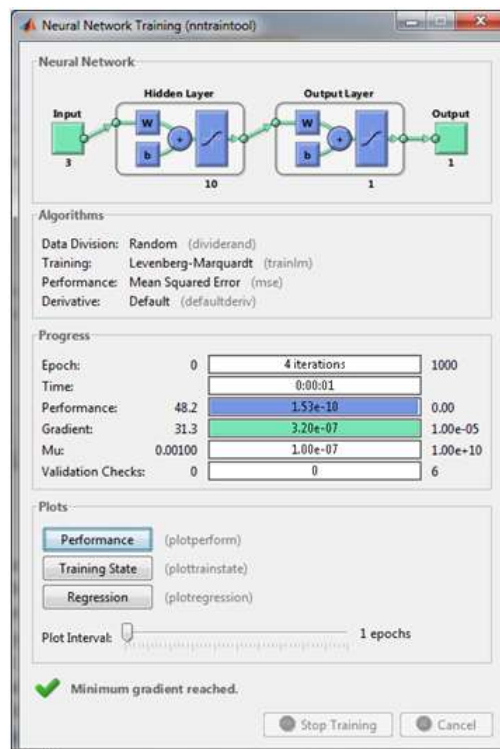
Fungsi “`uigetfile`” bermaksud mengeluarkan jendela mencari *file*. Di sini kita berikan rincian bahwa *file* yang akan dicari berekstensi `*.dat`. Dengan keterangan pada jendela “Ambil data Pelatihan”. Karena akan digunakan pada fungsi yang lain maka harus dibuatkan handles. Di sini variabel yang dibuatkan *handles*nya adalah “`handles.datatrain`”. Jadi variabel yang hanya bias digunakan pada fungsi yang lain (untuk *training*) adalah `handles.datatrain`. Instruksi terakhir merupakan instruksi yang harus diberikan sebagai aturan yang dibuat oleh matlab. Perhatikan penulisan *hObject* di mana “O” dengan huruf besar. Untuk pengambilan data target dapat dilakukan dengan cara yang sama yaitu:

```
target=uigetfile('*.dat','ambil data target');
target=load(target);
handles.target=target;
guidata(hObject,handles)
```

- 6) Perhatikan penjelasan untuk ambil data *training*. Sama saja, hanya variabelnya yang berbeda. Berikutnya untuk proses *training*, pertama-tama kita lakukan pembuatan *network* dengan fungsi “`newff`”. Dilanjutkan dengan *training* lewat fungsi “`train`”. Di sini jumlah neuron yang digunakan adalah 25 neuron untuk *Hidden Layer*-nya.

```
net=newff(handles.datatrain,handles.target,[25 1]);
net=train(net,handles.datatrain,handles.target);
simpan=uiputfile('*.mat','Simpan Hasil Pelatihan')
save(simpan)
```

- 7) Matlab akan memperlihatkan jendela proses *learning* seperti pada Gambar 2.13

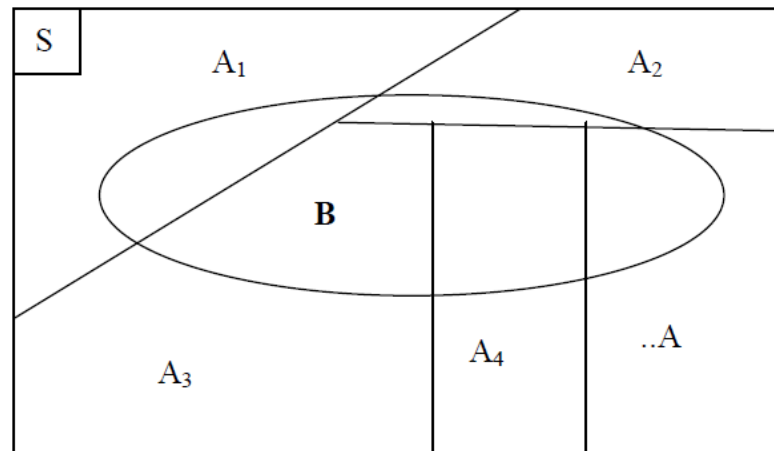


Gambar 2.13 Proses Pelatihan JST

2.7 Metode Naive Bayes

2.7.1 Teorema Bayes (soebanar, 2006:4.2)

Misal S adalah ruang sampel dari suatu eksperimen dan A_1, A_2, \dots, A_k adalah peristiwa-peristiwa di dalam S sedemikian sehingga A_1, A_2, \dots, A_k saling asing dan $\bigcup_{i=1}^k A_k = S$ dikatakan bahwa A_1, A_2, \dots, A_k membentuk partisi di dalam S . Dapat dilihat pada Gambar 2.14.



Gambar 2.14 Teorema Bayes

Jika k peristiwa A_1, A_2, \dots, A_k membentuk partisi di dalam S maka terlihat pada gambar 2.14 bahwa peristiwa-peristiwa $A_1 \cap B, A_2 \cap B, \dots, A_k \cap B$ membentuk partisi dalam B sehingga dapat ditulis $B = (A_1 \cap B) \cup (A_2 \cap B) \cup \dots \cup (A_k \cap B)$. Karena peristiwa-peristiwa di ruas kanan saling asing maka

$$P(B) = \sum_{i=1}^k P(A_i \cap B)$$

Jika $P(A_i) > 0$ untuk $i=1,2,\dots, k$ maka $P(A_i \cap B) = P(A_i)P(B|A_i)$ sehingga didapat $P(B) = \sum_{i=1}^k P(A_i)P(B|A_i)$. Misal peristiwa-peristiwa A_1, A_2, \dots, A_k membentuk partisi di dalam ruang sampel S sedemikian sehingga $A_i \cap A_j = \emptyset$; $i \neq j, i, j = 1, 2, \dots, k$ dan misalkan B sembarang peristiwa sedemikian hingga $P(B) > 0$ maka untuk $i=1,2,\dots,k$

$$P(A_i|B) = \frac{P(A_i)P(B|A_i)}{\sum_{i=1}^k P(A_i)P(B|A_i)}$$

Teorema Bayes memberikan aturan sederhana untuk menghitung probabilitas bersyarat peristiwa A_i jika B terjadi, jika masing-masing probabilitas tak bersyarat A_i dan probabilitas bersyarat B jika diberikan A_i .

2.7.2 Naive Bayes Classifier

Menurut Alfironi (2013:3), *Naive bayes classifier* mengestimasi peluang kelas bersyarat dengan mengasumsikan bahwa atribut adalah independen secara bersyarat yang diberikan dengan label kelas y . Asumsi independen bersyarat dapat dinyatakan dalam bentuk berikut :

$$P(X|Y = y) = \prod_{i=1}^d P(X_i|Y = y) \dots (1)$$

dengan tiap set atribut $X = \{X_1, X_2, \dots, X_d\}$ terdiri dari d atribut.

Sebelum menyelidiki lebih detail bagaimana *naive bayes classifier* bekerja, terlebih dahulu diuji notasi independensi bersyarat. Anggap X , Y , dan Z melambangkan tiga set variabel acak. Variabel di dalam X dikatakan independen secara bersyarat Y , yang diberikan Z , jika sesuai kondisi berikut.

$$P(X|Y, Z) = P(X|Z) \dots (2)$$

Contoh independensi bersyarat adalah hubungan panjang lengan manusia dengan kemampuan membacanya. Dapat diamati bahwa orang dengan lengan lebih panjang cenderung memiliki tingkat kemampuan membaca lebih tinggi. Hubungan ini dapat dijelaskan dengan kehadiran faktor *confounding*, yaitu usia. Seorang anak kecil cenderung memiliki lengan lebih pendek dan kemampuan membaca lebih sedikit dibanding orang dewasa. Jika usia seseorang ditetapkan, maka hubungan yang diamati antara panjang kengan dan kemampuan membaca akan hilang. Sehingga dapat disimpulkan bahwa panjang lengan dan kemampuan membaca adalah independen secara bersyarat ketika variabel usia ditetapkan.

Independensi bersyarat antara X dan Y juga dapat ditulis dalam bentuk serupa dengan Persamaan 1 :

$$\begin{aligned}
 P(X, Y|Z) &= \frac{P(X, Y, Z)}{P(Z)} \\
 &= \frac{P(X, Y, Z)}{P(Y, Z)} \times \frac{P(Y, Z)}{P(Z)} \\
 &= P(X|Y, Z) \times P(Y|Z) \\
 &= P(X|Z) \times P(Y|Z) \quad \dots (3)
 \end{aligned}$$

Persamaan 4 digunakan untuk memperoleh baris terakhir Persamaan.

Asumsi independen bersyarat, termasuk menghitung peluang bersyarat untuk setiap kombinasi X , hanya memerlukan mengestimasi peluang bersyarat untuk tiap X_i yang diberikan Y . pendekatan selanjutnya lebih praktis karena tidak mensyaratkan *training set* sangat besar untuk memperoleh estimasi peluang yang baik.

Untuk mengklasifikasi tes *record*, Naive Bayes *classifier* menghitung peluang posterior untuk tiap kelas Y :

$$P(Y|X) = \frac{P(Y) \prod_{i=1}^d P(X_i|Y)}{P(X)} \quad \dots (4)$$

$P(X)$ adalah tetap untuk seluruh Y , cukup untuk memilih kelas yang memaksimalkan istilah numerator, $P(Y) \prod_{i=1}^d P(X_i|Y)$.

Ada dua cara untuk mengestimasi peluang kelas bersyarat untuk mengestimasi atribut kontinyu pada Naive Bayes *classifiers*.

- 1) Mendiskritisasi tiap atribut kontinyu dan kemudian mengganti nilai atribut kontinyu dengan interval diskrit yang bersesuaian. Pendekatan ini mengubah atribut kontinyu ke dalam atribut ordinal. Peluang bersyarat diestimasi dengan menghitung pecahan *training record* yang dimiliki kelas y yang berada di dalam interval yang bersesuaian untuk X_i . Kesalahan estimasi tergantung pada strategi mendiskritisasi, sebagaimana halnya dengan jumlah interval diskrit. Jika jumlah interval terlalu besar, ada terlalu sedikit *training record* pada tiap interval untuk menyediakan estimasi yang *reliable* (dapat dipercaya) untuk $P(X_i|Y)$. Di sisi lain, jika jumlah interval terlalu kecil, maka beberapa interval dapat *aggregate records* dari kelas berbeda dan batas keputusan yang benar dapat hilang.
- 2) Diasumsikan bentuk tertentu distribusi peluang untuk variabel kontinyu dan mengestimasi parameter distribusi menggunakan *training* data. Distribusi *Gaussian* sering dipilih untuk merepresentasikan peluang kelas bersyarat untuk atribut kontinyu. Distribusi dikarakterisasi dengan dua parameter yaitu *mean*, μ , dan *varian*, σ^2 . Untuk tiap kelas y_j , peluang kelas bersyarat untuk atribut X_i adalah

$$P(X_i = x_i | Y = y_j) = \frac{1}{\sqrt{2\pi}\sigma_{ij}} e^{-\frac{(x_i - \mu_{ij})^2}{2\sigma_{ij}^2}} \quad \dots (5)$$

Parameter μ_{ij} dapat diestimasi berdasarkan sampel *mean* X_i (\bar{x}) untuk seluruh *training record* yang dimiliki kelas y_j . Dengan cara sama, σ^2_{ij} dapat diestimasi dari sampel varian (s^2) *training record* tersebut.

Persamaan 5 pada sisi kanan bersesuaian dengan fungsi densitas peluang, $f(X_i; \mu_{ij}, \sigma_{ij})$. Karena fungsi bernilai kontinyu, peluang bahwa variabel acak X_i mengambil nilai tertentu adalah nol. Sebagai gantinya, dihitung peluang bersyarat bahwa X_i berada pada beberapa interval, x_i dan $x_i + \epsilon$ dengan ϵ adalah konstanta kecil :

$$P(x_i \leq X_i \leq x_i + \epsilon | Y = y_j) = \int_{x_i}^{x_i + \epsilon} f(X_i; \mu_{ij}, \sigma_{ij}) dX_i$$

$$\approx f(x_i; \mu_{ij}, \sigma_{ij}) \epsilon \dots (6)$$

Karena muncul sebagai faktor pengali tetap untuk tiap kelas, maka dibatalkan ketika dinormalisasi peluang *posterior* untuk. Oleh karena itu, Persamaan masih dapat diterapkan untuk pendekatan peluang kelas bersyarat.

2.7.3 Keunggulan dan Kelemahan Metode Bayesian

Menurut Basuki (2006), Kelebihan metode Bayes sehingga digunakan dalam aplikasi ini adalah.

- 1) Metode *Find-S* tidak dapat digunakan untuk data yang tidak konsisten dan data yang bias, sehingga untuk bentuk data semacam ini salah satu metode sederhana yang dapat digunakan adalah metode bayes

- 2) Metode Bayes ini merupakan metode yang baik di dalam mesin pembelajaran berdasarkan data training, dengan menggunakan probabilitas bersyarat sebagai dasarnya

Sedangkan kelemahan dari metode Bayes adalah.

- 1) Metode Bayes hanya bisa digunakan untuk persoalan klasifikasi dengan *supervised learning* dan data-data kategorikal
- 2) Metode Bayes memerlukan pengetahuan awal untuk dapat mengambil suatu keputusan. Tingkat keberhasilan metode ini sangat tergantung pada pengetahuan awal yang diberikan

2.7.4 Pemanfaatan Metode Bayesian

Metode Naive Bayes dapat dimanfaatkan dalam beberapa bidang untuk menyelesaikan permasalahan seperti.

- 1) Menentukan diagnosa suatu penyakit berdasarkan data-data gejala (sebagai contoh hipertensi atau sakit jantung)
- 2) Mengenali buah berdasarkan fitur-fitur buah seperti warna, bentuk, rasa dan lain-lain
- 3) Mengenali warna berdasarkan fitur indeks warna RGB
- 4) Mendeteksi warna kulit (*skin detection*) berdasarkan fitur warna *chrominant*
- 5) Menentukan keputusan aksi (olahraga, art, psikologi) berdasarkan keadaan
- 6) Menentukan jenis pakaian yang cocok untuk keadaan-keadaan tertentu (seperti cuaca, musim, temperatur, acara, waktu, tempat dan lain-lain)

2.7.5 Penerapan Metode Bayes

Di sini contoh yang ambil adalah penanganan keputusan untuk bermain di luar ruangan menggunakan parameter cuaca, temperatur, dan kecepatan angin. Selengkapnya dapat dilihat seperti Tabel 2.5

Tabel 2.5 Tabel Contoh Data Naive Bayes

No	Cuaca	Temperatur	Kecepatan Angin	Berolah-raga
1	Cerah	Normal	Pelan	Ya
2	Cerah	Normal	Pelan	Ya
3	Hujan	Tinggi	Pelan	Tidak
4	Cerah	Normal	Kencang	Ya
5	Hujan	Tinggi	Kencang	Tidak
6	Cerah	Normal	Pelan	Ya

(Sumber: Basuki, 2006)

Apakah bila cuaca cerah dan kecepatan angin kencang, seseorang akan berolahraga?

Asumsi:

Y = berolah raga

X_1 = cuaca

X_2 = temperatur

X_3 = kecepatan angin

Fakta menunjukkan bahwa $P(Y = ya) = \frac{4}{6}$ maka di peroleh

$$P(Y = tidak) = \frac{2}{6}$$

$$P(X_1 = cerah | Y = ya) = 1$$

$$P(X_1 = cerah | Y = tidak) = 0$$

$$P(X_3 = kencang | Y = ya) = \frac{1}{4}$$

$$P(X_3 = \text{kencang} \mid Y = \text{tidak}) = \frac{1}{2}$$

HMAP dari keadaan ini dapat dihitung dengan:

$$\begin{aligned} P(X_1 = \text{cerah}, X_3 = \text{kencang} \mid Y = \text{ya}) \\ = \{P(X_1 = \text{cerah} \mid Y = \text{ya}).P((X_3 = \text{kencang} \mid Y = \text{ya}).P(Y = \text{ya}) \\ = \left\{(1) \cdot \frac{1}{4}\right\} \cdot \left(\frac{4}{6}\right) = \frac{1}{6} \end{aligned}$$

$$\begin{aligned} P(X_1 = \text{cerah}, X_3 = \text{kencang} \mid Y = \text{tidak}) \\ = \{P(X_1 = \text{cerah} \mid Y = \text{tidak}).P((X_3 = \text{kencang} \mid Y = \text{tidak}).P(Y = \text{tidak}) \\ = \left\{(0) \cdot \frac{1}{2}\right\} \cdot \left(\frac{2}{6}\right) = 0 \end{aligned}$$

Maka keputusan seseorang itu adalah berolahraga.

Metode ini dapat digunakan untuk data seperti pada kasus ini, ada atribut yang menggunakan data pilihan (ya atau tidak) seperti data mempunyai penyakit jantung atau tidak, menderita diabetes atau tidak, dan data numerik untuk mengolah data seperti umur, nadi, nafas, temperatur (Basuki, 2006).

2.8 Matrix Laboratory (Matlab)

Dalam deteksi dini penyakit stroke ini, diperlukan suatu *software* yang dapat membantu mengimplementasikan deteksi dini menjadi sebuah angka-angka yang selanjutnya dapat diolah menjadi suatu informasi. Dalam skripsi ini, penulis menggunakan *software Matrix Laboratory* (Matlab) untuk mencari penyelesaian masalah pengenalan pola dari suatu data yang telah disiapkan.

Menurut Iqbal (2009:2), Matlab adalah sebuah bahasa dengan (*high performance*) kinerja tinggi untuk komputasi masalah teknik. Matlab mengintegrasikan komputasi, visualisasi, dan pemrograman dalam suatu model yang sangat mudah untuk pakai dimana masalah-masalah dan penyelesaiannya diekspresikan dalam notasi matematika yang familiar. Penggunaan Matlab meliputi bidang-bidang :

- 1) Matematika dan komputasi
- 2) Pembentukan *algorithm*
- 3) Akusisi data
- 4) Pemodelan, simulasi, dan pembuatan *prototype*
- 5) Analisis data, eksplorasi, dan visualisasi
- 6) Grafik keilmuan dan bidang rekayasa

Menurut Iqbal (2009:2) sebagai sebuah sistem, MATLAB tersusun dari 5 bagian utama:

- 1) *Development Environment* merupakan sekumpulan perangkat dan fasilitas yang membantu anda untuk menggunakan fungsi-fungsi dan file-file MATLAB. Beberapa perangkat ini merupakan sebuah *graphical user interfaces* (GUI). Termasuk didalamnya adalah MATLAB desktop dan *Command Window*, *Command history*, sebuah *editor* dan *debugger*, dan *browsers* untuk melihat *help*, *workspace*, *files*, dan *search path*.
- 2) *MATLAB Mathematical Function Library* merupakan sekumpulan algoritma komputasi mulai dari fungsi-fungsi dasar seperti: *sum*, *sin*, *cos*, dan *complex*

arithmetic, sampai dengan fungsi-fungsi yang lebih kompek seperti *matrix inverse*, *matrix eigenvalues*, *Bessel functions*, dan *fast Fourier transforms*.

- 3) MATLAB *Language* merupakan suatu *high-level matrix/array language* dengan *control flow statements*, *functions*, *data structures*, *input/output*, dan *fitur-fitur object-oriented programming*. Ini memungkinkan bagi kita untuk melakukan kedua hal baik "pemrograman dalam lingkup sederhana " untuk mendapatkan hasil yang cepat, dan "pemrograman dalam lingkup yang lebih besar" untuk memperoleh hasil-hasil dan aplikasi yang komplek.
- 4) *Graphics*MATLAB memiliki fasilitas untuk menampilkan *vector* dan *matrices* sebagai suatu grafik. Didalamnya melibatkan *high-level functions* (fungsi-fungsi level tinggi) untuk visualisasi data dua dikensi dan data tiga dimensi, *image processing*, *animation*, dan *presentation graphics*.
- 5) MATLAB *Application Program Interface* (API) merupakan suatu *library* yang memungkinkan program yang telah anda tulis dalam bahasa C dan Fortran mampu berinterakasi dengan MATLAB. Ini melibatkan fasilitas untuk pemanggilan routines dari MATLAB (*dynamic linking*), pemanggilan MATLAB sebagai sebuah *computational engine*, dan untuk membaca danmenuliskan MAT-files.

2.8.1 Jendela-jendela pada Matlab

Ada beberapa macam jendela yang tersedia dalam Matlab, yang dapat dijelaskan sebagai berikut :

2.8.1.1 *Command Window/Editor*

Matlab *command window/editor* merupakan jendela di atas dapat dilakukan akses-akses ke *command-command* Matlab dengan mengetikkan barisan-barisan ekspresi Matlab, seperti mengakses *help window* dan lain-lainnya.

Command windows juga dapat digunakan untuk memanggil *tool* Matlab seperti *editor*, *debugger* atau fungsi. Ciri dari jendela ini adalah adanya *prompt* (*>>*) yang menyatakan Matlab siap menerima perintah. Perintah dapat berupa fungsi-fungsi pengaturan file (seperti perintah DOS/UNIX) maupun fungsi-fungsi bawaan atau *toolbox* Matlab sendiri.

2.8.1.2 *Current Directory*

Jendela ini menampilkan isi dari direktori kerja saat menggunakan Matlab. Direktori ini dapat diganti sesuai tempat direktori kerja yang diinginkan. *Default* dari alamat direktori berada dalam folder *works* tempat program *files* Matlab berada.

2.8.1.3 *Command History*

Jendela ini berfungsi untuk menyimpan perintah-perintah apa saja yang sebelumnya dilakukan oleh pengguna terhadap Matlab.

2.8.1.4 Editor/ Debugger (Editor M-File/ Pencarian Kesalahan)

Jendela ini merupakan tool yang disediakan oleh Matlab 5 ke atas. Berfungsi sebagai *editorscript* Matlab (M-file). Walaupun sebenarnya *script* ini untuk pemrograman Matlab, dapat saja menggunakan *editor* yang lain seperti *notepad*, *wordpad* bahkan *Microsoft Word*.

2.8.1.5 Workspace

Workspace berfungsi untuk menampilkan seluruh variabel-variabel yang sedang aktif pada saat pemakaian Matlab. Apabila variabel berupa data matriks berukuran besar maka pengguna dapat melihat isi dari seluruh data dengan melakukan *double* klik pada variabel tersebut. Matlab secara otomatis akan menampilkan jendela “*array editor*” yang berisikan data setiap variabel yang dipilih pengguna.

2.8.2 GUIDE Matlab

GUIDE atau GUI builder merupakan *Matlab script file* yang dibuat untuk analisis suatu permasalahan khusus. Penggunaan GUIDE memberikan/ menyediakan fasilitas, seperti *menu*, *pushbutton*, *slider*, dan sebagainya sesuai dengan program yang diinginkan atau digunakan tanpa *knowledge* dari Matlab. GUIDE juga memberikan cara efisien manajemen data.

2.8.2.1 Keunggulan GUIDE Matlab

GUIDE Matlab mempunyai kelebihan tersendiri dibandingkan dengan bahasa pemrograman lainnya, antara lain:

- 1) GUIDE Matlab banyak digunakan dan cocok untuk aplikasi-aplikasi berorientasi sains, sehingga banyak peneliti dan mahasiswa menggunakan GUIDE Matlab untuk menyelesaikan riset atau tugas akhirnya.
- 2) GUIDE Matlab mempunyai fungsi *built-in* yang siap digunakan dan pemakai tidak perlu repot membuatnya sendiri.
- 3) Ukuran file, baik *FIG-file* maupun *M-file* yang dihasilkan relatif kecil.
- 4) Kemampuan grafisnya cukup handal dan tidak kalah dibandingkan dengan bahasa pemrograman lainnya.

BAB III

METODE PENELITIAN

Pada penelitian ini metode atau langkah-langkah yang digunakan adalah sebagai berikut:

3.1 Studi Pustaka

Dalam studi pustaka ini digunakan sumber pustaka yang relevan digunakan untuk mengumpulkan informasi yang diperlukan dalam penelitian. Studi pustaka dengan mengumpulkan sumber pustaka yang dapat berupa buku, teks, makalah, dan sebagainya. Setelah sumber pustaka terkumpul dilanjutkan dengan penelaahan dari sumber pustaka tersebut. Pada akhirnya sumber pustaka itu dijadikan landasan untuk menganalisis permasalahan.

Dalam skripsi ini perlu adanya studi pustaka yang berkaitan dengan permasalahan yang ada, yaitu tentang apa saja gejala-gejala umum penyakit stroke, metode Jaringan Syaraf Tiruan, dan metode Naive Bayes yang akan dimanfaatkan untuk membantu menyelesaikan masalah yang ada.

3.2 Perumusan Masalah

Tahap ini dimaksudkan untuk memperjelas permasalahan sehingga mempermudah pembahasan selanjutnya. Selain itu, perumusan masalah juga menjadi sangat penting karena hal ini menjadi dasar dan tujuan akhir kenapa dilakukan penelitian ini. Masalah yang diangkat dalam penelitian ini adalah bagaimana

mengimplementasikan metode Jaringan Syaraf Tiruan dan metode Naive Bayes sebagai sistem deteksi penyakit stroke. Agar bisa dimanfaatkan sebagai upaya memberikan peringatan sedini mungkin berdasarkan pada data-data yang ada sebelumnya.

3.3 Pemecahan Masalah

3.3.1 Pengumpulan Data dan Pengolahan Data

Pada tahap ini dilakukan pengumpulan data-data baik untuk data *training* maupun data untuk pembuatan aplikasi. Teknik pengumpulan data ini adalah sebagai berikut :

a. Pengumpulan data Rekam Medis

Rekam medis adalah berkas yang berisikan catatan dan dokumen tentang pemeriksaan, pengobatan dan tindakan yang diberikan kepada pasien. Rekam medis yang dimaksud adalah tentang riwayat kesehatan yang termasuk dalam faktor risiko dan gejala penyakit stroke maupun penyakit yang mempunyai gejala dan faktor risikonya hampir mirip dengan stroke, dimana atribut dari data ditentukan menyesuaikan data rekam medis. Pengambilan data dilakukan di RSUD Tugurejo Semarang.

b. Metode Kepustakaan

Metode mengumpulkan data-data yang diperoleh dengan cara mengumpulkan data melalui buku-buku dan sumber-sumber lain (internet) yang relevan dengan permasalahan yang dihadapi, dalam hal ini tentang penyakit stroke dan faktor risikonya.

3.3.2 Perancangan, Pembuatan, Pengujian, dan Perbaikan Sistem

Metode-metode rekayasa perangkat lunak, memberikan teknik untuk membangun perangkat lunak. Berkaitan dengan serangkaian tugas yang luas yang menyangkut analisis kebutuhan, konstruksi program, desain, pengujian, dan pemeliharaan (Pressman, 2002:35). Metode-metode pengembangan perangkat lunak (model proses pengembangan perangkat lunak) yaitu metode sekuensial linier, *prototype*, RAD, *Fourth Generation Techniques* (4GT) dan spiral.

Pendekatan *prototyping* model digunakan jika pemakai hanya mendefinisikan objektif umum dari perangkat lunak tanpa merinci kebutuhan *input*, pemrosesan dan *outputnya*, sementara pengembang tidak begitu yakin akan efisiensi algoritma, adaptasi sistem operasi, atau bentuk antarmuka manusia-mesin yang harus diambil. Secara ideal *prototype* berfungsi sebagai sebuah mekanisme untuk mengidentifikasi kebutuhan perangkat lunak. Bila *prototype* yang sedang bekerja dibangun, pengembang harus menggunakan fragmen-fragmen program yang ada atau mengaplikasikan alat-alat bantu (contoh: *window manager*, dsb) yang memungkinkan program yang bekerja agar dimunculkan secara cepat (Pressman, 2002 : 36).

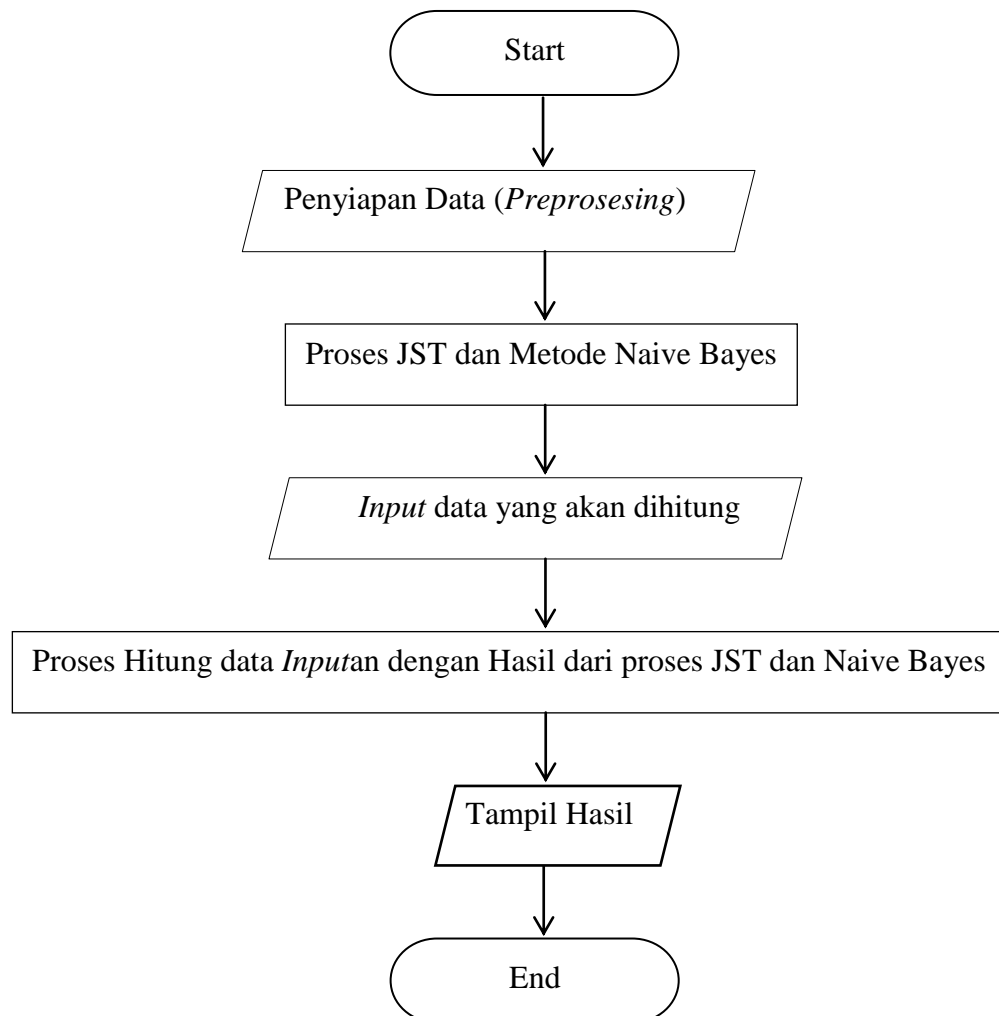
3.3.2.1 Perancangan Sistem

Setelah data-data terkait gejala-gejala stroke telah diperoleh, langkah selanjutnya dalam merancang sistem deteksi penyakit stroke. Sistem deteksi penyakit stroke ini dirancang sebagai tahap awal membuat sistem. Pada dasarnya tahap perancangan sistem ini merupakan kombinasi dari komponen-komponen

perangkat lunak dan perangkat keras dari sistem. Ada beberapa tahap yang dibuat dalam perancangan sistem ini adalah perancangan sistem pelatihan, arsitektur jaringan, dan perancangan sistem deteksi.

Rancangan sistem adalah rancangan dimana sistem dalam penelitian ini berjalan. Gambar 3.1 merupakan rancangan sistem yang akan dibuat. Rancangan sistem dalam proyek ini secara garis besar terdapat dalam tahap-tahap berikut.

- (1) Penyiapan Data
- (2) Proses Jaringan Syaraf Tiruan dan Metode Naive Bayes
- (3) *Inputan* data yang akan diproses
- (4) Penghitungan data *input* dengan nilai hasil dari masing-masing metode
- (5) Mendesain tampilan hasil



Gambar 3.1 Diagram Rancangan Sistem

3.3.2.2 Pembuatan Sistem

Dari tahap perancangan model, dilakukan realisasi pada perangkat lunak. *Inputan* yang digunakan yaitu berupa *inputan* dari form yang sudah disediakan pada program Matlab. Isi dari form tersebut memuat pertanyaan maupun isian yang harus diisi sesuai dengan aturan yang sudah dituliskan pada halaman tersebut. Dari form tersebut dapat diambil nilai-nilai yang akan diproses dengan data *training* yang ada kemudian akan muncul *ranking* kemungkinan penyakit

yang disebutkan di atas serta memberikan saran mengenai gejala dan faktor risiko yang di derita.

3.3.2.2.1 Rancangan *Input*

Rancangan *input* merupakan contoh data-data yang akan menjadi nilai *inputan* dari Jaringan Syaraf Tiruan maupun metode Naive Bayes. Untuk Jaringan Syaraf Tiruan data berupa numerik di transformasi menjadi data dengan besar antara nilai 0-1 menyesuaikan dengan nilai data bukan data numerik dengan menggunakan transformasi data tetapi untuk metode Naive Bayes data berupa numerik akan tetap diolah dalam numerik. Tabel 3.1 merupakan contoh *inputan* dan nilai yang akan diubah menyesuaikan database yang sudah dibuat.

