

IMPLEMENTASI METODE *SUPPORT VECTOR MACHINE* UNTUK REKOMENDASI PEMILIHAN TERAPI DEHIDRASI PADA ANAK

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Rani Anggi Nilam Sari
NIM: 135150109111001



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

PENGESAHAN

IMPLEMENTASI METODE *SUPPORT VECTOR MACHINE* UNTUK REKOMENDASI
PEMILIHAN TERAPI DEHIDRASI PADA ANAK

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Rani Anggi Nilam Sari
NIM: 135150109111001

Skripsi ini telah diuji dan dinyatakan lulus pada
19 Januari 2016

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Imam Cholissodin, S.Si, M.Kom
NIK: 850719 16 1 1 0422

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

Mengetahui
Ketua Program Studi Informatika/Illmu Komputer

Drs. Marji., M.T
NIP: 19670801 199203 1 001

PERNYATAAN ORISINALITAS

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

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

Malang, 19 Januari 2016



Rani Anggi Nilam Sari

NIM: 135150109111001

KATA PENGANTAR

Alhamdulillah rabbil 'alamin. Puji syukur penulis panjatkan kehadirat Allah SWT, karena atas segala rahmat dan limpahan hidayah-Nya, penulis dapat menyelesaikan laporan skripsi yang berjudul “Implementasi Metode *Support Vector Machine* Untuk Rekomendasi Pemilihan Terapi Dehidrasi Pada Anak”.

Dalam pelaksanaan dan penulisan laporan skripsi ini penulis mendapatkan banyak bantuan dari berbagai pihak baik secara moril maupun materil. Dalam kesempatan ini penulis ingin mengucapkan terimakasih yang sebesar-besarnya kepada :

1. Bapak Imam Cholissodin, S.Si., M.Kom dan Bapak Nurul Hidayat, S.Pd, M.Sc selaku dosen pembimbing skripsi yang telah dengan sabar membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini.
2. Bapak Ir. Sutrisno, M.T, Bapak Ir. Heru Nurwasito, M.Kom, Bapak Himawat Aryadita, S.T, M.Sc, dan Bapak Edy Santoso, S.Si, M.Kom selaku Ketua, Wakil Ketua 1, Wakil Ketua 2 dan Wakil Ketua 3 Fakultas Ilmu Komputer Universitas Brawijaya.
3. Bapak Drs. Marji, M.T dan Bapak Issa Arwani, S.Kom, M.Sc selaku Ketua dan Sekretaris Program Studi Teknik Informatika Universitas Brawijaya.
4. Ibu, Ayah, dan seluruh keluarga besar atas segala nasehat, kasih sayang, perhatian dan kesabarannya di dalam membesarkan dan mendidik penulis, serta yang senantiasa tiada henti-hentinya memberikan doa dan semangat demi terselesaikannya skripsi ini.
5. Ir. Tjutjuk Usmanhadi dan Putra Hermawan selaku Kabag Kemahasiswaan Bidang Kesejahteraan Mahasiswa Universitas Brawijaya dan bagian Kemahasiswaan Fakultas Ilmu Komputer Universitas Brawijaya.
6. Seluruh Dosen Teknik Informatika Universitas Brawijaya atas kesediaan membagi ilmunya kepada penulis.
7. Seluruh Civitas Akademika Teknik Informatika Universitas Brawijaya yang telah banyak memberi bantuan dan dukungan selama penulis menempuh studi di Teknik Informatika Universitas Brawijaya dan selama penyelesaian skripsi ini.
8. Teman-teman Konsentrasi Komputasi Cerdas dan Visualisasi serta seluruh Mahasiswa SAP Angkatan 2013 Teknik Informatika, terima kasih atas segala bantuannya selama menempuh studi di Teknik Informatika Universitas Brawijaya.
9. Seluruh pihak yang telah membantu kelancaran penulisan laporan skripsi yang tidak dapat kami sebutkan satu persatu.

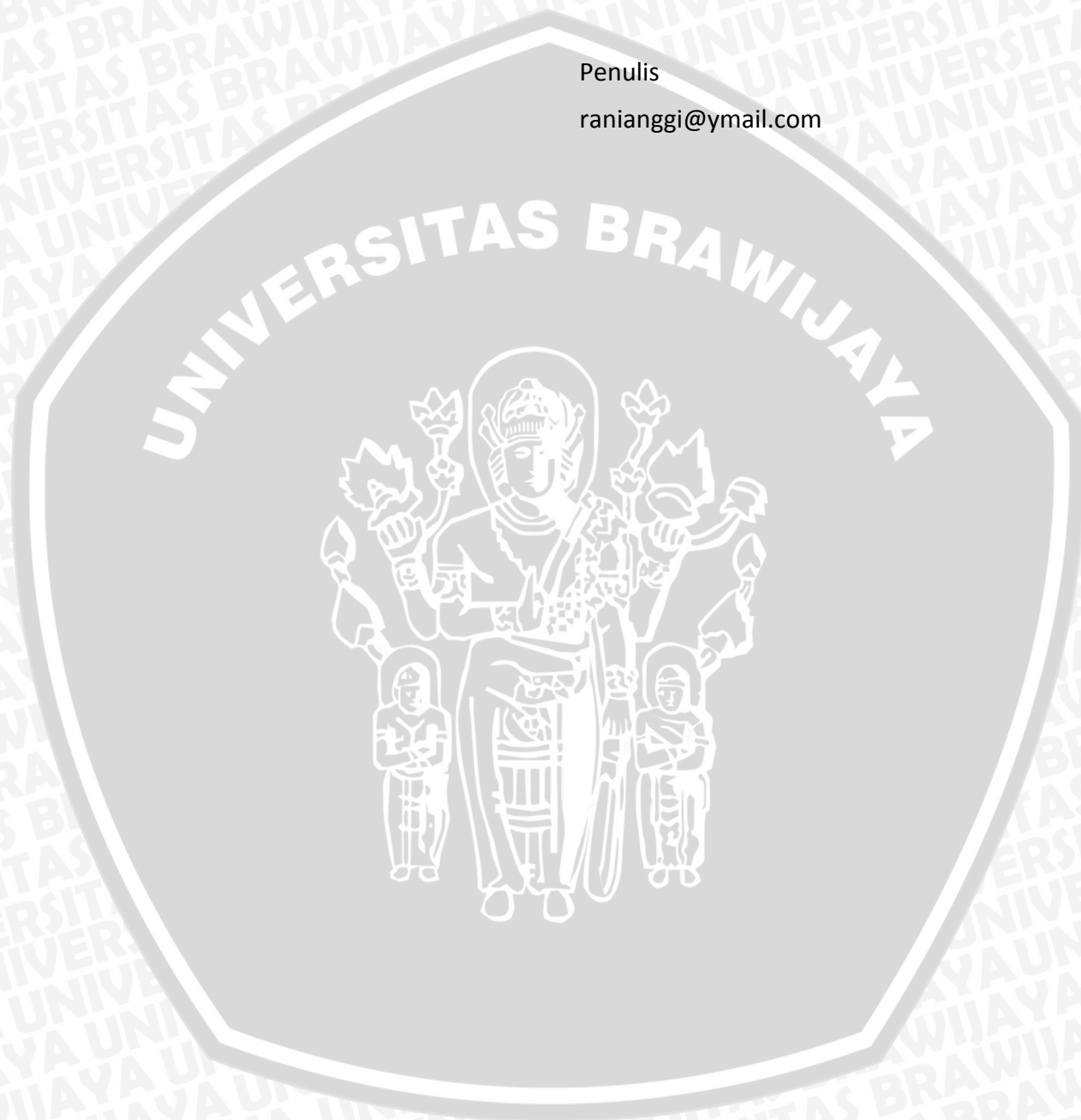
Penulis menyadari bahwa dalam penyusunan laporan skripsi ini masih banyak kekurangan baik format penulisan maupun isinya. Oleh karena itu, saran dan kritik

membangun dari para pembaca senantiasa penulis harapan guna perbaikan bagi laporan skripsi selanjutnya. Semoga laporan skripsi ini dapat memberikan manfaat bagi semua pihak, Amin.

Malang, 19 Januari 2016

Penulis

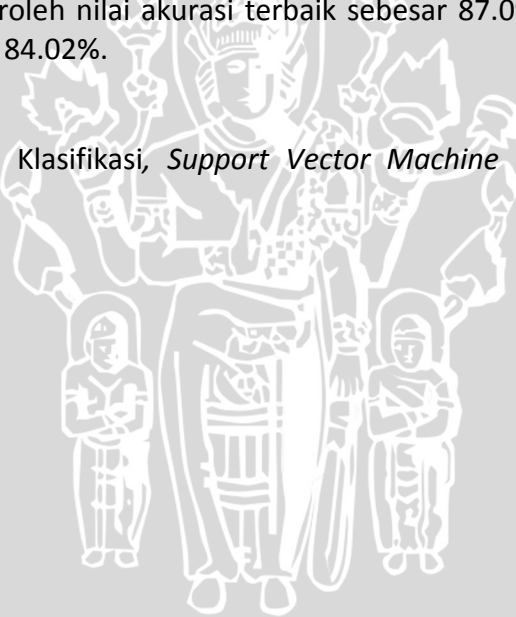
ranianggi@gmail.com



ABSTRAK

Dehidrasi adalah kehilangan cairan dari jaringan tubuh yang berlebihan. Dehidrasi merupakan gangguan yang umum terjadi pada bayi dan anak-anak ketika haluaran cairan total melebihi asupan cairan total. Hilangnya cairan yang berlebihan dikhawatirkan akan berdampak buruk pada kesehatan anak. Oleh karena itu, dibutuhkan sistem yang dapat mempercepat dan memudahkan perekomendasi terapi dehidrasi. Salah satu teknik yang dipakai adalah klasifikasi menggunakan metode *Support Vector Machine*. Metode *Support Vector Machine* bertujuan untuk membuat *hyperplane* atau garis pemisah yang optimal. Kemudian metode *Support Vector Machine* tersebut akan dilatih menggunakan metode *Sequential Training* agar mendapatkan posisi optimal dari *hyperplane* di ruang vektor. Pada penelitian ini digunakan sebanyak 150 data. Parameter yang digunakan dalam penelitian ini yaitu keadaan umum, nadi, pernapasan/RR, ubun-ubun, mata, elastisitas kulit, *Glasgow Coma Scale* (GCS), status gizi, dan suhu. Dari hasil pengujian semua skenario percobaan menggunakan metode SVM dengan kernel *Gaussian RBF* serta nilai parameter SVM $\lambda = 0.1$, $\gamma = 0.05$, $\sigma = 0.0001$, $itermax = 100$, dan $C = 1$, diperoleh nilai akurasi terbaik sebesar 87.09% serta rata-rata akurasi terbaik sebesar 84.02%.

Kata kunci: Dehidrasi, Klasifikasi, *Support Vector Machine* (SVM), *Sequential Training*.



ABSTRACT

Dehydration is excessive loss of body water. Dehydration is a common disorder that occurs in infants and children when the total fluid output exceeds the total fluid intake. Excessive fluid loss is feared to have a negative impact on children's health. Therefore, it needs a system that can accelerate and facilitate recommendation dehydration therapy. One technique used is the classification using Support Vector Machine. Support Vector Machine method aims to make hyperplane or optimal dividing line. Then the method of Support Vector Machine will be trained using Sequential Training to get the optimal position of the hyperplane in a vector space. In this research used 150 data. The parameters used in this research is general condition, pulse, respiration/RR, fontanel, eyes, skin elasticity, Glasgow Coma Scale (GCS), nutritional status, and temperature. From the results of experiments testing all scenarios using SVM with Gaussian RBF kernel and SVM parameter values $\lambda = 0.1$, $\gamma = 0.05$, $\sigma = 0.0001$, $\text{itermax} = 100$, and $C = 1$, the value of the best accuracy of 87.09% and the average accuracy is best for 84.02%.

Keywords: Dehydration, Classification, Support Vector Machine (SVM), Sequential Training.



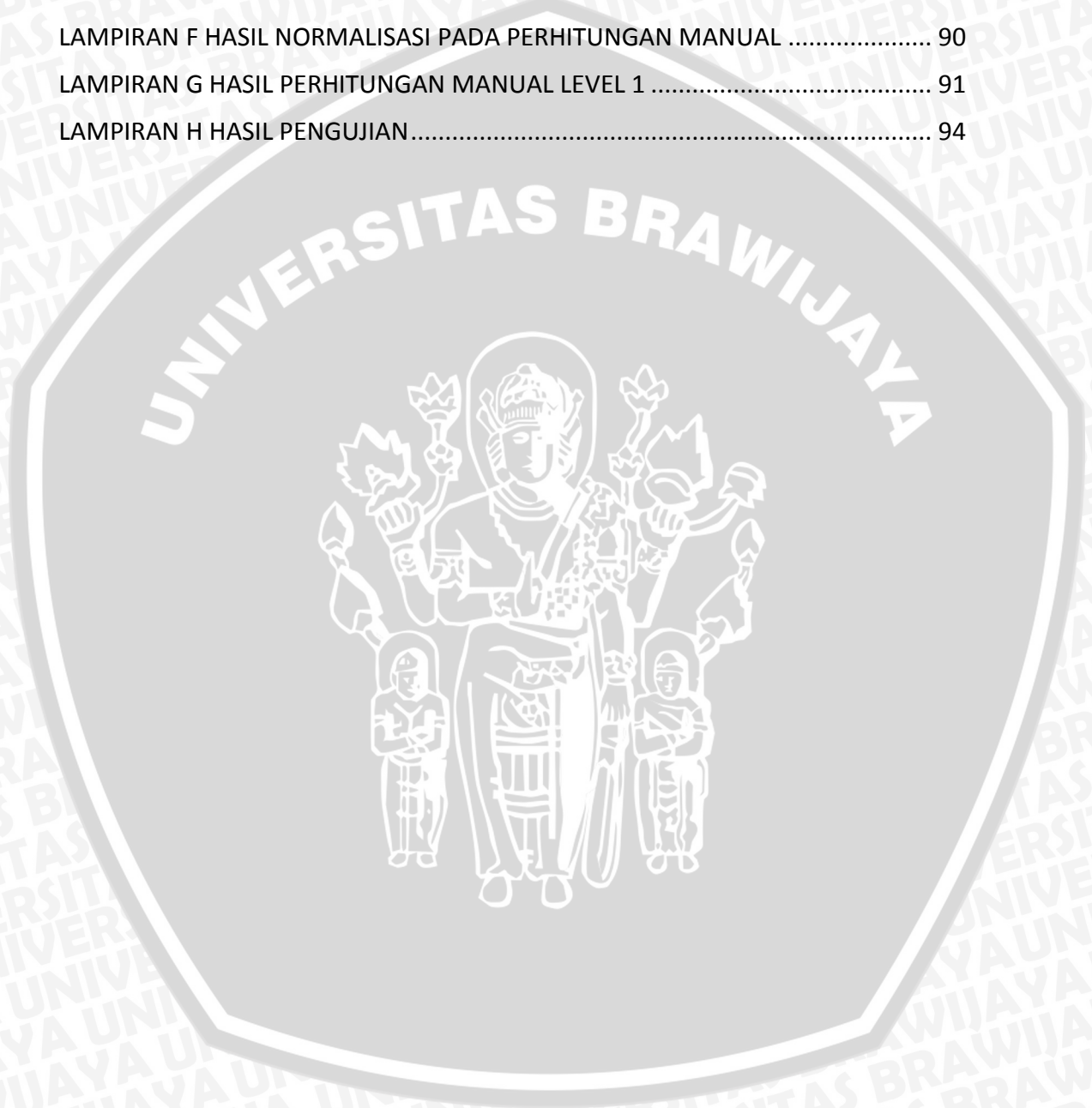
DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xiv
DAFTAR LAMPIRAN	xvi
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan	2
1.4 Manfaat.....	3
1.5 Batasan masalah	3
1.6 Sistematika pembahasan	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Tinjauan Pustaka.....	5
2.2 Dehidrasi	7
2.2.1 Jenis Dehidrasi.....	7
2.2.2 Etiologi Dehidrasi	8
2.2.3 Tanda-Tanda Dehidrasi	8
2.3 Klasifikasi.....	9
2.4 Normalisasi Data	9
2.5 <i>Support Vector Machine (SVM)</i>	10
2.5.1 <i>Support Vector Machine Linier</i>	11
2.5.2 <i>Support Vector Machine Non-Linier</i>	12
2.6 <i>Multiclass SVM</i>	13
2.6.1 <i>Metode One-Against-All</i>	13
2.6.2 <i>Metode One-Against-One</i>	14