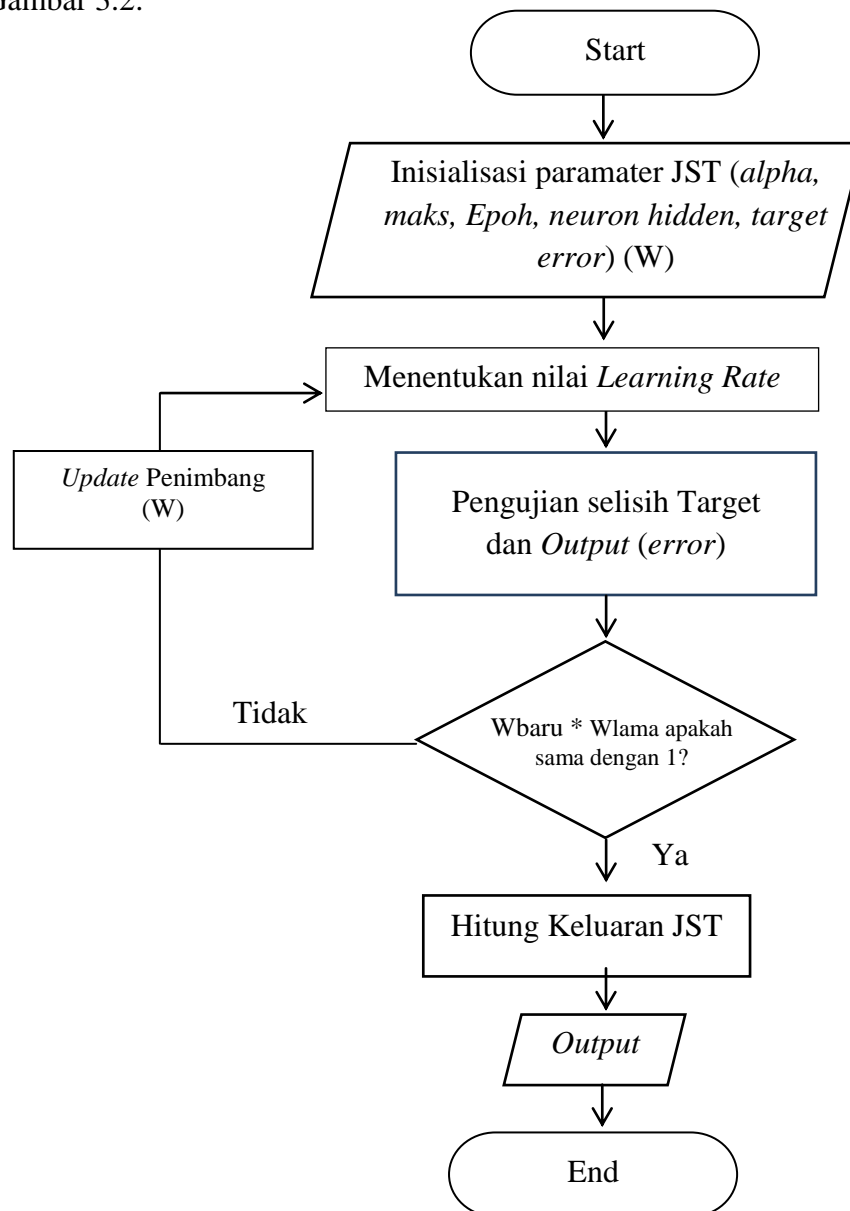
Tabel 3.1 Tabel Contoh Data Masukan

Variabel	Nilai ketika diproses
Umur	-
Nadi	-
Jenis Kelamin	(Laki-laki : 1, Perempuan : 0)
Kolesterol	-
Hipertensi	(Iya : 1, Tidak : 0)
Diabetes Militus	(Iya : 1, Tidak : 0)
Kesadaran	(Iya : 1, Tidak : 0)
Pusing	(Iya : 1, Tidak : 0)
Mual	(Iya : 1, Tidak : 0)
Kesemutan	(Iya : 1, Tidak : 0)
Riwayat Penyakit Stroke	(Iya : 1, Tidak : 0)

3.3.2.2.2 Metode Jaringan Syaraf Tiruan

Metode ini berfungsi untuk mencari nilai penimbang (W) faktor risiko sehingga nantinya hasilnya akan dijadikan perhitungan untuk menentukan apakah

Suspectstroke atau tidak. Diagram alir Jaringan Syaraf Tiruan dapat dilihat dalam Gambar 3.2.



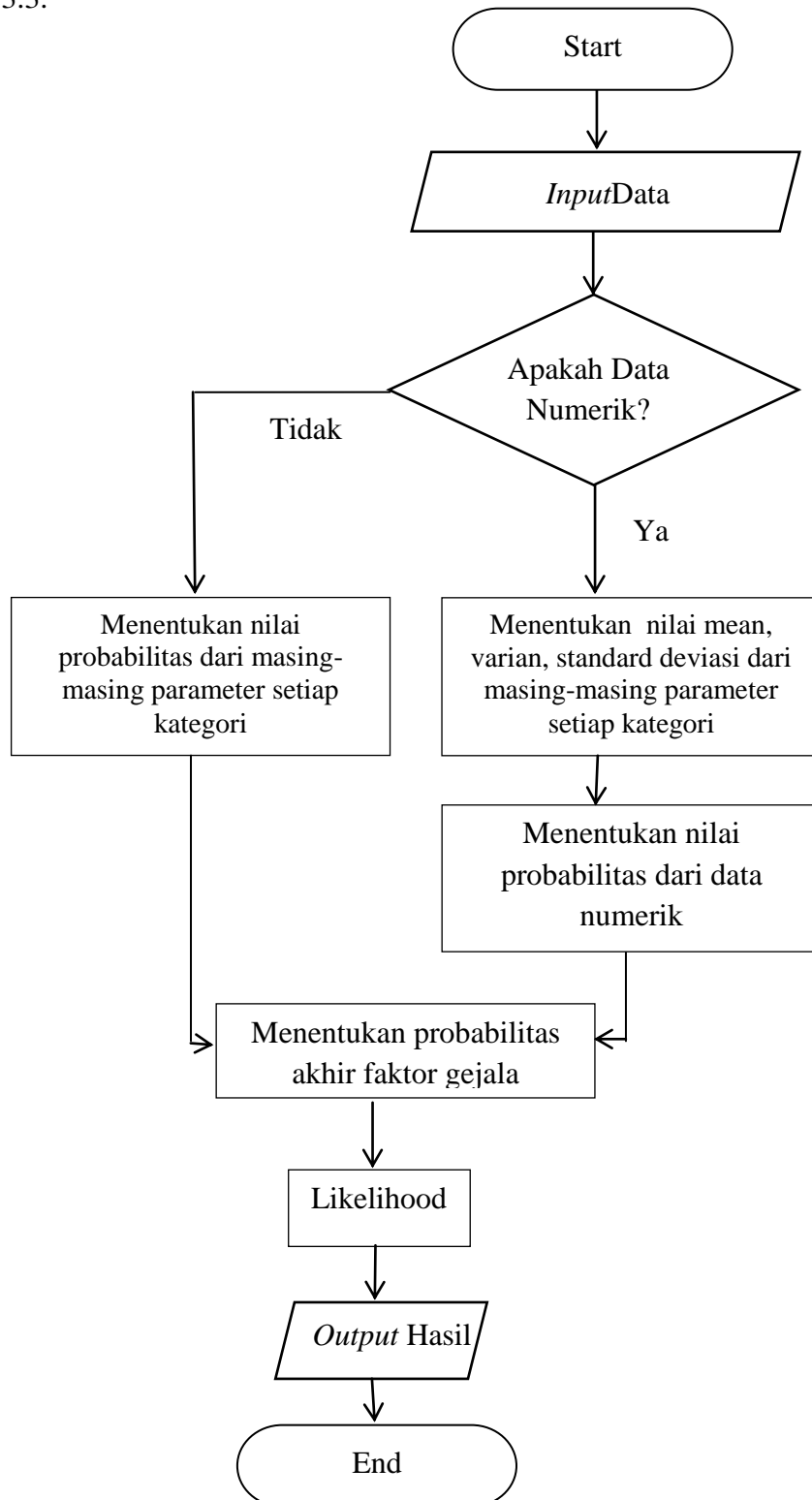
Gambar 3.2 Diagram Alir Jaringan Syaraf Tiruan

3.3.2.2.3 Metode Naive Bayes

Metode ini berfungsi untuk mencari nilai probabilitas tiap faktor risiko sehingga nantinya hasilnya akan dijadikan perhitungan untuk menentukan apakah

Suspectstroke atau tidak. Diagram alir Naive Bayes dapat dilihat dalam Gambar

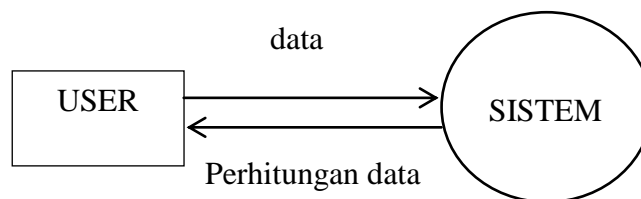
3.3.



Gambar 3.3 Diagram Alir Naive Bayes

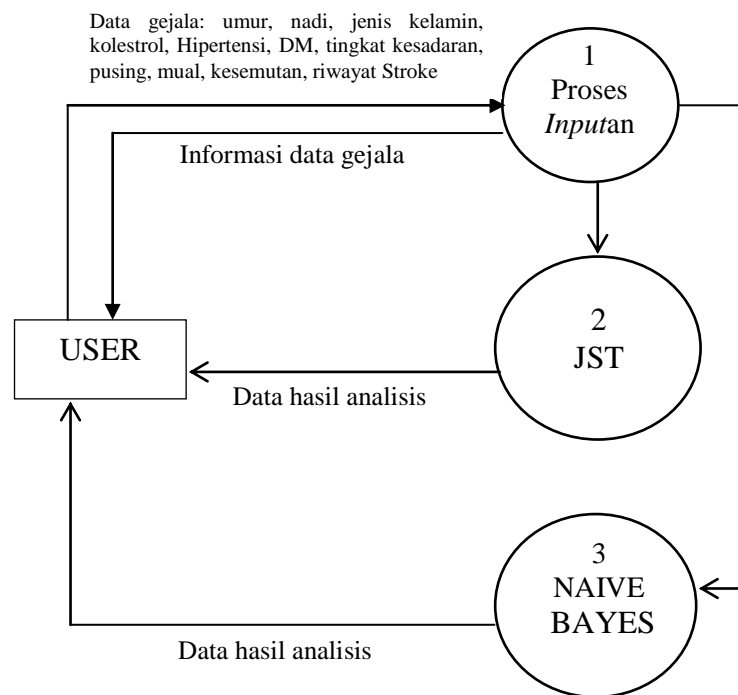
3.3.2.2.4 DFD (*Data Flow Diagram*)

Data Flow Diagram (DFD) merupakan suatu cara atau metode untuk membuat rancangan sebuah sistem yang mana berorientasi pada alur data yang bergerak pada sebuah sistem nantinya. DFD ini digunakan untuk melakukan proses *coding*. DFD dari sistem yang akan dibuat dalam skripsi ini dapat dilihat dalam Gambar 3.4. Pada Diagram tersebut memperlihatkan diagram awal *user* akan memasukkan data dan akan di proses oleh sistem sehingga menghasilkan perhitungan data.

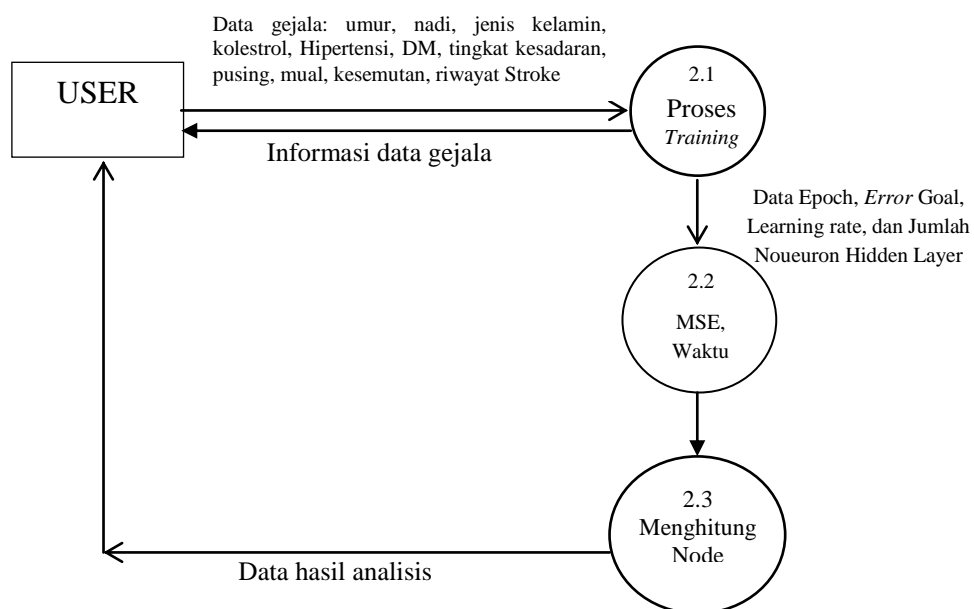


Gambar 3.4 Diagram Konteks (DFD level 0)

Dari Gambar 3.4 kemudian diperinci pada Gambar 3.5 dalam DFD level 1. Pada gambar tersebut memperlihatkan bahwa *user* akan memasukkan data gejala pasien. Data gejala tersebut meliputi umur, nadi, jenis kelamin, perhitungan kolesterol, Hepertensi yang diderita pasien, Diabetes Militus, tingkat kesadaran, pusing, mual, kesemutan, dan riwayat penyakit stroke. Dari data tersebut kemudian *user* memilih pemrosesan menggunakan pengujian metode JST atau metode Naive Bayes. Dari kedua metode tersebut akan menghasilkan data hasil analisis.

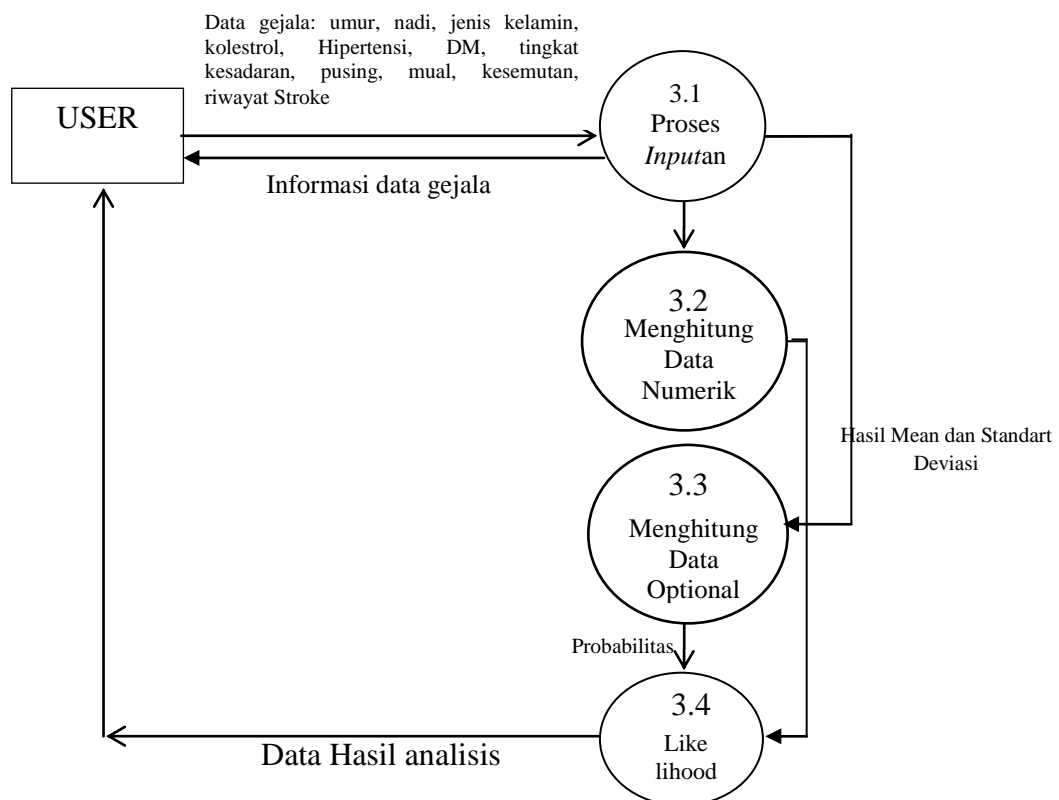


Gambar 3.5 *Data Flow Diagram* level 1



Gambar 3.6 *Data Flow Diagram* level 2 proses Jaringan Syaraf Tiruan

Pada Gambar 3.6 adalah DFD level 2 memperlihatkan proses dari Jaringan Syaraf Tiruan. Di dalam gambar *user* akan *menginputkan* gejala pasien terkena stroke dan kemudian dilakukan proses *training* dengan menentukan nilai dari *Maksimum Epoch*, *Error Goal*, *Learning rate*, dan *Jumlah Neuron Hidden Layer* yang akan dihitung. Dari proses *training* tersebut akan menghasilkan MSE (*Mean Squared Error*) dan waktu *training*. Dari situ sistem akan muncul *Output* bobot (*Node*) yang akan digunakan dalam perhitungan *Suspect* stroke atau tidak.



Gambar 3.7 Data Flow Diagram level 2 proses Naive Bayes

Pada Gambar 3.7 memperlihatkan DFD level 2 proses Naive Bayes. Di dalam gambar *user* akan *menginputkan* gejala pasien terkena stroke dan kemudian dilakukan proses pemisahan karena dalam *inputan* gejala terdapat data numerik

dan data opsional. Dalam perhitungan data numerik yang dihasilkan adalah *Mean* dan Standart Deviasi. Sedangkan dalam proses perhitungan data opsional akan menghasilkan nilai probabilitas. Kemudian dari hasil keseluruhan data akan dilakukan Likelihood. Dari hasil Likelihood tersebut akan menghasilkan analisis Naive Bayes.

3.3.2.3 Pengujian Sistem

Untuk mengetahui tingkat keakuratan perangkat lunak yang dibuat, maka pada tahap pengujian ini akan dilakukan dengan cara membandingkan nilai *error* dari hasil percobaan metode Jaringan Syaraf Tiruan dan metode Naive Bayes. Sehingga analisisnya berupa keterangan alasan kenapa salah satu metode tersebut bisa lebih baik dari metode satunya.

3.3.2.4 Perbaikan

Setelah sistem diuji coba dan terjadi beberapa kekurangan/ kelemahan, maka perlu diperbaiki sehingga sistem yang dibuat lebih baik.

3.4 Penarikan Kesimpulan

Langkah ini merupakan langkah terakhir dari penelitian. Penarikan kesimpulan didasarkan pada studi pustaka dan pembahasan permasalahan serta merupakan hasil analisis dari penelitian. Simpulan yang diperoleh adalah tentang bagaimana cara sistem deteksi penyakit stroke dengan metode Jaringan Syaraf Tiruan dan metode Naive Bayes. Sehingga dapat diketahui besar tingkat akurasi sistem sistem yang dihasilkan pada proses mendeteksi penyakit stroke.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil Penelitian

4.1.1 Tahap Pengambilan Data

Pada penelitian ini, dibuat sistem deteksi untuk mengetahui *Suspect* atau tidaknya pasien menderita penyakit stroke di RSUD Tugurejo Semarang. Untuk membuatnya diperlukan data-data terkait gejala-gejala umum penyakit stroke yang meliputi umur, nadi, jenis kelamin, perhitungan kolesterol, Hepertensi yang diderita pasien, Diabetes Militus, tingkat kesadaran, pusing, mual, kesemutan, dan riwayat penyakit stroke. Data-data tersebut digunakan untuk melakukan pelatihan program dan sebagai data latih dan data uji untuk sistem. Sebagai *outputnya* merupakan prediksi seseorang *Suspectstroke* atau tidak.

Data-data yang digunakan untuk melakukan pelatihan program dan sebagai data latih serta data uji diambil secara acak melalui data rekam medis setiap pasientahun 2013 di RSUD Tugurejo Semarang. Jumlah data yang diambil adalah 120 pasien yang digunakan sebanyak 89 pasien *Suspectstroke* dan 31 pasien *nonstroke*.

4.1.2 Pendefisian *Input* dan *Target*

Data gejala-gejala yang dialami oleh pasien selanjutnya akan diolah oleh jaringan. Supaya data dapat dikenali oleh jaringan, maka data harus

direpresentasikan ke dalam bentuk numerik antara 0 sampai dengan 1 jika data tersebut opsional, baik variabel maupun isinya yang merupakan masukan gejala penyakit stroke beserta kategori dan keluaran yang merupakan prediksi penyakit stroke. Nilai-nilai yang digunakan diperoleh berdasarkan kategori dari masing-masing variabel selain juga untuk memudahkan mengingat dalam pendefinisianannya.

4.1.2.1 Pendefisian Input

Gejala-gejala penyakit stroke diubah ke dalam variabel sedangkan kategori dari masing-masing gejala tersebut diubah ke bentuk numerik, yaitu:

- 1) Umur (X_1)
- 2) Nadi (X_2)
- 3) Jenis Kelamin (X_3) dengan kategori :
 - Laki-laki = 1
 - Perempuan = 0
- 4) Kolesterol (X_4)
- 5) Hipertensi (X_5) dengan kategori :
 - Menderita Hipertensi = 1
 - Tidak Hipertensi = 0
- 6) Diabetes Militus (X_6) dengan kategori :
 - Menderita Diabetes Militus = 1
 - Tidak Diabetes Militus = 0
- 7) Kesadaran (X_7) dengan kategori :

Pingsan = 1

Tidak Pingsan = 0

8) Pusing (X_8) dengan kategori :

Pusing = 1

Tidak Pusing = 0

9) Mual (X_9) dengan kategori :

Mual = 1

Tidak mual = 0

10) Kesemutan (X_{10}) dengan kategori :

Mengalami kesemutan = 1

Tidak mengalami kesemutan = 0

11) Riwayat Penyakit Stroke (X_{11}) dengan kategori :

Pernah mengalami penyakit Stroke = 1

Tidak pernah mengalami Stroke = 0

4.1.2.2 Penetapan Target

Hasil yang ingin diperoleh pada tahap ini, yaitu terdeteksinya suatu nilai untuk memprediksi seseorang *Suspectstroke* atau tidak. Hasil yang maksud adalah sebagai berikut.

- 1) Jika *output* bernilai 0 berarti pasien tidak *Suspectstroke*.
- 2) Jika *output* bernilai 1 berarti pasien *Suspectstroke*.

Tabel 4.1 Contoh Data Pelatihan Pasien dengan 80 Data

Pasien ke-	Variabel gejala penyakit Stroke											Target
	X ₁	X ₂	X ₃	X ₄	X ₅	X ₆	X ₇	X ₈	X ₉	X ₁₀	X ₁₁	
1	58	88	1	257	1	1	1	1	1	1	1	1
2	73	84	0	277	1	1	1	1	1	1	0	1
3	59	113	1	243	1	1	1	0	1	0	1	1
4	75	75	1	280	1	1	1	0	0	1	0	1
5	74	88	1	300	1	1	1	1	1	1	0	1
6	46	93	0	297	1	0	1	1	1	1	0	1
7	61	80	1	249	1	1	1	1	1	1	0	1
8	56	88	0	261	1	1	1	1	1	1	1	1
9	40	82	0	291	1	1	1	1	1	1	0	1
10	66	90	0	260	1	1	1	0	1	1	0	1
11	72	84	1	245	1	0	1	1	0	1	0	1
12	42	88	0	251	1	1	1	0	0	1	0	1
13	76	98	0	301	1	1	1	1	0	1	0	1
14	62	78	1	302	1	1	1	1	0	1	0	1
15	60	80	1	283	1	1	1	1	1	1	1	1
16	60	80	1	260	1	1	1	0	0	1	1	1
17	59	78	0	288	1	1	1	1	1	1	0	1
18	61	84	1	297	1	1	1	1	0	1	0	1
19	75	82	1	280	1	1	1	1	1	1	0	1
20	62	80	0	271	1	1	1	1	1	1	1	1
21	47	78	1	268	1	1	1	1	1	1	1	1
22	40	79	0	250	0	0	1	1	1	1	1	1
23	61	100	1	271	1	1	1	1	1	1	0	1
24	64	80	1	302	1	1	1	1	1	1	0	1
25	42	90	0	289	1	1	1	1	1	1	0	1
26	50	80	0	278	1	1	1	1	1	1	0	1
27	54	120	1	260	1	1	1	1	1	1	0	1
28	49	68	1	298	1	1	1	0	0	1	0	1
29	67	90	1	267	1	0	1	1	0	1	0	1
30	55	80	1	150	1	0	1	1	1	1	0	1
31	55	107	1	265	1	0	1	1	1	1	0	1
32	61	94	1	277	1	1	1	1	1	1	0	1
33	64	81	1	268	1	1	1	1	1	1	0	1
34	58	70	1	277	1	1	1	1	1	1	0	1
35	60	84	1	280	1	1	1	1	1	1	0	1
36	60	100	1	281	1	1	1	1	1	1	0	1
37	60	92	1	280	1	1	1	1	1	1	0	1
38	64	105	1	159	1	0	1	1	0	1	1	1
39	60	100	0	244	1	0	1	1	1	1	0	1
40	71	88	1	251	0	1	1	1	1	1	0	1
41	37	132	1	255	1	1	1	1	1	1	0	1
42	55	102	1	281	1	0	1	0	1	1	0	1
43	69	80	0	269	1	1	1	1	1	1	0	1
44	75	133	1	341	1	1	1	1	0	1	0	1
45	53	88	1	141	1	0	1	1	1	1	0	1
46	70	60	0	271	1	1	1	1	1	1	0	1
47	70	113	0	270	1	1	1	0	0	1	0	1
48	54	87	0	278	1	1	1	1	1	1	0	1
49	58	64	0	298	1	1	1	0	1	1	0	1
50	80	108	1	305	1	1	1	1	1	1	1	1
51	76	90	1	311	0	1	1	1	1	1	0	1
52	40	76	1	345	1	1	1	1	1	1	0	1
53	65	71	1	313	1	1	1	1	1	1	0	1
54	55	90	1	291	1	1	1	1	1	1	0	1
55	56	74	0	304	1	1	1	1	1	1	0	1
56	47	83	1	278	1	1	1	1	1	1	0	1
57	80	84	0	250	1	1	1	1	0	1	0	1
58	67	80	1	257	0	1	1	1	1	1	0	1
59	55	100	1	281	1	0	1	1	1	1	0	1
60	66	70	0	245	1	1	1	1	0	1	0	1
61	47	80	1	310	1	1	1	1	1	1	0	1
62	68	84	1	256	1	1	1	1	1	1	0	1
63	65	115	1	267	1	1	1	1	1	1	0	1
64	54	80	1	312	1	0	1	1	1	1	0	1
65	55	89	0	300	1	1	1	1	1	1	0	1
66	55	100	0	268	1	1	1	1	1	1	1	1
67	52	84	1	270	1	1	1	1	1	1	0	1
68	73	84	1	250	1	0	1	1	1	0	1	1
69	60	72	1	251	0	1	1	1	1	1	0	1
70	45	145	1	253	1	0	1	0	0	1	0	1
71	61	88	0	162	1	0	1	0	0	0	0	0
72	56	94	0	138	1	0	1	1	1	1	0	0
73	70	80	1	200	1	0	0	0	0	0	0	0
74	50	50	1	161	1	0	1	1	0	0	0	0
75	57	96	0	158	1	0	1	1	1	1	0	0
76	48	80	1	155	1	0	1	0	0	0	0	0
77	22	84	1	148	1	0	1	1	1	0	0	0
78	55	80	1	138	1	0	1	1	1	0	0	0
79	56	85	1	150	1	0	1	0	1	0	0	0
80	61	80	1	148	0	1	1	1	1	0	0	0

Pada Tabel 4.1 menunjukkan data pelatihan yang akan dipakai dalam analisis data yang dijadikan acuan sebagai data pelatihan di dalam Jaringan Syaraf Tiruan dan sebagai perhitungan acuan probabilitas dalam metode Naive Bayes. Data pelatihan yang digunakan berjumlah 80 data pasien. Sedang pada Tabel 4.2 adalah Data yang akan digunakan untuk menganalisis pengujian. Data akan diuji satu persatu dengan membandingkan antara Jaringan Syaraf Tiruan dan Naive Bayes. Data pengujian yang digunakan berjumlah 20 data pasien.

Tabel 4.2 Contoh Data Pengujian Pasien dengan 20 Data

Pasien ke-	Variabel gejala penyakit Stroke											Target
	X ₁	X ₂	X ₃	X ₄	X ₅	X ₆	X ₇	X ₈	X ₉	X ₁₀	X ₁₁	
1	77	112	1	300	1	0	0	1	1	1	1	0
2	69	81	0	200	1	1	1	1	0	1	0	1
3	67	75	0	244	1	0	1	1	0	1	0	1
4	42	90	1	311	1	1	1	1	1	1	0	1
5	64	94	0	180	0	0	1	1	1	0	0	0
6	69	84	1	290	1	1	1	1	1	1	0	1
7	46	92	1	291	1	1	1	1	0	1	0	1
8	47	83	0	278	1	0	1	1	0	1	0	1
9	68	88	0	168	1	0	1	0	1	1	0	1
10	75	75	0	175	1	1	0	0	1	0	0	0
11	45	100	1	158	1	0	1	1	1	0	1	0
12	40	92	1	155	0	0	0	1	1	0	0	0
13	46	82	0	155	1	0	1	0	0	1	0	0
14	55	79	0	155	1	0	1	0	0	0	0	0
15	54	135	0	148	0	0	0	1	1	0	0	0
16	76	100	0	145	0	0	0	1	1	0	0	0
17	54	80	0	150	0	1	1	0	1	1	0	0
18	54	90	0	167	1	1	1	1	1	0	0	0
19	57	88	0	200	1	1	0	1	1	1	0	0
20	44	88	1	270	1	1	1	1	0	1	0	0

*) Data pengujian lebih lengkap terdapat pada lampiran 8

4.2 Tahap Pengolahan Data

Dari data di atas, dilakukan pembuatan data latih dan data target yang digunakan sebagai *input* dan sistem. Data dengan gejala-gejala umum penyakit stroke akan dijadikan sebagai data “*Suspectstroke*” jika target 1 dan “*Negatif stroke*” jika target 0. Kedua data tersebut disimpan dengan format *excel*. Data yang telah disimpan dalam format *excel* akan *diconvert* menjadi format *.m* melalui *workspace* pada Matlab dengan cara *import* data.

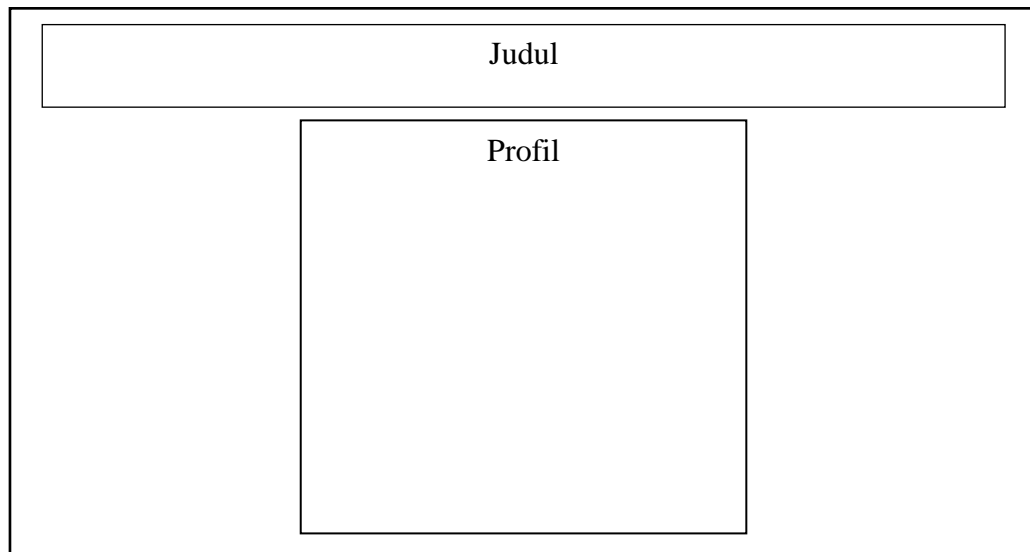
4.3 Perancangan Desain Sistem

Rancangan antar muka adalah rancangan tentang tampilan yang akan dilihat oleh *user* dari penelitian ini. Adapun rancangan tampilan yang akan dibuat pada penelitian ini adalah sebagai berikut:

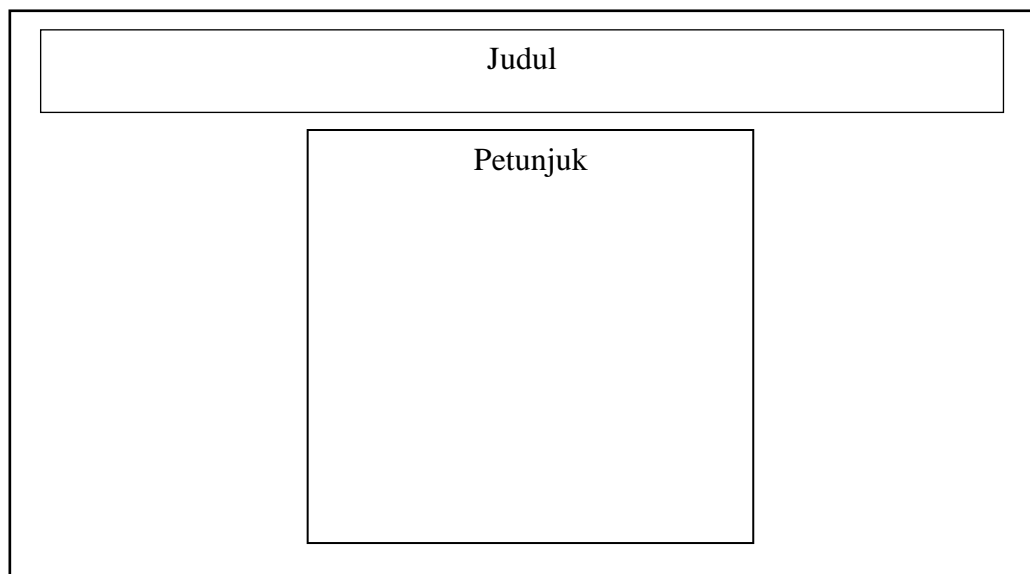
4.3.1 Tampilan Halaman Utama

Dalam halaman utama terdapat tiga bagian kolom yaitu meliputi.

- 1) Kolom judul digunakan untuk menampilkan judul program yang dibuat.
- 2) Kolom profil digunakan untuk menampilkan profil dari pembuat program.
- 3) Kolom petunjuk digunakan untuk menampilkan petunjuk penggunaan program tersebut.

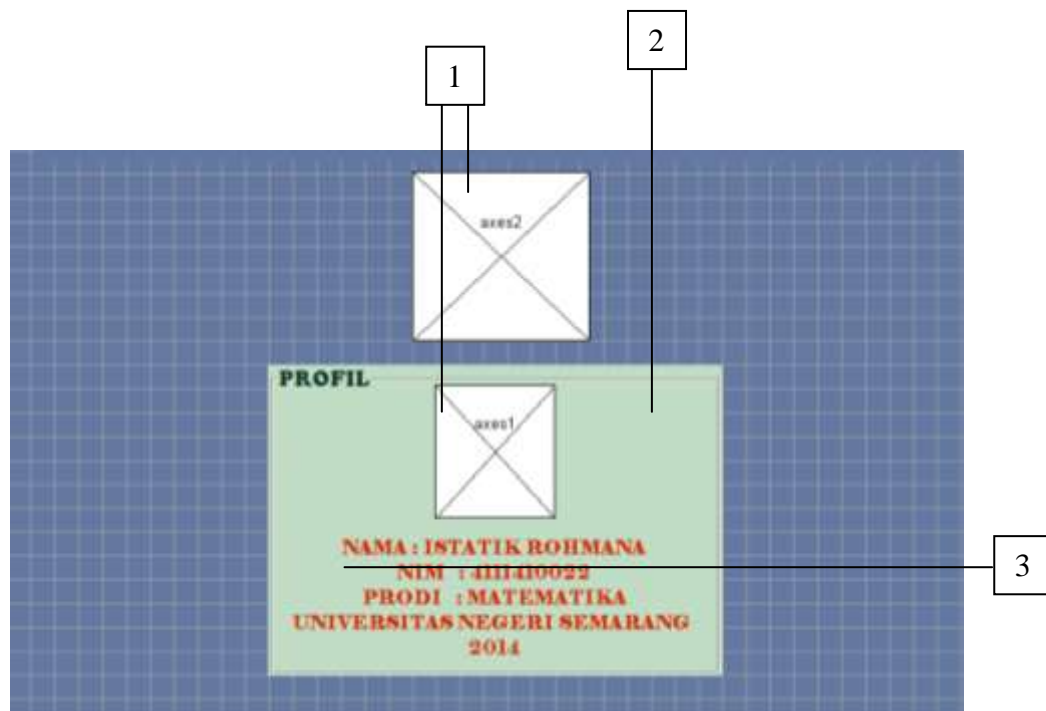


Gambar 4.1 Rancangan Halaman Utama

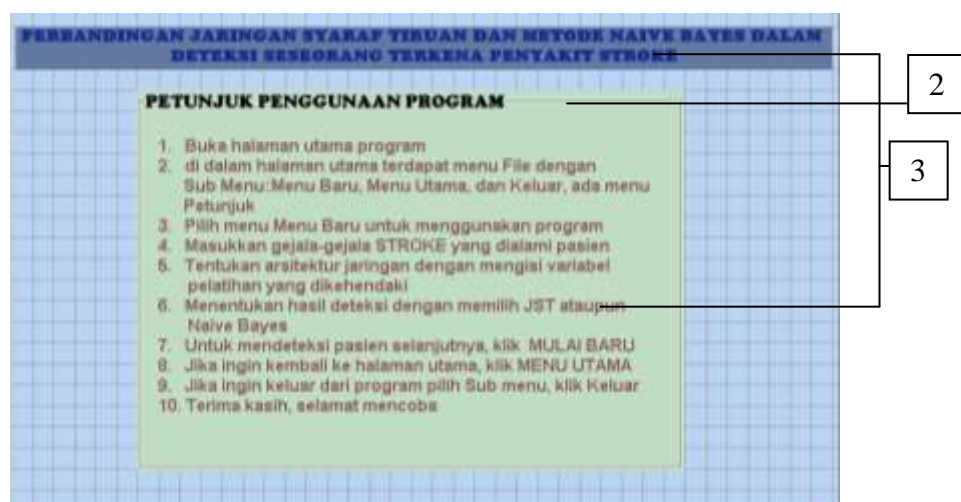


Gambar 4.2 Rancangan Halaman Petunjuk Penggunaan

Dari rancangan pada Gambar 4.1 dan Gambar 4.2 sesuai kegunaan dapat dilihat desain sebenarnya yang lebih jelas pada desain tampilan halaman yang digunakan untuk menghubungkan ke form *inputan* deteksi stroke sebagaimana ditunjukkan pada Gambar 4.3. Dan Gambar 4.4.



Gambar 4.3 Desain Tampilan Halaman Utama



Gambar 4.4 Desain Tampilan Petunjuk Penggunaan

Keterangan Gambar 4.3 dan Gambar 4.4

No.	Nama Komponen	Properti	Setting
1.	<i>Axes</i>	Tag	axes1, axes2
2.	<i>Panel</i>	Tag	uipanel1, uipanel2
3.	<i>Static text</i>	Tag	text3, text4, text5

*) Penjelasan lebih lanjut terdapat pada lampiran 1

4.3.2 Tampilan Form *Inputan* Deteksi

Dalam tampilan form *inputan* deteksi ini terdapat empat kolom sebagai berikut.

- 1) Kolom judul digunakan untuk menampilkan judul program yang dibuat.
- 2) Kolom form *inputan* digunakan untuk menampilkan form untuk *inputan user* sebagai hasil syarat untuk mengetahui *Suspectstroke* atau tidak.
- 3) Kolom perhitungan digunakan untuk menghitung analisis JST yang berupa *Maksimum Epoch*, *Error Goal*, *Learning Rate* dan jumlah *Neuron Hidden Layer*.
- 4) Kolom hasil digunakan untuk menampilkan form hasil JST dan Naive Bayes sesuai perintah.

The diagram illustrates the layout of the Detection Form Input. It is structured as follows:

- Judul**: A rectangular box at the top for the program title.
- Form *Inputan***: A large rectangular box below the title for user input.
- PERHITUNGAN**: A rectangular box below the input form for calculations.
- JST** and **NAIVE BAYES**: Two separate rectangular boxes positioned side-by-side below the calculation section.
- HASIL**: A rectangular box at the bottom, centered, for the final results.

Gambar 4.5 Rancangan Tampilan Form *Inputan* Deteksi

Dari rancangan pada Gambar 4.5 sesuai kegunaan dapat dilihat desain sebenarnya pada desain tampilan form deteksi penyakit stroke yang akan digunakan sebagai *input* variabel adalah sebagaimana pada Gambar 4.6

The image shows a software interface for stroke detection. It is divided into several sections:

- Data Input:** Contains fields for 'Usia' (Age), 'Jenis Kelamin' (Gender), 'Golongan Darah' (Blood Type), 'Tinggi Badan' (Body Height), 'Berat Badan' (Body Weight), 'Tekanan Darah' (Blood Pressure), 'Gula Darah' (Blood Sugar), 'Riwayat Stroke' (Stroke History), and 'Riwayat Penyakit Lain' (Other Diseases). Each field has a corresponding input box and a label.
- Perhitungan (Calculation):** Contains fields for 'Epoch', 'Learning Rate', 'Goal', and 'Jumlah Layer & Neuron'. There are also buttons for 'View Parameter JST' and 'Training'.
- Output:** A section labeled 'JST' with a 'Matrik Output' field and a 'Matrik Error' field.
- Footer:** A box containing the text 'KETERANGAN : 1YA = 1 dan TIDAK=0' and 'Jumlah Layer & Neuron = 1 dan 1000'.

 Numbered callouts point to specific elements:

- 1: Points to the 'Data Input' section header.
- 2: Points to the 'Perhitungan' section header.
- 3: Points to the 'Output' section header.
- 4: Points to the 'JST' section header.
- 5: Points to the 'Matrik Output' field.
- 6: Points to the 'KETERANGAN' box.
- 7: Points to the 'Matrik Error' field.

Gambar 4.6 Desain Tampilan Form Deteksi Penyakit Stroke

Keterangan Gambar 4.6

No.	Nama komponen	Properti	Setting
1	Static text	Tag	text1, text2, text3, text4, text5, text6, text7, text8, text9, text10, text11, text13, text14, text15, text16, text22, edit1, edit2, edit3, edit4, edit5, edit6, edit7, edit8, edit9, edit10, edit11, edit12, edit13, edit14, edit16, edit 18, uipanel1, unpanel2
2	Edit text	Tag	pushbutton1, pushbutton3, pushbutton5, pushbutton2
3	Panel	Tag	edit15
4	Pushbutton	Tag	text17, text18, text19
5	Edit	Tag	popupmenu1, popupmenu3, popupmenu4, popupmenu6, popupmenu7, popupmenu8, popupmenu9, popupmenu10
6	Static text	Tag	
7	Pop-up Menu	Tag	

*) Penjelasan lebih lanjut terdapat pada lampiran 3

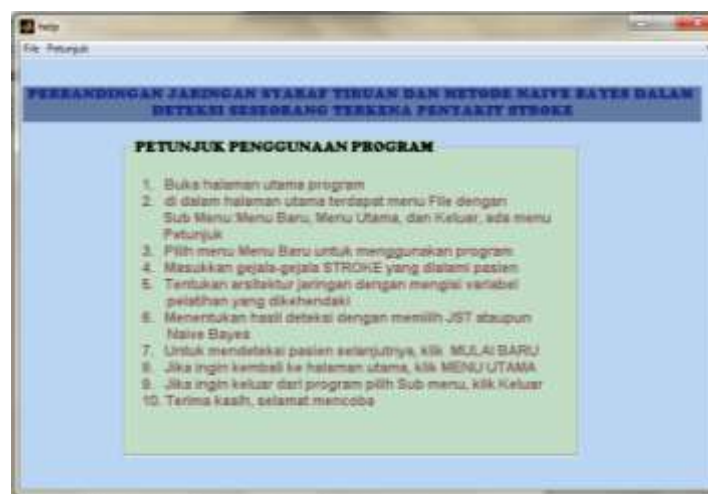
4.4 Tahap Implementasi Sistem

Setelah desain sistem dan *settingnya* dibuat, maka langkah selanjutnya yaitu mengimplementasikan sistem secara keseluruhan dengan menggunakan *software* Matlab.

4.4.1 Tampilan *Form* Halaman Utama



Gambar 4.7 Form Halaman Utama



Gambar 4.8 Form Petunjuk Penggunaan

Gambar 4.7 dan Gambar 4.8 merupakan tampilan awal dari form program agar mempermudah pengguna dalam mendeteksi penyakit stroke. Di dalam form

Naive Bayes, dan Mulai Baru. Berikut adalah penjelasan dari masing-masing *pushbutton*.

1) View Parameter JST

Command button View Parameter JST (pushbutton1) berfungsi untuk memunculkan parameter JST yang berupa *Epoch*, *learning rate*, jumlah *layer* & *neuron*, dan *goal* jika ingin ditampilkan dalam form. Berikut adalah *source code* Matlab yang menjalankan perintah *command button View Parameter JST (pushbutton1)*.

```
function pushbutton1_Callback(hObject, eventdata, handles)
set(handles.text13, 'visible', 'on')
set(handles.text14, 'visible', 'on')
set(handles.text15, 'visible', 'on')
set(handles.text16, 'visible', 'on')

set(handles.edit12, 'visible', 'on')
set(handles.edit13, 'visible', 'on')
set(handles.edit14, 'visible', 'on')
set(handles.edit16, 'visible', 'on')
set(handles.pushbutton3, 'enable', 'on')
```

2) Training

Command button training (pushbutton3) berfungsi untuk melakukan proses perhitungan sistem dari nilai maksimum *epoch*, *error goal*, *learning rate*, jumlah *neuron hidden layer* yang telah diinputkan. Berikut adalah *source code* Matlab yang menjalankan perintah *command button train (pushbutton3)*.

```
function pushbutton3_Callback(hObject, eventdata, handles)
%% data input
data_latih
clc
P=A(:,2:12);
T=A(:,13);
P=P';
T=T';
whos T
%% membangun jaringan saraf tiruan
Jumlah_Layer_Neuron=str2num(get(handles.edit14, 'string'));
```

```

net=newff(P,T,Jumlah_Layer_Neuron);

%% paramater jaringan
net.trainParam.epochs=str2num(get(handles.edit12,'string'));
net.trainParam.goal=str2num(get(handles.edit16,'string'));
net.trainParam.lr=str2num(get(handles.edit13,'string'));
net.trainParam.show=1;

%% pelatihan jaringan
net=train(net,P,T);

%% simulasi
an=abs(sim(net,P))'
whos an
%sim_result=[an T]
save netnet

%nntraintool('close')
%break
%% pengujian
data_uji
uji=U(:,2:12)';
target_uji=U(:,13);
%% hasil uji
hasil_uji=round(abs(sim(net,uji)))
Hasil_Uji_vs_Target=[hasil_uji target_uji ]
set(handles.text13,'visible','off')
set(handles.text14,'visible','off')
set(handles.text15,'visible','off')
set(handles.text16,'visible','off')

set(handles.edit12,'visible','off')
set(handles.edit13,'visible','off')
set(handles.edit14,'visible','off')
set(handles.edit16,'visible','off')

set(handles.pushbutton3,'enable','off')

```

3) JST

Command button JST (*pushbutton5* JST) berfungsi melakukan proses perhitungan deteksi penyakit stroke berdasarkan nilai variabel faktor gejala umur, nadi, jenis kelamin, kolesterol, hipertensi, diabetes militus, kesadaran, pusing, mual, kesemutan, dan riwayat penyakit stroke yang telah *diinputkan*. Hasil perhitungan deteksi akan terlihat pada hasil operasi yang berdasarkan pada *output* bobot (“*Suspectstroke* jika *Output* sama dengan 1, atau tidak *Suspectstroke* jika

sama dengan 0”). Berikut adalah *source code* Matlab yang menjalankan perintah *command button* JST (*pushbutton5* JST).

```
function pushbutton5_Callback(hObject, eventdata, handles)
load net
data_uji
x1=str2num(get(handles.edit1, 'string'));
x2=str2num(get(handles.edit2, 'string'));
x3=str2num(get(handles.edit3, 'string'));
x4=str2num(get(handles.edit4, 'string'));
x5=str2num(get(handles.edit5, 'string'));
x6=str2num(get(handles.edit6, 'string'));
x7=str2num(get(handles.edit7, 'string'));
x8=str2num(get(handles.edit8, 'string'));
x9=str2num(get(handles.edit9, 'string'));
x10=str2num(get(handles.edit10, 'string'));
x11=str2num(get(handles.edit11, 'string'));

uji=[x1;x2;x3;x4;x5;x6;x7;x8;x9;x10;x11];
target_uji=U(:,13);
%% hasil uji
hasil_uji=(abs(sim(net,uji)))
hasil_uji=round(abs(sim(net,uji)))

if hasil_uji ==0
    set(handles.edit15, 'string', 'Pasien tersebut tidak
SuspectStroke')
elseif hasil_uji ==1
    set(handles.edit15, 'string', 'Pasien tersebut SuspectStroke')
end
```

4) Naive Bayes

Command button Naive Bayes (*pushbutton2Naive Bayes*) berfungsi melakukan proses perhitungan deteksi penyakit stroke berdasarkan nilai variabel faktor gejala umur, nadi, jenis kelamin, kolesterol, hipertensi, diabetes militus, kesadaran, pusing, mual, kesemutan, dan riwayat penyakit stroke yang telah diinputkan dengan perhitungan probabilitas. Hasil perhitungan deteksi akan terlihat pada hasil operasi yang berdasarkan pada *output* bobot (“*Suspectstroke* jika *output* sama dengan 1, atau tidak *Suspectstroke* jika sama dengan 0”). Berikut

adalah *source code* Matlab yang menjalankan perintah *command button* Naive Bayes (*pushbutton2Naive Bayes*).

```
function pushbutton2_Callback(hObject, eventdata, handles)
data_latih
data=(A(:,[2:13]));
data_lat=data(:,1:end-1);
x1=str2num(get(handles.edit1,'string'));
x2=str2num(get(handles.edit2,'string'));
x3=str2num(get(handles.edit3,'string'));
x4=str2num(get(handles.edit4,'string'));
x5=str2num(get(handles.edit5,'string'));
x6=str2num(get(handles.edit6,'string'));
x7=str2num(get(handles.edit7,'string'));
x8=str2num(get(handles.edit8,'string'));
x9=str2num(get(handles.edit9,'string'));
x10=str2num(get(handles.edit10,'string'));
x11=str2num(get(handles.edit11,'string'));
data_uji=[x1;x2;x3;x4;x5;x6;x7;x8;x9;x10;x11]';
kelas_latih=data(:,end);
model=NaiveBayes.fit(data_lat, kelas_latih,'Distribution', 'mn');
hasil_uji=model.predict(data_uji)
if hasil_uji ==0
    set(handles.edit15,'string','Pasien tersebut tidak
    SuspectStroke')
elseif hasil_uji ==1
    set(handles.edit15,'string','Pasien tersebut SuspectStroke')
end
```

5) Mulai Baru

Command button mulai baru (*pushbutton6*) berfungsi untuk melakukan perintah mengosongkan semua isi form sebelumnya guna melakukan perhitungan deteksi penyakit stroke dengan data variabel yang baru. Berikut adalah *source code* Matlab yang menjalankan perintah *command button* mulai baru (*pushbutton6*).

```
function pushbutton6_Callback(hObject, eventdata, handles)
set(handles.edit1,'String','');
set(handles.edit2,'String','');
set(handles.edit3,'String','');
set(handles.edit4,'String','');
set(handles.edit5,'String','');
set(handles.edit6,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit9,'String','');
```

```
set(handles.edit10,'String','');
set(handles.edit11,'String','');
set(handles.edit15,'String','Hasil Deteksi...');
```

4.5 Analisis Jaringan Syaraf Tiruan

4.5.1 Penyiapan data untuk Jaringan Syaraf Tiruan

Pada Jaringan Syaraf Tiruan data yang diproses harus mempunyai *range* yang sama. Karena pada penelitian ini data yang digunakan dominan dengan batas nilai 0 sampai 1 maka semua data dinormalisasikan menjadi data dengan *range* 0 sampai 1. Pada tahap ini melakukan perubahan untuk data berupa numerik (data dengan *range* selain 1-0) untuk dinormalisasikan menjadi data dengan *range* 0 sampai 1. Pada penelitian ini ada tiga atribut data yang berupa numerik yaitu umur, nadi, dan kolesterol yang dipunyai pasien. Setelah proses normalisasi maka dihasilkan data normalisasi seperti ditunjukkan pada Tabel 4.3. Di bawah ini adalah rumus untuk menormalisasi data :

$$\left[\text{nilai baru} = \frac{nab - nbb}{naa - nba} (\text{nilai lama} - nba) + nbb \right]$$

Keterangan :

nab (batas atas nilai baru) = 1

nbb (batas bawah nilai baru) = 0

naa (batas atas nilai lama) = *max* dari database

nba (batas bawah nilai lama) = *min* dari database

Tabel 4.3 Tabel Contoh Hasil Normalisasi

Umur	Range 0-1	Nadi	Range 0-1	Kolesterol	Range 0-1
58	0,62069	88	0,4	257	0,574879
73	0,87931	84	0,357895	277	0,671498
59	0,637931	113	0,663158	243	0,507246
75	0,913793	75	0,263158	280	0,68599
74	0,896552	88	0,4	300	0,782609
46	0,413793	93	0,452632	297	0,768116
61	0,672414	80	0,315789	249	0,536232
56	0,586207	88	0,4	261	0,594203
40	0,310345	82	0,336842	291	0,73913
66	0,758621	90	0,421053	260	0,589372

*) Hasil Normalisasi lebih lengkap terdapat pada lampiran 8

4.5.2 Tahap Perancangan Sistem

Pada tahap perancangan ini, desain sistem mulai dibentuk untuk menentukan bagaimana suatu sistem akan menyelesaikan apa yang harus diselesaikan. Pada dasarnya tahap perancangan sistem ini merupakan kombinasi dari komponen-komponen perangkat keras dari suatu sistem. Sistem deteksi penyakit stroke ini secara garis besar akan menggunakan *software* Matlab.

4.5.2.1 Perancangan Sistem Pelatihan

Sistem pelatihan sebagai bagian awal dari sistem deteksi penyakit stroke. Sistem terdiri dari data-data yang telah diperoleh dan disimpan dalam bentuk *excel*. Sistem ini berfungsi untuk menyiapkan segala informasi sebagai bahan pelatihan dan pengenalan deteksi penyakit stroke. Ada beberapa tahapan yang harus dilakukan, antara lain menginputkan data-data yang sudah diolah yaitu data latih dan data target ke dalam sistem. Data latih terdiri atas umur, nadi, jenis

kelamin, kolesterol, Hipertensi, Diabetes Militus, kesadaran, pusing, mual, kesemutan, dan riwayat penyakit stroke. Sedangkan data target adalah data deteksi *Suspectstroke* atau tidak yang telah dibuat, jika *output* bobot bernilai lebih dari 0,5 maka nilai tersebut 1 dan jika *output* bobot kurang dari 0,5 maka bernilai 0. Nilai 1 untuk *Suspectstroke* dan nilai 0 untuk *negatifstroke*. Data yang telah *diinput* akan diproses oleh sistem sebagai hasil dari pelatihan data.

4.5.2.2 Perancangan Arsitektur Jaringan Syaraf Tiruan

Jaringan yang digunakan untuk deteksi penyakit stroke adalah jaringan syaraf tiruan *backpropagation* dengan langkah pembelajaran *feedforward*. Ini memiliki beberapa lapisan, yaitu lapisan masukan, lapisan keluaran dan beberapa lapisan tersembunyi. Lapisan tersembunyi tersebut membantu jaringan untuk dapat mengenali lebih banyak pola masukan dibandingkan dengan jaringan yang tidak memiliki lapisan tersembunyi. Parameter-parameter dalam pembentukan jaringan *backpropagation* adalah sebagai berikut.

4.5.2.2.1 Bobot dan Bias

Pemilihan bobot awal sangat mempengaruhi jaringan syaraf dalam mencapai minimum terhadap nilai *error*, serta cepat tidaknya proses pelatihan menuju kekonvergenan. Dalam hal ini, pemberian nilai bobot dan bias awal menggunakan bilangan acak kecil yang dilakukan oleh *software* Matlab.

4.5.2.2.2 Jumlah Neuron pada Lapisan Tersembunyi

Perancangan arsitektur *backpropagation* pertama adalah menentukan jumlah *hidden layer* dan menentukan banyaknya *neuron* dalam setiap *hidden layer*. Arsitektur jaringan yang digunakan dalam skripsi ini adalah arsitektur jaringan dengan satu *hidden layer*. Menurut teori, arsitektur ini disebut arsitektur jaringan layar jamak. Banyaknya *neuron hidden layer* ditentukan dengan cara *trial and error*, dalam arti hasil pembelajaran yang tercepat dan terbaik itulah yang akan menentukan banyaknya *neuron hidden layer* tersebut.

4.5.2.2.3 *Error Goal* (Kinerja Tujuan)

Kinerja tujuan adalah target nilai fungsi kinerja. Iterasi akan dihentikan apabila nilai fungsi kinerja kurang dari atau sama dengan kinerja tujuan (Kusumadewi, 2004:134).

Error goal atau galat ditentukan untuk membandingkan dengan galat pada jaringan saat pelatihan. Jaringan akan konvergen ketika *error* jaringan lebih kecil dari *error goal*. Dalam skripsi ini ditentukan *error goal* atau toleransi sebesar 0,001.

4.5.2.2.4 *Learning Rate* (Laju Pembelajaran)

Semakin besar nilai *learning rate* (α) akan berimplikasi pada semakin besarnya langkah pembelajaran. Jika *learning rate* diset terlalu besar, maka algoritma akan menjadi tidak stabil. Sebaliknya, jika *learning rate* diset terlalu kecil, maka algoritma akan konvergen dalam jangka waktu yang sangat lama (Kusumadewi, 2004: 134).

4.5.2.2.5 Fungsi Aktivasi

Dalam jaringan syaraf tiruan, fungsi aktivasi digunakan untuk menentukan keluaran suatu *neuron*. Fungsi aktivasi yang digunakan dalam penerapan JST untuk deteksi penyakit stroke, yaitu fungsi aktivasi *logsig* (sigmoid biner). Fungsi ini dipilih karena pada unit *output* hanya dirancang menampilkan 2 keputusan, yaitu apakah *Suspectstroke* dengan nilai 1 atau *negatifstroke* dengan nilai 0.

4.5.2.2.6 Maksimum Epoch

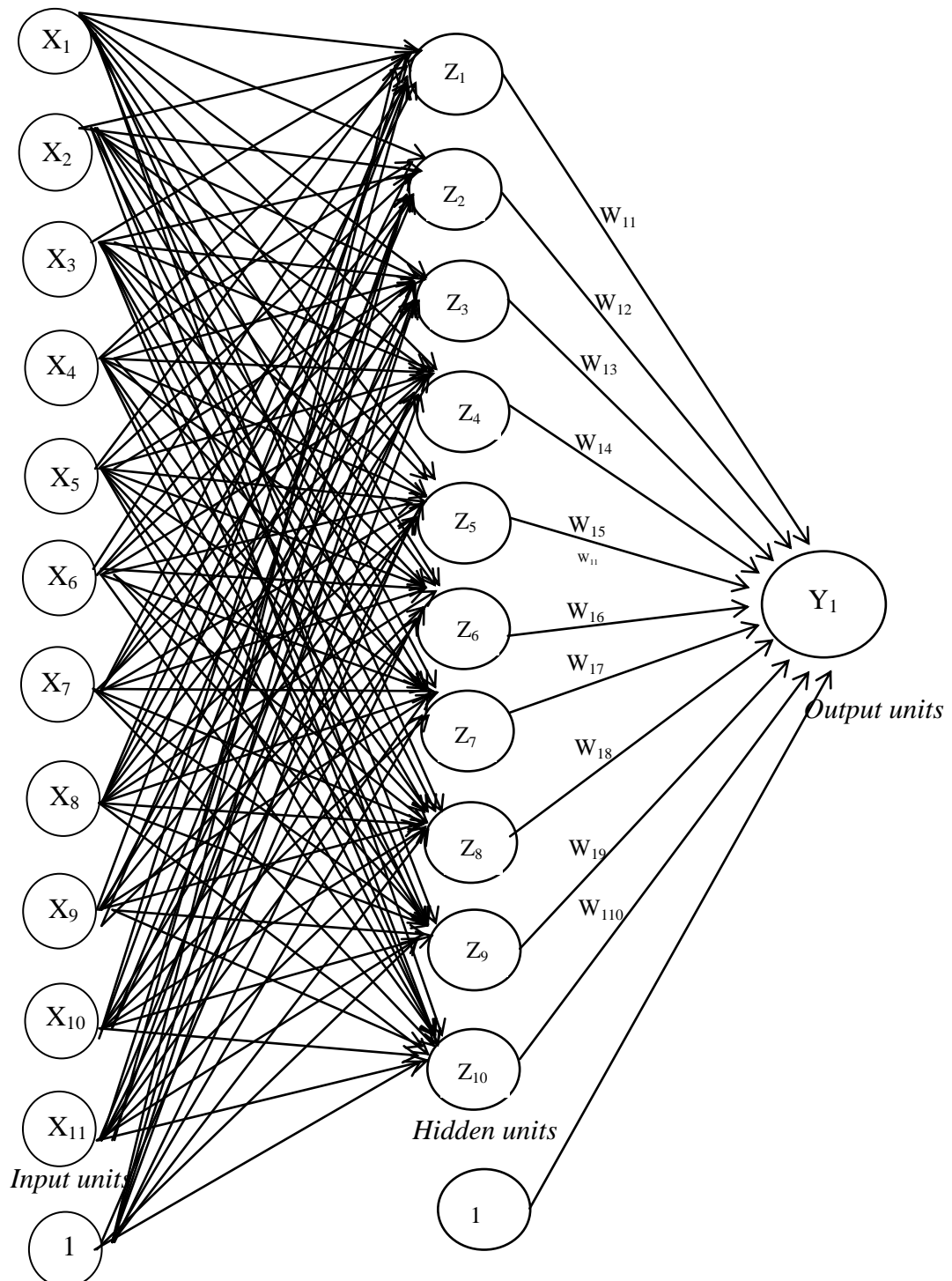
Jumlah epoh *maksimum* yang boleh dilakukan selama proses pelatihan. Iterasi akan dihentikan apabila epoh melebihi *maksimum* epoh. Berikut ini Tabel 4.4 adalah variasi yang dilakukan terhadap *learning rate* dan banyaknya *neuron hidden layer* yang akan digunakan untuk mengetahui seberapa besar tingkat akurasi sistem dalam melakukan pendeteksian terhadap penyakit stroke.

Tabel 4.4 Variansi *learning Rate* dan Banyaknya *Neuron Hidden Layer*

No.	<i>Learning rate</i> (α)	Banyaknya <i>neuron hidden layer</i>
1	0,001	1
2	0,001	5
3	0,001	10
4	0,002	1
5	0,002	5
6	0,002	10
7	0,005	1
8	0,005	5

9	0,005	10
10	0,0075	1
11	0,0075	5
12	0,0075	10
13	0,01	1
14	0,01	5
15	0,01	10

Gambar 4.10 Adalah arsitektur jaringan layar jamak dari sistem deteksi dini dalam sistem ini.



Bias Bernilai 1

Gambar 4.10 Arsitektur Jaringan dengan 10 *Neuron Hidden*

Keterangan:

$X_1, X_2, X_3, \dots, X_{11}$: nilai *input* variabel

$Z_1, Z_2, Z_3, \dots, Z_{10}$: *neuron hidden layer*

Y_1 : nilai target, bernilai 1 jika *Suspectstroke* dan bernilai 0 jika tidak stroke.

4.5.3 Hasil Arsitektur Jaringan Syaraf Tiruan *Backpropagation*

Akan dilihat pengaruh *learning rate* dan jumlah *neuron hidden layer* terhadap MSE, dan lama waktu pelatihan oleh sistem. Dari data latih dan data yang diujikan pada masing-masing variasi, jumlah data yang dikenali oleh sistem dengan benar sebagai berikut.

Jumlah iterasi (*epoch*) : 1000

Target *error* : 0,001

Berikut ini disajikan dalam tabel hasil dari variasi arsitektur jaringan yang dilakukan.

Tabel 4.5 Analisis Hasil Variasi Arsitektur Jaringan

No	A	Banyaknya <i>Neuron Hidden layer</i>	Hasil				
			Epoch	MSE	Waktu (detik)	Data Latih yang dikenali	Data <i>non</i> latih yang dikenali
1	0,001	1	1000	0,151	0:00:00	57	13
2	0,001	5	1000	0,0268	0:00:01	74	20
3	0,001	10	1000	0,00719	0:00:00	80	20
4	0,002	1	1000	3,61e ⁻⁰⁵	0:00:00	64	18
5	0,002	5	1000	0,00322	0:00:00	78	20
6	0,002	10	1000	0,00189	0:00:00	79	19
7	0,005	1	1000	0,0186	0:00:00	80	19
8	0,005	5	1000	2,05e ⁰⁷	0:00:00	79	20
9	0,005	10	1000	0,0293	0:00:00	80	20
10	0,0075	1	1000	7,73e ⁰⁷	0:00:00	79	20
11	0,0075	5	1000	0,0443	0:00:00	79	20
12	0,0075	10	1000	0,00176	0:00:00	80	20
13	0,01	1	1000	0,00134	0:00:00	80	20
14	0,01	5	1000	0,0183	0:00:00	80	20
15	0,01	10	1000	0,0700	0:00:00	79	19

Berdasarkan Tabel 4.5 didapatkan bahwa arsitektur jaringan yang paling optimal dalam deteksi penyakit stroke sebagai berikut.

Learning rate : 0,01

Banyaknya *neuron hidden layer* : 1

Jumlah iterasi (Epoch) : 1000

Target Error : 0,00134

Diperoleh tingkat akurasi sistem sebesar 100% dengan MSE sebesar 0,00134.

4.5.4 Hasil Kerja Sistem Jaringan Syaraf Tiruan

1) Variasi 1

Dari data pelatihan dengan 80 data dan 20 data pengujian diperoleh hasil analisis apakah seseorang terkena stroke atau tidak. Dari data tersebut akan diperoleh tingkat akurasi Jaringan Syaraf Tiruan. Data hasil dapat dilihat pada Tabel 4.6

Tabel 4.6 Hasil Uji Jaringan Syaraf Tiruan 20 Data Pengujian

No.	Target	Output bobot	Hasil Uji JST
1	0	0.644747	1
2	1	0.950923	1
3	1	0.534604	1
4	1	0.9997	1
5	0	0.284307	0
6	1	1.00127	1
7	1	1.00917	1
8	1	0.734548	1
9	1	0.0015657	0
10	0	0.0253761	0
11	0	0.767839	1
12	0	0.00207016	0
13	0	0.144154	0
14	0	0.328295	0
15	0	0.449879	0
16	0	0.369763	0
17	0	0.711609	1
18	0	0.0519255	0
19	0	0.989669	1
20	0	1.00984	1

Jaringan diuji dengan 20 data baru yang dimasukkan. Hasil pengujian terhadap 20 data baru menunjukkan bahwa tingkat akuratan data terhadap target sebesar 70 persen, yang diperoleh 14 data gejala pasien sesuai target dan 6 data gejala tidak sesuai dengan target.