2.6.3 Metode <i>Directed Acyclic Graph Support Vector Machine</i> (DAGSVM)	15
2.7 Metode <i>Sequential Training</i> pada SVM	16
2.8 Evaluasi	17
BAB 3 METODOLOGI PENELITIAN DAN PERANCANGAN.....	19
3.1 Studi Literatur	19
3.2 Pengumpulan Data	19
3.3 Analisis Kebutuhan	20
3.4 Perancangan Sistem.....	20
3.4.1 Normalisasi <i>Dataset</i>	21
3.4.2 Proses <i>Support Vector Machine</i> (SVM)	22
3.4.3 Perhitungan <i>Kernel SVM</i>	23
3.4.4 Perhitungan <i>Sequential Training SVM</i>	24
3.4.5 Perhitungan <i>Matriks Hessian</i>	25
3.4.6 Perhitungan Nilai E_i	26
3.4.7 Proses Perhitungan <i>Testing SVM</i>	27
3.4.8 Perhitungan Nilai $f(x)$ pada Data <i>Testing SVM</i>	28
3.4.9 <i>One Against All</i>	29
3.5 Implementasi Sistem	30
3.5.1 Perhitungan Manual.....	30
3.5.2 Perancangan Antarmuka.....	47
3.6 Pengujian Sistem.....	49
3.6.1 Pengujian terhadap Rasio Perbandingan	50
3.6.2 Pengujian terhadap Parameter <i>Kernel Gaussian RBF</i>	50
3.6.3 Pengujian terhadap Parameter <i>Sequential Training SVM</i>	51
3.7 Evaluasi Hasil.....	53
BAB 4 IMPLEMENTASI	54
4.1 Spesifikasi Sistem	54
4.1.1 Spesifikasi Perangkat Keras	54
4.1.2 Spesifikasi Perangkat Lunak	54
4.2 Batasan Implementasi	55
4.3 Implementasi Algoritma	55
4.3.1 Implementasi Algoritma Perhitungan <i>Kernel</i>	55

4.3.2 Implementasi Algoritma Perhitungan <i>Matriks Hessian</i>	56
4.3.3 Implementasi Algoritma Perhitungan Iterasi	56
4.3.4 Implementasi Algoritma Perhitungan Nilai wx^+ dan wx^-	57
4.3.5 Implementasi Algoritma Perhitungan Nilai $f(x)$	59
4.3.6 Implementasi Algoritma <i>Confusion Matrix</i>	60
4.4 Implementasi Antarmuka	61
4.4.1 Implementasi Antarmuka Halaman Perhitungan SVM Level 1... 61	
4.4.2 Implementasi Antarmuka Halaman Perhitungan SVM Level 2... 61	
4.4.3 Implementasi Antarmuka Halaman Data <i>Testing</i>	62
4.4.4 Implementasi Antarmuka Halaman Hasil	63
BAB 5 PENGUJIAN DAN ANALISIS	64
5.1 Pengujian Tingkat Akurasi terhadap Rasio Perbandingan	64
5.1.1 Skenario Pengujian Rasio Perbandingan	64
5.1.2 Analisis Pengujian Rasio Perbandingan	64
5.2 Pengujian terhadap Parameter <i>Kernel Gaussian RBF</i>	65
5.2.1 Skenario Pengujian <i>Sigma</i>	65
5.2.2 Analisis Pengujian <i>Sigma</i>	66
5.3 Pengujian terhadap Parameter <i>Sequential Training SVM</i>	67
5.3.1 Skenario Pengujian <i>Lamda</i>	67
5.3.2 Analisis Pengujian <i>Lamda</i>	67
5.3.3 Skenario Pengujian Konstanta <i>Gamma</i>	68
5.3.4 Analisis Pengujian Konstanta <i>Gamma</i>	69
5.3.5 Skenario Pengujian <i>Epsilon</i>	70
5.3.6 Analisis Pengujian <i>Epsilon</i>	70
5.3.7 Skenario Pengujian <i>Itermax</i>	71
5.3.8 Analisis Pengujian <i>Itermax</i>	72
5.3.9 Skenario Pengujian <i>Complexity</i>	73
5.3.10 Analisis Pengujain <i>Complexity</i>	73
BAB 6 Penutup	75
6.1 Kesimpulan	75
6.2 Saran	75
DAFTAR PUSTAKA	77

LAMPIRAN A SURAT KETERANGAN DARI RUMAH SAKIT	79
LAMPIRAN B SURAT KETERANGAN DARI KESBANGPOL	80
LAMPIRAN C SURAT IJIN PENELITIAN.....	81
LAMPIRAN D <i>DATASET</i>	82
LAMPIRAN E SAMPEL <i>DATASET</i> PADA PERHITUNGAN MANUAL.....	89
LAMPIRAN F HASIL NORMALISASI PADA PERHITUNGAN MANUAL	90
LAMPIRAN G HASIL PERHITUNGAN MANUAL LEVEL 1	91
LAMPIRAN H HASIL PENGUJIAN.....	94



DAFTAR TABEL

Tabel 2.1 Tinjauan pustaka	5
Tabel 2.2 Tanda dan gejala	8
Tabel 2.3 Metode <i>one-against-all</i> dengan 4 kelas.....	14
Tabel 2.4 Contoh 6 SVM <i>biner</i> dengan metode <i>one-against-one</i>	14
Tabel 2.5 Contoh 6 SVM biner dengan metode DAGSVM.....	15
Tabel 2.6 <i>Confusion matrix</i> 3x3.....	17
Tabel 3.1 <i>Range</i> normalisasi	30
Tabel 3.2 Tabel hasil normalisasi	31
Tabel 3.3 Data <i>training</i> 1.....	32
Tabel 3.4 Data <i>training</i> 2.....	32
Tabel 3.5 Data <i>training</i> 3.....	33
Tabel 3.6 Hasil perhitungan data <i>training</i> dengan <i>kernel</i> RBF	34
Tabel 3.7 Hasil perhitungan <i>matriks hessian</i>	35
Tabel 3.8 Hasil perhitungan <i>iterasi</i> ke 1 pada persamaan 2.20.....	36
Tabel 3.9 Hasil perhitungan <i>iterasi</i> ke 1 pada persamaan 2.20.....	36
Tabel 3.10 Data <i>training</i> 2.....	37
Tabel 3.11 Data <i>training</i> 6.....	38
Tabel 3.12 Hasil perhitungan $K(x_i, x^+)$ dan $K(x_i, x^-)$	39
Tabel 3.13 Hasil perhitungan $w \cdot x^+$ dan $w \cdot x^-$	40
Tabel 3.14 Data <i>training</i> 1.....	41
Tabel 3.15 Data <i>testing</i> 1	41
Tabel 3.16 Data <i>training</i> 2.....	42
Tabel 3.17 Hasil perhitungan $K(x_i, x)$ dan $\alpha_i y_i K(x_i, x)$	43
Tabel 3.18 Data <i>testing</i> 2	44
Tabel 3.19 hasil perhitungan $K(x_i, x)$ dan $\alpha_i y_i K(x_i, x)$	46
Tabel 3.20 Data hasil perhitungan $f(x)$ dan klasifikasi	46
Tabel 3.21 Model evaluasi <i>confusion matriks</i>	47
Tabel 3.22 Pengaruh rasio perbandingan terhadap rata-rata akurasi	50
Tabel 3.23 Pengaruh nilai σ (<i>sigma</i>) terhadap rata-rata akurasi.....	50
Tabel 3.24 Pengaruh nilai λ (<i>lamda</i>) terhadap rata-rata akurasi.....	51
Tabel 3.25 Pengaruh konstanta γ (<i>gamma</i>) terhadap rata-rata akurasi	51

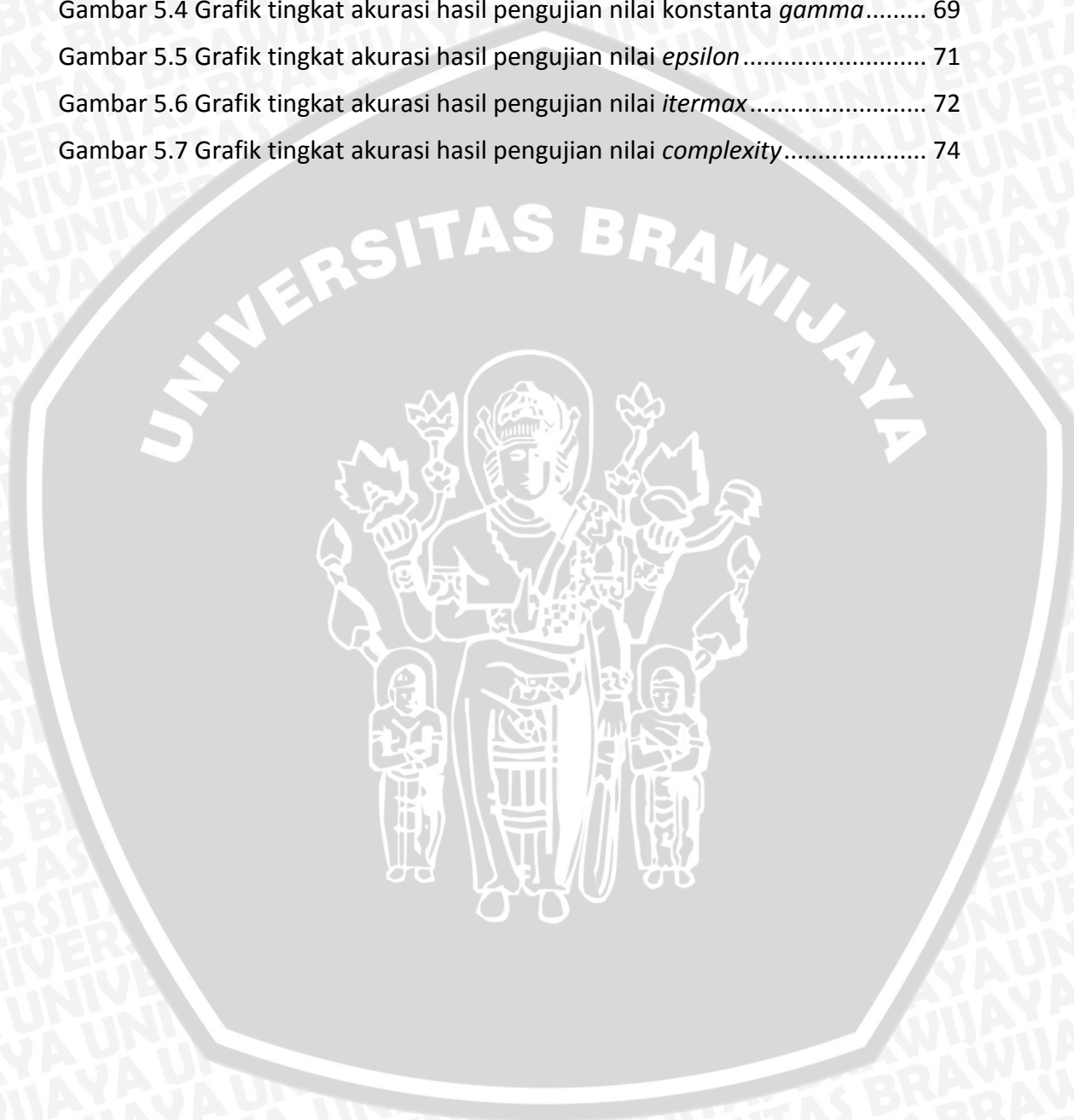
Tabel 3.26 Pengaruh nilai ϵ (<i>epsilon</i>) terhadap rata-rata akurasi.....	52
Tabel 3.27 Pengaruh nilai (<i>itermax</i>) terhadap rata-rata akurasi	52
Tabel 3.28 Pengaruh nilai c (<i>complexity</i>) terhadap rata-rata akurasi.....	52
Tabel 4.1 Spesifikasi perangkat keras	54
Tabel 4.2 Spesifikasi perangkat lunak	54
Tabel 5.1 Hasil pengujian rasio perbandingan	64
Tabel 5.2 Hasil pengujian <i>sigma</i>	66
Tabel 5.3 Hasil pengujian <i>lamda</i>	67
Tabel 5.4 Hasil pengujian konstanta γ (<i>gamma</i>).....	69
Tabel 5.5 Hasil pengujian <i>epsilon</i>	70
Tabel 5.6 Hasil pengujian <i>itermax</i>	72
Tabel 5.7 Hasil pengujian <i>complexity</i>	73



DAFTAR GAMBAR

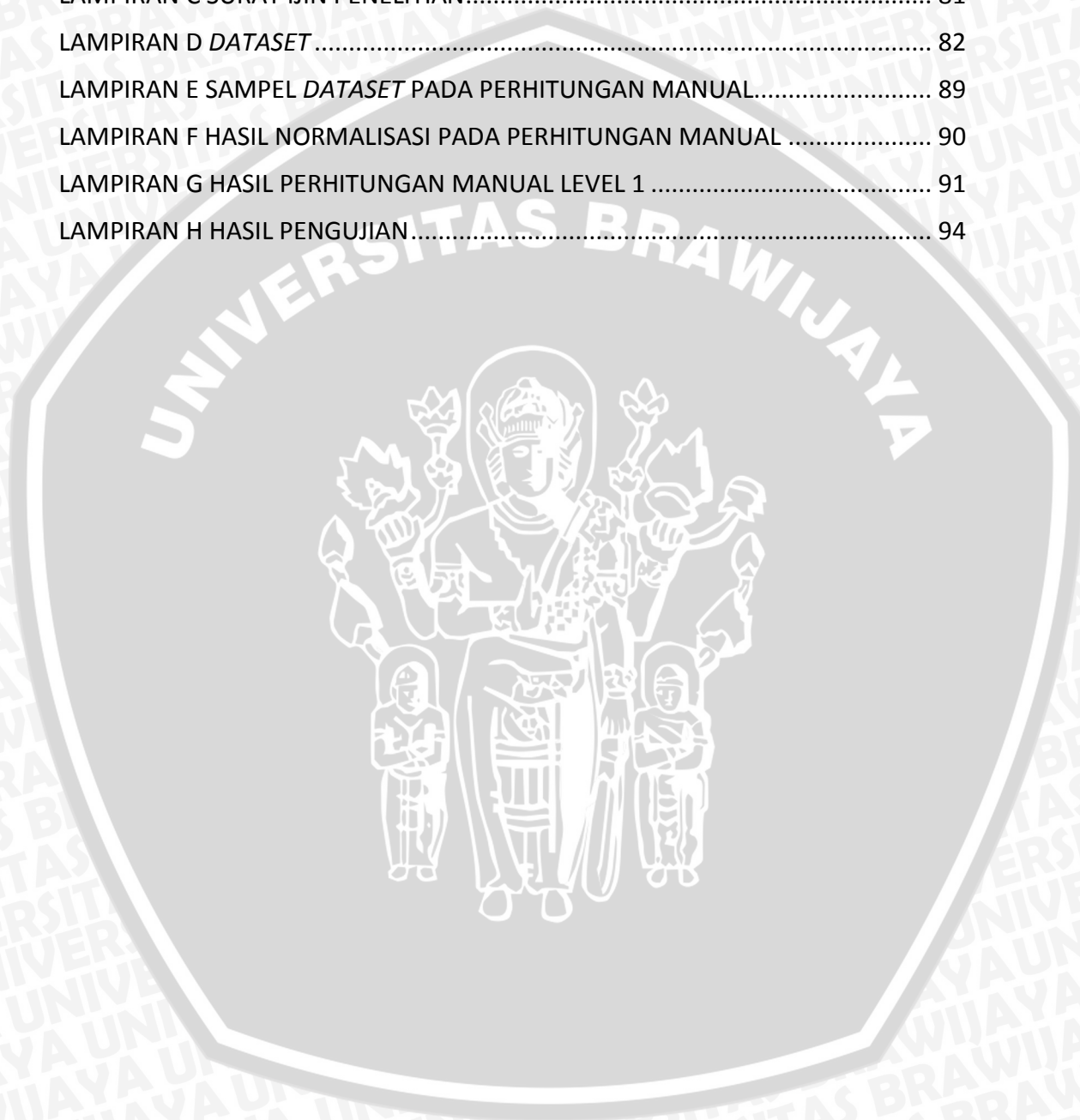
Gambar 2.1 Ide dasar penerapan SVM	10
Gambar 2.2 Contoh klasifikasi <i>one-against-all</i> untuk 4 kelas	14
Gambar 2.3 Contoh klasifikasi dengan metode <i>one-against-one</i>	15
Gambar 2.4 Contoh klasifikasi dengan metode DAGSVM	16
Gambar 3.1 Diagram alir penelitian	19
Gambar 3.2 Diagram alir perancangan sistem	21
Gambar 3.3 Diagram alir normalisasi <i>dataset</i>	21
Gambar 3.4 Diagram alir proses <i>Support Vector Machine</i>	22
Gambar 3.5 Diagram alir perhitungan <i>kernel</i> SVM	23
Gambar 3.6 Diagram alir perhitungan <i>sequential training SVM</i>	24
Gambar 3.7 Diagram alir perhitungan <i>matriks hessian</i>	26
Gambar 3.8 Diagram alir perhitungan nilai E_i	26
Gambar 3.9 Diagram alir perhitungan <i>testing SVM</i>	27
Gambar 3.10 Diagram alir perhitungan nilai $f(x)$	28
Gambar 3.11 Diagram alir perhitungan proses <i>one against all</i>	29
Gambar 3.12 <i>Form</i> perhitungan SVM level 1	48
Gambar 3.13 <i>Form</i> perhitungan SVM level 2	48
Gambar 3.14 <i>Form</i> data <i>testing</i>	49
Gambar 3.15 <i>Form</i> hasil	49
Gambar 4.1 Implementasi algoritma perhitungan <i>kernel</i> RBF	55
Gambar 4.2 Implementasi algoritma perhitungan matriks <i>hessian</i>	56
Gambar 4.3 Implementasi algoritma perhitungan E_i	56
Gambar 4.4 Implementasi algoritma perhitungan nilai δa_i	57
Gambar 4.5 Implementasi algoritma perhitungan nilai a_i baru	57
Gambar 4.6 Implementasi algoritma perhitungan nilai wx^+ dan wx^-	58
Gambar 4.7 Implementasi algoritma perhitungan nilai fungsi linier $f(x)$	59
Gambar 4.8 Implementasi algoritma <i>confusion matrix</i>	60
Gambar 4.9 Tampilan implementasi halaman perhitungan SVM level 1	61
Gambar 4.10 Tampilan implementasi halaman perhitungan SVM level 2	62
Gambar 4.11 Tampilan implementasi halaman data <i>testing</i>	62
Gambar 4.12 Tampilan implementasi halaman hasil	63

Gambar 5.1 Grafik tingkat akurasi hasil pengujian rasio perbandingan.....	65
Gambar 5.2 Grafik tingkat akurasi hasil pengujian nilai <i>sigma</i>	66
Gambar 5.3 Grafik tingkat akurasi hasil pengujian nilai <i>lamda</i>	68
Gambar 5.4 Grafik tingkat akurasi hasil pengujian nilai konstanta <i>gamma</i>	69
Gambar 5.5 Grafik tingkat akurasi hasil pengujian nilai <i>epsilon</i>	71
Gambar 5.6 Grafik tingkat akurasi hasil pengujian nilai <i>itermax</i>	72
Gambar 5.7 Grafik tingkat akurasi hasil pengujian nilai <i>complexity</i>	74



DAFTAR LAMPIRAN

LAMPIRAN A SURAT KETERANGAN DARI RUMAH SAKIT	79
LAMPIRAN B SURAT KETERANGAN DARI KESBANGPOL	80
LAMPIRAN C SURAT IJIN PENELITIAN.....	81
LAMPIRAN D <i>DATASET</i>	82
LAMPIRAN E SAMPEL <i>DATASET</i> PADA PERHITUNGAN MANUAL.....	89
LAMPIRAN F HASIL NORMALISASI PADA PERHITUNGAN MANUAL	90
LAMPIRAN G HASIL PERHITUNGAN MANUAL LEVEL 1	91
LAMPIRAN H HASIL PENGUJIAN.....	94



BAB 1 PENDAHULUAN

Bab ini menguraikan mengenai latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah dan sistematika pembahasan Implementasi Metode *Support Vector Machine* Untuk Rekomendasi Pemilihan Terapi Dehidrasi Pada Anak.

1.1 Latar belakang

Air adalah zat yang paling dibutuhkan tubuh manusia, karena 70% komponen tubuh manusia terdiri atas air (Dimiyati, 2014). Banyak masyarakat Indonesia yang belum mengetahui kebutuhan air minum per hari, baik pada orang dewasa maupun pada anak. Anak sangat rentan mengalami dehidrasi karena presentase kandungan air tubuh dan luas permukaan tubuh anak dan orang dewasa berbeda.

Komposisi air pada anak-anak sebesar 70-75% dari berat badan (Dimiyati, 2014). Luas permukaan tubuh anak relatif lebih besar sehingga kehilangan air melalui kulit pun menjadi lebih banyak. Fungsi konsentrasi air kemih oleh ginjal anak belum sempurna dan frekuensi nafas anak lebih cepat dari orang dewasa. Kebutuhan air per hari ditentukan oleh usia, berat badan dan aktivitas anak. Misalnya usia 7-9 tahun kebutuhan air per hari adalah 1,6 liter.

Ikatan Dokter Anak Indonesia menyampaikan bahwa air adalah zat yang berperan penting dalam pertumbuhan anak (Dimiyati, 2014). Cairan yang hilang melalui keringat dapat menyebabkan terjadi penurunan kemampuan fisik dan kognitif (Andayani, 2013). Seperti merasa mudah kepanasan, rewel, lesu, mudah menangis, dan marah, tidak dapat belajar dengan baik hingga mengalami masalah kesehatan.

Dehidrasi adalah ketidakseimbangan fisiologi cairan dan elektrolit yang disebabkan oleh kehilangan cairan dan elektrolit dalam jumlah besar karena diare, muntah, dan keringat (Muscari, 2005). Hilangnya cairan yang berlebihan dikhawatirkan akan berdampak buruk pada kesehatan anak. Cairan total tubuh pada anak terdiri atas cairan dan elektrolit yang didistribusikan di antara *kompartemen* cairan *intraselular* dan *ekstraselular*. Cairan *intraselular* (CIS) mencakup seluruh cairan di dalam dinding sel, elektrolit utama yang menyusun CIS adalah kalium. Cairan *ekstraselular* (CES) mencakup semua cairan yang berada di luar dinding sel, elektrolit utama yang menyusun CES adalah natrium.

Salah satu penyebab dehidrasi adalah diare. Penyakit diare masih merupakan masalah kesehatan masyarakat di negara berkembang seperti di Indonesia, karena morbiditas dan mortalitasnya yang masih tinggi. Survei morbiditas yang dilakukan oleh Sub Direktorat Diare, Departemen Kesehatan dari tahun 2000 s/d 2010 terlihat kecenderungan insiden naik. Pada tahun 2000 *Insiden Rate* (IR) penyakit Diare 301/ 1000 penduduk, tahun 2003 naik menjadi 374 /1000 penduduk, tahun 2006 naik menjadi 423 /1000 penduduk dan tahun 2010 menjadi 411/1000 penduduk (Depkes, 2014). RSUD Dr. Soegiri mencatat jumlah anak yang terserang diare pada 3 bulan terakhir ini yaitu September 24 pasien, Oktober 43 pasien, dan

November 36 pasien. Keterbatasan tenaga medis yang hanya berjumlah 18% dari total pasien, menjadi kendala untuk menangani pasien dehidrasi secara tepat dan cepat.

Penanganan yang tepat dan cepat terhadap penderita dehidrasi bisa menyelamatkan jiwa penderita. Dalam pemilihan terapi dehidrasi ada beberapa faktor yang harus diperhatikan, hal ini menyulitkan tenaga medis dalam menentukan terapi dehidrasi mana yang sesuai dengan keadaan pasien. Untuk mengidentifikasi terapi dehidrasi yang sesuai dengan kondisi pasien tidak mudah, karena faktor penyebabnya susah dideteksi dan dosis terapi yang dipilih harus sesuai dengan aturan-aturan yang ada. Ada tiga jenis terapi dehidrasi yaitu, terapi untuk dehidrasi ringan, dehidrasi sedang dan dehidrasi berat (Muscari, 2005).

Saat ini banyak dikembangkan teknologi dalam bidang kesehatan yang ditujukan untuk membantu tenaga medis. Salah satunya adalah pembuatan sistem pendukung keputusan untuk prediksi intubasi trakea (Qing Yan, 2009). Namun sistem tersebut masih terbatas, karena hanya digunakan untuk mengklasifikasikan pasien dalam 2 kelas, yaitu *Easy Tracheal Intubation* (ETI) dan *Difficult Tracheal Intubation* (DTI).

Berdasarkan latar belakang yang telah diuraikan di atas, penulis mencoba membangun sebuah sistem berbasis komputer untuk merekomendasikan terapi dehidrasi sesuai dengan kebutuhan pasien. Sistem pendukung keputusan dapat menjadi alat bantu untuk mengambil keputusan dan biasanya ditujukan untuk keputusan yang dipengaruhi oleh faktor-faktor yang semi terstruktur atau kurang terstruktur (Efraim, 2005). Pada penelitian sebelumnya metode *Support Vector Machine* digunakan untuk sistem pendukung keputusan pemilihan prediksi intubasi trakea dengan nilai rata-rata akurasi 90.53% (Qing Yan, 2009).

1.2 Rumusan masalah

Berdasarkan latar belakang yang telah dikemukakan di atas, dapat dibuat rumusan masalah sebagai berikut:

1. Bagaimana implementasi metode *Support Vector Machine* untuk rekomendasi pemilihan terapi dehidrasi pada anak.
2. Bagaimana tingkat akurasi hasil klasifikasi dengan metode *Support Vector Machine*.

1.3 Tujuan

Berdasarkan dari perumusan masalah, maka tujuan yang ingin dicapai dalam penelitian ini adalah :

1. Mengimplementasikan metode *Support Vector Machine* untuk rekomendasi pemilihan terapi dehidrasi pada anak.
2. Mengukur tingkat akurasi hasil implementasi metode *Support Vector Machine*.

1.4 Manfaat

Harapannya dalam penelitian ini dapat memberikan manfaat bagi semua pihak yang terkait. Adapun manfaat yang diperoleh dari pengimplementasian metode *Support Vector Machine* ini adalah:

1. Membantu tenaga medis untuk menentukan terapi dehidrasi yang sesuai dengan kebutuhan.
2. Memberikan rekomendasi terapi dehidrasi yang tepat dan cepat sehingga mengoptimalkan kinerja tenaga medis karena menyediakan cara perhitungan dan keluaran perhitungan yang siap pakai.
3. Memberikan panduan kepada tenaga medis dalam pemilihan terapi dehidrasi yang tepat agar dapat melakukan penanganan yang tepat dan cepat.
4. Membantu pemerintah dalam menurunkan angka kematian karena dehidrasi.

1.5 Batasan masalah

Batasan masalah yang akan dijadikan sebagai pedoman dalam pelaksanaan penelitian ini adalah sebagai berikut:

1. Data yang digunakan pada skripsi ini adalah 150 data dari Ruang Anggrek RSUD Dr. Soegiri Lamongan, Jawa Timur.
2. Gejala yang digunakan sebagai salah satu kriteria penentuan jenis terapi dehidrasi pada anak adalah keadaan umum, nadi, pernapasan/RR, ubun-ubun, mata, elastisitas kulit, GCS, status gizi, dan suhu.
3. Menggunakan metode *Support Vector Machine* dengan *kernel Radial Basis Function* (RBF).
4. Hasil implementasi metode SVM digunakan untuk menentukan pasien yang tergolong pada terapi dehidrasi tipe A, B dan C.

1.6 Sistematika pembahasan

Sistematika penulisan penelitian ditunjukkan untuk memberikan gambaran dan uraian dari penyusun laporan skripsi secara garis besar yang meliputi beberapa bab sebagai berikut :

BAB I : PENDAHULUAN

Menguraikan mengenai latar belakang, rumusan masalah, tujuan, manfaat dan batasan masalah, sistematika pembahasan.

BAB II : LANDASAN KEPUSTAKAAN

Menguraikan tentang kajian pustaka, dasar teori dan referensi yang mendasari pembuatan sistem Implementasi Metode *Support Vector Machine* Untuk Rekomendasi Pemilihan Terapi Dehidrasi Pada Anak.

BAB III : METODE PENELITIAN DAN PERANCANGAN

Menguraikan tentang metode dan langkah kerja yang dilakukan dalam penulisan laporan skripsi yang terdiri dari studi literatur, metode pengambilan data, analisis kebutuhan, perancangan sistem, implementasi pengujian dan analisis serta pengambilan kesimpulan.

BAB IV : IMPLEMENTASI

Membahas implementasi dari sistem Implementasi *Metode Support Vector Machine* Untuk Rekomendasi Pemilihan Terapi Dehidrasi Pada Anak yang sesuai dengan perancangan sistem yang telah dibuat.

BAB V : PENGUJIAN DAN ANALISIS

Memuat hasil pengujian dan analisis terhadap sistem rekomendasi yang telah direalisasikan.

BAB VI : PENUTUP

Memuat kesimpulan yang diperoleh dari pembuatan dan pengujian perangkat lunak yang dikembangkan dalam laporan skripsi ini serta saran-saran untuk pengembangan lebih lanjut.



BAB 2 LANDASAN KEPUSTAKAAN

Bab ini akan menguraikan tentang tinjauan pustaka dan dasar teori yang mendukung penelitian ini, sumbernya dari beberapa refrensi diantaranya kajian pustaka, dan teori Dehidrasi, Klasifikasi, Normalisasi Data, *Support Vector Machine* (SVM), *Multiclass SVM*, Metode *Sequential Training* pada SVM, Evaluasi.

2.1 Tinjauan Pustaka

Pada bab tinjauan pustaka, akan dijelaskan beberapa penelitian sebelumnya yang digunakan sebagai referensi dalam pengerjaan penelitian ini. Berikut adalah penjelasan tentang penelitian sebelumnya yang ditunjukkan pada Tabel 2.1.

Tabel 2.1 Tinjauan pustaka

No.	Judul	Object	Metode	Output
		Input dan Parameter	Proses	Hasil Penelitian
1.	<i>SVM-Based Decision Support System for Clinic Aided Tracheal Intubation Prediction with Multiple Features</i> (Qing Yan, 2009).	- Umur. - Jenis kelamin. - Berat badan. - <i>Bigonial distance</i>. - <i>Mandibular angle</i>. - <i>Mandibular length</i>. - <i>Neck circumference</i> - <i>Neck length</i>. - <i>Thyromental distance</i>. - <i>Extension of atlantoaxial joint</i>. - <i>Interincisor gap</i>. - <i>Modified mallampati test</i>. - <i>Upper lip bite test</i>.	SVM - Inisialisasi parameter SVM. - Menghitung <i>kernel</i> SVM. - Proses <i>training</i> SVM. - Proses <i>testing</i> SVM.	- <i>Easy tracheal intubation</i> (ETI). - <i>Difficult tracheal intubation</i> (DTI). - 90.53%
2.	Analisis Diagnosis Pasien Kanker Payudara Menggunakan Regresi Logistik dan <i>Support Vector Machine</i> (SVM) Berdasarkan Hasil Mamografi (Novianti, 2012).	- <i>Intermediate findings</i>. - <i>BIRADS category</i>. - <i>Suspicious for Malignancy</i>. - Usia. - Letak <i>abnormal</i>.	Regresi-SVM - Melakukan transformasi data sesuai dengan format <i>software</i> SVM yang akan digunakan. - Menentukan fungsi <i>kernel</i> untuk permodelan. - Menentukan nilai-nilai parameter <i>kernel</i> dan parameter <i>cost</i> untuk optimasi.	- <i>Benign</i> (jinak). - <i>Malignant</i>(ganas) - 91.9%

Tabel 2.1 Tinjauan pustaka (lanjutan)

No.	Judul	Object	Metode	Output
		Input dan Parameter	Proses	Hasil Penelitian
			- Memilih nilai parameter terbaik untuk optimasi data <i>training</i> untuk klasifikasi data <i>testing</i>. - Menghitung ketepatan klasifikasi.	
3.	Perbandingan Klasifikasi Tingkat Keganasan <i>Breast Cancer</i> Dengan Menggunakan Regresi Logistik Ordinal dan <i>Support Vector Machine</i> (SVM) (rachman, 2012).	- Ukuran tumor. - <i>Nodus</i>. - <i>Cemotheraphi</i>. - <i>Malignant/Benign</i>. - Letak kanker. - Usia pasien.	Regresi-SVM - Melakukan transformasi data sesuai dengan metode <i>multiclass</i> SVM. - Menentukan fungsi pemisah dengan metode <i>one against all</i>. - Menentukan nilai-nilai parameter. - Memilih nilai parameter terbaik. - Menghitung nilai ketepatan klasifikasi terbaik.	- Satdium I - Stadium II - Stadium III - 98.11%
4.	Implementasi Metode <i>Support Vector Machine</i> Untuk Rekomendasi Pemilihan Terapi Dehidrasi Pada Anak [usulan].	- Keadaan umum. - Nadi. - Pernapasan/RR. - Ubun-ubun. - Mata. - Elastisitas kulit. - <i>Glasgow Coma Scale</i> (GCS). - Status gizi. - Suhu.	SVM - Normalisasi <i>dataset</i>. - Inisialisasi parameter. - Menghitung <i>kernel</i> RBF. - Proses <i>sequential training</i> SVM - Proses <i>testing</i> SVM. - Proses <i>one against all</i>.	Dehidrasi - Ringan. - Sedang - Berat

Sumber: [Penelitian]

Pada Tabel 2.1 tinjauan pustaka disebutkan beberapa *paper* yang digunakan sebagai referensi penelitian ini. Diantaranya adalah “SVM-Based Decision Support System for Clinic Aided Tracheal Intubation Prediction with Multiple Features” Qing Yan dkk telah melakukan penelitian menggunakan metode *Support Vector Machine* (SVM) untuk memprediksi pemasangan intubasi pada trakea. Objek penelitian ini adalah pasien yang mengalami berbagai macam penyakit dan perlu dipasang alat bantu pernafasan. Metode *kernel* yang digunakan adalah *linier*, dengan 13 parameter *input* yaitu umur, jenis kelamin, berat badan, *bigonial distance* (jarak antara *gonion* kiri dan *gonion* kanan), *mandibular angle* (sudut wilayah segitiga dari *gonion* kiri sampai *gonion* kanan melalui *gnathion*), *mandibular length* (jarak antara *gonion* dan *gnathion*), *neck circumference* (diukur sekitar pertengahan leher), *neck length* (jarak antara *superior thyroid* dengan *head hypsokinesis*), *thyromental distance* (diukur dari *thyroid* ke ujung rahang dengan kepala diangkat), *extension of atlantoaxial joint*, *interincisor gap* (ruang antara gigi

seri bawah dan atas), *modified mallampati test*, *upper lip bite test*. Pada penelitian tersebut dapat digunakan untuk diagnosa intubasi trakea dengan akurasi 90.53% (Qing Yan, 2009).