2) Variasi 2

Dari data pelatihan dengan 80 data dan 30 data pengujian diperoleh hasil analisis apakah seseorang terkena stroke atau tidak. Dari data tersebut akan diperoleh tingkat akurasi Jaringan Syaraf Tiruan. Data hasil dapat dilihat pada Tabel 4.7

Tabel 4.7 Hasil Uji Jaringan Syaraf Tiruan 30 Data Pengujian

No.	Target	Output bobot	Hasil Uji JST
1	1	0.726441	1
2	1	1.00251	1
3	0	0.0443725	0
4	0	0.31275	0
5	1	0.997557	1
6	0	0.039963	0
7	1	0.966507	1
8	1	0.435219	0
9	1	0.953753	1
10	1	0.782923	1
11	0	0.644747	1
12	1	0.950923	1
13	1	0.534604	1
14	1	0.9997	1
15	0	0.284307	0
16	1	1.00127	1
17	1	1.00917	1
18	1	0.734548	1
19	1	0.0015657	0
20	0	0.0253761	0
21	0	0.767839	1
22	0	0.00207016	0
23	0	0.144154	0

24	0	0.328295	0
25	0	0.449879	0
26	0	0.369763	0
27	0	0.711609	1
28	0	0.0519255	0
29	0	0.989669	1
30	0	1.00984	1

Jaringan diuji dengan 30 data baru yang dimasukkan. Hasil pengujian terhadap 30 data baru menunjukkan bahwa tingkat akuratan data terhadap target sebesar 73,33 persen, yang diperoleh 22 data gejala pasien sesuai target dan 8 data gejala tidak sesuai dengan target.

3) Variasi 3

Dari data pelatihan dengan 80 data dan 40 data pengujian diperoleh hasil analisis apakah seseorang terkena stroke atau tidak. Dari data tersebut akan diperoleh tingkat akurasi Jaringan Syaraf Tiruan. Data hasil dapat dilihat pada Tabel 4.8

Tabel 4.8 Hasil Uji Jaringan Syaraf Tiruan 40 Data Pengujian

No.	Target	Output bobot	Hasil Uji JST
1	0	0.883497	1
2	1	0.904724	1
3	0	0.829274	1
4	0	0.788009	1
5	0	0.118203	0
6	1	1.048	1
7	1	0.191739	0
8	1	0.982576	1
9	1	1.02522	1
10	1	0.880448	1
11	1	0.726441	1
12	1	1.00251	1
13	0	0.0443725	0
14	0	0.31275	0
15	1	0.997557	1
16	0	0.039963	0

17	1	0.966507	1
18	1	0.435219	0
19	1	0.953753	1
20	1	0.782923	1
21	0	0.644747	1
22	1	0.950923	1

No.	Target	Output bobot	Hasil Uji JST
23	1	0.534604	1
24	1	0.9997	1
25	0	0.284307	0
26	1	1.00127	1
27	1	1.00917	1
28	1	0.734548	1
29	1	0.0015657	0
30	0	0.0253761	0
31	0	0.767839	1
32	0	0.00207016	0
33	0	0.144154	0
34	0	0.328295	0
35	0	0.449879	0
36	0	0.369763	0
37	0	0.711609	1
38	0	0.0519255	0
39	0	0.989669	1
40	0	1.00984	1

Jaringan diuji dengan 40 data baru yang dimasukkan. Hasil pengujian terhadap 40 data baru menunjukkan bahwa tingkat akuratan data terhadap target sebesar 72,5 persen, yang diperoleh 29 data gejala pasien sesuai target dan 11 data gejala tidak sesuai dengan target.

Hasil dari pelatihan digunakan untuk menentukan konfigurasi terbaik dari jaringan dengan metode *backpropagation*, *error goal* 0,001, *maksimum* epoh 1000 serta fungsi aktivasi menggunakan fungsi *logsig*, sedangkan konstanta belajar dan banyaknya neuron pada lapisan tersembunyi diubah secara *trial and error*. Dari hasil pelatihan telah dilakukan, jaringan mengenali 100% dari data yang

dilatihkan, ditunjukkan oleh hasil keluaran yang ditampilkan sesuai dengan target yang telah ditentukan yaitu contoh pada pasien nomer 2 target yang ditentukan adalah 1, dengan *Output* bobot menunjukkan 0.950923 maka mendekati 1, begitu pula dengan pasien nomer 5 dengan target yang ditentukan adalah 0, dengan *Output* bobot menunjukkan 0.284307 maka mendekati 0.

Pada tahap pelatihan maupun pengujian sistem Jaringan Syaraf Tiruan ini mempunyai beberapa kelebihan, antara lain proses yang akurat, cepat, serta meminimalisasi kesalahan. Pada jaringan syaraf tiruan yang perlu dilakukan adalah hanya melatih jaringan untuk “belajar” dengan cara memasukkan set data berisi sekumpulan kasus ke dalam jaringan. Dalam hal ini *software* yang cocok digunakan untuk membantu pemecahan masalah Jaringan Syaraf Tiruan adalah Matlab karena banyak model Jaringan Syaraf Tiruan yang memanipulasi *matriks* atau vektor dalam iterasinya. Adapun kelemahan sistem, belum ditemukannyacara terbaik untuk merepresentasikan data *input*, memilih arsitektur, serta jumlah node, dan jumlah lapisan. Cara yang digunakan hingga saat ini masih dengan cara coba-coba (*trial and error*).

4.6Naive Bayes

Metode Naive Bayes adalah proses pengklasifikasian data untuk menghasilkan keputusan sebagai hasil deteksi dengan menggunakan data *training*.

4.6.1 Tahap Perancangan Sistem

Pada tahap perancangan ini, desain sistem mulai dibentuk untuk menentukan bagaimana suatu sistem akan menyelesaikan apa yang harus

diselesaikan. Pada dasarnya tahap perancangan sistem ini merupakan kombinasi dari komponen-komponen perangkat keras dari suatu sistem. Sistem deteksi penyakit stroke ini secara garis besar akan menggunakan *software* Matlab.

4.6.1.1 Perancangan Sistem Pelatihan

Sistem pelatihan sebagai bagian awal dari sistem deteksi penyakit stroke. Sistem terdiri dari data-data yang telah diperoleh dan disimpan dalam bentuk *excel*. Sistem ini berfungsi untuk menyiapkan segala informasi sebagai bahan pelatihan dan pengenalan deteksi penyakit stroke. Ada beberapa tahapan yang harus dilakukan, antara lain menginputkan data-data yang sudah diolah yaitu data latih dan data target ke dalam sistem. Data latih terdiri atas umur, nadi, jenis kelamin, kolesterol, Hipertensi, Diabetes Militus, kesadaran, pusing, mual, kesemutan, dan riwayat penyakit stroke. Sedangkan data target adalah data deteksi *Suspectstroke* atau tidak yang telah dibuat, nilai 1 untuk *Suspectstroke* dan nilai 0 untuk *negatifstroke*. Data yang telah diinput akan diproses oleh sistem sebagai hasil dari pelatihan data.

4.6.2 Proses Perhitungan Naive Bayes

Proses yang dilakukan oleh metode ini meliputi proses penghitungan dari setiap inputan data dari *user* dengan memisahkan data numerik dan data opsional yang dapat dilihat pada Tabel 4.9. Untuk data berupa numerik (umur, nadi, dan kolesterol) dicari nilai *Mean* dan Standart Deviasi dari data *training* yang ada yang dapat dilihat pada Tabel 4.10, Sedangkan untuk data opsional menghitung nilai probabilitas yang diperoleh dari data *training* yang ada yang dilanjutkan

dengan membandingkan setiap hasilnya. Kemudian menghitung probabilitas bersyarat dari kedua bentuk data yang berbeda tersebut untuk melakukan pengambilan keputusan atau kesimpulan deteksi pasien apakah *Suspect* stroke atau tidak.

Tabel 4.9 Jenis Data Pada Variabel Gejala

Variabel	Jenis Data
Umur	Numerik
Nadi	Numerik
Jenis Kelamin	Opsional
Kolesterol	Numerik
Hipertensi	Opsional
Diabetes Militus	Opsional
Kesadaran	Opsional
Pusing	Opsional
Mual	Opsional
Kesemutan	Opsional
Riwayat Penyakit Stroke	Opsional

Tabel 4.10 Perhitungan Mean dan Standart Deviasi Pada Data Numerik

	Umur		Nadi		Kolesterol	
	Tidak <i>Suspect</i>	<i>Suspect</i>	Tidak <i>Suspect</i>	<i>Suspect</i>	Tidak <i>Suspect</i>	<i>Suspect</i>
Mean	53,6	59,72857	81,7	88,81429	155,8	271,8429
Variansi	144,64	105,3406	142,81	245,0369	281,36	1151,504
Standart Deviasi	12,02664	10,26356	11,95031	15,65366	16,77379	33,93382

Tabel 4.10 menunjukkan nilai dari Mean dan Standart deviasi dari umur, nadi, dan kolesterol dalam kategori *Suspect* maupun tidak. Data tersebut digunakan untuk perhitungan ketika *user* memasukkan data untuk masuk dalam perhitungan yang menghasilkan kesimpulan *Suspect* atau tidaknya *user* tersebut pada metode Naive Bayes. Untuk data opsional (0 atau 1) diperlukan nilai probabilitas yang diperoleh dari data *training* yang ada. Dapat dilihat pada Tabel

4.11 adalah tabel dari probabilitas jenis kelamin, Tabel 4.12 adalah tabel dari probabilitas Hipertensi. Tabel 4.13 adalah tabel dari probabilitas Diabetes Militus. Tabel 4.14 adalah tabel dari probabilitas kesadaran. Tabel 4.15 adalah tabel dari probabilitas pusing. Tabel 4.16 adalah tabel dari probabilitas mual. Tabel 4.17 adalah tabel dari probabilitas kesemutan. Tabel 4.18 adalah tabel dari probabilitas pernah stroke.

Tabel 4.11 Probabilitas Jenis Kelamin

		Tidak <i>Suspect</i>	<i>Suspect</i>
X_3	0	0,3	0,328571
	1	0,7	0,671429

Tabel 4.12 Probabilitas Hipertensi

		Tidak <i>Suspect</i>	<i>Suspect</i>
X_5	0	0,1	0,071429
	1	0,9	0,928571

Tabel 4.13 Probabilitas Diabetes Militus

		Tidak <i>Suspect</i>	<i>Suspect</i>
X_6	0	0,9	0,2
	1	0,1	0,8

Tabel 4.14 Probabilitas Kesadaran

		Tidak <i>Suspect</i>	<i>Suspect</i>
X_7	0	0,1	0
	1	0,9	1

Tabel 4.15 Probabilitas Pusing

		Tidak <i>Suspect</i>	<i>Suspect</i>
X_8	0	0,4	0,142857

1	0,6	0,857143
---	-----	----------

Tabel 4.16 Probabilitas Mual

		Tidak <i>Suspect</i>	<i>Suspect</i>
X ₉	0	0,4	0,214286
	1	0,6	0,785714

Tabel 4.17 Probabilitas Kesemutan

		Tidak <i>Suspect</i>	<i>Suspect</i>
X ₁₀	0	0,8	0,028571
	1	0,2	0,971429

Tabel 4.18 Probabilitas Pernah Stroke

		Tidak <i>Suspect</i>	<i>Suspect</i>
X ₁₁	0	1	0,828571
	1	0	0,171429

Tabel 4.10 sampai Tabel 4.18 merupakan tabel probabilitas data opsional untuk variabel – variabel masukan yang akan digunakan untuk penentuan *Suspect* atau tidak *Suspect* berdasarkan opsionalnya masing-masing. Nilai tersebut digunakan untuk perhitungan nilai probabilitas dari kedua kategori tersebut untuk menghasilkan kesimpulan *Suspect* atau tidak.

Dimisalkan terdapat data *test* untuk menghitung probabilitas bersyarat yang akan menghasilkan keputusan sebagai berikut ini.

1) Berdasarkan Data uji Tabel 4.2 pasien nomer 2:

- a. Umur : 69
- b. Nadi : 81
- c. Jenis kelamin : Perempuan
- d. Kolesterol : 200
- e. Hipertensi : ya
- f. Diabetes Militus : ya
- g. Kesadaran : ya
- h. Pusing : ya
- i. Mual : tidak
- j. Kesemutan : ya
- k. Riwayat Stroke : tidak

Perhitungan nilai probabilitas untuk data numerik yaitu:

$$P(\text{Umur} = 69|\text{Suspect}) = \frac{1}{\sqrt{2\pi}(10,26356)} e^{\frac{(69-59,72857)^2}{2(105,8406)}} = 0,0584667$$

$$P(\text{Nadi} = 81|\text{Suspect}) = \frac{1}{\sqrt{2\pi}(15,65366)} e^{\frac{(81-88,81429)^2}{2(245,0369)}} = 0,0288744$$

$$P(\text{Kolesterol} = 200|\text{Suspect}) = \frac{1}{\sqrt{2\pi}(33,933)} e^{\frac{(200-271,8429)^2}{2(1151,504)}} = 0,110590$$

$$P(\text{Umur} = 69|\text{Tidak}) = \frac{1}{\sqrt{2\pi}(12,02664)} e^{\frac{(69-53,6)^2}{2(144,64)}} = 0,0753216$$

$$P(\text{Nadi} = 81|\text{Tidak}) = \frac{1}{\sqrt{2\pi}(11,95031)} e^{\frac{(81-81,7)^2}{2(142,81)}} = 0,033449$$

$$P(\text{Kolesterol} = 200|\text{Tidak}) = \frac{1}{\sqrt{2\pi}(16,73379)} e^{\frac{(200-155,8)^2}{2(281,86)}} = 0,76588$$

Perhitungan probabilitas akhir untuk setiap faktor gejala stroke:

$$\begin{aligned}
 P(X|Suspect) &= P(X_1|Suspect) \times P(X_2|Suspect) \times P(X_3|Suspect) \times P(X_4|Suspect) \times \\
 &\quad P(X_5|Suspect) \times P(X_6|Suspect) \times P(X_7|Suspect) \times P(X_8|Suspect) \times \\
 &\quad P(X_9|Suspect) \times P(X_{10}|Suspect) \times P(X_{11}|Suspect) \\
 &= 0,058497 \times 0,0288744 \times 0,328571 \times 0,110590 \times 0,928571 \times 0,8 \times \\
 &\quad 1 \times 0,857143 \times 0,214286 \times 0,971429 \times 0,828571 \\
 &= 6,74038 \times 10^{-5}
 \end{aligned}$$

$$\begin{aligned}
 P(X|Tidak) &= P(X_1|Tidak) \times P(X_2|Tidak) \times P(X_3|Tidak) \times P(X_4|Tidak) \times \\
 &\quad P(X_5|Tidak) \times P(X_6|Tidak) \times P(X_7|Tidak) \times P(X_8|Tidak) \times \\
 &\quad P(X_9|Tidak) \times P(X_{10}|Tidak) \times P(X_{11}|Tidak) \\
 &= 0,0753216 \times 0,33449 \times 0,3 \times 0,76588 \times 0,9 \times 0,1 \times 1 \times 0,6 \times 0,4 \times \\
 &\quad 0,2 \times 1 \\
 &= 2,500739 \times 10^{-5}
 \end{aligned}$$

Nilai tersebut untuk mendapatkan probabilitas akhir:

$$P(Suspect|X) = \alpha \times 0,875 \times 6,74038 \times 10^{-5} = 5,89783 \times 10^{-5} \alpha$$

$$P(Tidak|X) = \alpha \times 0,125 \times 2,500739 \times 10^{-5} = 3,12592375 \times 10^{-5} \alpha$$

$\alpha = \frac{1}{P(X)}$ dengan ketentuan nilainya konstan sehingga tidak perlu diketahui karena

terbesar dari dua kelas tersebut tidak dapat dipengaruhi $P(X)$. Karena nilai probabilitas akhir terbesar berada di *Suspect* stroke dari hasil $5,89783 \times 10^{-5} \alpha$, maka data uji pada contoh di atas **pasien terkena penyakit stroke**. Dengan pernyataan tersebut maka data hasil uji sesuai dengan target.

2) Berdasarkan Data uji Tabel 4.2 pasien nomer 14:

- a. Umur : 55
- b. Nadi : 79
- c. Jenis kelamin : Perempuan
- d. Kolesterol : 155
- e. Hipertensi : ya
- f. Diabetes Militus : tidak
- g. Kesadaran : ya
- h. Pusing : tidak
- i. Mual : tidak
- j. Kesemutan : tidak
- k. Riwayat Stroke : tidak

Perhitungan nilai probabilitas untuk data numerik yaitu:

$$P(\text{Umur} = 55|\text{Suspect}) = \frac{1}{\sqrt{2\pi}(10,26356)} e^{-\frac{(55-59,72857)^2}{2(105,8406)}} = 0,0432332$$

$$P(\text{Nadi} = 79|\text{Suspect}) = \frac{1}{\sqrt{2\pi}(15,65366)} e^{-\frac{(79-88,81429)^2}{2(245,0369)}} = 0,031028354$$

$$P(\text{Kolesterol} = 155|\text{Suspect}) = \frac{1}{\sqrt{2\pi}(33,933)} e^{-\frac{(155-271,8429)^2}{2(1151,504)}} = 0,04144657$$

$$P(\text{Umur} = 69|\text{Tidak}) = \frac{1}{\sqrt{2\pi}(12,02664)} e^{-\frac{(69-58,6)^2}{2(144,64)}} = 0,033405436$$

$$P(\text{Nadi} = 81|\text{Tidak}) = \frac{1}{\sqrt{2\pi}(11,95031)} e^{-\frac{(81-81,7)^2}{2(142,81)}} = 0,033621725$$

$$P(\text{Kolesterol} = 155|\text{Tidak}) = \frac{1}{\sqrt{2\pi}(16,73379)} e^{-\frac{(155-155,8)^2}{2(281,86)}} = 0,023873634$$

Perhitungan probabilitas akhir untuk setiap faktor gejala stroke:

$$\begin{aligned}
 P(X|Suspect) &= P(X_1|Suspect) \times P(X_2|Suspect) \times P(X_3|Suspect) \times P(X_4|Suspect) \times \\
 &\quad P(X_5|Suspect) \times P(X_6|Suspect) \times P(X_7|Suspect) \times P(X_8|Suspect) \times \\
 &\quad P(X_9|Suspect) \times P(X_{10}|Suspect) \times P(X_{11}|Suspect) \\
 &= 0,043233 \times 0,031028354 \times 0,0328571 \times 0,04144657 \times 0,92857 \times \\
 &\quad 0,2 \times 1 \times 0,142857 \times 0,214286 \times 0,028571 \times 0,828571 \\
 &= 2,461090961 \times 10^{-9}
 \end{aligned}$$

$$\begin{aligned}
 P(X|Tidak) &= P(X_1|Tidak) \times P(X_2|Tidak) \times P(X_3|Tidak) \times P(X_4|Tidak) \times \\
 &\quad P(X_5|Tidak) \times P(X_6|Tidak) \times P(X_7|Tidak) \times P(X_8|Tidak) \times \\
 &\quad P(X_9|Tidak) \times P(X_{10}|Tidak) \times P(X_{11}|Tidak) \\
 &= 0,033405436 \times 0,033621725 \times 0,3 \times 0,023873634 \times 0,9 \times 1 \times 0,4 \times \\
 &\quad 0,4 \times 0,8 \times 1 \\
 &= 9,26679 \times 10^{-7}
 \end{aligned}$$

Nilai tersebut untuk mendapatkan probabilitas akhir:

$$P(Suspect|X) = \alpha \times 0,875 \times 2,461090961 \times 10^{-9} = 2,153454591 \times 10^{-9} \alpha$$

$$P(Tidak|X) = \alpha \times 0,125 \times 9,26679 \times 10^{-7} = 1,4826864 \times 10^{-8} \alpha$$

$\alpha = \frac{1}{P(X)}$ dengan ketentuan nilainya konstan sehingga tidak perlu diketahui karena

terbesar dari dua kelas tersebut tidak dapat dipengaruhi $P(X)$. Karena nilai probabilitas akhir terbesar berada di tidak *Suspect* stroke dari hasil $1,4826864 \times 10^{-8} \alpha$, maka data uji pada contoh di atas **pasien tidak terkena penyakit stroke**.

Dengan pernyataan tersebut maka data hasil uji sesuai dengan target.

3) Berdasarkan Data uji Tabel 4.2 pasien nomer 9:

- a. Umur : 68
- b. Nadi : 88
- c. Jenis kelamin : Perempuan
- d. Kolesterol : 168
- e. Hipertensi : ya
- f. Diabetes Militus : tidak
- g. Kesadaran : ya
- h. Pusing : tidak
- i. Mual : ya
- j. Kesemutan : ya
- k. Riwayat Stroke : tidak

Perhitungan nilai probabilitas untuk data numerik yaitu:

$$P(\text{Umur} = 68 | \text{Suspect}) = \frac{1}{\sqrt{2\pi}(10,26356)} e^{\frac{(68-59,72857)^2}{2(105,8406)}} = 0,053797$$

$$P(\text{Nadi} = 88 | \text{Suspect}) = \frac{1}{\sqrt{2\pi}(15,65366)} e^{\frac{(88-88,81429)^2}{2(245,0369)}} = 0,025526461$$

$$P(\text{Kolesterol} = 168 | \text{Suspect}) = \frac{1}{\sqrt{2\pi}(33,933)} e^{\frac{(168-271,8429)^2}{2(1151,504)}} = 0,12702485$$

$$P(\text{Umur} = 68 | \text{Tidak}) = \frac{1}{\sqrt{2\pi}(12,02664)} e^{\frac{(68-53,6)^2}{2(144,64)}} = 0,679489$$

$$P(\text{Nadi} = 88 | \text{Tidak}) = \frac{1}{\sqrt{2\pi}(11,95031)} e^{\frac{(88-81,7)^2}{2(142,81)}} = 0,383698$$

$$P(\text{Kolesterol} = 168 | \text{Tidak}) = \frac{1}{\sqrt{2\pi}(16,73379)} e^{\frac{(168-155,8)^2}{2(281,86)}} = 0,031066757$$

Perhitungan probabilitas akhir untuk setiap faktor gejala stroke:

$$\begin{aligned}
 P(X|Suspect) &= P(X_1|Suspect) \times P(X_2|Suspect) \times P(X_3|Suspect) \times P(X_4|Suspect) \times \\
 &\quad P(X_5|Suspect) \times P(X_6|Suspect) \times P(X_7|Suspect) \times P(X_8|Suspect) \times \\
 &\quad P(X_9|Suspect) \times P(X_{10}|Suspect) \times P(X_{11}|Suspect) \\
 &= 0,053797 \times 0,025526461 \times 0,328571 \times 0,127024855 \times 0,928571 \times \\
 &\quad 0,2 \times 1 \times 0,142857 \times 0,785714 \times 0,971429 \times 0,828571 \\
 &= 8,486363634 \times 10^{-7}
 \end{aligned}$$

$$\begin{aligned}
 P(X|Tidak) &= P(X_1|Tidak) \times P(X_2|Tidak) \times P(X_3|Tidak) \times P(X_4|Tidak) \times \\
 &\quad P(X_5|Tidak) \times P(X_6|Tidak) \times P(X_7|Tidak) \times P(X_8|Tidak) \times \\
 &\quad P(X_9|Tidak) \times P(X_{10}|Tidak) \times P(X_{11}|Tidak) \\
 &= 0,679489 \times 0,383698 \times 0,3 \times 0,031066757 \times 0,9 \times 0,9 \times 1 \times 0,4 \times \\
 &\quad 0,6 \times 0,2 \times 1 \\
 &= 9,4474673 \times 10^{-5}
 \end{aligned}$$

Nilai tersebut untuk mendapatkan probabilitas akhir:

$$P(Suspect|X) = \alpha \times 0,875 \times 8,486363634 \times 10^{-7} = 7,42556818 \times 10^{-7} \alpha$$

$$P(Tidak|X) = \alpha \times 0,125 \times 9,4474673 \times 10^{-5} = 1,180933412 \times 9,447 \times 10^{-5} \alpha$$

$\alpha = \frac{1}{P(X)}$ dengan ketentuan nilainya konstan sehingga tidak perlu diketahui karena

terbesar dari dua kelas tersebut tidak dapat dipengaruhi $P(X)$. Karena nilai probabilitas akhir terbesar berada di tidak *Suspect* stroke dari hasil $1,180933412 \times 9,447 \times 10^{-5} \alpha$, maka data uji pada contoh di atas **pasien tidak terkena penyakit stroke**. Dengan pernyataan tersebut maka data hasil uji tidak sesuai dengan target.

4.6.3 Hasil Kerja Sistem Naive Bayes

1) Variasi 1

Dari data pelatihan dengan 80 data dan 20 data pengujian diperoleh hasil analisis apakah seseorang terkena stroke atau tidak. Dari data tersebut akan diperoleh tingkat akurasi Naive Bayes. Data hasil dapat dilihat pada Tabel 4.19.

Tabel 4. 19 Tabel Hasil Uji Naive Bayes 20 Data Pengujian

No	Tar get	Probabilitas <i>Suspect</i>	Probabilitas Tidak <i>Suspect</i>	Hasil Probabilitas Terbesar	<i>Output</i> Bobot	Hasil Uji Naive Bayes
1	0	$2,483160411 \times 10^{-7} \alpha$	0	$2,483160411 \times 10^{-7} \alpha$	1	1
2	1	$5,89783 \times 10^{-5} \alpha$	$3,12592375 \times 10^{-5} \alpha$	$5,89783 \times 10^{-5} \alpha$	1	1
3	1	$4,867329 \times 10^{-6} \alpha$	$2,8193229 \times 10^{-7} \alpha$	$4,867329 \times 10^{-6} \alpha$	1	1
4	1	$1,16106995 \times 10^{-5} \alpha$	$1,1806323 \times 10^{-6} \alpha$	$1,16106995 \times 10^{-5} \alpha$	1	1
5	0	$3,433284716 \times 10^{-7} \alpha$	$3,7218224 \times 10^{-6} \alpha$	$3,72182241 \times 10^{-6} \alpha$	0	0
6	1	$5,01632141 \times 10^{-7} \alpha$	$2,7481511 \times 10^{-8} \alpha$	$5,01632141 \times 10^{-7} \alpha$	1	1
7	1	$2,202623 \times 10^{-5} \alpha$	$2,2636312 \times 10^{-6} \alpha$	$2,202623 \times 10^{-5} \alpha$	1	1
8	1	$8,38282206 \times 10^{-5} \alpha$	$1,3197646 \times 10^{-6} \alpha$	$8,38282206 \times 10^{-5} \alpha$	1	1
9	1	$7,42556818 \times 10^{-7} \alpha$	$1,1809334 \times 10^{-5} \alpha$	$1,1809334 \times 10^{-5} \alpha$	0	0
10	0	0	$3,907477 \times 10^{-9} \alpha$	$3,907477 \times 10^{-9} \alpha$	0	0
11	0	$4,138439245 \times 10^{-7} \alpha$	$2,27202797 \times 10^{-5} \alpha$	$2,27202797 \times 10^{-5} \alpha$	0	0
12	0	0	$1,64922083 \times 10^{-6} \alpha$	$1,64922083 \times 10^{-6} \alpha$	0	0
13	0	$1,453317739 \times 10^{-7} \alpha$	$2,82613858 \times 10^{-7} \alpha$	$2,82613858 \times 10^{-7} \alpha$	0	0
14	0	$2,1535 \times 10^{-9} \alpha$	$1,482686 \times 10^{-8} \alpha$	$1,482686 \times 10^{-8} \alpha$	0	0
15	0	0	$1,33919961 \times 10^{-7} \alpha$	$1,33919961 \times 10^{-7} \alpha$	0	0
16	0	0	$1,44787031 \times 10^{-6} \alpha$	$1,44787031 \times 10^{-6} \alpha$	0	0
17	0	$4,592520434 \times 10^{-9} \alpha$	$1,49308626 \times 10^{-7} \alpha$	$1,49308626 \times 10^{-7} \alpha$	0	0
18	0	$6,285942614 \times 10^{-7} \alpha$	$2,94265366 \times 10^{-6} \alpha$	$2,94265366 \times 10^{-6} \alpha$	0	0
19	0	$1,779649085 \times 10^{-6} \alpha$	$1,64457243 \times 10^{-8} \alpha$	$1,779649085 \times 10^{-6} \alpha$	1	1
20	0	$2,153478453 \times 10^{-6} \alpha$	$1,65049453 \times 10^{-5} \alpha$	$2,153478453 \times 10^{-6} \alpha$	1	1

Berdasarkan Tabel 4.19 maka diperoleh tingkat keakuratan dari sistem yang telah dibuat. Sistem diuji dengan 20 data baru yang dimasukkan. Hasil pengujian terhadap 20 data baru menunjukkan bahwa tingkat akuratan data

terhadap target sebesar 80 persen, yang diperoleh 16 data gejala pasien sesuai target dan 4 data gejala tidak sesuai dengan target.

2) Variasi 2

Dari data pelatihan dengan 80 data dan 30 data pengujian diperoleh hasil analisis apakah seseorang terkena stroke atau tidak. Dari data tersebut akan diperoleh tingkat akurasi Naive Bayes. Data hasil dapat dilihat pada Tabel 4.20

Tabel 4. 20 Tabel Hasil Uji Naive Bayes 30 Data Pengujian

No	Tar get	Probabilitas <i>Suspect</i>	Probabilitas Tidak <i>Suspect</i>	Hasil Probabilitas Terbesar	<i>Output</i> Bobot	Hasil Uji Naive Bayes
1	1	$3,544355896 \times 10^{-7} \alpha$	$1,264053004 \times 10^{-6} \alpha$	$1,264053004 \times 10^{-6} \alpha$	0	0
2	1	$8,386573052 \times 10^{-7} \alpha$	$1,101107614 \times 10^{-8} \alpha$	$8,386573052 \times 10^{-7} \alpha$	1	1
3	1	$2,799204229 \times 10^{-7} \alpha$	$2,502534293 \times 10^{-7} \alpha$	$2,799204229 \times 10^{-7} \alpha$	1	1
4	1	$7,843397156 \times 10^{-8} \alpha$	$1,635319152 \times 10^{-6} \alpha$	$1,635319152 \times 10^{-6} \alpha$	0	0
5	1	$2,299550723 \times 10^{-6} \alpha$	$7,81192902 \times 10^{-10} \alpha$	$2,299550723 \times 10^{-6} \alpha$	1	1
6	1	0	$5,348496635 \times 10^{-7} \alpha$	$5,348496635 \times 10^{-7} \alpha$	0	0
7	1	$3,331158239 \times 10^{-6} \alpha$	$5,810390135 \times 10^{-6} \alpha$	$3,331158239 \times 10^{-6} \alpha$	1	1
8	1	$2,778742768 \times 10^{-7} \alpha$	0	$2,778742768 \times 10^{-7} \alpha$	1	1
9	1	$4,26844795 \times 10^{-7} \alpha$	$2,364127668 \times 10^{-9} \alpha$	$4,26844795 \times 10^{-7} \alpha$	1	1
10	1	$5,252496084 \times 10^{-8} \alpha$	$4,698189791 \times 10^{-8} \alpha$	$5,252496084 \times 10^{-8} \alpha$	1	1
11	0	$2,483160411 \times 10^{-7} \alpha$	0	$2,483160411 \times 10^{-7} \alpha$	1	1
12	1	$5,89783 \times 10^{-5} \alpha$	$3,12592375 \times 10^{-5} \alpha$	$5,89783 \times 10^{-5} \alpha$	1	1
13	1	$4,867329 \times 10^{-6} \alpha$	$2,8193229 \times 10^{-7} \alpha$	$4,867329 \times 10^{-6} \alpha$	1	1
14	1	$1,16106995 \times 10^{-5} \alpha$	$1,1806323 \times 10^{-6} \alpha$	$1,16106995 \times 10^{-5} \alpha$	1	1
15	0	$3,433284716 \times 10^{-7} \alpha$	$3,7218224 \times 10^{-6} \alpha$	$3,7218224 \times 10^{-6} \alpha$	0	0
16	1	$5,01632141 \times 10^{-7} \alpha$	$2,7481511 \times 10^{-8} \alpha$	$5,01632141 \times 10^{-7} \alpha$	1	1
17	1	$2,202623 \times 10^{-5} \alpha$	$2,2636312 \times 10^{-6} \alpha$	$2,202623 \times 10^{-5} \alpha$	1	1
18	1	$8,38282206 \times 10^{-5} \alpha$	$1,3197646 \times 10^{-6} \alpha$	$8,38282206 \times 10^{-5} \alpha$	1	1
19	1	$7,42556818 \times 10^{-7} \alpha$	$1,1809334 \times 10^{-5} \alpha$	$1,1809334 \times 10^{-5} \alpha$	0	0
20	0	0	$3,907477 \times 10^{-9} \alpha$	$3,907477 \times 10^{-9} \alpha$	0	0
21	0	$4,138439245 \times 10^{-7} \alpha$	$2,27202797 \times 10^{-5} \alpha$	$2,27202797 \times 10^{-5} \alpha$	0	0
22	0	0	$1,64922083 \times 10^{-6} \alpha$	$1,64922083 \times 10^{-6} \alpha$	0	0
23	0	$1,453317739 \times 10^{-7} \alpha$	$2,82613858 \times 10^{-7} \alpha$	$2,82613858 \times 10^{-7} \alpha$	0	0
24	0	$2,1535 \times 10^{-9} \alpha$	$1,482686 \times 10^{-8} \alpha$	$1,482686 \times 10^{-8} \alpha$	0	0
25	0	0	$1,33919961 \times 10^{-7} \alpha$	$1,33919961 \times 10^{-7} \alpha$	0	0
26	0	0	$1,44787031 \times 10^{-6} \alpha$	$1,44787031 \times 10^{-6} \alpha$	0	0
27	0	$4,592520434 \times 10^{-9} \alpha$	$1,49308626 \times 10^{-7} \alpha$	$1,49308626 \times 10^{-7} \alpha$	0	0
28	0	$6,285942614 \times 10^{-7} \alpha$	$2,94265366 \times 10^{-6} \alpha$	$2,94265366 \times 10^{-6} \alpha$	0	0
29	0	$1,779649085 \times 10^{-6} \alpha$	$1,64457243 \times 10^{-8} \alpha$	$1,779649085 \times 10^{-6} \alpha$	1	1

30	0	$2,153478453 \times 10^{-6} \alpha$	$1,65049453 \times 10^{-5} \alpha$	$2,153478453 \times 10^{-6} \alpha$	1	1
----	---	-------------------------------------	------------------------------------	-------------------------------------	---	---

Berdasarkan Tabel 4.20 maka diperoleh tingkat keakuratan dari sistem yang telah dibuat. Sistem diuji dengan 30 data baru yang dimasukkan. Hasil pengujian terhadap 30 data baru menunjukkan bahwa tingkat akuratan data terhadap target sebesar 80 persen, yang diperoleh 24 data gejala pasien sesuai target dan 6 data gejala tidak sesuai dengan target.