Pada *paper* selanjutnya, membahas tentang “Analisis Diagnosis Pasien Kanker Payudara Menggunakan *Regresi Logistik* dan *Support Vector Machine* (SVM) Berdasarkan Hasil Mamografi” yang memiliki nilai rata-rata akurasi sebesar 91.9%. Data yang digunakan sebanyak 267 data yaitu 100 data pasien kanker payudara *benign* dan 167 data pasien kanker payudara *malignant* dengan 5 fitur. Pada hasil pengujian, didapatkan rata-rata akurasi metode SVM lebih besar daripada metode regresi logistik (Novianti, 2012).

Paper yang terakhir adalah penelitian tentang “Perbandingan Klasifikasi Tingkat Keganasan *Breast Cancer* Dengan Menggunakan Regresi Logistik Ordinal dan *Support Vector Machine* (SVM)” menghasilkan nilai rata-rata akurasi sebesar 98.11%. Data yang digunakan sebanyak 178 data dengan 6 fitur. SVM memiliki ketepatan klasifikasi lebih besar dari regresi logistik ordinal, dimana hasil akurasi klasifikasi tingkat keganasan *breast cancer* dengan metode regresi sebesar 56.6% (rachman, 2012).

Pada *paper* pertama membahas tentang pemilihan jenis intubasi trakea, memisahkan 2 *class*. Permasalahan yang ada pada *paper* pertama adalah, ketika data yang dipisahkan adalah 3 *class*. *Paper* ketiga penelitian tentang perbandingan klasifikasi 3 *class* menggunakan metode regresi logistik ordinal dengan SVM menggunakan fungsi *kernel* RBF dan *polynomial*. Hasil akurasi menunjukkan metode SVM menghasilkan rata-rata akurasi yang lebih tinggi dari metode regresi yaitu 98.11%. Pada penelitian ini membahas tentang Implementasi Metode *Support Vector Machine* yang diterapkan pada kasus pemilihan rekomendasi terapi dehidrasi. Data dari hasil observasi berupa data *non-linier* dan terdapat 3 *class*, permasalahan ini diatasi dengan perhitungan fungsi *kernel* dan fungsi *multiclass one against all*. *Kernel* yang digunakan pada penelitian ini adalah RBF.

2.2 Dehidrasi

Dehidrasi adalah kehilangan cairan dari jaringan tubuh yang berlebihan. Dehidrasi merupakan gangguan yang umum terjadi pada bayi dan anak-anak ketika haluaran cairan total melebihi asupan cairan total (Muscari, 2005).

2.2.1 Jenis Dehidrasi

Dehidrasi dapat digolongkan berdasarkan derajatnya yaitu:

1. Dehidrasi ringan dicirikan dengan kehilangan 5% dari berat badan sebelum sakit.
2. Dehidrasi sedang dicirikan dengan kehilangan 5% sampai 10% dari berat badan sebelum sakit.
3. Dehidrasi berat dicirikan dengan kehilangan lebih dari 10% berat badan sebelum sakit.

2.2.2 Etiologi Dehidrasi

Dehidrasi dapat disebabkan oleh kehilangan air yang tidak disadari pada kulit dan saluran pernapasan, peningkatan eksresi cairan pada ginjal dan *gastrointestinal* (GI), atau penurunan asupan cairan.

Kemungkinan penyebab dehidrasi antara lain:

1. Muntah dan diare yang berlebihan
2. Asupan cairan yang tidak cukup
3. Ketoasidosis diabetik
4. Luka bakar berat
5. Demam tinggi berkepanjangan
6. Hiperventilasi

2.2.3 Tanda-Tanda Dehidrasi

Tanda-tanda dehidrasi bergantung pada derajat dehidrasi.

1. Haus
2. Keletihan
3. Penurunan berat badan
4. Membran mukosa kering
5. Penurunan atau hilangnya produksi air mata
6. Turgor kulit tidak elastis dan waktu pengisian kembali (*capillary refill*) kapiler meningkat
7. Mata cekung
8. Depresi *fontanel*
9. Penurunan haluaran urine
10. *Takikardia* adalah denyut jantung yang lebih cepat daripada denyut jantung normal.
11. *Takipnea* adalah pernapasan abnormal cepat dan dangkal, biasanya didefinisikan lebih dari 60 hembusan per menit.
12. Penurunan tekanan darah
13. Rasa haus berlebihan

Tabel 2.2 Tanda dan gejala

No.	Tanda dan Gejala	Dehidrasi Ringan	Dehidrasi Sedang	Dehidrasi Berat
1	Keadaan Umum	Cukup	sedang	lemah
2	Nadi	$100 < x \leq 120$ /menit	$x > 120$	$x \leq 100$
3	Pernapasan/RR	$20 < x \leq 24$ /menit	$x > 24$	$x < 24$
4	Uzun-uzun	tidak cekung	tidak cekung	cekung
5	Mata	tidak cowong	tidak cowong	cowong
6	Elastisitas kulit	kembali dalam 2 detik	kembali > 2 detik(lambat)	kembali > 2 detik(sangat lambat)
7	<i>Glasgow Coma Scale</i> (GCS)	456 <i>composmetis</i>	345 <i>apatis</i>	111 <i>somnolen</i>
8	Status Gizi	cukup	kurang	buruk
9	Suhu	$36,4 < x \leq 37,4$	$x \leq 36,4$	$x \leq 35,4$

Sumber: [wawancara]

2.3 Klasifikasi

Menurut Han dan Kamber klasifikasi adalah proses untuk menemukan kumpulan model atau fungsi yang menggambarkan dan membedakan kelas data atau konsep, agar model tersebut dapat digunakan untuk memprediksi kelas objek yang belum diketahui. Klasifikasi merupakan suatu cara untuk memberikan klasifikasi pada data baru dengan menggunakan sejumlah aturan yang telah dihasilkan dari manipulasi data yang telah diklasifikasi. Teknik ini menggunakan *supervised induction*, yang memanfaatkan kumpulan pengujian dari record data sebelumnya yang diklasifikasi untuk menentukan kelas data baru (Kusrini, 2007).

Teknik klasifikasi merupakan pendekatan secara sistematis untuk membuat model klasifikasi dari inputan *dataset*. Adapun beberapa contoh teknik klasifikasi diantaranya adalah *Naive Bayes*, *Decision Tree based Methods*, *Rule-based Methods*, *Support Vector Mechines*, *Neural Network*, *K-Nearest Neighbor (K-NN)*, banyak juga yang dikembangkan dengan mengkombinasi 2 atau lebih teknik klasifikasi seperti *Fuzzy K-Nearest Neighbor*, *Adaptive K-Nearest Neighbor* dan lain sebagainya.

2.4 Normalisasi Data

Normalisasi data adalah teknik untuk mengelompokkan data atau nilai atribut sesuai dengan skala tertentu. Normalisasi bertujuan untuk menghindari beberapa atribut yang memiliki nilai besar mempengaruhi atribut lain yang diukur dengan skala yang lebih kecil (Yessivirna, 2013).

Metode aktivasi *sigmoid biner* (s') untuk menyelesaikan permasalahan dunia nyata yang kompleks dan biasanya bersifat *non-linier* dengan mentransformasikan data pada interval $[0.1, 0.9]$ karena fungsi ini tidak pernah mencapai nilai 0 atau 1. Persamaan 2.1 merupakan rumus normalisasi *sigmoid biner*. Selain aktivasi *sigmoid biner*, ada aktivasi untuk normalisasi data interval $[-1, 1]$ disebut fungsi *sigmoid bipolar* (sp'). Persamaan 2.2 merupakan metode aktivasi *sigmoid bipolar* (Putri, 2015).

$$s' = \frac{0.8(s - a)}{b - a} + 0.1 \quad 2.1$$

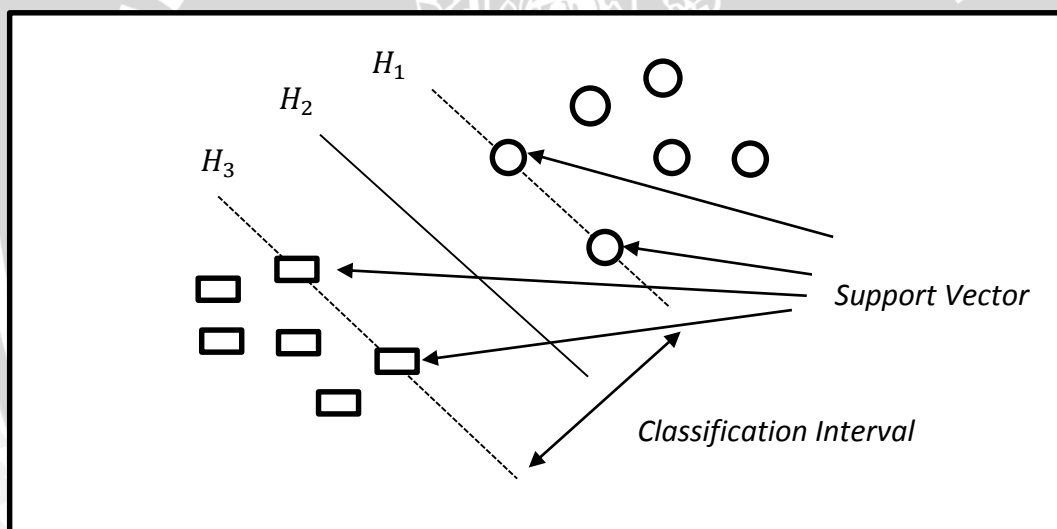
$$sp' = \frac{1 - e^{-s}}{1 + e^{-s}} \quad 2.2$$

Dimana s' dan sp' adalah nilai baru dengan rentang nilai antara 0 sampai 1 untuk *sigmoid biner*, tetapi lebih baik ditransformasikan ke interval yang lebih kecil yaitu 0.1 sampai 0.9, hal ini dilakukan untuk menghindari nilai 0 dan rentang -1 sampai 1 untuk *sigmoid bipolar*, s adalah nilai awal atribut yang akan dinormalisasikan, a adalah nilai minimum atribut dan b adalah nilai maksimum atribut dan e adalah nilai eksponensial.

2.5 Support Vector Machine (SVM)

SVM (*Support Vector Machine*) merupakan metode *learning machine* yang bekerja atas prinsip *Structural Risk Minimization* (SRM) dengan tujuan menemukan *hyperplane* terbaik yang memisahkan dua buah kelas pada *input space* (rachman, 2012). SVM adalah seperangkat metode pembelajaran mesin teori statistik (Kartal, 2013). Teknik SVM dilakukan secara analisis berdasarkan hitungan matematis sehingga dapat membantu dalam pemecahan suatu masalah dengan menentukan suatu *hyperplane* terbaik. *Hyperplane* adalah suatu bidang pemisah yang terletak ditengah antara dua set objek dari dua kelas.

Proses generalisasi dalam SVM dapat digunakan untuk menentukan pola tertentu berdasarkan data *testing* dan data *training*. SVM merupakan metode yang sangat cepat dalam menyelesaikan problem karena strukturnya yang sederhana. Metode SVM ini mempunyai kemampuan menemukan fungsi pemisah (*classifier*) yang optimal (Damayanti, 2010). SVM dapat digunakan untuk kasus yang hanya melakukan klasifikasi dua kelas dan selalu memberikan hasil yang sama dalam setiap kali melakukan proses *running*. Pada Gambar 2.1 dijelaskan SVM dengan dua dimensi.



Gambar 2.1 Ide dasar penerapan SVM

Sumber: (Yi, 2010)

Simbol bulat dan persegi masing-masing mewakili dua jenis sampel yaitu H_1 dan H_2 yang dipisahkan oleh *hyperplane* (H). Jarak antara dua jenis sampel yang dipisahkan oleh *hyperplane* adalah *classification interval* atau margin. Sedangkan vector terdekat yang memisahkan *hyperplane* disebut dengan *support vector*.

Berdasarkan karakteristiknya metode SVM dibagi menjadi dua, yaitu SVM *linier* dan *non-linier* (Yi, 2010). SVM *linier* adalah metode SVM yang memisahkan kedua *class* oleh *hyperplane* yaitu dengan *soft margin*. Sedangkan SVM *non-linier* adalah metode SVM yang menerapkan fungsi *kernel trick* pada ruang berdimensi tinggi (rachman, 2012).

2.5.1 Support Vector Machine Linier

SVM linier dibagi menjadi dua, yaitu *separable* dan *non-separable* (rachman, 2012). SVM melakukan klasifikasi himpunan data *training* berpasangan dari dua kelas (Novianti, 2012). Klasifikasi linier adalah klasifikasi data yang memisahkan data secara linier. Misal diketahui $x_i = \{x_1, \dots, x_n\}$ adalah titik pada dataset, dan $y_i = y \in \{+1, -1\}$ adalah *class* pada data x_i . Bidang pemisah terbaik adalah yang dapat memisahkan semua data set yang sesuai dengan *class* dan memiliki margin paling besar. Bidang pemisah pertama membatasi *class* pertama, sedangkan bidang pemisah kedua membatasi *class* kedua, sehingga diperoleh Persamaan 2.3 dan Persamaan 2.4.

$$(w \cdot x_i + b) \geq +1 \text{ untuk } y_i = +1 \quad (2.3)$$

$$(w \cdot x_i + b) \leq -1 \text{ untuk } y_i = -1 \quad (2.4)$$

Dimana variabel w merupakan bidang normal antara bidang pemisah terhadap pusat koordinat, dan variabel b adalah posisi bidang relatif terhadap pusat koordinat. Jarak antara *hyperplane* dan titik terdekat untuk menemukan margin terbesar adalah dengan memaksimalkan fungsi $\frac{1}{||w||}$ atau dengan meminimalkan $||w||^2$. Untuk bidang pemisah atau *hyperplane* kedua *class* dijelaskan pada Persamaan 2.5.

$$y_i(w \cdot x_i + b) \geq 1, i = 1, 2, \dots, n \quad (2.5)$$

Persamaan diatas dapat dirumuskan sebagai *Quadratic Programming* dengan mencari titik minimal. *Hyperplane* terbaik dengan margin terbesar dapat dirumuskan menjadi masalah optimasi *constraint* pada Persamaan 2.6.

$$\text{Min} = \frac{1}{2} ||w||^2 \quad (2.6)$$

Dimana syarat $y_i(w \cdot x_i + b) \geq 1$. Persamaan 2.6 digunakan untuk menyelesaikan permasalahan optimasi yang dapat diselesaikan dengan menggunakan fungsi *Lagrange*. Fungsi *Lagrange* yang digunakan adalah *Lagrange multiplier* dijelaskan pada Persamaan 2.7.

$$\text{Min } L(w, b, \alpha) = \frac{1}{2} ||w||^2 - \sum_{i=1}^n \alpha_i [y_i(w \cdot x_i + b) - 1] \quad (2.7)$$

Dimana α_i merupakan nilai bobot dari setiap titik data. Notasi ini bernilai nol atau positif ($\alpha_i \geq 0$), dengan meminimalkan nilai L terhadap w dan b , maka untuk menghitung w dan b dapat menggunakan Persamaan 2.8.

$$w = \sum_{i=1}^n \alpha_i y_i x_i \quad \text{dan} \quad b = -\frac{1}{2} (w \cdot x^+ + w \cdot x^-) \quad (2.8)$$

Dimana x^+ merupakan nilai dari salah satu *support vector* dari *class positive* dan x^- merupakan nilai dari salah satu *support vector* dari *class negative*. Persamaan 2.8 dapat dimodifikasi menjadi fungsi maksimal untuk mencari *hyperplane* terbaik. Dapat dilihat pada Persamaan 2.9.

$$\max \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{j=1}^n \alpha_i \alpha_j y_i y_j x_i x_j \quad (2.9)$$

Dengan batasan

$$\alpha_i \geq 0 (i = 1, 2, \dots, n) \text{ dan } \sum_{i,j=1}^n \alpha_i y_i = 0$$

Hasil dari perhitungan pada Persamaan 2.9 dapat diperoleh nilai α_i bernilai positif, dan selanjutnya akan digunakan untuk menemukan w . Pada setiap data *training*, terdapat nilai α_i . *Support vector* adalah data *training* yang mempunyai nilai $\alpha_i > 0$, sedangkan sisanya memiliki nilai $\alpha_i = 0$. Jadi, fungsi keputusan yang dihasilkan hanya dipengaruhi oleh *support vector*. Untuk mendapatkan klasifikasi yang optimal data *testing* x dapat digunakan Persamaan 2.10. Hasil klasifikasi dapat dilihat pada Persamaan 2.11.

$$f(x) = (w \cdot x + b) \quad \text{atau} \quad f(x) = \sum_{i=1}^m \alpha_i y_i x_i \cdot x + b \quad (2.10)$$

$$\text{Fungsi Klasifikasi} = \text{sign}(f(x)) \quad (2.11)$$

Keterangan variabel m adalah jumlah *support vector* atau titik data yang memiliki $\alpha_i > 0$, dan notasi x merupakan data yang akan diklasifikasikan.