3) Variasi 3

Dari data pelatihan dengan 80 data dan 40 data pengujian diperoleh hasil analisis apakah seseorang terkena stroke atau tidak. Dari data tersebut akan diperoleh tingkat akurasi Naive Bayes. Data hasil dapat dilihat pada Tabel 4.21

Tabel 4.21 Tabel Hasil Uji Naive Bayes 40 Data Pengujian

No.	Targ et	Probabilitas <i>Suspect</i>	Probabilitas Tidak <i>Suspect</i>	Hasil Probabilitas Terbesar	<i>Out put Bob ot</i>	Hasil Uji Naive Bayes
1	1	$3,4835701 \times 10^{-5} \alpha$	$1,335768758 \times 10^{-7} \alpha$	$3,4835701 \times 10^{-5} \alpha$	1	1
2	1	$8,532689076 \times 10^{-7} \alpha$	$3,759367587 \times 10^{-7} \alpha$	$8,532689076 \times 10^{-7} \alpha$	1	1
3	1	$5,786750235 \times 10^{-7} \alpha$	$6,5075971 \times 10^{-7} \alpha$	$6,5075971 \times 10^{-7} \alpha$	0	0
4	1	$2,486068034 \times 10^{-8} \alpha$	$3,465849346 \times 10^{-7} \alpha$	$3,465849346 \times 10^{-7} \alpha$	0	0
5	1	$8,38456357 \times 10^{-8} \alpha$	$7,35867344 \times 10^{-6} \alpha$	$7,35867344 \times 10^{-6} \alpha$	0	0
6	1	$4,9632856 \times 10^{-6} \alpha$	$3,8098651 \times 10^{-6} \alpha$	$4,9632856 \times 10^{-6} \alpha$	1	1
7	1	$9,761312746 \times 10^{-7} \alpha$	$5,0284365 \times 10^{-7} \alpha$	$9,761312746 \times 10^{-7} \alpha$	1	1
8	1	$4,89042725 \times 10^{-7} \alpha$	$9,47357084 \times 10^{-8} \alpha$	$4,89042725 \times 10^{-7} \alpha$	1	1
9	1	$3,903567164 \times 10^{-7} \alpha$	$2,039472957 \times 10^{-9} \alpha$	$3,903567164 \times 10^{-7} \alpha$	1	1
10	1	$1,973687269 \times 10^{-7} \alpha$	$7,9817322602 \times 10^{-9} \alpha$	$1,973687269 \times 10^{-7} \alpha$	1	1
11	1	$3,544355896 \times 10^{-7} \alpha$	$1,264053004 \times 10^{-6} \alpha$	$1,264053004 \times 10^{-6} \alpha$	0	0
12	1	$8,386573052 \times 10^{-7} \alpha$	$1,101107614 \times 10^{-8} \alpha$	$8,386573052 \times 10^{-7} \alpha$	1	1
13	1	$2,799204229 \times 10^{-7} \alpha$	$2,502534293 \times 10^{-7} \alpha$	$2,799204229 \times 10^{-7} \alpha$	1	1
14	1	$7,843397156 \times 10^{-8} \alpha$	$1,635319152 \times 10^{-6} \alpha$	$1,635319152 \times 10^{-6} \alpha$	0	0
15	1	$2,299550723 \times 10^{-6} \alpha$	$7,81192902 \times 10^{-10} \alpha$	$2,299550723 \times 10^{-6} \alpha$	1	1
16	1	0	$5,348496635 \times 10^{-7} \alpha$	$5,348496635 \times 10^{-7} \alpha$	0	0
17	1	$3,331158239 \times 10^{-6} \alpha$	$5,810390135 \times 10^{-6} \alpha$	$3,331158239 \times 10^{-6} \alpha$	1	1
18	1	$2,778742768 \times 10^{-7} \alpha$	0	$2,778742768 \times 10^{-7} \alpha$	1	1
19	1	$4,26844795 \times 10^{-7} \alpha$	$2,364127668 \times 10^{-9} \alpha$	$4,26844795 \times 10^{-7} \alpha$	1	1
20	1	$5,252496084 \times 10^{-8} \alpha$	$4,698189791 \times 10^{-8} \alpha$	$5,252496084 \times 10^{-8} \alpha$	1	1
21	0	$2,483160411 \times 10^{-7} \alpha$	0	$2,483160411 \times 10^{-7} \alpha$	1	1
22	1	$5,89783 \times 10^{-5} \alpha$	$3,12592375 \times 10^{-5} \alpha$	$5,89783 \times 10^{-5} \alpha$	1	1
23	1	$4,867329 \times 10^{-6} \alpha$	$2,8193229 \times 10^{-7} \alpha$	$4,867329 \times 10^{-6} \alpha$	1	1

24	1	$1,16106995 \times 10^{-5} \alpha$	$1,1806323 \times 10^{-6} \alpha$	$1,16106995 \times 10^{-5} \alpha$	1	1
25	0	$3,433284716 \times 10^{-7} \alpha$	$3,7218224 \times 10^{-6} \alpha$	$3,72182241 \times 10^{-6} \alpha$	0	0
26	1	$5,01632141 \times 10^{-7} \alpha$	$2,7481511 \times 10^{-8} \alpha$	$5,01632141 \times 10^{-7} \alpha$	1	1
27	1	$2,202623 \times 10^{-5} \alpha$	$2,2636312 \times 10^{-6} \alpha$	$2,202623 \times 10^{-5} \alpha$	1	1
28	1	$8,38282206 \times 10^{-5} \alpha$	$1,3197646 \times 10^{-6} \alpha$	$8,38282206 \times 10^{-5} \alpha$	1	1
29	1	$7,42556818 \times 10^{-7} \alpha$	$1,1809334 \times 10^{-5} \alpha$	$1,1809334 \times 10^{-5} \alpha$	0	0
30	0	0	$3,907477 \times 10^{-9} \alpha$	$3,907477 \times 10^{-9} \alpha$	0	0
31	0	$4,138439245 \times 10^{-7} \alpha$	$2,27202797 \times 10^{-5} \alpha$	$2,27202797 \times 10^{-5} \alpha$	0	0
32	0	0	$1,64922083 \times 10^{-6} \alpha$	$1,64922083 \times 10^{-6} \alpha$	0	0
33	0	$1,453317739 \times 10^{-7} \alpha$	$2,82613858 \times 10^{-7} \alpha$	$2,82613858 \times 10^{-7} \alpha$	0	0

No.	Targ et	Probabilitas <i>Suspect</i>	Probabilitas Tidak <i>Suspect</i>	Hasil Probabilitas Terbesar	<i>Out put Bob ot</i>	Hasil Uji Naive Bayes
34	0	$2,1535 \times 10^{-9} \alpha$	$1,482686 \times 10^{-8} \alpha$	$1,482686 \times 10^{-8} \alpha$	0	0
35	0	0	$1,33919961 \times 10^{-7} \alpha$	$1,33919961 \times 10^{-7} \alpha$	0	0
36	0	0	$1,44787031 \times 10^{-6} \alpha$	$1,44787031 \times 10^{-6} \alpha$	0	0
37	0	$4,592520434 \times 10^{-9} \alpha$	$1,49308626 \times 10^{-7} \alpha$	$1,49308626 \times 10^{-7} \alpha$	0	0
38	0	$6,285942614 \times 10^{-7} \alpha$	$2,94265366 \times 10^{-6} \alpha$	$2,94265366 \times 10^{-6} \alpha$	0	0
39	0	$1,779649085 \times 10^{-6} \alpha$	$1,64457243 \times 10^{-8} \alpha$	$1,779649085 \times 10^{-6} \alpha$	1	1
40	0	$2,153478453 \times 10^{-6} \alpha$	$1,65049453 \times 10^{-5} \alpha$	$2,153478453 \times 10^{-6} \alpha$	1	1

Berdasarkan Tabel 4.22 maka diperoleh tingkat keakuratan dari sistem yang telah dibuat. Sistem diuji dengan 40 data baru yang dimasukkan. Hasil pengujian terhadap 40 data baru menunjukkan bahwa tingkat akuratan data terhadap target sebesar 80 persen, yang diperoleh 32 data gejala pasien sesuai target dan 8 data gejala tidak sesuai dengan target.

Dari hasil pelatihan telah dilakukan, sistem mengenali 100% dari data yang dilatihkan, ditunjukkan oleh hasil keluaran yang ditampilkan sesuai dengan target yang telah ditentukan yaitu contoh pada pasien nomer 2 target yang ditentukan adalah 1 dan *Output* bobot menunjukkan 1 dengan perhitungan manual Probabilitas *Suspect* stroke $5,89783 \times 10^{-5} \alpha$ sedang nilai dari probabilitas tidak *Suspect* stroke sebesar $3,12592375 \times 10^{-5} \alpha$ maka hasil probabilitas yang diambil adalah terbesar dengan hasil $5,89783 \times 10^{-5} \alpha$ maka data gejala menunjukkan

Suspectstroke, begitu pula dengan pasien nomer 5 dengan target yang ditentukan adalah 0 dan *Output* bobot menunjukkan dengan perhitungan manual Probabilitas *Suspectstroke* $3,433284716 \times 10^{-7}$ α sedang nilai dari probabilitas tidak *Suspectstroke* sebesar $3,72182241 \times 10^{-6}$ α maka hasil probabilitas yang diambil adalah terbesar dengan hasil $3,72182241 \times 10^{-6}$ α maka data gejala menunjukkan tidak *Suspectstroke*. Hasil pengujian terhadap 20 data baru menunjukkan bahwa tingkat akuratan data terhadap target sebesar 80 persen, yang diperoleh 16 data gejala pasien sesuai target dan 4 data gejala tidak sesuai dengan target.

Metode Naive Bayes merupakan metode yang baik di dalam mesin pembelajaran berdasarkan data *training*, dengan menggunakan probabilitas bersyarat sebagai acuannya, sedangkan kelemahan dari metode tersebut adalah hanya digunakan untuk persoalan klasifikasi dengan *supervised learning* dan data-data kategorial. *Software* Matlab juga mempermudah *user* untuk informasi hasil dari deteksi penyakit stroke dengan menggunakan metode Naive Bayes. Hasil perhitungan didapatkan dari perhitungan menggunakan rumus Naive Bayes dengan menghitung jumlah gejala ya dan tidak kemudian dari hasil perhitungan disajikan perbandingan untuk mendapatkan kesimpulannya.

4.7 Hasil Perbandingan Jaringan Syaraf Tiruan dan Metode Naive Bayes

Dari hasil analisis sistem maka diperoleh perbandingan persentase tingkat akuratan dari Jaringan Syaraf Tiruan dan Naive Bayes. Perbandingan hasil kerja

sistem dari Jaringan Syaraf Tiruan dan Metode Naive Bayes dapat dilihat pada Tabel 4.22.

$$\text{Persentase akurasi} = \frac{\text{jumlah pasien yang berhasil diuji}}{\text{jumlah pasien yang diuji}} 100 \%$$

Tabel 4.22 Hasil Perbandingan Jaringan Syaraf Tiruan dan Naive Bayes

Banyak Data Latih	Banyak Data Uji	Jaringan Syaraf Tiruan			Naive Bayes		
		Persentase akurasi	Sesuai Target	Tidak Sesuai Target	Persentase akurasi	Sesuai Target	Tidak Sesuai Target
80	20	70	14	6	80	16	4
80	30	73,33	24	6	80	25	5
80	40	72,5	27	13	80	34	6
70	20	75	15	5	85	17	3
70	30	80	24	6	83,33	25	5
70	40	80	32	8	87,5	35	5
60	20	50	10	10	75	15	5
60	30	66,67	20	10	76,67	23	9
60	40	72,5	29	11	77,5	32	8
Rata-rata		71,11			80,555		

Dari Tabel 4.22 maka nilai akurasi dari setiap data Naive Bayes yang paling akurat dalam mendeteksi penyakit stroke sebesar 80,555 persen sedangkan dalam Jaringan Syaraf Tiruan sebesar 71,11 persen. Dengan jumlah pengujian yang berbeda Naive Bayes memiliki tingkat akurasi tinggi dibandingkan dengan Jaringan Syaraf Tiruan secara stabil. Nilai persentase akurasi tertinggi dalam Jaringan Syaraf Tiruan maupun Naive Bayes terdapat pada data pelatihan 70 data. Semakin banyak data yang diujikan maka semakin tinggi nilai akurasi deteksi penyakit stroke tersebut.

4.7.1 Hasil Analisis Faktor Risiko yang Berpengaruh

4.7.1.1 Hasil Analisis Faktor Risiko dengan Jenis Data Opsional yang Berpengaruh

Pada pengujian yang akan dilakukan terdapat 8 macam data masukan dengan 1 faktor risiko. Pengujian bertujuan untuk mengamati faktor risiko apa yang paling berpengaruh untuk menentukan *Suspect* atau tidak pada setiap metode dalam proses pembelajaran. *Inputan* untuk pengujian dapat dilihat pada Tabel 4.23.

Tabel 4.23 *Inputan* Pengujian dengan 1 Faktor Risiko

Variabel	Gejala Penyakit Stroke										
	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11
Umur	50	50	50	50	50	50	50	50	50	50	50
Nadi	80	80	80	80	80	80	80	80	80	80	80
Kelamin	1	0	0	0	0	0	0	0	0	0	1
Kolesterol	250	250	250	250	250	250	250	250	250	250	250
Hipertensi	0	0	1	0	0	0	0	0	0	0	1
DM	0	0	0	1	0	0	0	0	0	0	1
Kesadaran	0	0	0	0	1	0	0	0	0	0	1
Pusing	0	0	0	0	0	1	0	0	0	0	1
Mual	0	0	0	0	0	0	1	0	0	0	1
Kesemutan	0	0	0	0	0	0	0	1	0	0	1
Pernah Stroke	0	0	0	0	0	0	0	0	1	0	1

Berdasarkan pengujian yang sudah dilakukan, dengan memasukan *input* data gejala yang ada pada Tabel 4.23 maka diperoleh hasil faktor yang paling berpengaruh dalam metode. Hasil Pengujian faktor yang paling berpengaruh dalam pengujian dapat dilihat pada Tabel 4.24.

Tabel 4.24 Hasil pengujian Faktor Variabel Opsional yang Paling Berpengaruh

Pengujian data ke-	<i>Output</i> bobot JST	Hasil Uji JST	Probabilitas Variabel	<i>Output</i> Bobot	Hasil Uji Naive Bayes
V1	0.725645	1	0,34832591	1	1
V2	0.668511	1	0,4858935	1	1
V3	0.562209	1	0,671429	1	1
V4	1.04659	1	0,4348583	1	1
V5	1.43556	1	0,9285711	1	1
V6	1.66473	1	0,8	1	1
V7	0,557718	1	1	1	1
V8	1.67905	1	0,857143	1	1
V9	0.701214	1	0,785714	1	1
V10	1.43556	1	0,971429	1	1
V11	0.917612	1	0,171429	1	1

Berdasarkan data gejala pasien yang telah dibuat dapat dilihat faktor risiko yang memiliki gejala terbesar dalam Jaringan Syaraf Tiruan adalah kesemutan dengan nilai bobot terbesar yaitu 1,67905. Sedangkan, menggunakan Naive Bayes faktor risiko yang paling berpengaruh adalah mengalami mual dengan hasil probabilitas sebesar 1.

4.7.1.2 Hasil Analisis Faktor Risiko dengan Jenis Data Numerik yang Berpengaruh

Pengujian bertujuan untuk mengamati faktor risiko apa yang paling berpengaruh untuk menentukan *Suspect* atau tidak pada setiap metode dalam proses pembelajaran. Pengujian yang akan dilakukan adalah pengujian dengan

menggunakan data numerik yang berbeda. *Inputan* untuk pengujian dapat dilihat pada Tabel 4.25 untuk pengujian umur dengan *input* data umur 35 tahun sampai 85. Tabel 4.27 untuk pengujian nadi dengan *input* data nadi 50 kali per menit sampai 95 kali/menit. Tabel 4.28 untuk pengujian kolesterol dengan *input* data 150 mg/dL sampai 350 mg/dL. Berdasarkan Tabel 4.1 data pelatihan menunjukkan bahwa data umur yang didapatkan dari penelitian adalah 35 tahun sampai 85 tahun. Data nadi pada data pelatihan yang didapatkan dari penelitian adalah 50-100 kali/menit. Data kolesterol pada data pelatihan yang didapatkan dari penelitian adalah 150 mg/dL sampai 350 mg/dL.

Tabel 4.25 *Inputan* Pengujian pada Umur

Variabel	Gejala Penyakit Stroke									
	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
Umur	35	40	45	50	55	60	65	70	75	80
Nadi	70	70	70	70	70	70	70	70	70	70
Kelamin	0	0	0	0	0	0	0	0	0	0
Kolesterol	200	200	200	200	200	200	200	200	200	200
Hipertensi	0	0	0	0	0	0	0	0	0	0
DM	0	0	0	0	0	0	0	0	0	0
Kesadaran	0	0	0	0	0	0	0	0	0	0
Pusing	0	0	0	0	0	0	0	0	0	0
Mual	0	0	0	0	0	0	0	0	0	0
Kesemutan	0	0	0	0	0	0	0	0	0	0
Pernah Stroke	0	0	0	0	0	0	0	0	0	0

Berdasarkan pengujian yang sudah dilakukan, dengan memasukan *input* data gejala yang ada maka diperoleh hasil faktor yang paling berpengaruh dalam metode yang dapat dilihat pada Tabel 4.26.

Tabel 4.26 Hasil pengujian Faktor Variabel Numerik Umur yang Paling Berpengaruh

Pengujian data ke-	<i>Output</i> bobot JST	Hasil Uji JST	<i>Output</i> Bobot	Perhitungan Probabilitas terbesar	Hasil Uji Naive Bayes
--------------------	-------------------------	---------------	---------------------	-----------------------------------	-----------------------

V1	0.00148163	0	0	$2,4325324 \times 10^{-7} \alpha$	0
V2	0.00148163	0	0	$3,1294812 \times 10^{-7} \alpha$	0
V3	0.00148162	0	0	$3,540241 \times 10^{-7} \alpha$	0
V4	0.00148161	0	1	$5,914091 \times 10^{-7} \alpha$	1
V5	0.0014816	0	1	$8,2364654 \times 10^{-7} \alpha$	1
V6	0.00148158	0	1	$5,891235 \times 10^{-7} \alpha$	1
V7	0.00148156	0	1	$5,51471202 \times 10^{-7} \alpha$	1
V8	0.00148153	0	1	$5,374114 \times 10^{-7} \alpha$	1
V9	0.00148149	0	1	$5,02471245 \times 10^{-7} \alpha$	1
V10	0.00148144	0	1	$4,842686 \times 10^{-7} \alpha$	1

Dapat dilihat pada Tabel 4.28 dengan analisis Jaringan Syaraf Tiruan

output bobot terbesar terdapat pada umur 35 tahun, tetapi tidak berpengaruh dalam deteksi penyakit stroke karena selisih *output* bobot yang hampir sama di setiap data pengujian. Sedangkan dalam Naive Bayes probabilitas umur yang paling besar sebesar $7,2847812 \times 10^{-6} \alpha$ dengan umur 55 tahun.

Tabel 4.27 *Inputan* Pengujian pada Nadi

Variabel	Gejala Penyakit Stroke									
	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
Umur	50	50	50	50	50	50	50	50	50	50
Nadi	50	55	60	65	70	75	80	85	90	95
Kelamin	0	0	0	0	0	0	0	0	0	0
Kolesterol	200	200	200	200	200	200	200	200	200	200
Hipertensi	0	0	0	0	0	0	0	0	0	0
DM	0	0	0	0	0	0	0	0	0	0
Kesadaran	0	0	0	0	0	0	0	0	0	0
Pusing	0	0	0	0	0	0	0	0	0	0
Mual	0	0	0	0	0	0	0	0	0	0
Kesemutan	0	0	0	0	0	0	0	0	0	0
Pernah Stroke	0	0	0	0	0	0	0	0	0	0

Berdasarkan pengujian yang sudah dilakukan, dengan memasukan *input* data gejala yang ada maka diperoleh hasil faktor yang paling berpengaruh dalam metode yang dapat dilihat pada Tabel 4.28.

Pusing	0	0	0	0	0	0	0	0	0	0
Mual	0	0	0	0	0	0	0	0	0	0
Kesemutan	0	0	0	0	0	0	0	0	0	0
Pernah Stroke	0	0	0	0	0	0	0	0	0	0

Berdasarkan pengujian yang sudah dilakukan, dengan memasukan *input* data gejala yang ada maka diperoleh hasil faktor yang paling berpengaruh dalam metode yang dapat dilihat pada Tabel 4.30.

Tabel 4.30 Hasil pengujian Faktor Variabel Numerik Kolesterol yang Paling Berpengaruh

Pengujian data ke-	<i>Output</i> bobot JST	Hasil Uji JST	<i>Output</i> Bobot	Perhitungan Probabilitas terbesar	Hasil Uji Naive Bayes
V1	0.00148165	0	0	$5,098491 \times 10^{-7} \alpha$	0
V2	0.00148164	0	0	$5,0948132 \times 10^{-7} \alpha$	0
V3	0.00148163	0	0	$6,04812 \times 10^{-7} \alpha$	1
V4	0.00148161	0	0	$6,91247912 \times 10^{-7} \alpha$	1
V5	0.0014815	0	0	$5,1948129 \times 10^{-7} \alpha$	1
V6	0.0014811	0	0	$6,847812 \times 10^{-7} \alpha$	1
V7	0.00148008	0	0	$5,294729 \times 10^{-7} \alpha$	1
V8	0.00147403	0	0	$5,01839127 \times 10^{-7} \alpha$	1
V9	0.00145325	0	0	$4,9824712 \times 10^{-7} \alpha$	1
V10	0.00137585	0	0	$4,28461023 \times 10^{-7} \alpha$	1

Dapat dilihat pada Tabel 4.29 dengan analisis Jaringan Syaraf Tiruan *output* bobot terbesar terdapat pada kolesterol 150 mg/dL, tetapi nadi tidak berpengaruh dalam deteksi penyakit stroke. Sedangkan dalam Naive Bayes probabilitas kolesterol yang paling besar sebesar $6,847812 \times 10^{-7} \alpha$ dengan kolesterol 200 mg/dL.

Berdasarkan hasil di atas menunjukkan bahwa data opsional memiliki pengaruh di dalam penentuan *Suspect* atau tidaknya pada Jaringan Syaraf Tiruan. Sedangkan di dalam hasil data numerik besar kecil nilai dari data tersebut tidak

memiliki pengaruh yang cukup besar, dapat dilihat pada hasil *output* bobot kerja sistem yang memiliki nilai hampir sama dan selisih yang sama di setiap data. Maka Jaringan Syaraf Tiruan baik digunakan untuk perhitungan satu jenis data. Tetapi, di dalam perhitungan Naive Bayes kedua jenis data (data numerik dan data opsional) yang berbeda memiliki pengaruh yang sama dalam penentuan *Suspect* stroke atau tidak karena di setiap data tersebut memiliki nilai probabilitas dan hasil likelihood yang akan menjadi penentu *Suspect* stroke atau tidak.

Di dalam hasil penelitian menunjukkan bahwa Naive Bayes memiliki tingkat akuratan lebih baik dibandingkan dengan Jaringan Syaraf Tiruan, tetapi Jaringan Syaraf Tiruan memiliki teknik yang lebih bagus dibandingkan dengan Naive Bayes. Jaringan Syaraf Tiruan mempunyai karakteristik yang adaptif yaitu belajar dari data sebelumnya (data latih) dan mengenal pola data yang selalu berubah tanpa harus memperdulikan target yang ada. Sedangkan di dalam penelitian menunjukkan bahwa data target pada Naive Bayes dan hasil probabilitas dari data yang diinputkan sangat berpengaruh terhadap hasil *output*. Semakin besar nilai probabilitas dari *input* yang dimasukkan kecenderungan nilai hasil semakin tidak konsisten.

BAB V

PENUTUP

Pada bab ini diuraikan beberapa kesimpulan dari pengujian aplikasi dan beberapa saran yang dapat menyempurnakan perancangan dalam yang telah dibuat.

5.1 Simpulan

Dari hasil pembahasan pada bab IV dapat ditarik kesimpulan sebagai berikut.

1. Sistem deteksi penyakit stroke diciptakan menggunakan *software* Matlab. Data diolah dengan menggunakan data dari gejala-gejala umum pada penyakit stroke yang dijadikan sebagai data latih dan data *Suspectstroke* atau tidak digunakan sebagai target. Dalam pembuatan sistem, tahap yang dilakukan adalah membuat desain program dengan menggunakan *Graphics User Interface* (GUI) kemudian dilanjutkan dengan menggunakan *coding* Matlab supaya desain program berfungsi. Setelah proses-proses tersebut, sistem dapat digunakan mendeteksi *Suspectstroke* atau tidak.
2. Berdasarkan hasil pengujian pada data uji yang telah dilakukan maka dapat diketahui bahwa metode Jaringan Syaraf Tiruan diperoleh prosentase keberhasilan sebesar 71,11 persen, sedangkan dalam Naive Bayes diperoleh hasil persentase sebesar 80,555 persen.
3. Di dalam hasil penelitian menunjukkan bahwa Naive Bayes memiliki tingkat akuratan lebih baik dibandingkan dengan Jaringan Syaraf Tiruan, namun

Jaringan Syaraf Tiruan memiliki teknik yang lebih bagus dibandingkan dengan Naive Bayes. Jaringan Syaraf Tiruan mempunyai karakteristik yang adaptif yaitu belajar dari data sebelumnya (data latih) dan mengenal pola data yang selalu berubah tanpa harus memperdulikan data target yang ada. Sedangkan data target pada Naive Bayes dan hasil probabilitas dari data yang diinputkan sangat berpengaruh terhadap hasil *output*. Semakin besar nilai probabilitas dari *input* yang dimasukkan kecenderungan nilai hasil semakin tidak konsisten.

5.2 Saran

Saran yang dapat ditulis untuk penelitian ini sebagai berikut.

1. Sistem ini memungkinkan untuk dikembangkan dengan menggunakan *software* lain selain *software* Matlab supaya lebih mudah digunakan, karena *software* Matlab memerlukan memori yang cukup banyak di dalam komputer.
2. Penggunaan metode lain perlu dilakukan dalam penanganan masalah yang sama agar dapat menilai metode mana yang paling cocok dengan permasalahan yang ada.
3. Untuk penelitian selanjutnya diharapkan adanya variabel *input* yang lebih rinci sebagai data uji dan data target karena mungkin masih banyak faktor yang menyebabkan seseorang terdeteksi penyakit stroke.
4. Penggunaan satu jenis data saja perlu dilakukan dalam penanganan masalah yang sama agar mengetahui perbandingan tingkat akuratan kerja sistem.

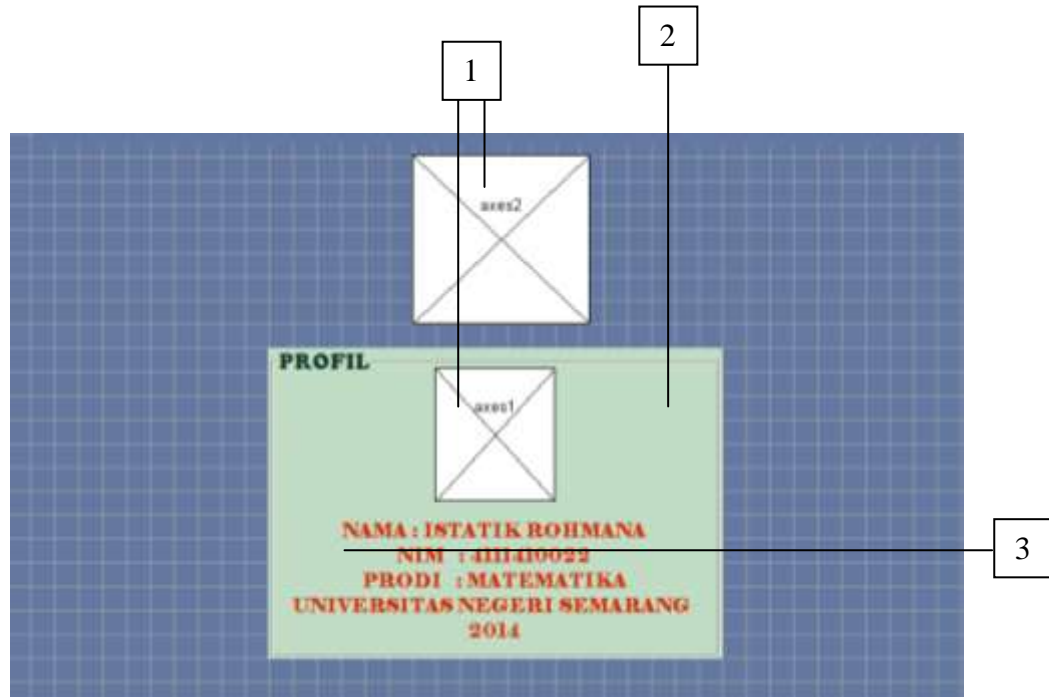
DAFTAR PUSTAKA

- Amin, S. 2012. *Sistem Deteksi Dini Hama Wereng Batang Coklat Menggunakan Jaringan Syaraf Tiruan Backpropagation*. Skripsi. Semarang: FMIPA Universitas Negeri Semarang.
- Alfironi, 2013. *Implementasi Data Mining Dengan Naive Bayes Classifier Untuk Mendukung Strategi Pemasaran Di Bagian Humas Stmik Amikom Yogyakarta*. Skripsi. Yogyakarta: Amikom Yogyakarta.
- Basuki, A. 2006. *Metode Bayes*. Surabaya : PENS-ITS.
- Brunser, A.M., dkk. 2013. Accuracy of Diffusion- Weighted Imaging in the Deteksi of Stroke in Patiens With *Suspected Cerebral Infarct*. *Stroke*. American Heart Association, 44(4) : 1169-1171. Tersedia di <http://Stroke.ahajournals.org/content/44/4/1169> [diakses 26 April 2014].
- Demetgul, M., I.N. Tansel, & S. Taskin. 2009. Fault Deteksi Of Pneumatic Sitemns With Artificial Neural Network Algorithms. *Expert Sitemns with Applications*, 36(8) : 10512-10519. Tersedia di <http://www.elsevier.com/locate/eswa> [diakses 20-09-2013].
- Departemen Kesehatan RI Provinsi Jawa Tengah. 2007. Riset Kesehatan Dasar-Dinkes Jateng. Tersedia di <http://www.dinkesjatengprov.go.id/mi/riskesda.jateng2007.html> [diakses 18-01-2014].
- Departemen Kesehatan RI Provinsi Jawa Tengah. 2009. Daftar Tabel Profil Kesehatan Provinsi Jawa Tengah Tahun 2009. Tersedia di http://dinkesjatengprov.go.id/dokumen/profil/2009/profil_2009br.pdf [diakses 18-01-2014].
- Eliyani. 2005. Pengantar Jaringan Syaraf Tiruan. Tersedia di materikuliaah.com [diakses 20-11-2013].
- Hermawan,A. 2006. *Jaringan Syaraf Tiruan Teori dan Aplikasi*. Yogyakarta: ANDI.
- Iqbal, M.2009. *Dasar Pengolahan Citra Menggunakan MATLAB*. Department Ilmu dan Tehnologi Kelautan IPB.
- Kusrini. 2006. *Sistem Pakar, Teori dan Aplikasi*. Yogyakarta: ANDI.
- Kusumadewi, S. 2002. *Membangun Jaringan Syaraf Tiruan dengan Matlab dan Exel Link*. Yogyakarta: Graha Ilmu.
- Lingga, L. 2013. *All About Stroke*.Jakarta: Elex Media Komputindo.