2.5.2 Support Vector Machine Non-Linier

Permasalahan *non-linier* dapat diselesaikan dengan mentransformasikan ke dalam permasalahan *linier* dengan menggunakan dimensi yang tinggi (Kartal, 2013). Fungsi *kernel* dapat didefinisikan menjadi *input kernel trick*. *Kernel trick* adalah bagian dari pembelajaran dalam metode SVM, dimana untuk mengetahui fungsi *kernel* tanpa harus mengetahui wujud dari fungsi *non-linier*. Untuk menghitung *kernel trick* ditunjukkan pada Persamaan 2.12.

$$K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j) \quad (2.12)$$

Notasi x_i, x_j dengan *dot product* dapat diganti dengan simbol K . Model *quadratic problem* dapat digunakan untuk mendapatkan *hyperplane* yang optimal dengan menggunakan persamaan klasifikasi data *non-linier*. Dapat ditunjukkan pada Persamaan 2.13.

$$\min \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \cdot y_i \cdot K(x_i, x_j) \quad (2.13)$$

Dengan batasan berikut $\sum_{i=0}^n \alpha_i y_i = 0, 0 \leq \alpha_i \leq C, \text{ dan } i = 1, 2, \dots, n$

Nilai $K(x, y)$ merupakan fungsi *kernel* yang menunjukkan pemetaan *non-linier* pada *feature space* (rachman, 2012). Pada Persamaan 2.14 merupakan fungsi keputusan klasifikasi optimal.

$$f(x) = \sum_{i=1}^m \alpha_i \cdot y_i \cdot K(x_i, x_j) + b \quad (2.14)$$

Berikut adalah beberapa fungsi *kernel* (Novianti, 2012) :

- Fungsi *Linier* ditunjukkan pada Persamaan 2.15.

$$K(x, y) = x \cdot y \quad (2.15)$$

- Fungsi *Polynomial* ditunjukkan pada Persamaan 2.16.

$$K(x, y) = (x \cdot y)^d \quad (2.16)$$

- Fungsi *Gaussian* RBF ditunjukkan pada Persamaan 2.17.

$$K(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right) \quad (2.17)$$

- Fungsi *Sigmoid* didefinisikan pada Persamaan 2.18.

$$K(x, y) = \tanh(\sigma(x, y) + C) \quad (2.18)$$

Kernel linier adalah *kernel* yang digunakan ketika data yang diklasifikasikan dipisahkan oleh sebuah garis atau *hyperline*. Sedangkan *kernel non-linier* adalah *kernel* yang digunakan ketika data yang diklasifikasikan dipisahkan dengan garis lengkung atau sebuah bidang pada ruang yang berdimensi tinggi.

2.6 Multiclass SVM

SVM adalah sebuah metode untuk mengatasi masalah klasifikasi dua kelas. Untuk menyelesaikan masalah *multi-class problem*, dikembangkan dua metode pengambilan keputusan yaitu klasifikasi *multi-class* dengan pendekatan mengkombinasikan beberapa *binary classifier* atau menggabungkan semua data yang terdiri dari beberapa kelas ke dalam permasalahan optimasi (Sembiring, 2007).

Biasanya yang digunakan adalah pengkombinasian beberapa *binary classifier*. Metode-metode yang dimaksud adalah *one-against-all*, *one-against-one*, dan *Directed Acyclic Graph Support Vector Machine* (DAGSVM).

2.6.1 Metode One-Against-All

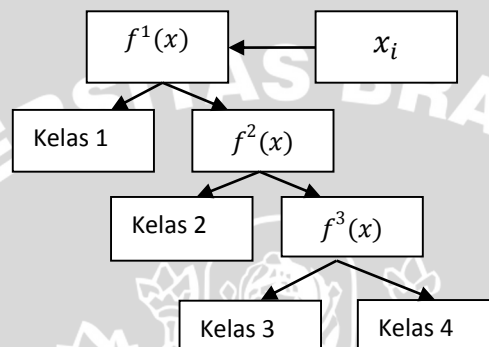
Metode ini dikembangkan berdasarkan sejumlah k SVM biner (k adalah jumlah kelas) (Hsu, 2002). Misalkan terdapat permasalahan klasifikasi dengan 4 buah jumlah kelas, digunakan 3 buah SVM biner sebagai pelatihan seperti yang dapat

terlihat pada Tabel 2.2 dan penggunaannya pada pengklasifikasian data baru dapat dilihat pada Gambar 2.2.

Tabel 2.3 Metode *one-against-all* dengan 4 kelas

$y_i = 1$	$y_i = -1$	Hipotesis
Kelas 1	Bukan kelas 1	$f^1(x) = (w^1)x + b^1$
Kelas 2	Bukan kelas 2	$f^2(x) = (w^2)x + b^2$
Kelas 3	Kelas 4	$f^3(x) = (w^3)x + b^3$

Sumber: (Sembiring, 2007)



Gambar 2.2 Contoh klasifikasi *one-against-all* untuk 4 kelas

Sumber: (Sembiring, 2007)

2.6.2 Metode *One-Against-One*

Metode ini dikembangkan berdasarkan rumus $\frac{k(k-1)}{2}$ SVM biner (k adalah jumlah kelas) (Hsu, 2002). Misalkan terdapat permasalahan klasifikasi dengan 4 buah jumlah kelas, digunakan 6 buah SVM *biner* sebagai pelatihan seperti yang dapat terlihat pada Tabel 2.4 dan penggunaannya pada pengklasifikasian data baru dapat dilihat pada Gambar 2.3.

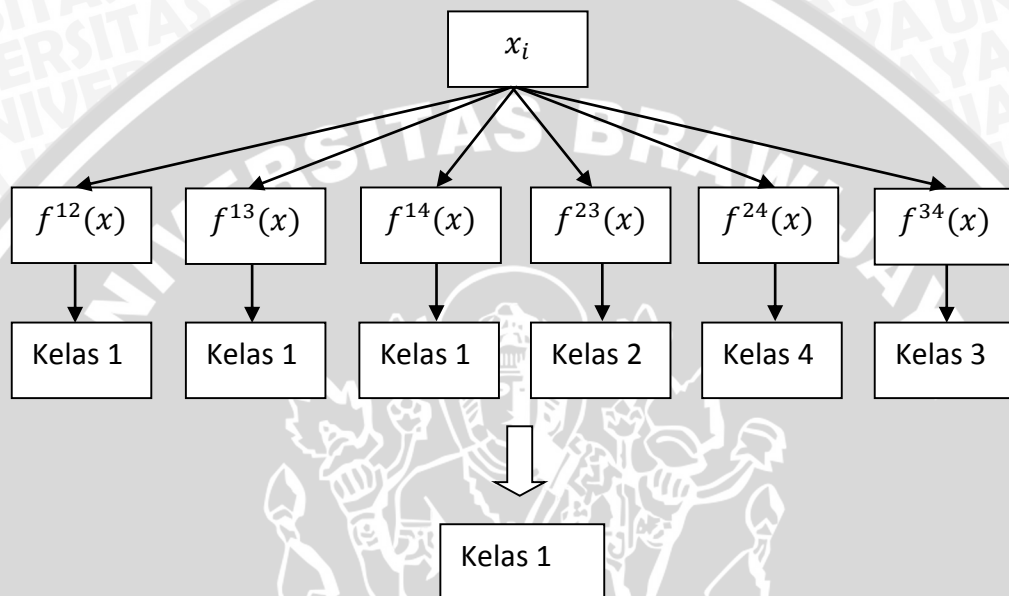
Tabel 2.4 Contoh 6 SVM *biner* dengan metode *one-against-one*

$y_i = 1$	$y_i = -1$	Hipotesis
Kelas 1	Bukan kelas 2	$f^{12}(x) = (w^{12})x + b^{12}$
Kelas 1	Bukan kelas 3	$f^{13}(x) = (w^{13})x + b^{13}$
Kelas 1	Bukan kelas 4	$f^{14}(x) = (w^{14})x + b^{14}$
Kelas 2	Bukan kelas 3	$f^{23}(x) = (w^{23})x + b^{23}$
Kelas 2	Bukan kelas 4	$f^{24}(x) = (w^{24})x + b^{24}$
Kelas 3	Bukan kelas 4	$f^{34}(x) = (w^{34})x + b^{34}$

Sumber: (Sembiring, 2007)

Pada Tabel 2.4 dijelaskan pada fungsi f^{12} sampai f^{14} digunakan untuk memisahkan kelas 1, dimana ketika nilai $y_i = 1$ maka hasilnya adalah kelas 1 dan

ketika $y_i = -1$, maka hasilnya adalah bukan kelas dari nilai fungsi pada digit terakhir. Sebagai contoh fungsi f^{12} ketika nilainya positif, maka akan menghasilkan kelas 1 dan nilai negatif mengklasifikasikan pada bukan kelas 2. Untuk fungsi f^{23} dan f^{24} digunakan untuk memisahkan kelas 2, dimana ketika nilai $y_i = 1$ maka hasilnya adalah kelas 2 dan ketika $y_i = -1$, maka hasilnya adalah bukan kelas dari nilai fungsi pada digit terakhir, seperti pada fungsi sebelumnya. Dan untuk fungsi f^{34} seperti fungsi-fungsi sebelumnya. Ketika nilai $y_i = 1$ maka hasil klasifikasi pada kelas 3 dan nilai $y_i = -1$ hasil klasifikasinya bukan kelas 4.



Gambar 2.3 Contoh klasifikasi dengan metode *one-against-one*
Sumber: (Sembiring, 2007)

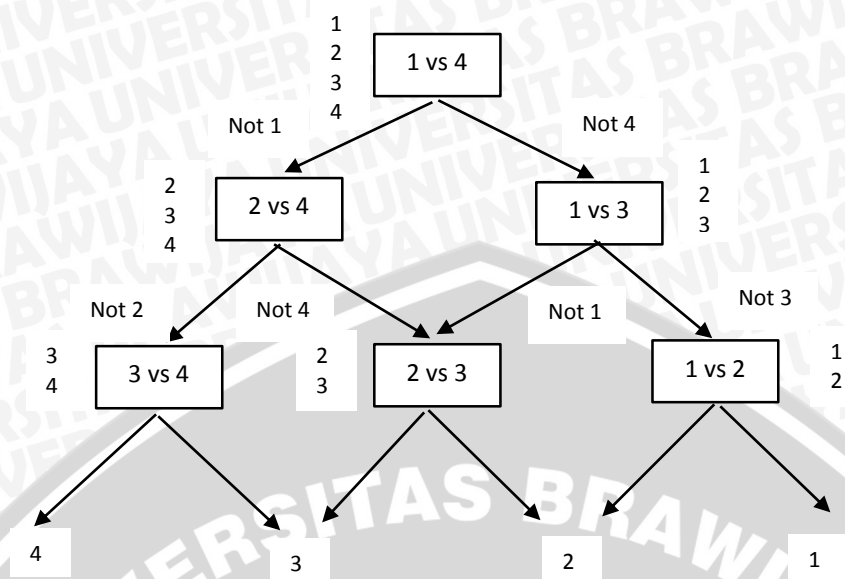
2.6.3 Metode *Directed Acyclic Graph Support Vector Machine* (DAGSVM)

Metode ini dikembangkan sama dengan *one-against-one*, dibangun berdasarkan $\frac{k(k-1)}{2}$ SVM biner (k adalah jumlah kelas) (Hsu, 2002). Tetapi, pada saat pengujian digunakan *binary directed acyclic graph* dapat dilihat pada Gambar 2.4 dan pelatihan seperti yang dapat terlihat pada Tabel 2.5.

Tabel 2.5 Contoh 6 SVM biner dengan metode DAGSVM

$y_i = 1$	$y_i = -1$	Hipotesis
Bukan kelas 2	Bukan kelas 1	$f^{12}(x) = (w^{12})x + b^{12}$
Bukan kelas 3	Bukan kelas 1	$f^{13}(x) = (w^{13})x + b^{13}$
Bukan kelas 4	Bukan kelas 1	$f^{14}(x) = (w^{14})x + b^{14}$
Bukan kelas 3	Bukan kelas 2	$f^{23}(x) = (w^{23})x + b^{23}$
Bukan kelas 4	Bukan kelas 2	$f^{24}(x) = (w^{24})x + b^{24}$
Bukan kelas 4	Bukan kelas 3	$f^{34}(x) = (w^{34})x + b^{34}$

Sumber: (Sembiring, 2007)



Gambar 2.4 Contoh klasifikasi dengan metode DAGSVM
Sumber: (Kurniawati, 2015)

2.7 Metode *Sequential Training* pada SVM

Vijayakumar mengembangkan metode *quadratic problem* untuk menemukan *hyperplane* yang optimal dalam SVM (Vijayakumar, 1999). Untuk mencari nilai, dapat dijelaskan pada tahapan-tahapan berikut:

1. Proses yang dilakukan setelah perhitungan kernel adalah inisialisasi untuk parameter SVM, misalnya nilai $\alpha_i = 0$, nilai $\lambda = 0.5$, $\gamma = 0.01$, $\varepsilon = 0.0001$ dan $C = 1$.
2. Menghitung komponen matriks dengan menggunakan Persamaan 2.19.

$$D_{ij} = y_i y_j (K(x_i, x_j) + \lambda^2) \quad (2.19)$$

Notasi x adalah dari data ke i dan data ke j .

Notasi y adalah kelas dari data ke i dan data ke j .

$K(x_i, x_j)$ adalah fungsi *kernel* yang digunakan.

3. Untuk setiap $i, = 1, 2, \dots, n$ dihitung dengan menggunakan Persamaan 2.20 sampai Persamaan 2.23.

- a. $E_i = \sum_{j=1}^n \alpha_j D_{ij}$ (2.20)

- b. $\delta \alpha_i = \min\{\max[\gamma(1 - E_i), -\alpha_i], C - \alpha_i\}$ (2.21)

$\delta \alpha_i$ adalah variabel tunggal, bukan bentuk perkalian δ dan α_i

- c. $\alpha_i = \alpha_i + \delta \alpha_i$ (memperbarui nilai α_i) (2.22)

Notasi γ adalah parameter untuk mengontrol kecepatan proses *learning* (*learning rate*). Nilai konstanta untuk parameter γ , ditunjukkan pada persamaan 2.23.

$$\gamma = \frac{\text{konstanta } \gamma}{\max_{\{i\}} D_{ii}} \quad (2.23)$$

$\max_{\{i\}} D_{ii}$ adalah nilai maximum yang diperoleh dari nilai diagonal pada matriks. Dengan menggunakan nilai konstanta untuk nilai yang diinputkan oleh *user* dalam pengujian adalah 0.001.

Fungsi $\delta\alpha_i$ adalah fungsi konvergensi untuk memantau perubahan fungsi *Lagrange multiplier*.

Ketika data *training* mencapai nilai *konvergen* maks ($|\delta\alpha_i|$) $< \varepsilon$, dan jika iterasi mencapai nilai yang ditentukan maka iterasi akan dihentikan.

4. Tahapan nomor 3 diulang sampai α mencapai nilai konvergen. Konvergen dapat didefinisikan dari tingkat perubahan pada nilai α .
5. Nilai *Support Vector* diperoleh dari $SV = (\alpha_i > \text{ThresholdSV})$. Nilai *ThresholdSV* ditentukan dari beberapa kali dilakukan uji coba. Nilai *ThresholdSV* tersebut adalah $\text{ThresholdSV} \geq 0$.

2.8 Evaluasi

Evaluasi adalah proses mengukur tingkat keberhasilan atau membandingkan hasil dari proses implementasi dengan standar yang telah ditetapkan. Akurasi adalah pengukuran kedekatan kualitas untuk nilai aktual bahwa kualitas sistem tersebut benar (Istara, 2013). *Classification accuracy* adalah salah satu model evaluasi untuk menghitung ketepatan klasifikasi yang diperoleh. Untuk menghitung *classification accuracy* ditunjukkan pada Persamaan 2.24.

$$\text{Akurasi} = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.24)$$

Classification accuracy ditentukan dengan menggunakan nilai yang terdapat dalam *confusion matrix*. Evaluasi sistem dapat dilakukan dengan cara membuat *confusion matrix*, dimana *confusion matrix* 3x3 bisa dilihat pada Tabel 2.6.

Tabel 2.6 *Confusion matrix* 3x3

Actual Class	Predicted Class		
	Class 1	Class 2	Class 3
Class 1	$C_{1,1}$	$C_{1,2}$	$C_{1,3}$
Class 2	$C_{2,1}$	$C_{2,2}$	$C_{2,3}$
Class 3	$C_{3,1}$	$C_{3,2}$	$C_{3,3}$

Sumber: (Hermaduanti, 2008)

Istilah-istilah dalam proses pengujian untuk klasifikasi:

a. TP (*True Positive*)

Apabila *actual class* dan sistem sama menghasilkan *positive class*, jika:

- *Actual class* "Ringan" sistem juga memutuskan kelas "Ringan".
- *Actual class* "Sedang" sistem juga memutuskan kelas "Sedang".
- *Actual class* "Berat" sistem juga memutuskan kelas "Berat".

Adapun persamaanya dapat dilihat pada Persamaan 2.25

$$TP = C_{1,1} + C_{2,2} + C_{3,3} \quad (2.25)$$

b. TN (*True Negative*)

Apabila *actual class* dan sistem sama menghasilkan *negative class*, jika:

- *Actual class* tidak *class* "Ringan" sistem juga memutuskan *class* bukan *class* "Ringan".
- *Actual class* tidak *class* "Sedang" sistem juga memutuskan *class* bukan *class* "Sedang".
- *Actual class* tidak *class* "Berat" sistem juga memutuskan *class* bukan *class* "Berat".

Adapun persamaanya dapat dilihat pada Persamaan 2.26

$$TN = (C_{1,1} + C_{2,2}) + (C_{1,1} + C_{3,3}) + (C_{2,2} + C_{3,3}) \quad (2.26)$$

c. FP (*False Positive*)

Apabila *actual class negative*, tetapi sistem memutuskan *positive class*, jika:

- *Actual class* bukan *class* "Ringan", tetapi sistem memutuskan *class* "Ringan".
- *Actual class* bukan *class* "Sedang", tetapi sistem memutuskan *class* "Sedang".
- *Actual class* bukan *class* "Berat", tetapi sistem memutuskan *class* "Berat".

Adapun persamaanya dapat dilihat pada Persamaan 2.27

$$FP = (C_{2,1} + C_{3,1}) + (C_{1,2} + C_{3,2}) + (C_{1,3} + C_{2,3}) \quad (2.27)$$

d. FN (*False Negative*)

Apabila *actual class positive*, tetapi sistem memutuskan *negative class*, jika:

- *Actual class* "Ringan", tetapi sistem memutuskan bukan *class* "Ringan".
- *Actual class* "Sedang", tetapi sistem memutuskan bukan *class* "Sedang".
- *Actual class* "Berat", tetapi sistem memutuskan bukan *class* "Berat".

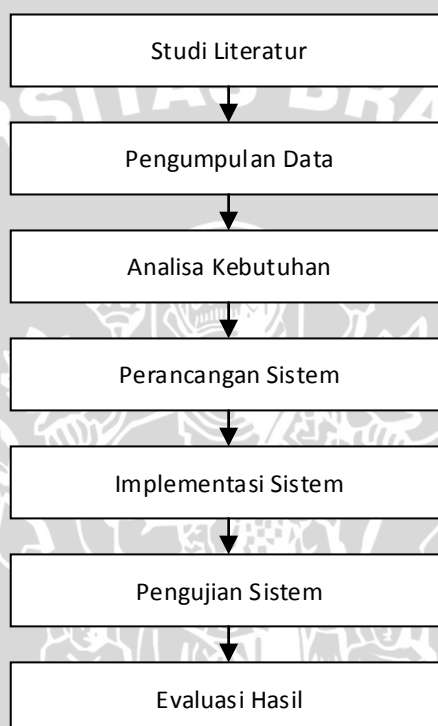
Adapun persamaanya dapat dilihat pada Persamaan 2.28

$$FN = (C_{1,2} + C_{1,3}) + (C_{2,1} + C_{2,3}) + (C_{3,1} + C_{3,2}) \quad (2.28)$$

BAB 3 METODOLOGI PENELITIAN DAN PERANCANGAN

Pada bab metodologi dijelaskan langkah-langkah yang akan ditempuh dalam penyusunan skripsi Implementasi Metode *Support Vector Machine* Untuk Rekomendasi Pemilihan Terapi Dehidrasi Pada Anak yaitu perancangan, implementasi dan pengujian dari sistem yang akan dibuat.

Tahap penelitian disini akan membahas langkah-langkah Implementasi Metode *Support Vector Machine* Untuk Rekomendasi Pemilihan Terapi Dehidrasi Pada Anak. Langkah-langkah dalam pengerjaan penelitian ini dapat dilihat pada Gambar 3.1.



Gambar 3.1 Diagram alir penelitian
Sumber: [Perancangan]

3.1 Studi Literatur

Studi literatur digunakan dalam mempelajari dasar-dasar teori dan literatur terkait dengan penelitian yang akan dilakukan. Pada penelitian ini data penyakit dehidrasi diolah menggunakan algoritma *Support Vector Machine* (SVM). Teori dan literatur tersebut diperoleh dari buku, jurnal, *ebook*, internet, penelitian sebelumnya, dan sumber pustaka lain yang dapat dipertanggungjawabkan.

3.2 Pengumpulan Data

Pengumpulan data pada penelitian ini adalah data pasien dehidrasi akibat diare yang diambil dari Ruang Anggrek RSUD Dr. Soegiri Lamongan Tahun 2015. Jumlah data yang digunakan dalam penelitian ini adalah 150 dimana data tersebut dikelompokkan menjadi tiga *class* yang berbeda yaitu kelas ringan, sedang, dan

berat. Fitur yang digunakan ada 9 yaitu, keadaan umum, nadi, pernapasan/RR, ubun-ubun, mata, elastisitas kulit, *Glasgow Coma Scale* (GCS), status gizi, dan suhu.

3.3 Analisis Kebutuhan

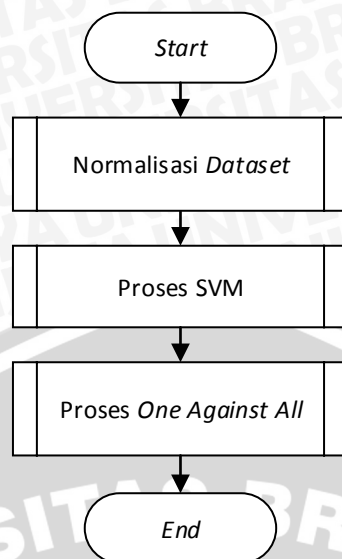
Analisis kebutuhan bertujuan untuk menganalisis dan mendapatkan semua kebutuhan yang diperlukan dalam pembuatan Implementasi Metode *Support Vector Machine* Untuk Rekomendasi Pemilihan Terapi Dehidrasi Pada Anak. Analisis kebutuhan disesuaikan dengan lokasi dan variabel penelitian, menentukan kebutuhan data yang akan digunakan, dan mempersiapkan alat dan bahan penelitian.

Metode analisis yang digunakan adalah *Procedural Analysis* dengan menggunakan bahasa pemodelan prosedural. Pemrograman berbasis prosedur merupakan teknik pemrograman yang dikembangkan berdasarkan algoritma untuk memecahkan suatu masalah. Algoritma merupakan cara-cara yang ditempuh dalam memanipulasi data sehingga masalah yang dihadapi bisa dipecahkan. Dalam hal ini, menggunakan metode *Support Vector Machine* dalam pengimplementasiannya. Secara keseluruhan, kebutuhan yang digunakan dalam pembuatan sistem ini meliputi:

1. Kebutuhan *Hardware*, meliputi:
 - Laptop
2. Kebutuhan *Software*, meliputi:
 - Microsoft Windows 8.1 sebagai sistem operasi
 - MySQL sebagai server *Database Management System*
 - Notepad++ sebagai *editor* untuk pembuatan sistem menggunakan Bahasa pemrograman PHP.
 - XAMPP
3. Data yang dibutuhkan meliputi:
 - Data pasien dehidrasi Ruang Anggrek RSUD Dr. Soegiri Lamongan Tahun 2015.
 - Data kriteria bobot untuk masing-masing terapi dehidrasi.

3.4 Perancangan Sistem

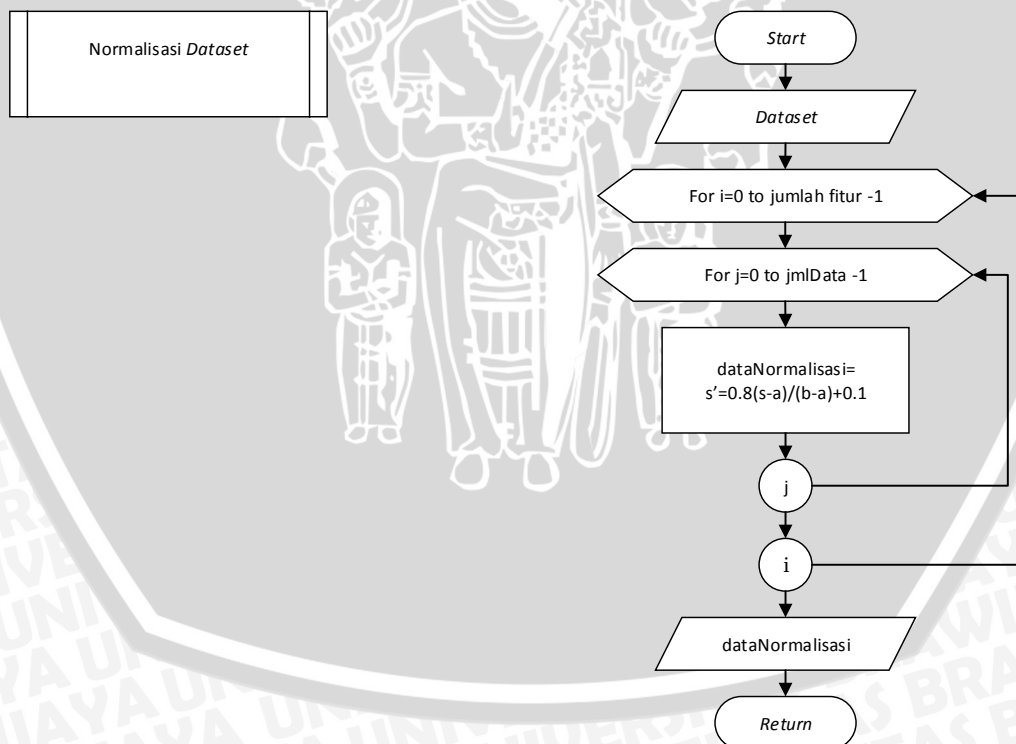
Tahap perancangan sistem dilakukan untuk mengetahui mekanisme proses pengklasifikasian terhadap data penyakit dehidrasi. Data yang digunakan berupa format *Microsoft Excel* (.csv). Terdapat 3 tahap dalam subbab perancangan sistem ini. Tahap pertama adalah normalisasi *dataset* dengan menggunakan *Sigmoid Biner*. Tahap kedua adalah proses SVM. Sebelum masuk proses SVM, konsep *one against all* digunakan untuk menentukan *class data training* dilevel 1, maupun level selanjutnya. Tahap ketiga adalah proses *one against all*. Pada gambar 3.2 akan dijelaskan perancangan dari sistem yang dibangun.



Gambar 3.2 Diagram alir perancangan sistem
Sumber: [Perancangan]

3.4.1 Normalisasi *Dataset*

Pada Gambar 3.3 adalah diagram alir proses normalisasi *dataset* menggunakan *Sigmoid Biner*



Gambar 3.3 Diagram alir normalisasi *dataset*
Sumber: [Perancangan]

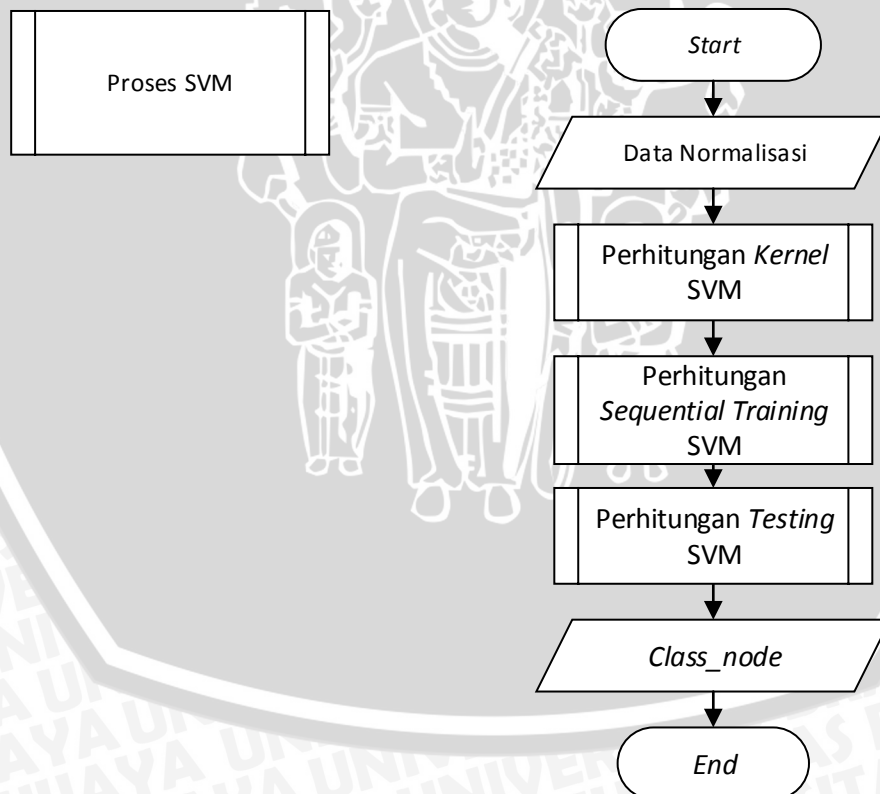
Pada Gambar 3.3 dijelaskan tahapan dari normalisasi *dataset*. Berikut adalah penjelasan dari tahapan-tahapan dari proses normalisasi.

1. Meng-input-kan *dataset* awal pasien dehidrasi yang akan digunakan.
2. Perulangan terhadap banyaknya kriteria terpilih dan perulangan terhadap jumlah data untuk proses perhitungan normalisasi.
3. Menghitung nilai normalisasi dengan menggunakan Persamaan 2.1.
4. *Output* yang dihasilkan adalah data yang telah ternormalisasi.

3.4.2 Proses *Support Vector Machine* (SVM)

Pada tahap proses SVM, langkah pertama yang dilakukan adalah mengambil *dataset*. Selanjutnya dilakukan dengan proses pembuatan *kernel* yang terdiri dari beberapa tahapan. Tahap pertama adalah menentukan data yang akan digunakan, tahap selanjutnya adalah menentukan *kernel*. Pada penelitian ini digunakan *kernel Radial Basis Fuction* (RBF) yang akan menghasilkan matriks *kernel*. Pada Gambar 3.4 akan dijelaskan tahapan SVM.

1. Melakukan *input dataset* ternormalisasi.
2. Melakukan perhitungan *kernel RBF Gaussian SVM*.
3. Melakukan perhitungan data *training SVM*, menggunakan metode *sequential training SVM*.
4. Melakukan perhitungan data *testing* untuk dilakukan *testing*.
5. *Output* yang dihasilkan adalah *predicted class*.