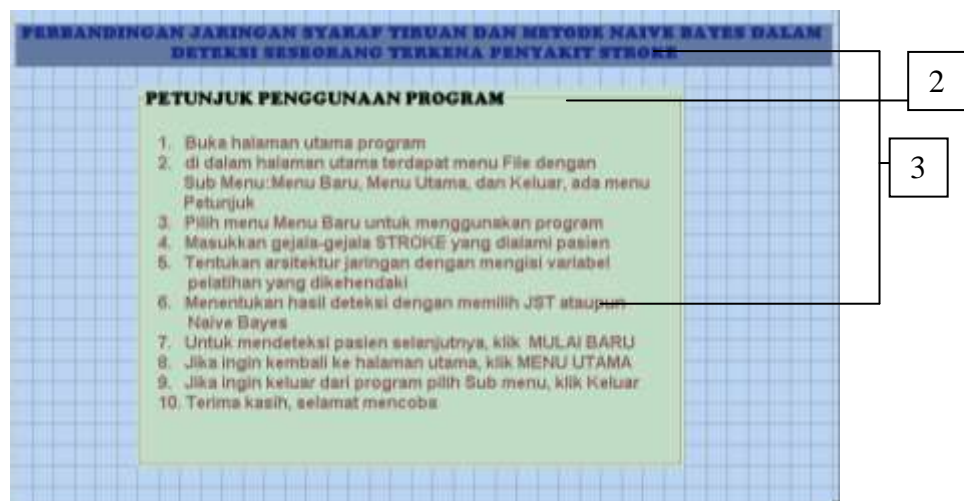
- Maes S., dkk. 2007. Credit Card. Fraud Detection Using Bayesian and Neural Networks. *Prosiding* Department of Computer Science Computational Modeling Lab (COMO). Brussel : Vrije Universiteit Brussel.
- Michael L.A., & P. Wilding. 1992. Application of Neural Networks to the Interpretation of Laboratory Data in Cancer Deteksi. *Clinical Chemistry*, 38(1): 34-38.
- Pandjaitan, L.W. 2007. *Dasar-dasar Komputasi Cerdas*. Yogyakarta: ANDI.
- Prabowo, dkk. 2013. *Penerapan Data Mining Dengan Matlab*. Bandung: Rekayasa SAINS.
- Pressman, R.S. 1997. *Rekayasa Perangkat Lunak Pendekatan Praktisi*. Translated by Harnaningrum.2002. Indonesia: ANDI and McGraw-Hill Book Co.
- Purnamasari, R.W. 2013. *Implementasi Jaringan Syaraf Tiruan Backpropagation Sebagai Sistem Deteksi Penyakit Tuberculosis (TBC)*. Skripsi. Semarang: Universitas Negeri Semarang.
- Puspitaningrum, D. 2006. *Pengantar Jaringan Syaraf Tiruan*. Yogyakarta: ANDI.
- Ratna, D. A. 2011. *Sistem Kendali Cerdas*. Yogyakarta: Graha Ilmu.
- Siang, J.J. 2009. *Jaringan Syaraf Tiruan Dan Pemrogramannya Menggunakan Matlab*. Yogyakarta: ANDI.
- Soebanar. 2008. *Inferensi Bayesian*. Jakarta: Universitas Terbuka.
- WHO. 2004. Stroke, Cerebrovascular accident. Tersedia di http://www.who.int/topics/cerebrovascular_accident/en/[di akses 26 April 2014].
- Widjaja D. 2005. Stroke-Masa Kini dan Masa Yang Akan Datang. Tersedia di <http://www.kalbe.co.id/files/cdk/files/13Stroke102.pdf/13Stroke102.html>[di akses 26 April 2014].

Lampiran 1

Desain Tampilan *Interface* untuk Halaman Utama



Gambar Desain Tampilan Halaman Utama



Gambar Desain Tampilan Petunjuk Penggunaan

Keterangan Gambar

No.	Nama Komponen	Title	Setting
1.	<i>Axes</i>	-	axes1
		-	axes2
2.	<i>Panel</i>	PROFIL	uipanel1
		PETUNJUK PENGGUNAAN ...	uipanel2
3.	<i>Static text</i>	NAMA ISTATIK ROHMANA ...	text3
		1. Buka halaman utama program ...	text4

*Lampiran 2***Kode Program untuk halaman Depan**

```

function help_Callback(hObject, eventdata, handles)
% hObject    handle to help (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
help
close(halutama)

% -----
----
function new_Callback(hObject, eventdata, handles)
% hObject    handle to new (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
soft_main
close(halutama)

% -----
----
function awal_Callback(hObject, eventdata, handles)
% hObject    handle to awal (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
----
function exit_Callback(hObject, eventdata, handles)
% hObject    handle to exit (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
close

```

Lampiran 3

Desain Tampilan Program Deteksi Penyakit Stroke

The image shows the user interface of a Stroke Disease Detection Program. It is divided into several sections with numbered callouts:

- 1**: Points to the 'Data Input' section, which includes fields for 'Umur' (Age), 'Jenis Kelamin' (Gender), 'Kolesterol', 'Tekanan Darah' (Blood Pressure), 'Gula Darah' (Blood Sugar), 'Merokok' (Smoking), 'Minum Alkohol' (Alcohol Consumption), 'Riwayat Stroke' (Stroke History), 'Riwayat Penyakit Jantung' (Heart Disease History), 'Riwayat Penyakit Ginjal' (Kidney Disease History), 'Riwayat Penyakit Diabetes' (Diabetes History), 'Riwayat Penyakit Hipertensi' (Hypertension History), 'Riwayat Penyakit Asam Lambung' (Stomach Acid History), 'Riwayat Penyakit Osteoporosis' (Osteoporosis History), 'Riwayat Penyakit Demensia' (Dementia History), 'Riwayat Penyakit Alzheimer' (Alzheimer's History), 'Riwayat Penyakit Parkinson' (Parkinson's History), 'Riwayat Penyakit Multiple Sclerosis' (Multiple Sclerosis History), 'Riwayat Penyakit Epilepsi' (Epilepsy History), 'Riwayat Penyakit Skizofrenia' (Schizophrenia History), 'Riwayat Penyakit Bipolar' (Bipolar History), 'Riwayat Penyakit Depresi' (Depression History), 'Riwayat Penyakit Kecemasan' (Anxiety History), 'Riwayat Penyakit Gangguan Obsesif-Kompulsif' (Obsessive-Compulsive Disorder History), 'Riwayat Penyakit Gangguan Perilaku' (Behavioral Disorder History), 'Riwayat Penyakit Gangguan Kepribadian' (Personality Disorder History), 'Riwayat Penyakit Gangguan Psikotik' (Psychotic Disorder History), 'Riwayat Penyakit Gangguan Psikosis' (Psychosis History), 'Riwayat Penyakit Gangguan Psikotik' (Psychotic Disorder History), 'Riwayat Penyakit Gangguan Psikosis' (Psychosis History).
- 2**: Points to the 'Perhitungan' (Calculation) section, which includes buttons for 'View Parameter JST', 'Training', 'Epoch', 'Learning Rate', 'Goal', 'Jumlah Layer & Neuron', and 'Jumlah Neuron per Layer'.
- 3**: Points to the 'JST' (Jaringan Saraf Tiruan) section, which includes a 'JST' button and a 'Native Output' button.
- 4**: Points to the 'Hasil deteksi' (Detection Result) section, which includes a text area for the result and a 'Duplikat Hasil' (Duplicate Result) button.
- 5**: Points to the 'Output' section, which includes a 'Output' button and a 'MULAI KALKULASI' (Start Calculation) button.
- 6**: Points to the 'KETERANGAN' (Explanation) section, which includes a text box with the following text:

KETERANGAN :
 FYA = 1 dan TIDAK=0
 JENIS PENYAKIT LAIN LAIN = 1 dan TIDAK=0
- 7**: Points to the 'Data Input' section, which includes fields for 'Umur', 'Jenis Kelamin', 'Kolesterol', 'Tekanan Darah', 'Gula Darah', 'Merokok', 'Minum Alkohol', 'Riwayat Stroke', 'Riwayat Penyakit Jantung', 'Riwayat Penyakit Ginjal', 'Riwayat Penyakit Diabetes', 'Riwayat Penyakit Hipertensi', 'Riwayat Penyakit Asam Lambung', 'Riwayat Penyakit Osteoporosis', 'Riwayat Penyakit Demensia', 'Riwayat Penyakit Alzheimer', 'Riwayat Penyakit Parkinson', 'Riwayat Penyakit Multiple Sclerosis', 'Riwayat Penyakit Epilepsi', 'Riwayat Penyakit Skizofrenia', 'Riwayat Penyakit Bipolar', 'Riwayat Penyakit Depresi', 'Riwayat Penyakit Kecemasan', 'Riwayat Penyakit Gangguan Obsesif-Kompulsif', 'Riwayat Penyakit Gangguan Perilaku', 'Riwayat Penyakit Gangguan Kepribadian', 'Riwayat Penyakit Gangguan Psikotik', 'Riwayat Penyakit Gangguan Psikosis'.

Gambar Desain Tampilan Form Deteksi Penyakit Stroke

Keterangan Gambar

No.	Nama komponen	Title	Setting
1	<i>Static text</i>	Umur Nadi Kelamin Kolesterol Hipertensi Diabetes Militus Kesadaran Pusing Mual Kesemutan Riwayat Stroke Epoch Learning Rate Jumlah Layer & Neuron Goal <i>Output Bobot</i>	text1, text2, text21, text4, text5, text6, text7, text8, text9, text10, text11, text13, text14, text15, text16, text22
2	<i>Edit text</i>	- - - - - - - - - - - - 1000 0.001 5 1e-6 -	edit1, edit2, edi3, edit4, edit5, edit6, edit7, edit8, edit9, edit10, edit11, edit12, edit13, edit14, edit16, edit 18
3	<i>Panel</i>	Data <i>Input</i> Perhitungan	uipanel1, unpanel2
4	<i>Pushbutton</i>	View Parameter JST Training JST Naive Bayes MULAI BARU -	pushbutton1, pushbutton3, pushbutton5, pushbutton2, pushbutton6 edit15
5	<i>Edit</i>	KETERANGAN:	text17,
6	<i>Static text</i>	IYA =1 DAN TIDAK = 0 Jenis Kelamin Laki-laki =1, Perempuan=0 Laki-laki, perempuan	text18, text19
7	<i>Pop-up Menu</i>	Hipertensi, tidak hipertensi DM, tidak DM, Pingsan, tidak pingsan Pusing, tidak pusing Mual, tidak mual Kesemutan, tidak kesemutan Pernah Stroke, tidak Stroke	popupmenu1, popupmenu3, popupmenu4, popupmenu6, popupmenu7, popupmenu8, popupmenu9, popupmenu10

*Lampiran 4***Kode Program Deteksi Penyakit Stroke**

```

function varargout = soft_main(varargin)
% SOFT_MAIN M-file for soft_main.fig
%     SOFT_MAIN, by itself, creates a new SOFT_MAIN or raises the
existing
%     singleton*.
%
%     H = SOFT_MAIN returns the handle to a new SOFT_MAIN or the
handle to
%     the existing singleton*.
%
%     SOFT_MAIN('CALLBACK',hObject,eventData,handles,...) calls
the local
%     function named CALLBACK in SOFT_MAIN.M with the given input
arguments.
%
%     SOFT_MAIN('Property','Value',...) creates a new SOFT_MAIN
or raises the
%     existing singleton*. Starting from the left, property
value pairs are
%     applied to the GUI before soft_main_OpeningFcn gets called. An
unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to soft_main_OpeningFcn via
varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help soft_main

% Last Modified by GUIDE v2.5 26-Aug-2014 17:59:14

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
'gui_Singleton',  gui_Singleton, ...
'gui_OpeningFcn', @soft_main_OpeningFcn, ...
'gui_OutputFcn',  @soft_main_OutputFcn, ...
'gui_LayoutFcn',  [] , ...
'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

```

```

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

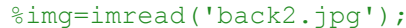

% --- Executes just before soft_main is made visible.
function soft_main_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to soft_main (see VARARGIN)

% Choose default command line output for soft_main
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes soft_main wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = soft_main_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;
;
%axes(handles.axes2)

set(handles.text13, 'visible', 'off')
set(handles.text14, 'visible', 'off')
set(handles.text15, 'visible', 'off')
set(handles.text16, 'visible', 'off')

set(handles.edit12, 'visible', 'off')
set(handles.edit13, 'visible', 'off')
set(handles.edit14, 'visible', 'off')
set(handles.edit16, 'visible', 'off')

set(handles.pushbutton3, 'enable', 'off')

```



```

function edit1_Callback(hObject, eventdata, handles)
% hObject      handle to edit1 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%         str2double(get(hObject,'String')) returns contents of
edit1 as a double

% --- Executes during object creation, after setting all
properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit1 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)
% hObject      handle to edit2 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%         str2double(get(hObject,'String')) returns contents of
edit2 as a double

% --- Executes during object creation, after setting all
properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit2 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

end

```
function edit3_Callback(hObject, eventdata, handles)
% hObject      handle to edit3 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text
%         str2double(get(hObject,'String')) returns contents of
edit3 as a double

% --- Executes during object creation, after setting all
properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit3 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function edit4_Callback(hObject, eventdata, handles)
% hObject      handle to edit4 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit4 as text
%         str2double(get(hObject,'String')) returns contents of
edit4 as a double

% --- Executes during object creation, after setting all
properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit4 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
```

```

%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit5_Callback(hObject, eventdata, handles)
% hObject      handle to edit5 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit5 as text
%          str2double(get(hObject,'String')) returns contents of
edit5 as a double

% --- Executes during object creation, after setting all
properties.
function edit5_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit5 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit6_Callback(hObject, eventdata, handles)
% hObject      handle to edit6 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit6 as text
%          str2double(get(hObject,'String')) returns contents of
edit6 as a double

% --- Executes during object creation, after setting all
properties.
function edit6_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit6 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB

```

```

% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit7_Callback(hObject, eventdata, handles)
% hObject    handle to edit7 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit7 as text
%         str2double(get(hObject,'String')) returns contents of
edit7 as a double

% --- Executes during object creation, after setting all
properties.
function edit7_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit7 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit8_Callback(hObject, eventdata, handles)
% hObject    handle to edit8 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit8 as text
%         str2double(get(hObject,'String')) returns contents of
edit8 as a double

% --- Executes during object creation, after setting all
properties.

```

```

function edit8_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit8 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function edit9_Callback(hObject, eventdata, handles)
% hObject      handle to edit9 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit9 as text
%         str2double(get(hObject,'String')) returns contents of
edit9 as a double

```

```

% --- Executes during object creation, after setting all
properties.
function edit9_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit9 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function edit10_Callback(hObject, eventdata, handles)
% hObject      handle to edit10 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit10 as text
%         str2double(get(hObject,'String')) returns contents of
edit10 as a double

```

```

% --- Executes during object creation, after setting all
properties.
function edit10_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit10 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit11_Callback(hObject, eventdata, handles)
% hObject    handle to edit11 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit11 as text
%         str2double(get(hObject,'String')) returns contents of
edit11 as a double

% --- Executes during object creation, after setting all
properties.
function edit11_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit11 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

set(handles.text13,'visible','on')
set(handles.text14,'visible','on')
set(handles.text15,'visible','on')
set(handles.text16,'visible','on')

set(handles.edit12,'visible','on')
set(handles.edit13,'visible','on')
set(handles.edit14,'visible','on')
set(handles.edit16,'visible','on')

set(handles.pushbutton3,'enable','on')
% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
data_latih
data=(A(:,[2:13]));
data_lat=data(:,1:end-1);
x1=str2num(get(handles.edit1,'string'));
x2=str2num(get(handles.edit2,'string'));
x3=str2num(get(handles.edit3,'string'));
x4=str2num(get(handles.edit4,'string'));
x5=str2num(get(handles.edit5,'string'));
x6=str2num(get(handles.edit6,'string'));
x7=str2num(get(handles.edit7,'string'));
x8=str2num(get(handles.edit8,'string'));
x9=str2num(get(handles.edit9,'string'));
x10=str2num(get(handles.edit10,'string'));
x11=str2num(get(handles.edit11,'string'));
data_uji=[x1;x2;x3;x4;x5;x6;x7;x8;x9;x10;x11]';
kelas_latih=data(:,end);
model=NaiveBayes.fit(data_lat, kelas_latih,'Distribution', 'mn');
hasil_uji=model.predict(data_uji)
set(handles.edit18,'string',hasil_uji)
if hasil_uji ==0
    set(handles.edit15,'string','Pasien tersebut tidak Suspect
Stroke')
elseif hasil_uji ==1
    set(handles.edit15,'string','Pasien tersebut Suspect Stroke')
end

function edit12_Callback(hObject, eventdata, handles)
% hObject    handle to edit12 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit12 as text
%        str2double(get(hObject,'String')) returns contents of
edit12 as a double

```

```

% --- Executes during object creation, after setting all
properties.
function edit12_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit12 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit13_Callback(hObject, eventdata, handles)
% hObject    handle to edit13 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit13 as text
%       str2double(get(hObject,'String')) returns contents of
edit13 as a double

% --- Executes during object creation, after setting all
properties.
function edit13_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit13 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit14_Callback(hObject, eventdata, handles)
% hObject    handle to edit14 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit14 as text

```



```

%          str2double(get(hObject,'String')) returns contents of
edit14 as a double

% --- Executes during object creation, after setting all
properties.
function edit14_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit14 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit15_Callback(hObject, eventdata, handles)
% hObject    handle to edit15 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit15 as text
%         str2double(get(hObject,'String')) returns contents of
edit15 as a double

% --- Executes during object creation, after setting all
properties.
function edit15_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit15 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit16_Callback(hObject, eventdata, handles)
% hObject    handle to edit16 (see GCBO)

```

```

% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit16 as text
% str2double(get(hObject,'String')) returns contents of
edit16 as a double

% --- Executes during object creation, after setting all
properties.
function edit16_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit16 (see GCBO)
% eventdata reserved - to be defined in a future version of
MATLAB
% handles empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton3 (see GCBO)
% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see GUIDATA)

%% data input
data_latih
clc
P=A(:,2:12);
T=A(:,13);
P=P';
T=T';
whos T
%% membangun jaringan saraf tiruan
Jumlah_Layer_Neuron=str2num(get(handles.edit14,'string'));
net=newff(P,T,Jumlah_Layer_Neuron);

%% paramater jaringan
net.trainParam.epochs=str2num(get(handles.edit12,'string'));
net.trainParam.goal=str2num(get(handles.edit16,'string'));
net.trainParam.lr=str2num(get(handles.edit13,'string'));
net.trainParam.show=1;

%% pelatihan jaringan
net=train(net,P,T);

```

```

%% simulasi
an=abs(sim(net,P))'
whos an
%sim_result=[an T]
save netnet

%nntraintool('close')
%break
%% pengujian
data_uji
uji=U(:,2:12)';
target_uji=U(:,13);
%% hasil uji
hasil_uji=round(abs(sim(net,uji)))
Hasil_Uji_vs_Target=[hasil_uji target_uji ]
set(handles.text13,'visible','off')
set(handles.text14,'visible','off')
set(handles.text15,'visible','off')
set(handles.text16,'visible','off')

set(handles.edit12,'visible','off')
set(handles.edit13,'visible','off')
set(handles.edit14,'visible','off')
set(handles.edit16,'visible','off')

set(handles.pushbutton3,'enable','off')

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton5 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)
%% pengujian
load net
data_uji
x1=str2num(get(handles.edit1,'string'));
x2=str2num(get(handles.edit2,'string'));
x3=str2num(get(handles.edit3,'string'));
x4=str2num(get(handles.edit4,'string'));
x5=str2num(get(handles.edit5,'string'));
x6=str2num(get(handles.edit6,'string'));
x7=str2num(get(handles.edit7,'string'));
x8=str2num(get(handles.edit8,'string'));
x9=str2num(get(handles.edit9,'string'));
x10=str2num(get(handles.edit10,'string'));
x11=str2num(get(handles.edit11,'string'));

uji=[x1;x2;x3;x4;x5;x6;x7;x8;x9;x10;x11];
target_uji=U(:,13);
%% hasil uji
hasil_uji=(abs(sim(net,uji)))
hasil_uji=round(abs(sim(net,uji)))

```

```

set(handles.edit18,'string',hasil_ujio)
if hasil_uji ==0
    set(handles.edit15,'string','Pasien tersebut tidak Suspect
Stroke')
elseif hasil_uji ==1
    set(handles.edit15,'string','Pasien tersebut Suspect Stroke')
end

% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton6 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.edit1,'String','');
set(handles.edit2,'String','');
set(handles.edit3,'String','');
set(handles.edit4,'String','');
set(handles.edit5,'String','');
set(handles.edit6,'String','');
set(handles.edit7,'String','');
set(handles.edit8,'String','');
set(handles.edit9,'String','');
set(handles.edit10,'String','');
set(handles.edit11,'String','');
set(handles.edit15,'String','Hasil Deteksi...');
set(handles.edit18,'String','');

% --- Executes on selection change in popupmenu5.
function popupmenu5_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu5 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns popupmenu5
contents as cell array
%         contents{get(hObject,'Value')} returns selected item from
popupmenu5

% --- Executes during object creation, after setting all
properties.
function popupmenu5_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu5 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: popupmenu controls usually have a white background on
Windows.
%         See ISPC and COMPUTER.

```

```

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in popupmenu1.
function popupmenu1_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns popupmenu1
contents as cell array
%         contents{get(hObject,'Value')} returns selected item from
popupmenu1
v=get(handles.popupmenu1,'Value')
switch v
case 1
set(handles.edit3,'string','1');
case 2
set(handles.edit3,'string','0');
end

% --- Executes during object creation, after setting all
properties.
function popupmenu1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: popupmenu controls usually have a white background on
Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in popupmenu3.
function popupmenu3_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu3 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns popupmenu3
contents as cell array
%         contents{get(hObject,'Value')} returns selected item from
popupmenu3

```

```

v=get(handles.popupmenu3,'Value')
switch v
case 1
set(handles.edit5,'string','1');
case 2
set(handles.edit5,'string','0');
end

% --- Executes during object creation, after setting all
properties.
function popupmenu3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu3 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: popupmenu controls usually have a white background on
Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in popupmenu4.
function popupmenu4_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu4 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns popupmenu4
contents as cell array
%         contents{get(hObject,'Value')} returns selected item from
popupmenu4
v=get(handles.popupmenu4,'Value')
switch v
case 1
set(handles.edit6,'string','1');
case 2
set(handles.edit6,'string','0');
end

% --- Executes during object creation, after setting all
properties.
function popupmenu4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu4 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

```

```

% Hint: popupmenu controls usually have a white background on
Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in popupmenu6.
function popupmenu6_Callback(hObject, eventdata, handles)
% hObject      handle to popupmenu6 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns popupmenu6
contents as cell array
%       contents{get(hObject,'Value')} returns selected item from
popupmenu6
v=get(handles.popupmenu6,'Value')
switch v
case 1
    set(handles.edit7,'string','1');
case 2
    set(handles.edit7,'string','0');
end

% --- Executes during object creation, after setting all
properties.
function popupmenu6_CreateFcn(hObject, eventdata, handles)
% hObject      handle to popupmenu6 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: popupmenu controls usually have a white background on
Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in popupmenu7.
function popupmenu7_Callback(hObject, eventdata, handles)
% hObject      handle to popupmenu7 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns popupmenu7
contents as cell array

```

```

%           contents{get(hObject,'Value')} returns selected item from
popupmenu7
v=get(handles.popupmenu7,'Value')
switch v
case 1
set(handles.edit8,'string','1');
case 2
set(handles.edit8,'string','0');
end

% --- Executes during object creation, after setting all
properties.
function popupmenu7_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu7 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: popupmenu controls usually have a white background on
Windows.
%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in popupmenu8.
function popupmenu8_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu8 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns popupmenu8
contents as cell array
%           contents{get(hObject,'Value')} returns selected item from
popupmenu8
v=get(handles.popupmenu8,'Value')
switch v
case 1
set(handles.edit9,'string','1');
case 2
set(handles.edit9,'string','0');
end

% --- Executes during object creation, after setting all
properties.
function popupmenu8_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu8 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB

```



```

% handles    empty - handles not created until after all
CreateFcns called

% Hint: popupmenu controls usually have a white background on
Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in popupmenu9.
function popupmenu9_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu9 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns popupmenu9
contents as cell array
%         contents{get(hObject,'Value')} returns selected item from
popupmenu9
v=get(handles.popupmenu9,'Value')
switch v
case 1
    set(handles.edit10,'string','1');
case 2
    set(handles.edit10,'string','0');
end

% --- Executes during object creation, after setting all
properties.
function popupmenu9_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu9 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: popupmenu controls usually have a white background on
Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in popupmenu10.
function popupmenu10_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu10 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

% Hints: contents = get(hObject,'String') returns popupmenu10
contents as cell array
%         contents{get(hObject,'Value')} returns selected item from
popupmenu10
v=get(handles.popupmenu10,'Value')
switch v
case 1
set(handles.edit11,'string','1');
case 2
set(handles.edit11,'string','0');
end

% --- Executes during object creation, after setting all
properties.
function popupmenu10_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu10 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: popupmenu controls usually have a white background on
Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit18_Callback(hObject, eventdata, handles)
% hObject    handle to edit18 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit18 as text
%         str2double(get(hObject,'String')) returns contents of
edit18 as a double

% --- Executes during object creation, after setting all
properties.
function edit18_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit18 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.

```

```

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all
properties.
function text22_CreateFcn(hObject, eventdata, handles)
% hObject    handle to text22 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% -----
----
function file_Callback(hObject, eventdata, handles)
% hObject    handle to file (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
----
function help_Callback(hObject, eventdata, handles)
% hObject    handle to help (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
help
close (soft_main)

% -----
----
function new_Callback(hObject, eventdata, handles)
% hObject    handle to new (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
----
function awal_Callback(hObject, eventdata, handles)
% hObject    handle to awal (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
halutama
close (soft_main)

```

```
% -----  
-----  
function exit_Callback(hObject, eventdata, handles)  
% hObject    handle to exit (see GCBO)  
% eventdata  reserved - to be defined in a future version of  
MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
close
```

Lampiran 5

Kode Jaringan Syaraf Tiruan

```
%% data input
data_latih
clc
P=A(:,2:12);
T=A(:,13);
P=P';
T=T';
whos T
%% membangun jaringan saraf tiruan
Jumlah_Layer_Neuron=str2num(get(handles.edit14,'string'));
net=newff(P,T,Jumlah_Layer_Neuron);

%% paramater jaringan
net.trainParam.epochs=str2num(get(handles.edit12,'string'));
net.trainParam.goal=str2num(get(handles.edit16,'string'));
net.trainParam.lr=str2num(get(handles.edit13,'string'));
net.trainParam.show=1;

%% pelatihan jaringan
net=train(net,P,T);

%% simulasi
an=abs(sim(net,P))'
whos an
%sim_result=[an T]
save netnet

%nntraintool('close')
%break
%% pengujian
data_uji
uji=U(:,2:12)';
target_uji=U(:,13);
%% hasil uji
hasil_uji=round(abs(sim(net,uji)))
Hasil_Uji_vs_Target=[hasil_uji target_uji ]
set(handles.text13,'visible','off')
set(handles.text14,'visible','off')
set(handles.text15,'visible','off')
set(handles.text16,'visible','off')

set(handles.edit12,'visible','off')
set(handles.edit13,'visible','off')
set(handles.edit14,'visible','off')
set(handles.edit16,'visible','off')

set(handles.pushbutton3,'enable','off')
```

```

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
%% pengujian
load net
data_uji
x1=str2num(get(handles.edit1,'string'));
x2=str2num(get(handles.edit2,'string'));
x3=str2num(get(handles.edit3,'string'));
x4=str2num(get(handles.edit4,'string'));
x5=str2num(get(handles.edit5,'string'));
x6=str2num(get(handles.edit6,'string'));
x7=str2num(get(handles.edit7,'string'));
x8=str2num(get(handles.edit8,'string'));
x9=str2num(get(handles.edit9,'string'));
x10=str2num(get(handles.edit10,'string'));
x11=str2num(get(handles.edit11,'string'));

uji=[x1;x2;x3;x4;x5;x6;x7;x8;x9;x10;x11];
target_uji=U(:,13);
%% hasil uji
hasil_ujio=(abs(sim(net,uji)))
hasil_uji=round(abs(sim(net,uji)))
set(handles.edit18,'string',hasil_ujio)
if hasil_uji ==0
    set(handles.edit15,'string','Pasien tersebut tidak
SuspectStroke')
elseif hasil_uji ==1
    set(handles.edit15,'string','Pasien tersebut SuspectStroke')
end

```

Lampiran 6

Kode Naive Bayes