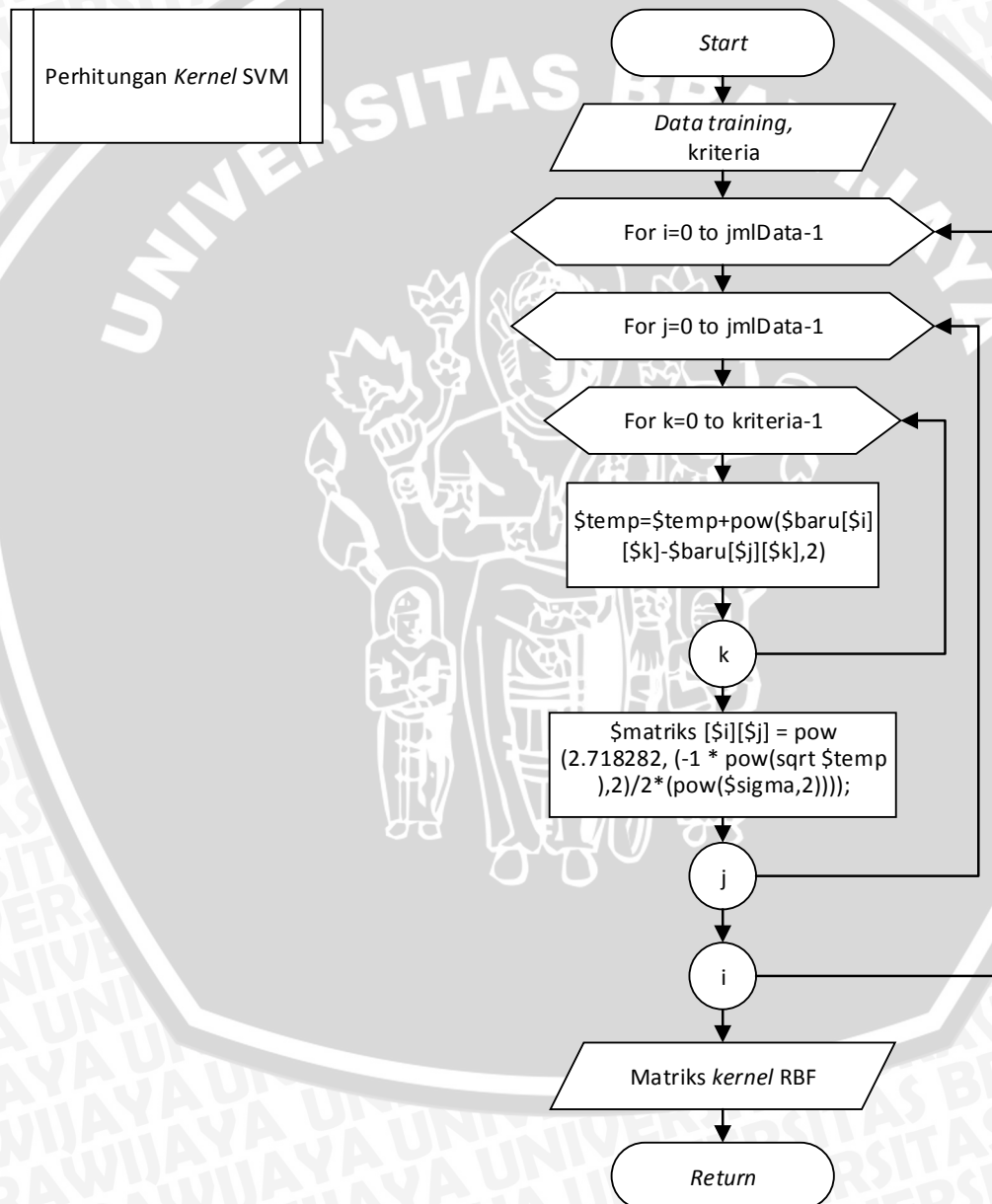


Gambar 3.4 Diagram alir proses *Support Vector Machine*
Sumber: [Perancangan]

3.4.3 Perhitungan *Kernel SVM*

Pada tahap perhitungan *kernel SVM* langkah pertama yang dilakukan adalah mengambil *data training*, yang selanjutnya dilakukan pengolahan data menggunakan persamaan 2.15. Pada Gambar 3.5 akan dijelaskan proses perhitungan *kernel SVM*.

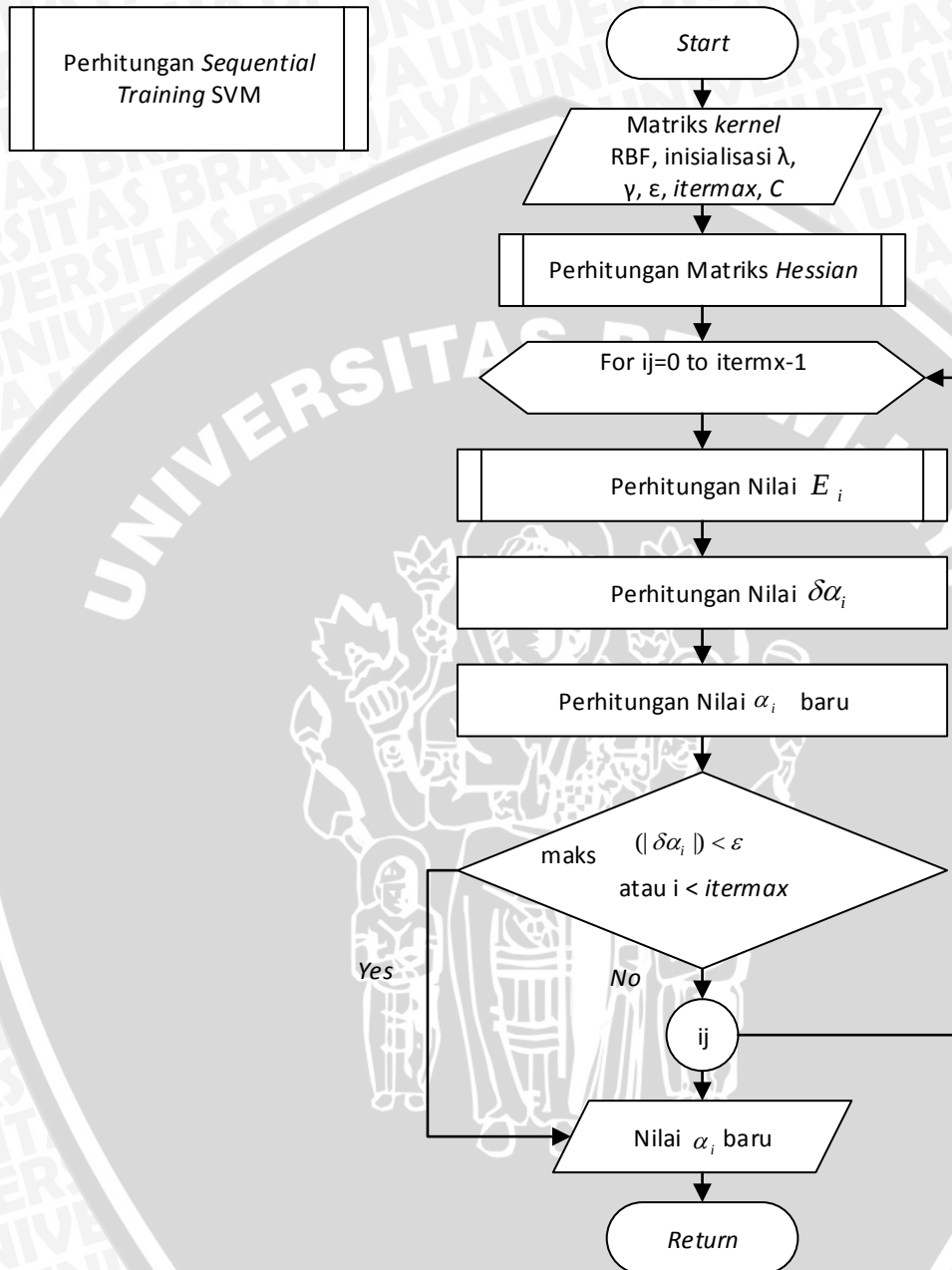
1. Melakukan *input data training*.
2. Perulangan terhadap jumlah *data training*.
3. Melakukan proses perhitungan *kernel Gaussian RBF*.
4. *Output* yang dihasilkan adalah matriks *kernel RBF*.



Gambar 3.5 Diagram alir perhitungan *kernel SVM*
Sumber: [Perancangan]

3.4.4 Perhitungan *Sequential Training SVM*

Langkah perhitungan *Sequential Training* dapat dilihat pada Gambar 3.6 Diagram Alir *Sequential Training SVM*.



Gambar 3.6 Diagram alir perhitungan *sequential training SVM*

Sumber: [Perancangan]

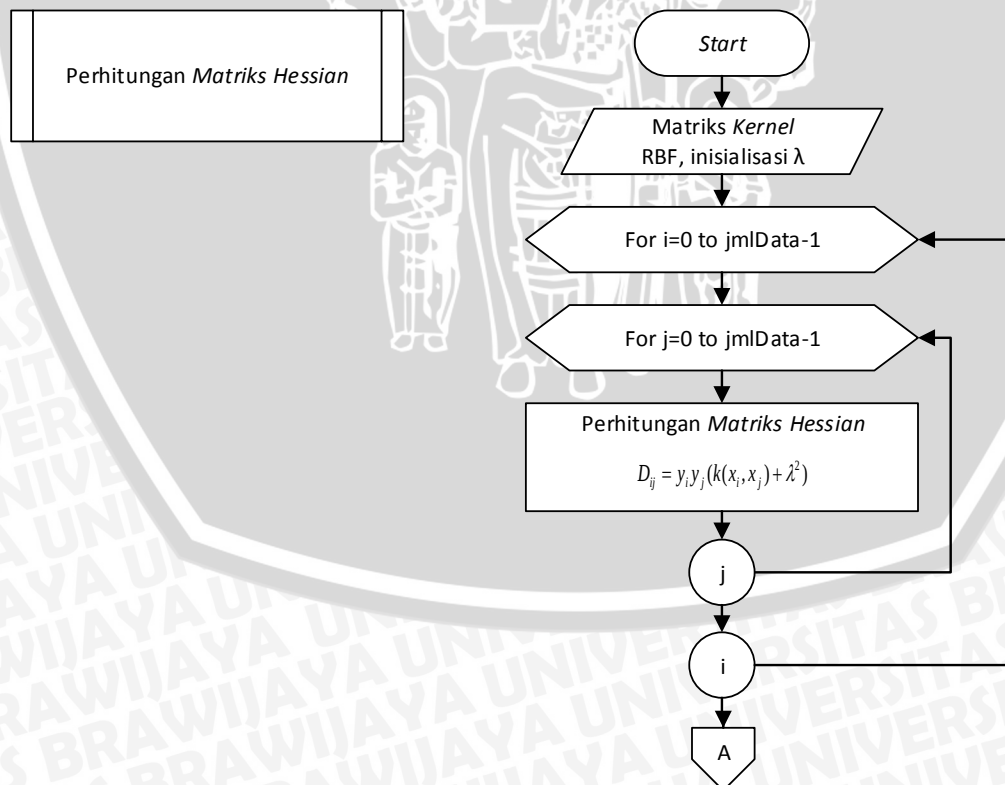
Pada Gambar 3.6 dijelaskan tahapan-tahapan perhitungan *sequential training SVM*. Berikut adalah penjelasannya.

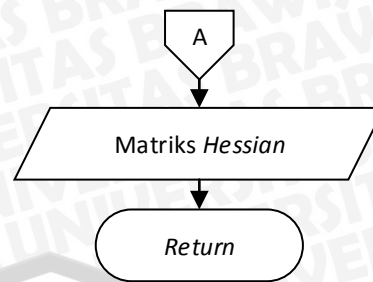
1. Inisialisasi parameter *Support Vector Machine* yaitu α_i (fungsi lagrange), λ (lamda), γ (gamma), ϵ (epsilon), Maximum Iterasi (*itermax*), C (complexity). *Kernel* yang digunakan adalah *kernel Gaussian RBF*.

2. Langkah pertama dalam proses *sequential training* adalah menghitung *matriks hessian* sesuai dengan Persamaan 2.19.
3. Langkah selanjutnya adalah menentukan nilai α_i untuk menghitung nilai E_i menggunakan Persamaan 2.20.
4. Langkah selanjutnya adalah menghitung nilai $\delta\alpha_i$ menggunakan Persamaan 2.21.
5. Setelah didapat nilai $\delta\alpha_i$, melakukan iterasi sejumlah data untuk proses *sequential training* SVM. Banyaknya iterasi ditentukan ketika *user* menginputkan nilai ϵ (*epsilon*).
6. Tahap perhitungan *sequential training* yang selanjutnya adalah, mencari nilai α_i menggunakan Persamaan 2.22.
7. Iterasi yang dilakukan pada data *training* jika telah mencapai nilai *konvergen* maks ($|\delta\alpha_i| < \epsilon$), atau mencapai nilai *itermax* maka iterasi akan dihentikan. Proses diulangi sampai α_i mencapai nilai konvergen. Konvergen terjadi ketika, nilai α_i mengalami perubahan.
8. *Output* yang dihasilkan adalah variabel α_i , fungsi *hyperplane* (pemisah) yang optimal yaitu perubahan nilai α_i yang akan digunakan untuk menentukan *Support Vector*. Fungsi pemisah tersebut akan digunakan dalam proses *testing*.

3.4.5 Perhitungan *Matriks Hessian*

Langkah perhitungan *Matriks Hessian* dapat dilihat pada Gambar 3.7 Diagram Alir Perhitungan *Matriks Hessian*.





Gambar 3.7 Diagram alir perhitungan *matriks hessian*

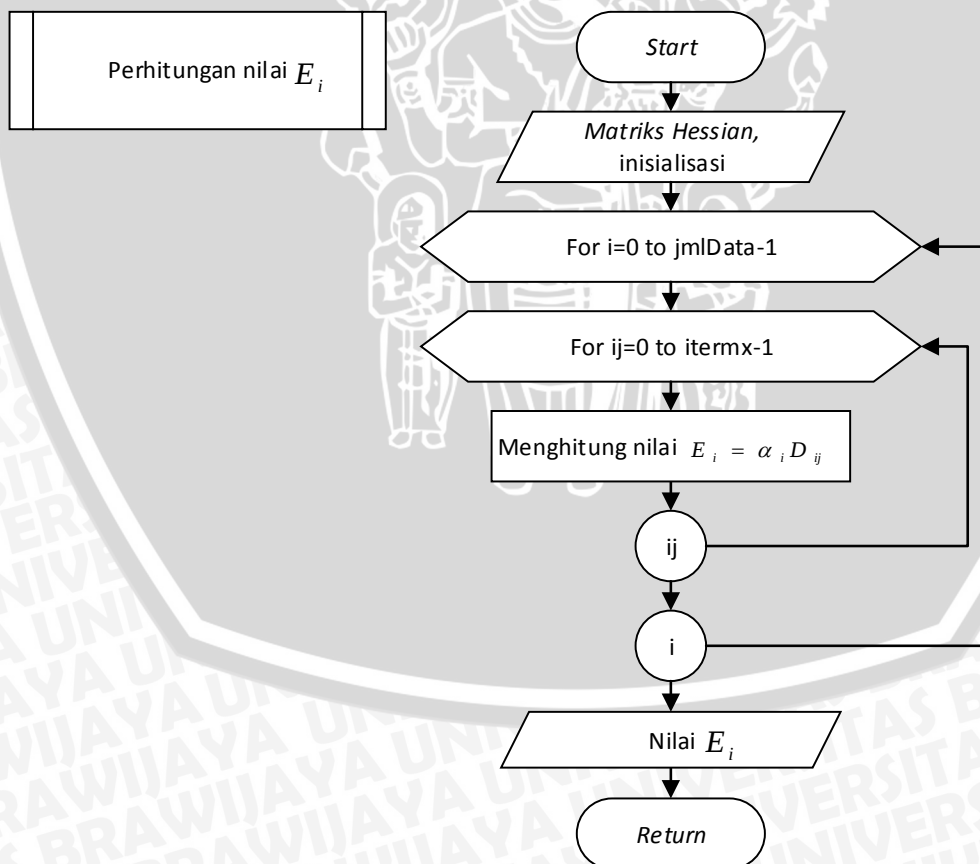
Sumber: [Perancangan]

Pada Gambar 3.7 dijelaskan tahapan-tahapan perhitungan *Matriks Hessian*. Berikut adalah penjelasannya.

1. Meng-*input*-kan data *training*.
2. Inisialisasi parameter *Matriks Hessian* yaitu nilai λ (*lamda*).
3. Perhitungan *Matriks Hessian* menggunakan Persamaan 2.19.
4. Perulangan terhadap jumlah data *training*.
5. *Output* yang dihasilkan adalah nilai *Matriks Hessian*.

3.4.6 Perhitungan Nilai E_i

Langkah perhitungan nilai E_i dapat dilihat pada Gambar 3.8 Diagram Alir Perhitungan Nilai E_i .