```

data_latih
data=(A(:,[2:13]));
data_lat=data(:,1:end-1);
x1=str2num(get(handles.edit1,'string'));
x2=str2num(get(handles.edit2,'string'));
x3=str2num(get(handles.edit3,'string'));
x4=str2num(get(handles.edit4,'string'));
x5=str2num(get(handles.edit5,'string'));
x6=str2num(get(handles.edit6,'string'));
x7=str2num(get(handles.edit7,'string'));
x8=str2num(get(handles.edit8,'string'));
x9=str2num(get(handles.edit9,'string'));
x10=str2num(get(handles.edit10,'string'));
x11=str2num(get(handles.edit11,'string'));
data_uji=[x1;x2;x3;x4;x5;x6;x7;x8;x9;x10;x11]';
kelas_latih=data(:,end);
model=NaiveBayes.fit(data_lat, kelas_latih,'Distribution', 'mn');
hasil_uji=model.predict(data_uji)
set(handles.edit18,'string',hasil_uji)
if hasil_uji ==0
    set(handles.edit15,'string','Pasien tersebut tidak
    SuspectStroke')
elseif hasil_uji ==1
    set(handles.edit15,'string','Pasien tersebut SuspectStroke')
end

```

*Lampiran 7***Matriks Data Pelatihan****Matriks Data Pelatihan dengan 80 Data**

A= [. . .

1	58	88	1	257	1	1	1	1	1	1	1	1
2	73	84	0	277	1	1	1	1	1	1	0	1
3	59	113	1	243	1	1	1	0	1	0	1	1
4	75	75	1	280	1	1	1	0	0	1	0	1
5	74	88	1	300	1	1	1	1	1	1	0	1
6	46	93	0	297	1	0	1	1	1	1	0	1
7	61	80	1	249	1	1	1	1	1	1	0	1
8	56	88	0	261	1	1	1	1	1	1	1	1
9	40	82	0	291	1	1	1	1	1	1	0	1
10	66	90	0	260	1	1	1	0	1	1	0	1
11	72	84	1	245	1	0	1	1	0	1	0	1
12	42	88	0	251	1	1	1	0	0	1	0	1
13	76	98	0	301	1	1	1	1	0	1	0	1
14	62	78	1	302	1	1	1	1	0	1	0	1
15	60	80	1	283	1	1	1	1	1	1	1	1
16	60	80	1	260	1	1	1	0	0	1	1	1
17	59	78	0	288	1	1	1	1	1	1	0	1
18	61	84	1	297	1	1	1	1	0	1	0	1
19	75	82	1	280	1	1	1	1	1	1	0	1
20	62	80	0	271	1	1	1	1	1	1	1	1
21	47	78	1	268	1	1	1	1	1	1	1	1
22	40	79	0	250	0	0	1	1	1	1	1	1
23	61	100	1	271	1	1	1	1	1	1	0	1
24	64	80	1	302	1	1	1	1	1	1	0	1
25	42	90	0	289	1	1	1	1	1	1	0	1
26	50	80	0	278	1	1	1	1	1	1	0	1
27	54	120	1	260	1	1	1	1	1	1	0	1
28	49	68	1	298	1	1	1	0	0	1	0	1
29	67	90	1	267	1	0	1	1	0	1	0	1
30	55	80	1	150	1	0	1	1	1	1	0	1
31	55	107	1	265	1	0	1	1	1	1	0	1
32	61	94	1	277	1	1	1	1	1	1	0	1
33	64	81	1	268	1	1	1	1	1	1	0	1
34	58	70	1	277	1	1	1	1	1	1	0	1
35	60	84	1	280	1	1	1	1	1	1	0	1
36	60	100	1	281	1	1	1	1	1	1	0	1
37	60	92	1	280	1	1	1	1	1	1	0	1
38	64	105	1	159	1	0	1	1	0	1	1	1
39	60	100	0	244	1	0	1	1	1	1	0	1
40	71	88	1	251	0	1	1	1	1	1	0	1
41	37	132	1	255	1	1	1	1	1	1	0	1
42	55	102	1	281	1	0	1	0	1	1	0	1
43	69	80	0	269	1	1	1	1	1	1	0	1
44	75	133	1	341	1	1	1	1	0	1	0	1
45	53	88	1	141	1	0	1	1	1	1	0	1

46	70	60	0	271	1	1	1	1	1	1	0	1
47	70	113	0	270	1	1	1	0	0	1	0	1
48	54	87	0	278	1	1	1	1	1	1	0	1
49	58	64	0	298	1	1	1	0	1	1	0	1
50	80	108	1	305	1	1	1	1	1	1	1	1
51	76	90	1	311	0	1	1	1	1	1	0	1
52	40	76	1	345	1	1	1	1	1	1	0	1
53	65	71	1	313	1	1	1	1	1	1	0	1
54	55	90	1	291	1	1	1	1	1	1	0	1
55	56	74	0	304	1	1	1	1	1	1	0	1
56	47	83	1	278	1	1	1	1	1	1	0	1
57	80	84	0	250	1	1	1	1	0	1	0	1
58	67	80	1	257	0	1	1	1	1	1	0	1
59	55	100	1	281	1	0	1	1	1	1	0	1
60	66	70	0	245	1	1	1	1	0	1	0	1
61	47	80	1	310	1	1	1	1	1	1	0	1
62	68	84	1	256	1	1	1	1	1	1	0	1
63	65	115	1	267	1	1	1	1	1	1	0	1
64	54	80	1	312	1	0	1	1	1	1	0	1
65	55	89	0	300	1	1	1	1	1	1	0	1
66	55	100	0	268	1	1	1	1	1	1	1	1
67	52	84	1	270	1	1	1	1	1	1	0	1
68	73	84	1	250	1	0	1	1	1	0	1	1
69	60	72	1	251	0	1	1	1	1	1	0	1
70	45	145	1	253	1	0	1	0	0	1	0	1
71	61	88	0	162	1	0	1	0	0	0	0	0
72	56	94	0	138	1	0	1	1	1	1	0	0
73	70	80	1	200	1	0	1	0	0	0	0	0
74	50	50	1	161	1	0	1	1	0	0	0	0
75	57	96	0	158	1	0	1	1	1	1	0	0
76	48	80	1	155	1	0	1	0	0	0	0	0
77	22	84	1	148	1	0	1	1	1	0	0	0
78	55	80	1	138	1	0	1	1	1	0	0	0
79	56	85	1	150	1	0	1	0	1	0	0	0
80	61	80	1	148	0	1	1	1	1	0	0	0]

Matriks Data Pelatihan dengan 70 Data

A= [. . .

1	58	88	1	257	1	1	1	1	1	1	1	1
2	73	84	0	277	1	1	1	1	1	1	0	1
3	59	113	1	243	1	1	1	0	1	0	1	1
4	75	75	1	280	1	1	1	0	0	1	0	1
5	74	88	1	300	1	1	1	1	1	1	0	1
6	46	93	0	297	1	0	1	1	1	1	0	1
7	61	80	1	249	1	1	1	1	1	1	0	1
8	56	88	0	261	1	1	1	1	1	1	1	1
9	40	82	0	291	1	1	1	1	1	1	0	1
10	66	90	0	260	1	1	1	0	1	1	0	1
11	72	84	1	245	1	0	1	1	0	1	0	1
12	42	88	0	251	1	1	1	0	0	1	0	1
13	76	98	0	301	1	1	1	1	0	1	0	1

14	62	78	1	302	1	1	1	1	0	1	0	1
15	60	80	1	283	1	1	1	1	1	1	1	1
16	60	80	1	260	1	1	1	0	0	1	1	1
17	59	78	0	288	1	1	1	1	1	1	0	1
18	61	84	1	297	1	1	1	1	0	1	0	1
19	75	82	1	280	1	1	1	1	1	1	0	1
20	62	80	0	271	1	1	1	1	1	1	1	1
21	47	78	1	268	1	1	1	1	1	1	1	1
22	40	79	0	250	0	0	1	1	1	1	1	1
23	61	100	1	271	1	1	1	1	1	1	0	1
24	64	80	1	302	1	1	1	1	1	1	0	1
25	42	90	0	289	1	1	1	1	1	1	0	1
26	50	80	0	278	1	1	1	1	1	1	0	1
27	54	120	1	260	1	1	1	1	1	1	0	1
28	49	68	1	298	1	1	1	0	0	1	0	1
29	67	90	1	267	1	0	1	1	0	1	0	1
30	55	80	1	150	1	0	1	1	1	1	0	1
31	55	107	1	265	1	0	1	1	1	1	0	1
32	61	94	1	277	1	1	1	1	1	1	0	1
33	64	81	1	268	1	1	1	1	1	1	0	1
34	58	70	1	277	1	1	1	1	1	1	0	1
35	60	84	1	280	1	1	1	1	1	1	0	1
36	60	100	1	281	1	1	1	1	1	1	0	1
37	60	92	1	280	1	1	1	1	1	1	0	1
38	64	105	1	159	1	0	1	1	0	1	1	1
39	60	100	0	244	1	0	1	1	1	1	0	1
40	71	88	1	251	0	1	1	1	1	1	0	1
41	37	132	1	255	1	1	1	1	1	1	0	1
42	55	102	1	281	1	0	1	0	1	1	0	1
43	69	80	0	269	1	1	1	1	1	1	0	1
44	75	133	1	341	1	1	1	1	0	1	0	1
45	53	88	1	141	1	0	1	1	1	1	0	1
46	70	60	0	271	1	1	1	1	1	1	0	1
47	70	113	0	270	1	1	1	0	0	1	0	1
48	54	87	0	278	1	1	1	1	1	1	0	1
49	58	64	0	298	1	1	1	0	1	1	0	1
50	80	108	1	305	1	1	1	1	1	1	1	1
51	76	90	1	311	0	1	1	1	1	1	0	1
52	40	76	1	345	1	1	1	1	1	1	0	1
53	65	71	1	313	1	1	1	1	1	1	0	1
54	55	90	1	291	1	1	1	1	1	1	0	1
55	56	74	0	304	1	1	1	1	1	1	0	1
56	47	83	1	278	1	1	1	1	1	1	0	1
57	80	84	0	250	1	1	1	1	0	1	0	1
58	67	80	1	257	0	1	1	1	1	1	0	1
59	55	100	1	281	1	0	1	1	1	1	0	1
60	66	70	0	245	1	1	1	1	0	1	0	1
61	61	88	0	162	1	0	1	0	0	0	0	0
62	56	94	0	138	1	0	1	1	1	1	0	0
63	70	80	1	200	1	0	1	0	0	0	0	0
64	50	50	1	161	1	0	1	1	0	0	0	0
65	57	96	0	158	1	0	1	1	1	1	0	0
66	48	80	1	155	1	0	1	0	0	0	0	0
67	22	84	1	148	1	0	1	1	1	0	0	0
68	55	80	1	138	1	0	1	1	1	0	0	0
69	56	85	1	150	1	0	1	0	1	0	0	0

70 61 80 1 148 0 1 1 1 1 0 0 0]

Matriks Data Pelatihan dengan 60 Data

```

A= [ . . .
1   58  88  1   257 1   1   1   1   1   1   1   1
2   73  84  0   277 1   1   1   1   1   1   0   1
3   59 113  1   243 1   1   1   0   1   0   1   1
4   75  75  1   280 1   1   1   0   0   1   0   1
5   74  88  1   300 1   1   1   1   1   1   0   1
6   46  93  0   297 1   0   1   1   1   1   0   1
7   61  80  1   249 1   1   1   1   1   1   0   1
8   56  88  0   261 1   1   1   1   1   1   1   1
9   40  82  0   291 1   1   1   1   1   1   0   1
10  66  90  0   260 1   1   1   0   1   1   0   1
11  72  84  1   245 1   0   1   1   0   1   0   1
12  42  88  0   251 1   1   1   0   0   1   0   1
13  76  98  0   301 1   1   1   1   0   1   0   1
14  62  78  1   302 1   1   1   1   0   1   0   1
15  60  80  1   283 1   1   1   1   1   1   1   1
16  60  80  1   260 1   1   1   0   0   1   1   1
17  59  78  0   288 1   1   1   1   1   1   0   1
18  61  84  1   297 1   1   1   1   0   1   0   1
19  75  82  1   280 1   1   1   1   1   1   0   1
20  62  80  0   271 1   1   1   1   1   1   1   1
21  47  78  1   268 1   1   1   1   1   1   1   1
22  40  79  0   250 0   0   1   1   1   1   1   1
23  61 100  1   271 1   1   1   1   1   1   0   1
24  64  80  1   302 1   1   1   1   1   1   0   1
25  42  90  0   289 1   1   1   1   1   1   0   1
26  50  80  0   278 1   1   1   1   1   1   0   1
27  54 120  1   260 1   1   1   1   1   1   0   1
28  49  68  1   298 1   1   1   0   0   1   0   1
29  67  90  1   267 1   0   1   1   0   1   0   1
30  55  80  1   150 1   0   1   1   1   1   0   1
31  55 107  1   265 1   0   1   1   1   1   0   1
32  61  94  1   277 1   1   1   1   1   1   0   1
33  64  81  1   268 1   1   1   1   1   1   0   1
34  58  70  1   277 1   1   1   1   1   1   0   1
35  60  84  1   280 1   1   1   1   1   1   0   1
36  60 100  1   281 1   1   1   1   1   1   0   1
37  60  92  1   280 1   1   1   1   1   1   0   1
38  64 105  1   159 1   0   1   1   0   1   1   1
39  60 100  0   244 1   0   1   1   1   1   0   1
40  71  88  1   251 0   1   1   1   1   1   0   1
41  37 132  1   255 1   1   1   1   1   1   0   1
42  55 102  1   281 1   0   1   0   1   1   0   1
43  69  80  0   269 1   1   1   1   1   1   0   1
44  75 133  1   341 1   1   1   1   0   1   0   1
45  53  88  1   141 1   0   1   1   1   1   0   1
46  70  60  0   271 1   1   1   1   1   1   0   1
47  70 113  0   270 1   1   1   0   0   1   0   1
48  54  87  0   278 1   1   1   1   1   1   0   1
49  58  64  0   298 1   1   1   0   1   1   0   1

```

50	80	108	1	305	1	1	1	1	1	1	1	1
51	61	88	0	162	1	0	1	0	0	0	0	0
52	56	94	0	138	1	0	1	1	1	1	0	0
53	70	80	1	200	1	0	1	0	0	0	0	0
54	50	50	1	161	1	0	1	1	0	0	0	0
55	57	96	0	158	1	0	1	1	1	1	0	0
56	48	80	1	155	1	0	1	0	0	0	0	0
57	22	84	1	148	1	0	1	1	1	0	0	0
58	55	80	1	138	1	0	1	1	1	0	0	0
59	56	85	1	150	1	0	1	0	1	0	0	0
60	61	80	1	148	0	1	1	1	1	0	0	0]

1) Variasi 1

Matriks Data Pengujian dengan 20 Data

U=

1	77	112	1	300	1	0	0	1	1	1	1	0
2	69	81	0	200	1	1	1	1	0	1	0	1
3	67	75	0	244	1	0	1	1	0	1	0	1
4	42	90	1	311	1	1	1	1	1	1	0	1
5	64	94	0	180	0	0	1	1	1	0	0	0
6	69	84	1	290	1	1	1	1	1	1	0	1
7	46	92	1	291	1	1	1	1	0	1	0	1
8	47	83	0	278	1	0	1	1	0	1	0	1
9	68	88	0	168	1	0	1	0	1	1	0	1
10	75	75	0	175	1	1	0	0	1	0	0	0
11	45	100	1	158	1	0	1	1	1	0	1	0
12	40	92	1	155	0	0	0	1	1	0	0	0
13	46	82	0	155	1	0	1	0	0	1	0	0
14	55	79	0	155	1	0	1	0	0	0	0	0
15	54	135	0	148	0	0	0	1	1	0	0	0
16	76	100	0	145	0	0	0	1	1	0	0	0
17	54	80	0	150	0	1	1	0	1	1	0	0
18	54	90	0	167	1	1	1	1	1	0	0	0
19	57	88	0	200	1	1	0	1	1	1	0	0
20	44	88	1	270	1	1	1	1	0	1	0	0

2) Variasi 2

Matriks Data Pengujian dengan 30 Data

U=	[
	1	56	80	1	155	1	0	1	1	1	1	0	1
	2	63	80	1	260	1	1	1	1	1	1	1	1
	3	50	80	0	158	1	0	1	0	0	0	0	0
	4	55	84	0	148	1	0	1	1	1	0	0	0
	5	60	110	1	270	1	1	1	1	1	1	0	1
	6	70	80	1	210	1	0	1	0	0	0	0	0
	7	62	88	0	300	0	1	1	1	1	1	0	1
	8	45	76	0	250	1	0	1	1	1	0	1	1
	9	46	90	0	270	1	1	1	1	0	1	0	1
	10	65	75	0	245	1	0	1	1	0	1	0	1
	11	77	112	1	300	1	0	0	1	1	1	1	0
	12	69	81	0	200	1	1	1	1	0	1	0	1
	13	67	75	0	244	1	0	1	1	0	1	0	1
	14	42	90	1	311	1	1	1	1	1	1	0	1
	15	64	94	0	180	0	0	1	1	1	0	0	0
	16	69	84	1	290	1	1	1	1	1	1	0	1
	17	46	92	1	291	1	1	1	1	0	1	0	1
	18	47	83	0	278	1	0	1	1	0	1	0	1
	19	68	88	0	168	1	0	1	0	1	1	0	1
	20	75	75	0	175	1	1	0	0	1	0	0	0
	21	45	100	1	158	1	0	1	1	1	0	1	0

22	40	92	1	155	0	0	0	1	1	0	0	0
23	46	82	0	155	1	0	1	0	0	1	0	0
24	55	79	0	155	1	0	1	0	0	0	0	0
25	54	135	0	148	0	0	0	1	1	0	0	0
26	76	100	0	145	0	0	0	1	1	0	0	0
27	54	80	0	150	0	1	1	0	1	1	0	0
28	54	90	0	167	1	1	1	1	1	0	0	0
29	57	88	0	200	1	1	0	1	1	1	0	0
30	44	88	1	270	1	1	1	1	0	1	0	0]

3) Variasi 3

Matriks Data Pengujian dengan 40 Data

U= [

1	78	110	1	300	1	0	0	1	1	1	1	0
2	67	80	0	200	1	1	1	1	0	1	0	1
3	70	80	0	170	1	1	0	0	1	0	0	0
4	50	80	0	150	0	1	1	0	1	1	0	0
5	45	100	1	150	1	0	1	1	1	0	1	0
6	65	88	1	310	1	1	1	1	1	1	0	1
7	75	84	1	250	1	0	1	1	1	0	1	1
8	45	80	1	270	1	1	1	1	1	1	0	1
9	76	89	1	300	0	1	1	1	1	1	0	1
10	55	90	1	280	1	0	1	1	1	1	0	1
11	56	80	1	155	1	0	1	1	1	1	0	1
12	63	80	1	260	1	1	1	1	1	1	1	1
13	50	80	0	158	1	0	1	0	0	0	0	0
14	55	84	0	148	1	0	1	1	1	0	0	0
15	60	110	1	270	1	1	1	1	1	1	0	1
16	70	80	1	210	1	0	1	0	0	0	0	0
17	62	88	0	300	0	1	1	1	1	1	0	1
18	45	76	0	250	1	0	1	1	1	0	1	1
19	46	90	0	270	1	1	1	1	0	1	0	1
20	65	75	0	245	1	0	1	1	0	1	0	1
21	77	112	1	300	1	0	0	1	1	1	1	0
22	69	81	0	200	1	1	1	1	0	1	0	1
23	67	75	0	244	1	0	1	1	0	1	0	1
24	42	90	1	311	1	1	1	1	1	1	0	1
25	64	94	0	180	0	0	1	1	1	0	0	0
26	69	84	1	290	1	1	1	1	1	1	0	1
27	46	92	1	291	1	1	1	1	0	1	0	1
28	47	83	0	278	1	0	1	1	0	1	0	1
29	68	88	0	168	1	0	1	0	1	1	0	1
30	75	75	0	175	1	1	0	0	1	0	0	0
31	45	100	1	158	1	0	1	1	1	0	1	0
32	40	92	1	155	0	0	0	1	1	0	0	0
33	46	82	0	155	1	0	1	0	0	1	0	0
34	55	79	0	155	1	0	1	0	0	0	0	0
35	54	135	0	148	0	0	0	1	1	0	0	0
36	76	100	0	145	0	0	0	1	1	0	0	0
37	54	80	0	150	0	1	1	0	1	1	0	0
38	54	90	0	167	1	1	1	1	1	0	0	0

39 57 88 0 200 1 1 0 1 1 1 0 0
 40 44 88 1 270 1 1 1 1 0 1 0 0]

Lampiran 8

Tabel Pelatihan Pasien

Data Pelatihan dengan 70 Data Pasien

Pasien	Variabel gejala penyakit Stroke											Target
ke-	X ₁	X ₂	X ₃	X ₄	X ₅	X ₆	X ₇	X ₈	X ₉	X ₁₀	X ₁₁	
1	58	88	1	257	1	1	1	1	1	1	1	1
2	73	84	0	277	1	1	1	1	1	1	0	1
3	59	113	1	243	1	1	1	0	1	0	1	1
4	75	75	1	280	1	1	1	0	0	1	0	1
5	74	88	1	300	1	1	1	1	1	1	0	1
6	46	93	0	297	1	0	1	1	1	1	0	1
7	61	80	1	249	1	1	1	1	1	1	0	1
8	56	88	0	261	1	1	1	1	1	1	1	1
9	40	82	0	291	1	1	1	1	1	1	0	1
10	66	90	0	260	1	1	1	0	1	1	0	1
11	72	84	1	245	1	0	1	1	0	1	0	1
12	42	88	0	251	1	1	1	0	0	1	0	1
13	76	98	0	301	1	1	1	1	0	1	0	1
14	62	78	1	302	1	1	1	1	0	1	0	1
15	60	80	1	283	1	1	1	1	1	1	1	1
16	60	80	1	260	1	1	1	0	0	1	1	1
17	59	78	0	288	1	1	1	1	1	1	0	1
18	61	84	1	297	1	1	1	1	0	1	0	1
19	75	82	1	280	1	1	1	1	1	1	0	1
20	62	80	0	271	1	1	1	1	1	1	1	1
21	47	78	1	268	1	1	1	1	1	1	1	1
22	40	79	0	250	0	0	1	1	1	1	1	1
23	61	100	1	271	1	1	1	1	1	1	0	1
24	64	80	1	302	1	1	1	1	1	1	0	1
25	42	90	0	289	1	1	1	1	1	1	0	1
26	50	80	0	278	1	1	1	1	1	1	0	1
27	54	120	1	260	1	1	1	1	1	1	0	1
28	49	68	1	298	1	1	1	0	0	1	0	1
29	67	90	1	267	1	0	1	1	0	1	0	1
30	55	80	1	150	1	0	1	1	1	1	0	1
31	55	107	1	265	1	0	1	1	1	1	0	1
32	61	94	1	277	1	1	1	1	1	1	0	1
33	64	81	1	268	1	1	1	1	1	1	0	1
34	58	70	1	277	1	1	1	1	1	1	0	1
35	60	84	1	280	1	1	1	1	1	1	0	1
36	60	100	1	281	1	1	1	1	1	1	0	1
37	60	92	1	280	1	1	1	1	1	1	0	1

38	64	105	1	159	1	0	1	1	0	1	1	1
39	60	100	0	244	1	0	1	1	1	1	0	1
40	71	88	1	251	0	1	1	1	1	1	0	1
41	37	132	1	255	1	1	1	1	1	1	0	1
42	55	102	1	281	1	0	1	0	1	1	0	1
43	69	80	0	269	1	1	1	1	1	1	0	1
44	75	133	1	341	1	1	1	1	0	1	0	1
45	53	88	1	141	1	0	1	1	1	1	0	1
46	70	60	0	271	1	1	1	1	1	1	0	1
47	70	113	0	270	1	1	1	0	0	1	0	1
48	54	87	0	278	1	1	1	1	1	1	0	1
49	58	64	0	298	1	1	1	0	1	1	0	1
50	80	108	1	305	1	1	1	1	1	1	1	1
51	76	90	1	311	0	1	1	1	1	1	0	1
52	40	76	1	345	1	1	1	1	1	1	0	1
53	65	71	1	313	1	1	1	1	1	1	0	1
54	55	90	1	291	1	1	1	1	1	1	0	1
55	56	74	0	304	1	1	1	1	1	1	0	1
56	47	83	1	278	1	1	1	1	1	1	0	1
57	80	84	0	250	1	1	1	1	0	1	0	1
58	67	80	1	257	0	1	1	1	1	1	0	1
59	55	100	1	281	1	0	1	1	1	1	0	1
60	66	70	0	245	1	1	1	1	0	1	0	1
61	47	80	1	310	1	1	1	1	1	1	0	1
62	68	84	1	256	1	1	1	1	1	1	0	1
63	65	115	1	267	1	1	1	1	1	1	0	1
64	54	80	1	312	1	0	1	1	1	1	0	1
65	55	89	0	300	1	1	1	1	1	1	0	1
66	55	100	0	268	1	1	1	1	1	1	1	1
67	52	84	1	270	1	1	1	1	1	1	0	1
68	73	84	1	250	1	0	1	1	1	0	1	1
69	60	72	1	251	0	1	1	1	1	1	0	1
70	45	145	1	253	1	0	1	0	0	1	0	1

Data Pelatihan dengan 60 Data Pasien

Pasien ke-	Variabel gejala penyakit Stroke											Target
	X ₁	X ₂	X ₃	X ₄	X ₅	X ₆	X ₇	X ₈	X ₉	X ₁₀	X ₁₁	
1	58	88	1	257	1	1	1	1	1	1	1	1
2	73	84	0	277	1	1	1	1	1	1	0	1
3	59	113	1	243	1	1	1	0	1	0	1	1
4	75	75	1	280	1	1	1	0	0	1	0	1
5	74	88	1	300	1	1	1	1	1	1	0	1
6	46	93	0	297	1	0	1	1	1	1	0	1
7	61	80	1	249	1	1	1	1	1	1	0	1
8	56	88	0	261	1	1	1	1	1	1	1	1
9	40	82	0	291	1	1	1	1	1	1	0	1
10	66	90	0	260	1	1	1	0	1	1	0	1
11	72	84	1	245	1	0	1	1	0	1	0	1
12	42	88	0	251	1	1	1	0	0	1	0	1
13	76	98	0	301	1	1	1	1	0	1	0	1
14	62	78	1	302	1	1	1	1	0	1	0	1
15	60	80	1	283	1	1	1	1	1	1	1	1
16	60	80	1	260	1	1	1	0	0	1	1	1
17	59	78	0	288	1	1	1	1	1	1	0	1
18	61	84	1	297	1	1	1	1	0	1	0	1
19	75	82	1	280	1	1	1	1	1	1	0	1
20	62	80	0	271	1	1	1	1	1	1	1	1
21	47	78	1	268	1	1	1	1	1	1	1	1
22	40	79	0	250	0	0	1	1	1	1	1	1
23	61	100	1	271	1	1	1	1	1	1	0	1
24	64	80	1	302	1	1	1	1	1	1	0	1
25	42	90	0	289	1	1	1	1	1	1	0	1
26	50	80	0	278	1	1	1	1	1	1	0	1
27	54	120	1	260	1	1	1	1	1	1	0	1
28	49	68	1	298	1	1	1	0	0	1	0	1
29	67	90	1	267	1	0	1	1	0	1	0	1
30	55	80	1	150	1	0	1	1	1	1	0	1
31	55	107	1	265	1	0	1	1	1	1	0	1
32	61	94	1	277	1	1	1	1	1	1	0	1
33	64	81	1	268	1	1	1	1	1	1	0	1
34	58	70	1	277	1	1	1	1	1	1	0	1
35	60	84	1	280	1	1	1	1	1	1	0	1
36	60	100	1	281	1	1	1	1	1	1	0	1
37	60	92	1	280	1	1	1	1	1	1	0	1
38	64	105	1	159	1	0	1	1	0	1	1	1
39	60	100	0	244	1	0	1	1	1	1	0	1
40	71	88	1	251	0	1	1	1	1	1	0	1

41	37	132	1	255	1	1	1	1	1	1	0	1
42	55	102	1	281	1	0	1	0	1	1	0	1
43	69	80	0	269	1	1	1	1	1	1	0	1
44	75	133	1	341	1	1	1	1	0	1	0	1
45	53	88	1	141	1	0	1	1	1	1	0	1
46	70	60	0	271	1	1	1	1	1	1	0	1
47	70	113	0	270	1	1	1	0	0	1	0	1
48	54	87	0	278	1	1	1	1	1	1	0	1
49	58	64	0	298	1	1	1	0	1	1	0	1
50	80	108	1	305	1	1	1	1	1	1	1	1
51	76	90	1	311	0	1	1	1	1	1	0	1
52	40	76	1	345	1	1	1	1	1	1	0	1
53	65	71	1	313	1	1	1	1	1	1	0	1
54	55	90	1	291	1	1	1	1	1	1	0	1
55	56	74	0	304	1	1	1	1	1	1	0	1
56	47	83	1	278	1	1	1	1	1	1	0	1
57	80	84	0	250	1	1	1	1	0	1	0	1
58	67	80	1	257	0	1	1	1	1	1	0	1
59	55	100	1	281	1	0	1	1	1	1	0	1
60	66	70	0	245	1	1	1	1	0	1	0	1

Lampiran 9

Tabel Pengujian Pasien

1) Variasi 2

Tabel Contoh Data Pengujian Pasien dengan 30 Data

Pasien	Variabel gejala penyakit Stroke											Target
ke-	X ₁	X ₂	X ₃	X ₄	X ₅	X ₆	X ₇	X ₈	X ₉	X ₁₀	X ₁₁	
1	56	80	1	155	1	0	1	1	1	1	0	1
2	63	80	1	260	1	1	1	1	1	1	1	1
3	50	80	0	158	1	0	1	0	0	0	0	0
4	55	84	0	148	1	0	1	1	1	0	0	0
5	60	110	1	270	1	1	1	1	1	1	0	1
6	70	80	1	210	1	0	1	0	0	0	0	0
7	62	88	0	300	0	1	1	1	1	1	0	1
8	45	76	0	250	1	0	1	1	1	0	1	1
9	46	90	0	270	1	1	1	1	0	1	0	1
10	65	75	0	245	1	0	1	1	0	1	0	1
11	77	112	1	300	1	0	0	1	1	1	1	0
12	69	81	0	200	1	1	1	1	0	1	0	1
13	67	75	0	244	1	0	1	1	0	1	0	1
14	42	90	1	311	1	1	1	1	1	1	0	1
15	64	94	0	180	0	0	1	1	1	0	0	0
16	69	84	1	290	1	1	1	1	1	1	0	1
17	46	92	1	291	1	1	1	1	0	1	0	1
18	47	83	0	278	1	0	1	1	0	1	0	1
19	68	88	0	168	1	0	1	0	1	1	0	1
20	75	75	0	175	1	1	0	0	1	0	0	0
21	45	100	1	158	1	0	1	1	1	0	1	0
22	40	92	1	155	0	0	0	1	1	0	0	0
23	46	82	0	155	1	0	1	0	0	1	0	0
24	55	79	0	155	1	0	1	0	0	0	0	0
25	54	135	0	148	0	0	0	1	1	0	0	0
26	76	100	0	145	0	0	0	1	1	0	0	0
27	54	80	0	150	0	1	1	0	1	1	0	0
28	54	90	0	167	1	1	1	1	1	0	0	0
29	57	88	0	200	1	1	0	1	1	1	0	0
30	44	88	1	270	1	1	1	1	0	1	0	0

2) Variasi 3

Tabel Contoh Data Pengujian Pasien dengan 40 Data

Pasien ke-	Variabel gejala penyakit Stroke											Target
	X ₁	X ₂	X ₃	X ₄	X ₅	X ₆	X ₇	X ₈	X ₉	X ₁₀	X ₁₁	
1	78	110	1	300	1	0	0	1	1	1	1	1
2	67	80	0	200	1	1	1	1	0	1	0	1
3	70	80	0	170	1	1	0	0	1	0	0	0
4	50	80	0	150	0	1	1	0	1	1	0	0
5	45	100	1	150	1	0	1	1	1	0	1	0
6	65	88	1	310	1	1	1	1	1	1	0	1
7	75	84	1	250	1	0	1	1	1	0	1	1
8	45	80	1	270	1	1	1	1	1	1	0	1
9	76	89	1	300	0	1	1	1	1	1	0	1
10	55	90	1	280	1	0	1	1	1	1	0	1
11	56	80	1	155	1	0	1	1	1	1	0	1
12	63	80	1	260	1	1	1	1	1	1	1	1
13	50	80	0	158	1	0	1	0	0	0	0	0
14	55	84	0	148	1	0	1	1	1	0	0	0
15	60	110	1	270	1	1	1	1	1	1	0	1
16	70	80	1	210	1	0	1	0	0	0	0	0
17	62	88	0	300	0	1	1	1	1	1	0	1
18	45	76	0	250	1	0	1	1	1	0	1	1
19	46	90	0	270	1	1	1	1	0	1	0	1
20	65	75	0	245	1	0	1	1	0	1	0	1
21	77	112	1	300	1	0	0	1	1	1	1	0
22	69	81	0	200	1	1	1	1	0	1	0	1
23	67	75	0	244	1	0	1	1	0	1	0	1
24	42	90	1	311	1	1	1	1	1	1	0	1
25	64	94	0	180	0	0	1	1	1	0	0	0
26	69	84	1	290	1	1	1	1	1	1	0	1
27	46	92	1	291	1	1	1	1	0	1	0	1
28	47	83	0	278	1	0	1	1	0	1	0	1
29	68	88	0	168	1	0	1	0	1	1	0	1
30	75	75	0	175	1	1	0	0	1	0	0	0
31	45	100	1	158	1	0	1	1	1	0	1	0
32	40	92	1	155	0	0	0	1	1	0	0	0
33	46	82	0	155	1	0	1	0	0	1	0	0
34	55	79	0	155	1	0	1	0	0	0	0	0
35	54	135	0	148	0	0	0	1	1	0	0	0
36	76	100	0	145	0	0	0	1	1	0	0	0
37	54	80	0	150	0	1	1	0	1	1	0	0
38	54	90	0	167	1	1	1	1	1	0	0	0
39	57	88	0	200	1	1	0	1	1	1	0	0

40	44	88	1	270	1	1	1	1	0	1	0	0
----	----	----	---	-----	---	---	---	---	---	---	---	---

*Lampiran 10***Hasil Normalisasi Data Numerik Pada Data Pelatihan**

1) Variasi 1

Hasil Normalisasi Data Numerik Pada Data 80 Pelatihan

No.	Umur		Nadi		Kolesterol	
1	58	0,62069	88	0,4	257	0,574879
2	73	0,87931	84	0,357895	277	0,671498
3	59	0,637931	113	0,663158	243	0,507246
4	75	0,913793	75	0,263158	280	0,68599
5	74	0,896552	88	0,4	300	0,782609
6	46	0,413793	93	0,452632	297	0,768116
7	61	0,672414	80	0,315789	249	0,536232
8	56	0,586207	88	0,4	261	0,594203
9	40	0,310345	82	0,336842	291	0,73913
10	66	0,758621	90	0,421053	260	0,589372
11	72	0,862069	84	0,357895	245	0,516908
12	42	0,344828	88	0,4	251	0,545894
13	76	0,931034	98	3	301	0,78744
14	62	0,689655	78	0,263158	302	0,792271
15	60	0,655172	80	0,315789	283	0,700483
16	60	0,655172	80	0,315789	260	0,589372
17	59	0,637931	78	0,294737	288	0,724638
18	61	0,672414	84	0,357895	297	0,768116
19	75	0,913793	82	0,336842	280	0,68599
20	62	0,689655	80	0,315789	271	0,642512
21	47	0,431034	78	0,294737	268	0,628019
22	40	0,310345	79	0,305263	250	0,541063
23	61	0,672414	100	0,526316	271	0,642512
24	64	0,724138	80	0,315789	302	0,792271
25	42	0,344828	90	0,421053	289	0,729469
26	50	0,482759	80	0,315789	278	0,676329
27	54	0,551724	120	0,736842	260	0,589372
28	49	0,465517	68	0,189474	298	0,772947
29	67	0,775862	90	0,421053	267	0,623188
30	55	0,568966	80	0,315789	150	0,057971
31	55	0,568966	107	0,6	265	0,613527
32	61	0,672414	94	0,463158	277	0,671498
33	64	0,724138	81	0,326316	268	0,628019
34	58	0,62069	70	0,210526	277	0,671498
35	60	0,655172	84	0,357895	280	0,68599
36	60	0,655172	100	0,526316	281	0,690821

37	60	0,655172	92	0,442105	280	0,68599
38	64	0,724138	105	0,578947	159	0,101449
39	60	0,655172	100	0,526316	244	0,512077
40	71	0,844828	88	0,4	251	0,545894
41	37	0,258621	132	0,863158	255	0,565217
42	55	0,568966	102	0,863158	281	0,690821
43	69	0,810345	80	0,315789	269	0,63285
44	75	0,913793	133	0,873684	341	0,980676
45	53	0,534483	88	0,4	141	0,014493
46	70	0,827586	60	0,105263	271	0,642512
47	70	0,827586	113	0,663158	270	0,637681
48	54	0,551724	87	0,389474	278	0,676329
49	58	0,62069	64	0,147368	298	0,772947
50	80	1	108	0,610526	305	0,806763
51	76	0,931034	90	0,421053	311	0,835749
52	40	0,310345	76	0,273684	345	1
53	65	0,741379	71	0,221053	313	0,845411
54	55	0,568966	90	0,421053	291	0,73913
55	56	0,586207	74	0,252632	304	0,801932
56	47	0,431034	83	0,347368	278	0,676329
57	80	1	84	0,357895	250	0,541063
58	67	0,775862	80	0,315789	257	0,574879
59	55	0,568966	100	0,526316	281	0,690821
60	66	0,758621	70	0,210526	245	0,516908
61	47	0,431034	80	0,315789	310	0,830918
62	68	0,793103	84	0,357895	256	0,570048
63	65	0,741379	115	0,684211	267	0,623188
64	54	0,551724	80	0,315789	312	0,84058
65	55	0,568966	89	0,410526	300	0,782609
66	55	0,568966	100	0,526316	268	0,628019
67	52	0,517241	84	0,357895	270	0,637681
68	73	0,87931	84	0,357895	250	0,541063
69	60	0,655172	72	0,231579	251	0,545894
70	45	0,396552	145	1	253	0,555556
71	61	0,672414	88	0,4	162	0,115942
72	56	0,586207	94	0,463158	138	0
73	70	0,827586	80	0,315789	200	0,299517
74	50	0,482759	50	0	161	0,111111
75	57	0,603448	96	0,484211	158	0,096618
76	48	0,448276	80	0,315789	155	0,082126
77	22	0	84	0,357895	148	0,048309
78	55	0,568966	80	0,315789	138	0
79	56	0,586207	85	0,368421	150	0,057971
80	61	0,672414	80	0,315789	148	0,048309

Hasil Normalisasi Data Numerik Pada Data 70 Pelatihan

No.	Umur		Nadi		Kolesterol	
1	58	0,62069	88	0,457831	257	0,574879
2	73	0,87931	84	0,409639	277	0,671498
3	59	0,637931	113	0,759036	243	0,507246
4	75	0,913793	75	0,301205	280	0,68599
5	74	0,896552	88	0,457831	300	0,782609
6	46	0,413793	93	0,518072	297	0,768116
7	61	0,672414	80	0,361446	249	0,536232
8	56	0,586207	88	0,457831	261	0,594203
9	40	0,310345	82	0,385542	291	0,73913
10	66	0,758621	90	0,481928	260	0,589372
11	72	0,862069	84	0,409639	245	0,516908
12	42	0,344828	88	0,457831	251	0,545894
13	76	0,931034	98	0,578313	301	0,78744
14	62	0,689655	78	0,337349	302	0,792271
15	60	0,655172	80	0,361446	283	0,700483
16	60	0,655172	80	0,361446	260	0,589372
17	59	0,637931	78	0,337349	288	0,724638
18	61	0,672414	84	0,409639	297	0,768116
19	75	0,913793	82	0,385542	280	0,68599
20	62	0,689655	80	0,361446	271	0,642512
21	47	0,431034	78	0,337349	268	0,628019
22	40	0,310345	79	0,349398	250	0,541063
23	61	0,672414	100	0,60241	271	0,642512
24	64	0,724138	80	0,361446	302	0,792271
25	42	0,344828	90	0,481928	289	0,729469
26	50	0,482759	80	0,361446	278	0,676329
27	54	0,551724	120	0,843373	260	0,589372
28	49	0,465517	68	0,216867	298	0,772947
29	67	0,775862	90	0,481928	267	0,623188
30	55	0,568966	80	0,361446	150	0,057971
31	55	0,568966	107	0,686747	265	0,613527
32	61	0,672414	94	0,53012	277	0,671498
33	64	0,724138	81	0,373494	268	0,628019
34	58	0,62069	70	0,240964	277	0,671498
35	60	0,655172	84	0,409639	280	0,68599
36	60	0,655172	100	0,60241	281	0,690821
37	60	0,655172	92	0,506024	280	0,68599
38	64	0,724138	105	0,662651	159	0,101449
39	60	0,655172	100	0,60241	244	0,512077
40	71	0,844828	88	0,457831	251	0,545894
41	37	0,258621	132	0,987952	255	0,565217

42	55	0,568966	102	0,626506	281	0,690821
43	69	0,810345	80	0,361446	269	0,63285
44	75	0,913793	133	1	341	0,980676
45	53	0,534483	88	0,457831	141	0,014493
46	70	0,827586	60	0,120482	271	0,642512
47	70	0,827586	113	0,759036	270	0,637681
48	54	0,551724	87	0,445783	278	0,676329
49	58	0,62069	64	0,168675	298	0,772947
50	80	1	108	0,698795	305	0,806763
51	76	0,931034	90	0,481928	311	0,835749
52	40	0,310345	76	0,313253	345	1
53	65	0,741379	71	0,253012	313	0,845411
54	55	0,568966	90	0,481928	291	0,73913
55	56	0,586207	74	0,289157	304	0,801932
56	47	0,431034	83	0,39759	278	0,676329
57	80	1	84	0,409639	250	0,541063
58	67	0,775862	80	0,361446	257	0,574879
59	55	0,568966	100	0,60241	281	0,690821
60	66	0,758621	70	0,240964	245	0,516908
61	61	0,672414	88	0,457831	162	0,115942
62	56	0,586207	94	0,53012	138	0
63	70	0,827586	80	0,361446	200	0,299517
64	50	0,482759	50	0	161	0,111111
65	57	0,603448	96	0,554217	158	0,096618
66	48	0,448276	80	0,361446	155	0,082126
67	22	0	84	0,409639	148	0,048309
68	55	0,568966	80	0,361446	138	0
69	56	0,586207	85	0,421687	150	0,057971
70	61	0,672414	80	0,361446	148	0,048309

Hasil Normalisasi Data Numerik Pada Data 60 Pelatihan

No.	Umur		Nadi		Kolesterol	
1	58	0,62069	88	0,457831	257	0,586207
2	73	0,87931	84	0,409639	277	0,684729
3	59	0,637931	113	0,759036	243	0,517241
4	75	0,913793	75	0,301205	280	0,699507
5	74	0,896552	88	0,457831	300	0,79803
6	46	0,413793	93	0,518072	297	0,783251
7	61	0,672414	80	0,361446	249	0,546798
8	56	0,586207	88	0,457831	261	0,605911
9	40	0,310345	82	0,385542	291	0,753695
10	66	0,758621	90	0,481928	260	0,600985
11	72	0,862069	84	0,409639	245	0,527094
12	42	0,344828	88	0,457831	251	0,55665
13	76	0,931034	98	0,578313	301	0,802956
14	62	0,689655	78	0,337349	302	0,807882
15	60	0,655172	80	0,361446	283	0,714286
16	60	0,655172	80	0,361446	260	0,600985
17	59	0,637931	78	0,337349	288	0,738916
18	61	0,672414	84	0,409639	297	0,783251
19	75	0,913793	82	0,385542	280	0,699507
20	62	0,689655	80	0,361446	271	0,655172
21	47	0,431034	78	0,337349	268	0,640394
22	40	0,310345	79	0,349398	250	0,551724
23	61	0,672414	100	0,60241	271	0,655172
24	64	0,724138	80	0,361446	302	0,807882
25	42	0,344828	90	0,481928	289	0,743842
26	50	0,482759	80	0,361446	278	0,689655
27	54	0,551724	120	0,843373	260	0,600985
28	49	0,465517	68	0,216867	298	0,788177
29	67	0,775862	90	0,481928	267	0,635468
30	55	0,568966	80	0,361446	150	0,059113
31	55	0,568966	107	0,686747	265	0,625616
32	61	0,672414	94	0,53012	277	0,684729
33	64	0,724138	81	0,373494	268	0,640394
34	58	0,62069	70	0,240964	277	0,684729
35	60	0,655172	84	0,409639	280	0,699507
36	60	0,655172	100	0,60241	281	0,704433
37	60	0,655172	92	0,506024	280	0,699507
38	64	0,724138	105	0,662651	159	0,103448
39	60	0,655172	100	0,60241	244	0,522167
40	71	0,844828	88	0,457831	251	0,55665
41	37	0,258621	132	0,987952	255	0,576355

42	55	0,568966	102	0,626506	281	0,704433
43	69	0,810345	80	0,361446	269	0,64532
44	75	0,913793	133	1	341	1
45	53	0,534483	88	0,457831	141	0,014778
46	70	0,827586	60	0,120482	271	0,655172
47	70	0,827586	113	0,759036	270	0,650246
48	54	0,551724	87	0,445783	278	0,689655
49	58	0,62069	64	0,168675	298	0,788177
50	80	1	108	0,698795	305	0,82266
51	61	0,672414	88	0,457831	162	0,118227
52	56	0,586207	94	0,53012	138	0
53	70	0,827586	80	0,361446	200	0,305419
54	50	0,482759	50	0	161	0,1133
55	57	0,603448	96	0,554217	158	0,098522
56	48	0,448276	80	0,361446	155	0,083744
57	22	0	84	0,409639	148	0,049261
58	55	0,568966	80	0,361446	138	0
59	56	0,586207	85	0,421687	150	0,059113
60	61	0,672414	80	0,361446	148	0,049261

Lampiran 11

Perhitungan Probabilitas Naive Bayes

1) Perhitungan Probabilitas Naive Bayes dengan 70 Data Pelatihan

Perhitungan Mean dan Standart Deviasi Pada Data Numerik

	Umur		Nadi		Kolesterol	
	Tidak <i>Suspect</i>	<i>Suspect</i>	Tidak <i>Suspect</i>	<i>Suspect</i>	Tidak <i>Suspect</i>	<i>Suspect</i>
Mean	53,6	60,11667	81,7	88,06667	155,8	271,5333
Variansi	144,64	109,6031	142,81	210,7289	281,36	1252,982
Standart Deviasi	12,02664	10,46915	11,95031	14,5165	16,77379	35,39749

Probabilitas Jenis kelamin

		tidak <i>suspect</i>	<i>suspect</i>
X ₃	0	0,3	0,35
	1	0,7	0,65

Probabilitas Hipertensi

		tidak <i>suspect</i>	<i>suspect</i>
X ₅	0	0,1	0,066667
	1	0,9	0,933333

Probabilitas Diabetes Militus

		tidak <i>suspect</i>	<i>suspect</i>
X ₆	0	0,9	0,183333
	1	0,1	0,816667

Probabilitas Kesadaran

		tidak <i>suspect</i>	<i>suspect</i>

X_7	0	0,1	0
	1	0,9	1

Probabilitas Pusing

		tidak <i>suspect</i>	<i>suspect</i>
X_8	0	0,4	0,15
	1	0,6	0,85

Probabilitas Mual

		tidak <i>suspect</i>	<i>suspect</i>
X_9	0	0,4	0,233333
	1	0,6	0,766667

Probabilitas Kesemutan

		tidak <i>suspect</i>	<i>suspect</i>
X_{10}	0	0,8	0,016667
	1	0,2	0,983333

Probabilitas Pernah Stroke

		tidak <i>suspect</i>	<i>suspect</i>
X_{11}	0	1	0,833333
	1	0	0,166667

2) Perhitungan Probabilitas Naive Bayes dengan 60 Data Pelatihan

Perhitungan Mean dan Standart Deviasi Pada Data Numerik

	Umur		Nadi		Kolesterol	
	Tidak <i>Suspect</i>	<i>Suspect</i>	Tidak <i>Suspect</i>	<i>Suspect</i>	Tidak <i>Suspect</i>	<i>Suspect</i>
Mean	53,6	60	81,7	89,32	155,8	268,34
Variansi	144,64	103,52	142,81	226,9376	281,36	1261,024
Standart Deviasi	12,02664	10,17448	11,95031	15,06445	16,77379	35,51091

Probabilitas Jenis kelamin

		tidak	<i>suspect</i>
		<i>suspect</i>	
X_3	0	0,3	0,36
	1	0,7	0,64

Probabilitas Hipertensi

		tidak	<i>suspect</i>
		<i>suspect</i>	
X_5	0	0,1	0,04
	1	0,9	0,96

Probabilitas Diabetes Militus

		tidak	<i>suspect</i>
		<i>suspect</i>	
X_6	0	0,9	0,2
	1	0,1	0,8

Probabilitas Kesadaran

		tidak	<i>suspect</i>
		<i>suspect</i>	
X_7	0	0,1	0
	1	0,9	1

Probabilitas Pusing

		tidak <i>suspect</i>	<i>suspect</i>
X ₈	0	0,4	0,18
	1	0,6	0,82

Probabilitas Mual

		tidak <i>suspect</i>	<i>suspect</i>
X ₉	0	0,4	0,24
	1	0,6	0,76

Probabilitas Kesemutan

		tidak <i>suspect</i>	<i>suspect</i>
X ₁₀	0	0,8	0,02
	1	0,2	0,98

Probabilitas Pernah Stroke

		tidak <i>suspect</i>	<i>suspect</i>
X ₁₁	0	1	0,8
	1	0	0,2

Lampiran 12

Hasil Analisis Sistem

1) Data Pelatihan dengan 70 Data Latih

Data Pengujian dengan 20 Data Uji

No.	Target	Output bobot JST	Hasil Uji JST	Hasil Uji Naive Bayes
1	0	0.875534	1	1
2	1	0.908574	1	1
3	1	0.785434	1	1
4	1	0.985761	1	1
5	0	0.035121	0	0
6	1	1.04866	1	1
7	1	0.968993	1	1
8	1	0.778725	1	1
9	1	0.846991	1	1
10	0	0.357074	0	0
11	0	0.128443	0	0
12	0	0.041726	0	0
13	0	0.76452	1	0
14	0	0.0336847	0	0
15	0	0.245024	0	0
16	0	0.0363139	0	0
17	0	0.771394	1	0
18	0	0.343445	0	0
19	0	0.79698	1	1
20	0	0.947851	1	1
Jumlah data sesuai target			15	17

Data Pengujian dengan 30 Data Uji

No.	Target	Output bobot JST	Hasil Uji JST	Hasil Uji Naive Bayes
1	1	0.726441	1	0
2	1	1.00251	1	1
3	0	0.0336624	0	0
4	0	0.349774	0	0
5	1	0.997557	1	1
6	0	0.039963	1	1
7	1	0.966507	1	1
8	1	0.435219	0	1

9	1	0.953753	1	1
10	1	0.782923	1	1
11	0	0.875534	1	1
12	1	0.908574	1	1
13	1	0.785434	1	1
14	1	0.985761	1	1
15	0	0.035121	0	0
16	1	1.04866	1	1
17	1	0.968993	1	1
18	1	0.778725	1	1
19	1	0.846991	1	1
20	0	0.357074	0	0
21	0	0.128443	0	0
22	0	0.041726	0	0
23	0	0.76452	1	0
24	0	0.0336847	0	0
25	0	0.245024	0	0
26	0	0.0363139	0	0
27	0	0.771394	1	0
28	0	0.343445	0	0
29	0	0.79698	1	1
30	0	0.947851	1	1
Jumlah data sesuai target			24	25

Data Pengujian dengan 40 Data Uji

No.	Target	Output bobot JST	Hasil Uji JST	Hasil Uji Naive Bayes
1	1	0.883497	1	1
2	1	0.904724	1	1
3	0	0.356721	0	0
4	0	0.788009	1	0
5	0	0.118203	0	0
6	1	1.048	1	1
7	1	0.191739	0	1
8	1	0.982576	1	1
9	1	1.02522	1	1
10	1	0.880448	1	1
11	1	0.726441	1	0
12	1	1.00251	1	1
13	0	0.0336624	0	0
14	0	0.349774	0	0
15	1	0.997557	1	1
16	0	0.039963	1	1
17	1	0.966507	1	1

18	1	0.435219	0	1
19	1	0.953753	1	1
20	1	0.782923	1	1
21	0	0.875534	1	1
22	1	0.908574	1	1
23	1	0.785434	1	1
24	1	0.985761	1	1
25	0	0.035121	0	0
26	1	1.04866	1	1
27	1	0.968993	1	1
28	1	0.778725	1	1
29	1	0.846991	1	1
30	0	0.357074	0	0
31	0	0.128443	0	0
32	0	0.041726	0	0
33	0	0.76452	1	0
34	0	0.0336847	0	0
35	0	0.245024	0	0
36	0	0.0363139	0	0
37	0	0.771394	1	0
38	0	0.343445	0	0
39	0	0.79698	1	1
40	0	0.947851	1	1
Jumlah data sesuai target			32	35

2) Data Pelatihan dengan 60 Data Latih

Data Pengujian dengan 20 Data Uji

No.	Target	Output bobot JST	Hasil Uji JST	Hasil Uji Naive Bayes
1	0	0,896136	1	1
2	1	0,764612	1	1
3	1	0,682022	1	1
4	1	0,879721	1	1
5	0	0,790222	1	1
6	1	0,929012	1	1
7	1	0,782081	1	1
8	1	0,783711	1	1
9	1	0,989034	1	0
10	0	0,762786	1	0
11	0	0,9289112	1	0
12	0	0,672518	1	0
13	0	0,202103	0	0
14	0	0,828129	1	0
15	0	0,102165	0	0

16	0	0,936763	1	0
17	0	0,109832	0	0
18	0	0,653852	1	0
19	0	0,848573	1	1
20	0	0,973732	1	1
Jumlah data sesuai target			10	15

Data Pengujian dengan 30 Data Uji

No.	Target	<i>Output</i> bobot JST	Hasil Uji JST	Hasil Uji Naive Bayes
1	1	0,557385	1	0
2	1	0,896342	1	1
3	0	0,568326	1	0
4	0	0,752142	1	0
5	1	0,974573	1	0
6	0	0,975343	1	1
7	1	0,954321	1	1
8	1	0,953251	1	1
9	1	0,656594	1	1
10	1	0,953421	1	1
11	0	0,649685	1	1
12	1	0,745243	1	1
13	1	0,875832	1	1
14	1	0,525709	1	1
15	0	0.605303	1	1
16	1	0,774365	1	1
17	1	0,758905	1	1
18	1	0,982543	1	1
19	1	0,954392	1	0
20	0	0,232493	0	0
21	0	0,024211	0	0
22	0	0,432524	0	0
23	0	0,912738	1	0
24	0	0,121313	0	0
25	0	0,791421	1	0
26	0	0,123417	0	0
27	0	0,124677	0	0
28	0	0,973862	1	0
29	0	0,987321	1	1
30	0	0,872148	1	1
Jumlah data sesuai target			20	22

Data Pengujian dengan 40 Data Uji

No.	Target	Output bobot JST	Hasil Uji JST	Hasil Uji Naive Bayes
1	0	0,784752	1	1
2	1	0,832562	1	1
3	0	0,748649	1	0
4	0	0,870365	1	0
5	0	0,904759	1	0
6	1	0,760474	1	1
7	1	0,707349	1	1
8	1	0,878142	1	1
9	1	0,909322	1	1
10	1	0,724641	1	1
11	1	0,870832	1	0
12	1	0,803259	1	1
13	0	0,902302	1	0
14	0	0,032578	0	0
15	1	0,572852	1	0
16	0	0,709735	1	1
17	1	0,870325	1	1
18	1	0,864809	1	1
19	1	0,805832	1	1
20	1	0,738701	1	1
21	0	0,840329	1	0
22	1	0,935924	1	1
23	1	0,852355	1	1
24	1	0,973532	1	1
25	0	0,128378	0	1
26	1	0,777235	1	1
27	1	0,935702	1	1
28	1	0,973273	1	1
29	1	0,932759	1	0
30	0	0,131344	0	0
31	0	0,129374	0	0
32	0	0,959454	1	0
33	0	0,493986	0	0
34	0	0,734882	1	0
35	0	0,343433	0	0
36	0	0,621532	1	0
37	0	0,216492	0	0
38	0	0,148262	0	0
39	0	0,878024	1	1
40	0	0,837483	1	1

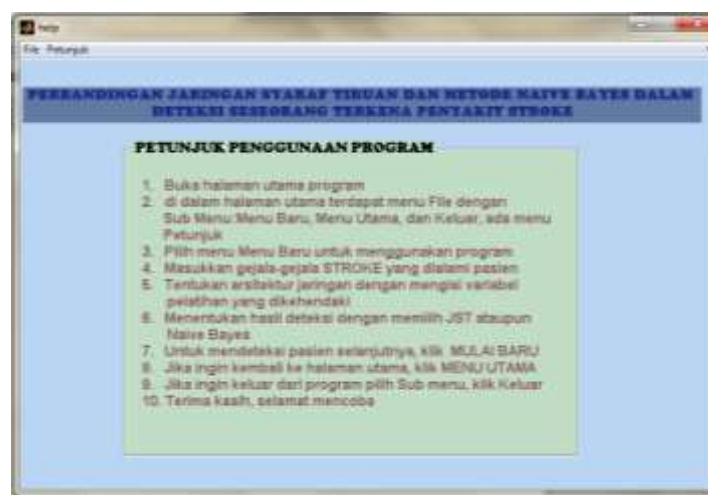
Jumlah data sesuai target	29	32
---------------------------	----	----

Lampiran 13

Contoh Program Saat Dijalankan



Gambar Desain Halaman Utama



Gambar Desain Petunjuk Penggunaan

Neural Network Training (nntraintool)

Neural Network:

Input → [Layer 1: W, b, Sum, Activation] → [Layer 2: W, b, Sum, Activation] → Output

Algorithms:

- Training: Levenberg-Marquardt (training)
- Performance: Mean Squared Error (mse)
- Data Division: Random (Standard)

Progress:

Metric	Value	Target/Range
Epoch	0	10 iterations / 1000
Time	0:00:17	
Performance	0.617	1.00e-16
Gradient	1.00	1.00e-10
Max	0.00100	1.00e+18
Validation Checks	0	6

Plots:

- ☒ Performance (splitperformance)
- ☐ Training Data (splittrainingdata)
- ☐ Regression (splitregression)

Plot Interval: 1 epochs

Validation step

Buttons: Stop Training, Cancel

Gambar Proses *Training* JST

Lampiran 14**Data Pasien Suspect Penyakit Stroke****Di RS. Tugurejo Semarang****No. Kode RS : 337**

No.	No. Pasien	Seks	Lama di rawat	Kelas Perawatan	Status Pulang
1	188549	L	5	KELAS III	Pulang hidup
2	404006	P	2	KELAS II	Pulang hidup
3	404177	L	2	KELAS II	Meninggal kurang 48 jam
4	205757	L	2	KELAS I	Meninggal kurang 48 jam
5	118952	L	2	KELAS II	Meninggal kurang 48 jam
6	254501	P	4	KELAS II	Meninggal lebih 48 jam
7	404510	L	5	KELAS III	Pulang hidup
8	404645	P	2	KELAS II	Meninggal kurang 48 jam
9	404910	P	9	KELAS III	Pulang hidup
10	233025	P	2	KELAS III	Meninggal kurang 48 jam
11	285939	L	2	KELAS I	Meninggal kurang 48 jam
12	405257	P	7	KELAS II	Pulang hidup
13	405295	P	1	KELAS I	Pulang hidup
14	405347	L	9	KELAS III	Pulang hidup
15	405350	L	2	KELAS III	Meninggal kurang 48 jam
16	405457	L	5	KELAS I	Ke rumah sakit
17	405461	P	8	KELAS III	Pulang hidup
18	210759	L	5	KELAS III	Aps/paksa
19	273409	L	7	KELAS III	Pulang hidup
20	405828	P	14	KELAS III	Pulang hidup
21	235538	L	7	KELAS III	Pulang hidup
22	238699	P	6	KELAS II	Pulang hidup
23	405911	L	2	KELAS III	Meninggal kurang 48 jam
24	406034	L	6	KELAS III	Pulang hidup
25	406376	P	8	KELAS I	Pulang hidup
26	406625	P	6	KELAS III	Pulang hidup

27	220226	L	6	KELAS III	Pulang hidup
28	253376	L	5	KELAS II	Pulang hidup
29	406760	L	3	KELAS I	Meninggal lebih 48 jam
30	398024	L	6	KELAS III	Pulang hidup
31	167048	L	7	KELAS III	Pulang hidup
32	243367	L	16	KELAS III	Pulang hidup
33	407157	L	7	KELAS III	Pulang hidup
34	121995	L	9	KELAS I	Pulang hidup
35	239462	L	11	KELAS I	Meninggal kurang 48 jam
36	407197	L	1	KELAS I	Meninggal kurang 48 jam
37	151392	L	13	KELAS II	Pulang hidup
38	149602	L	1	KELAS I	Meninggal lebih 48 jam
39	407496	P	13	KELAS III	Pulang hidup
40	101672	L	3	KELAS II	Meninggal lebih 48 jam
41	234252	L	13	KELAS III	Pulang hidup
42	259397	L	8	KELAS III	Pulang hidup
43	406522	P	10	KELAS III	Pulang hidup
44	223096	L	1	KELAS III	Meninggal kurang 48 jam
45	407869	L	7	KELAS III	Pulang hidup
46	407924	P	7	KELAS II	Pulang hidup
47	241338	P	1	KELAS II	Meninggal kurang 48 jam
48	119792	P	9	KELAS I	Pulang hidup
49	38375	P	11	KELAS III	Pulang hidup
50	390182	L	5	KELAS III	Pulang hidup
51	139758	P	9	KELAS III	Pulang hidup
52	403079	L	4	KELAS III	Meninggal lebih 48 jam
53	408216	L	2	KELAS III	Meninggal kurang 48 jam
54	408303	L	7	KELAS III	Aps/paksa
55	408343	P	6	KELAS III	Pulang hidup
56	276034	L	9	KELAS I	Pulang hidup
57	211554	P	2	KELAS II	Meninggal kurang 48 jam
58	408575	L	3	KELAS II	Meninggal lebih 48 jam
59	408614	L	6	KELAS III	Pulang hidup
60	408630	P	9	KELAS III	Pulang hidup
61	170494	L	7	KELAS III	Pulang hidup

62	408767	L	9	KELAS III	Pulang hidup
63	231082	P	3	KELAS I	Meninggal lebih 48 jam
64	195143	L	2	KELAS II	Meninggal kurang 48 jam
65	408959	P	13	KELAS III	Meninggal lebih 48 jam
66	398147	P	17	KELAS II	Meninggal lebih 48 jam
67	409183	L	6	KELAS II	Meninggal lebih 48 jam
68	409247	L	6	KELAS III	Pulang hidup
69	409258	L	7	KELAS III	Pulang hidup
70	210172	L	2	KELAS II	Meninggal kurang 48 jam
71	119233	L	2	KELAS II	Meninggal kurang 48 jam
72	220078	P	1	KELAS II	Meninggal kurang 48 jam
73	51397	P	4	KELAS II	Meninggal lebih 48 jam
74	409574	L	4	KELAS II	Meninggal lebih 48 jam
75	409634	L	6	KELAS III	Pulang hidup
76	409671	P	3	KELAS III	Aps/paksa
77	409728	P	1	KELAS I	Meninggal kurang 48 jam
78	165789	P	2	KELAS I	Meninggal kurang 48 jam
79	200250	L	7	KELAS III	Pulang hidup
80	143947	L	11	KELAS III	Pulang hidup
81	434644	L	5	KELAS I	Pulang hidup
82	232718	L	8	KELAS III	Pulang hidup
83	96361	L	7	KELAS III	Pulang hidup
84	49034	L	2	KELAS II	Meninggal kurang 48 jam
85	180260	L	2	KELAS II	Meninggal kurang 48 jam
86	435184	L	9	KELAS III	Pulang hidup
87	432235	P	18	KELAS III	Pulang hidup
88	435833	P	15	KELAS I	Pulang hidup
89	429375	P	7	KELAS III	Pulang hidup

Lampiran 15

Data Pasien Tidak *Suspect* Penyakit Stroke

Di RS. Tugurejo Semarang

No. Kode RS : 337

No.	No. Pasien	Seks	Lama di rawat	Kelas Perawatan	Status Pulang
1	409780	P	7	VIP	Pulang hidup
2	409790	P	7	KELAS III	Pulang hidup
3	409889	L	6	KELAS III	Pulang hidup
4	409916	L	10	KELAS III	Pulang hidup
5	410028	P	2	KELAS II	Meninggal kurang 48 jam
6	66660	L	5	KELAS I	Meninggal lebih 48 jam
7	150618	L	9	KELAS III	Pulang hidup
8	142676	L	8	KELAS III	Pulang hidup
9	410323	L	11	KELAS I	Pulang hidup
10	144503	L	6	KELAS III	Pulang hidup
11	395997	L	6	KELAS II	Pulang hidup
12	262372	P	10	KELAS III	Dirujuk ke panti
13	410510	P	2	KELAS II	Meninggal kurang 48 jam
14	176602	L	2	KELAS III	Pulang hidup
15	410613	L	7	KELAS II	Pulang hidup
16	171426	P	5	KELAS III	Pulang hidup
17	411174	P	7	KELAS III	Pulang hidup
18	411178	P	8	KELAS III	Pulang hidup
19	411285	P	6	VIP	Pulang hidup
20	411375	P	9	KELAS I	Pulang hidup
21	411482	P	1	KELAS II	Meninggal kurang 48 jam
22	411504	P	1	KELAS I	Meninggal kurang 48 jam
23	283089	L	8	KELAS III	Pulang hidup
24	433568	L	7	KELAS III	PULANG HIDUP
25	208702	P	10	VIP	PULANG HIDUP
26	433761	P	5	KELAS III	PULANG HIDUP
27	433834	L	7	KELAS I	APS/PAKSA
28	433855	P	2	VIP	PULANG HIDUP
29	432913	P	3	KELAS I	PULANG HIDUP
30	433928	L	7	KELAS I	PULANG HIDUP
31	433941	L	7	KELAS III	PULANG HIDUP

Lampiran 16



**KEPUTUSAN
DEKAN FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS NEGERI SEMARANG**

Nomor: 725/P/2013

**Tentang
PENETAPAN DOSEN PEMBIMBING SKRIPSI/TUGAS AKHIR SEMESTER
GASAL/GENAP
TAHUN AKADEMIK 2013/2014**

- Menimbang** : Bahwa untuk memperlancar mahasiswa Jurusan/Prodi Matematika/Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam membuat Skripsi/Tugas Akhir, maka perlu menetapkan Dosen-dosen Jurusan/Prodi Matematika/Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam UNNES untuk menjadi pembimbing.
- Mengingat** : 1. SK. Rektor UNNES No. 164/O/2004 tentang Pedoman penyusunan Skripsi/Tugas Akhir Mahasiswa Strata Satu (S1) UNNES;
2. SK Rektor UNNES No.162/O/2004 tentang penyelenggaraan Pendidikan UNNES;
3. Undang-undang No.20 Tahun 2003 tentang Sistem Pendidikan Nasional (Tambahan Lembaran Negara RI No.4301, penjelasan atas Lembaran Negara RI Tahun 2003, Nomor 78)
- Menimbang** : Usulan Ketua Jurusan/Prodi Matematika/Matematika Tanggal 29 Oktober 2013

MEMUTUSKAN

- Menetapkan** :
PERTAMA : Menunjuk dan menugaskan kepada:
- | | |
|------------------|-------------------------------|
| Nama | : Riza Arifudin, S.Pd., M.Cs. |
| NIP | : 198005252005011001 |
| Pangkat/Golongan | : III/A |
| Jabatan Akademik | : Lektor |
- Sebagai Pembimbing
- Untuk membimbing mahasiswa penyusun skripsi/Tugas Akhir :
- | | |
|---------------|--|
| Nama | : ISTATIK ROHMANA |
| NIM | : 4111410022 |
| Jurusan/Prodi | : Matematika/Matematika |
| Topik | : ANALISIS PREDIKSI SESEORANG TERKENA STOKE DENGAN MENGGUNAKAN JARINGAN SYARAF TIRUAN DAN METODE NAIVE BAYES |
- KEDUA** : Keputusan ini mulai berlaku sejak tanggal ditetapkan.

Tembusan
1. Pembantu Dekan Bidang Akademik
2. Ketua Jurusan
3. Dosen Pembimbing
4. Petinggal



4111410022

FM-03-AKD-24/Rev. 00

**DITETAPKAN DI : SEMARANG
PADA TANGGAL : 29 Oktober 2013
DEKAN**

Prof. Dr. Wiyanto, M.Si.
NIP 196310121988031001



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS NEGERI SEMARANG

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

Gedung D5 Lt.1 Kampus Sekaran Gunungpati Semarang Kode Pos 50229, Telp. (024)8508112
Telp. Dekan (024)8508005; Jurusan: Matematika (024)8508032; Fisika (024)8508034; Kimia (024)8508035; Biologi (024)8508033
Fax. (024)8508005; Website: <http://mipa.unnes.ac.id>; Email: mipa@unnes.ac.id

No : 9523 /UN37.1.4/LT/2013

Lamp : -

Hal : Ijin Penelitian

Kepada

Yth. Direktur Utama RSUD Tugurejo Semarang

Di Semarang

Dengan hormat,

Bersama ini, kami mohon ijin pelaksanaan penelitian untuk penyusunan skripsi/Tugas Akhir oleh mahasiswa sebagai berikut:

Nama : Istatik Rohmana
NIM : 4111410022
Prodi : Matematika, S1
Judul : Analisis Prediksi Seseorang Terkena Stroke Dengan Menggunakan Jaringan Syaraf Tiruan dan Metode Naive Bayes
Tempat : RSUD Tugurejo Semarang
Waktu : 24 Desember 2013 s/d 24 Januari 2014

Atas perhatian dan kerjasamanya disampaikan terima kasih.



24 Desember 2013

Dr. Wiyanto, M.Si

NIP. 19631012 198803 1 001

FM-05-AKD-24



PEMERINTAH PROVINSI JAWA TENGAH
RUMAH SAKIT UMUM DAERAH TUGUREJO
 Alamat Kantor : Jalan Raya Tugurejo, Semarang Telepon. 7605378, 7605297 Faksimile. 7604398
 Email: tugurejo@jatengprov.go.id Website: www.rstugurejo.com

Semarang, 30 Desember 2013

Nomor : 423.4 / 6538
 Lampiran : -
 Perihal : Ijin Penelitian

Kepada Yth,
 Dekan Fakultas Matematika dan Ilmu
 Pengetahuan Alam
 Universitas Negeri Semarang
 di


SEMARANG

Menindaklanjuti surat Saudara nomor : 6033/UN7.3.4/D1/PP/2013 tanggal 20 Desember 2013 perihal tersebut pada pokok surat, pada dasarnya kami **tidak keberatan dan memberi ijin** untuk melakukan Pengambilan Data Awal di Rumah Sakit Umum Daerah Tugurejo Provinsi Jawa Tengah kepada mahasiswa yang Saudara ajukan :

NAMA : Istatik Rohmana
 NIM : 4111410022
 PRODI : S1 Matematika
 JUDUL : " Analisis Prediksi Seseorang Terkena Stroke dengan Menggunakan Jaringan Saraf Tiruan dan Metode *Naive Bayes*."

Demikian atas perhatian dan kerjasamanya kami ucapkan terima kasih.

a.n. DIREKTUR RSUD TUGUREJO
 PROVINSI JAWA TENGAH
 Wadiv. Umum dan Keuangan


Drs. RETNO SUDEWI, Apt., MSI, MM
 Pembina Tingkat I
 NIP. 19681124 199310 2 001

TEMBUSAN :

1. Direktur RSUD Tugurejo (sebagai laporan)