Gambar 3.8 Diagram alir perhitungan nilai E_i

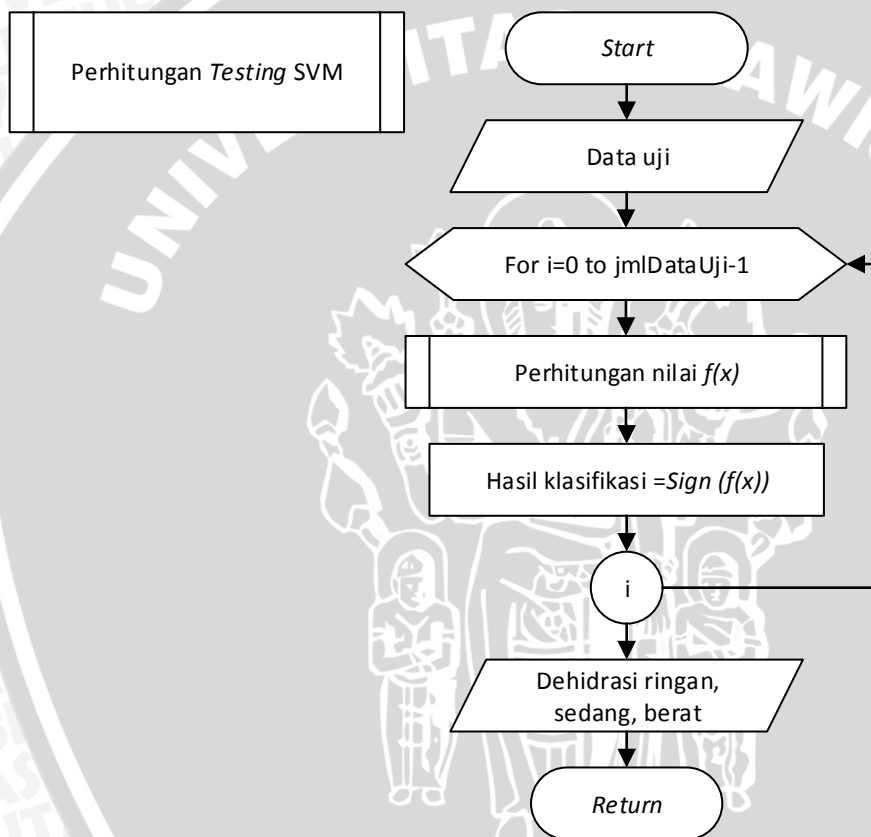
Sumber: [Perancangan]

Pada Gambar 3.8 dijelaskan tahapan-tahapan perhitungan Nilai E_i . Berikut adalah penjelasannya.

1. Meng-input-kan nilai *Matriks Hessian*.
2. Inisialisasi nilai $\alpha_i = 0$
3. Menghitung nilai E_i menggunakan Persamaan 2.20.
4. Melakukan perulangan data *training* pada proses perhitungan nilai E_i .
5. *Output* yang dihasilkan adalah nilai E_i .

3.4.7 Proses Perhitungan *Testing* SVM

Langkah perhitungan *testing* SVM dapat dilihat pada Gambar 3.10 Diagram Alir Perhitungan *Testing* SVM.



Gambar 3.9 Diagram alir perhitungan *testing* SVM

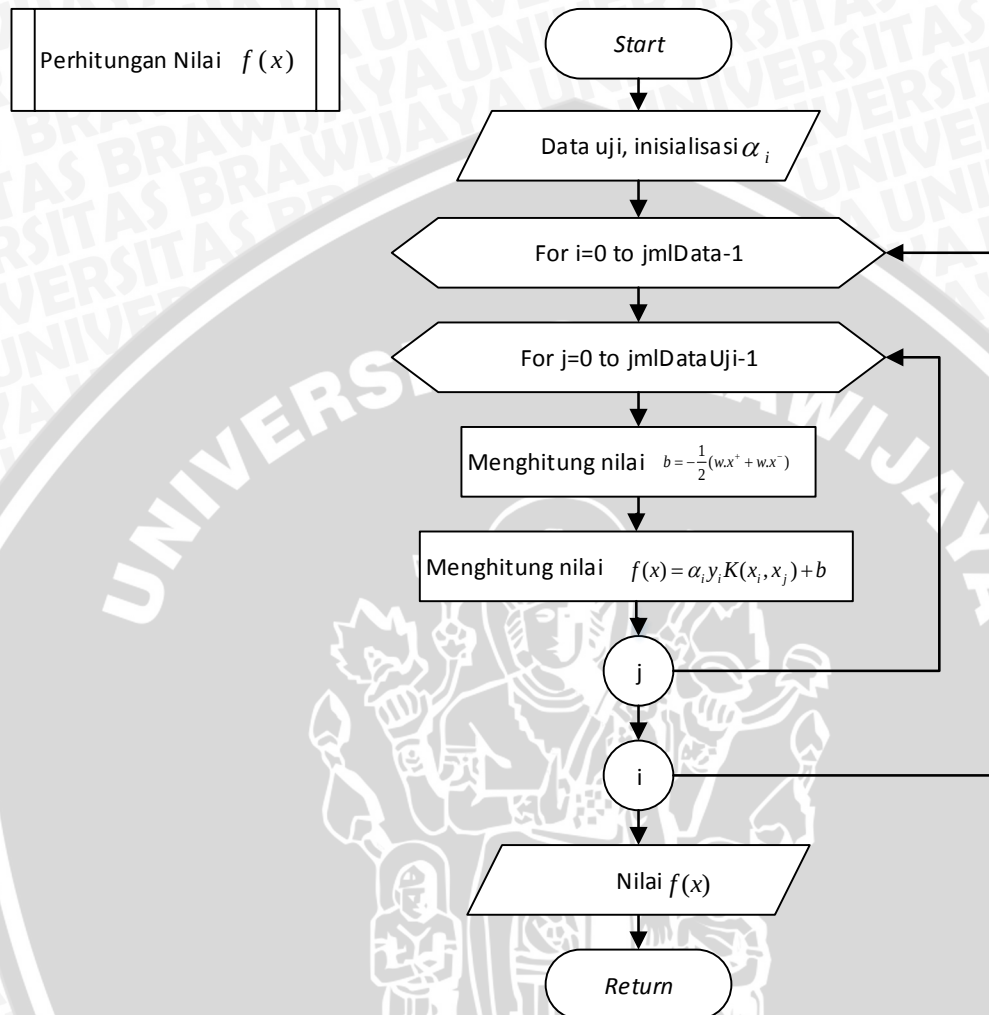
Sumber: [Perancangan]

Pada Gambar 3.10 dijelaskan tahapan-tahapan perhitungan data *testing*. Berikut adalah penjelasannya.

1. Meng-input-kan data uji.
2. Nilai tersebut akan diuji dengan fungsi *hyperplane* yang diperoleh dari *sequential training* SVM.
3. Menghitung hasil klasifikasi yaitu dengan fungsi $\text{sign}(f(x))$ pada masing-masing data *testing* terpilih.
4. *Output* yang dihasilkan adalah nilai klasifikasi *class* dari setiap data *testing*, untuk *class negative* dilakukan perhitungan pada *level 2*.

3.4.8 Perhitungan Nilai $f(x)$ pada Data Testing SVM

Langkah perhitungan $f(x)$ dapat dilihat pada Gambar 3.9 Diagram Alir Perhitungan Nilai $f(x)$.



Gambar 3.10 Diagram alir perhitungan nilai $f(x)$

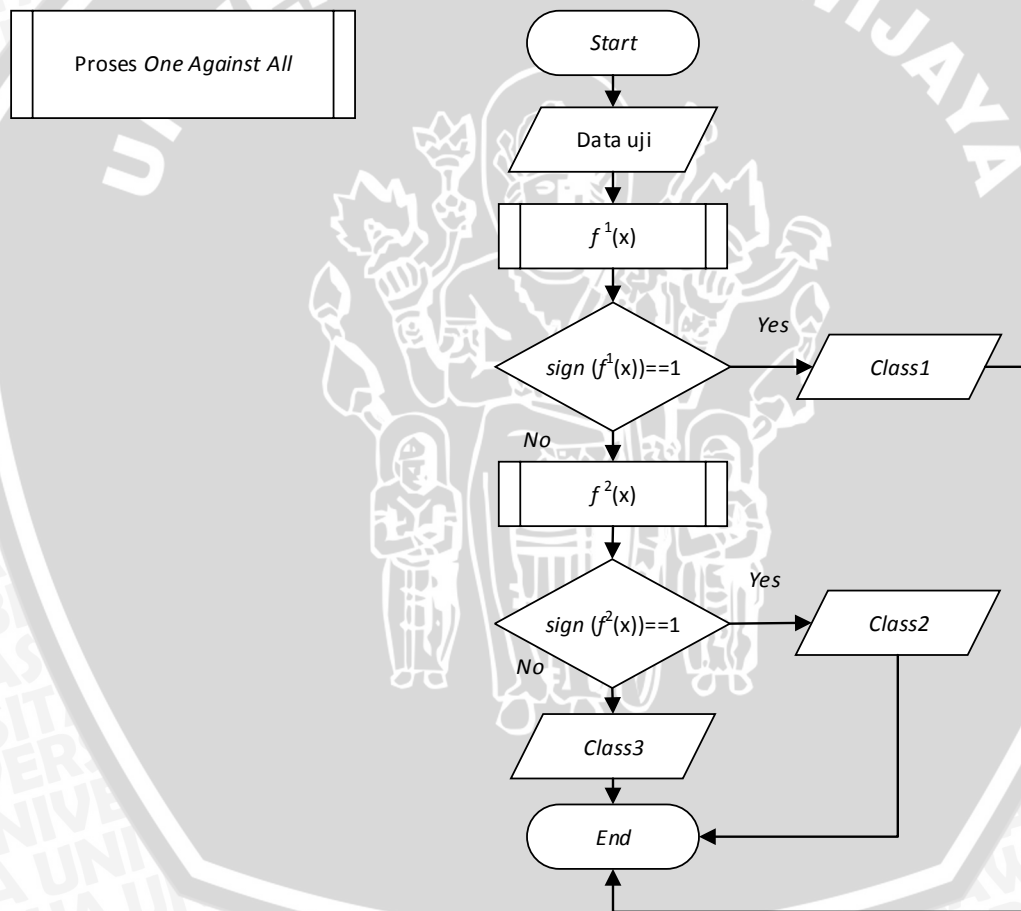
Sumber: [Perancangan]

Pada Gambar 3.9 dijelaskan tahapan-tahapan perhitungan Nilai $f(x)$. Berikut adalah penjelasannya.

1. Meng-input-kan *dataset testing* terpilih.
2. Inisialisasi terhadap nilai α_i , yang diperoleh dari proses iterasi terakhir perhitungan *sequential training*.
3. Perhitungan nilai $K(x_i, x_j)$ terhadap nilai data *testing*.
4. Perhitungan $f(x) = \alpha_i \cdot y_i \cdot K(x_i, x_j)$. Nilai y_i adalah nilai *class* dari data *testing*.
5. *Output* yang dihasilkan adalah nilai $f(x)$ data *testing*.

3.4.9 One Against All

Tahapan *One Against All* (OAA) adalah tahapan untuk mengatasi kelemahan metode SVM yang hanya memisahkan 2 kelas (+1, -1) menjadi *multi-class*. Metode ini membantu dalam mengatasi masalah dari data yang digunakan dalam skripsi ini. Dari klasifikasi dehidrasi terdapat 3 kelas yaitu kelas ringan, kelas sedang, dan kelas berat dimana metode ini membuat 2 fungsi. Fungsi pertama memisahkan kelas 1 adalah positif (+1) dan kelas 2 dan 3 adalah negatif (-1). Sedangkan fungsi kedua adalah memisahkan kelas 2 adalah positif (+1) dan kelas 1 dan 3 adalah negatif (-1). Maka jika pada fungsi pertama nilai yang didapatkan positif (+1) maka otomatis data tersebut termasuk kelas 1 sedangkan jika negatif (-1) maka dilakukan pengecekan pada fungsi kedua. Pada fungsi kedua jika didapatkan nilai positif (+1) maka data tersebut termasuk kelas 2 sedangkan jika negatif (-1) maka data tersebut termasuk kelas 3. Pada Gambar 3.11 akan dijelaskan tahapan-tahapan *one against all*.



Gambar 3.11 Diagram alir perhitungan proses *one against all*
Sumber: [Perancangan]

3.5 Implementasi Sistem

Implementasi program dengan menggunakan algoritma *Support Vector Machine* metode *sequential training* dilakukan dengan mengacu pada perancangan sistem. Implementasi program ini menggunakan bahasa pemrograman PHP, *Microsoft Excel*, dan *tools* pendukung lainnya. Implementasi sistem tersebut meliputi :

1. Pembuatan hitungan manual.
2. Pembuatan antarmuka.

3.5.1 Perhitungan Manual

Perhitungan manual pada penelitian ini, menggunakan *dataset* dari data *sample* data pasien yang terdiri dari 9 atribut atau kriteria dan akan dibagi menjadi tiga kelas, yaitu dehidrasi tipe ringan, sedang dan berat. Data tersebut berasal dari hasil observasi pada Ruang Anggrek RSUD Dr. Soegiri Lamongan. Data hasil observasi dapat dilihat pada Lampiran D Tabel Data pasien. Data tersebut diubah berdasarkan parameter penilaian tertentu dari masing-masing kriteria. Dengan *range* parameter antara 1-3 dari setiap kriteria, dan data dinormalisasi dengan *interval* antara [0.1, 0.9] sesuai dengan Persamaan 2.1. Kelas pasien yang mengalami dehidrasi ringan diberi nilai 1, kelas pasien yang mengalami dehidrasi sedang diberi nilai 2, dan kelas pasien yang mengalami dehidrasi berat diberi nilai 3. Pada perhitungan manual data yang digunakan terdiri dari 13 *dataset*, yang terdiri dari 10 *dataset training* dan 3 *dataset testing*. Data *sample* yang akan digunakan sebagai *dataset training* dan *dataset testing* dapat dilihat pada Tabel 3.2 *Sample Dataset*. Untuk tabel data secara lengkap dapat dilihat pada Lampiran E Tabel *Data Sample* untuk *data training* dan *data testing*.

3.5.1.1 Perhitungan Manual Proses Normalisasi Data

Pada penelitian ini digunakan *sigmoid biner* dengan *interval* [0.1, 0.9] untuk menormalisasikan data. Hal ini digunakan untuk menghindari hasil bernilai nol, sehingga hasil minimum bernilai 0.1 dan hasil maksimum bernilai 0.9. Dimana nilai *range* diambil dari nilai maksimum dikurangi nilai minimum pada keseluruhan data. Berikut adalah contoh perhitungannya menggunakan Persamaan 2.1, misalnya diambil 1 data dari *data training* dengan atribut F1 yaitu nomer 1 yang bernilai 3. Detailnya ada pada Lampiran D.

Tabel 3.1 Range normalisasi

	F1	F2	F3	F4	F5	F6	F7	F8	F9
Min	1	90	20	1	1	1	2	1	35
Max	3	140	40	2	2	3	3	3	39
Range	2	50	20	1	1	2	1	2	4

Sumber: [Perancangan]

$$s' = \frac{0,8(s - a)}{(b - a)} + 0.1$$

Dimana s' adalah nilai hasil normalisasi, s adalah data awal, a adalah nilai minimum, dan b adalah nilai maksimum.

$$s' = \frac{0.8(3-1)}{(3-1)} + 0.1 = \frac{0.8(2)}{(2)} + 0.1 = \frac{1.6}{2} + 0.1 = 0.8 + 0.1 = 0.9$$

Pada Tabel 3.2 adalah tabel hasil normalisasi dengan menggunakan *sigmoid biner* dengan *interval* [0.1, 0.9].

Tabel 3.2 Tabel hasil normalisasi

ID	F1	F2	...	F8	F9	Kelas
1	0.900000	0.260000	...	0.900000	0.900000	1
2	0.900000	0.580000	...	0.900000	0.820000	1
3	0.900000	0.260000	...	0.900000	0.900000	1
4	0.500000	0.900000	...	0.500000	0.300000	2
5	0.500000	0.580000	...	0.500000	0.300000	2
7	0.500000	0.580000	...	0.500000	0.300000	2
6	0.100000	0.100000	...	0.100000	0.100000	3
8	0.100000	0.100000	...	0.100000	0.180000	3
11	0.100000	0.100000	...	0.100000	0.220000	3
30	0.100000	0.100000	...	0.100000	0.180000	3

Sumber: [Perancangan]

3.5.1.2 Perhitungan Manual Proses Training SVM

Pada penelitian ini, akan digunakan metode *sequential training*. Pada proses perhitungan manual proses *training* SVM, data dari sekumpulan objek akan diklasifikasikan. Diketahui *hyperplane* yang merupakan fungsi pemisah dengan nilai *margin* yang maksimal. *Hyperplane* adalah suatu bidang pemisah yang terletak ditengah antara dua set objek dari dua kelas. Untuk mendapatkan nilai *margin* yang maksimal dapat digunakan metode *sequential*.

Parameter yang terdapat pada metode SVM dan digunakan dalam proses *training* antara lain adalah w (*weight*), b (posisi bidang relatif terhadap pusat koordinat) dan C (*complexity*). Setelah data dinormalisasi menggunakan Persamaan 2.1

Data hasil normalisasi digunakan untuk menghitung nilai *kernel*. *Kernel* yang digunakan adalah *Gaussian RBF* (*Radial Basic Function*). Hasil perhitungan *kernel* RBF dapat dilihat pada Tabel 3.6. Berikut adalah contoh perhitungan *kernel* RBF menggunakan Persamaan 2.17

Pada Tabel 3.3 adalah tabel data *training* dengan ID pasien 1.

Data 1 (x_1)

Tabel 3.3 Data training 1

No.	Keadaan Umum	Nadi	Pernapasan /RR	...	Elastisitas Kulit	Status Gizi	Suhu
1	0.900000	0.260000	0.100000	...	0.900000	0.900000	0.900000

Sumber: [Perancangan]

$k(x, y)$, misal $x = x_1, y = x_1$ maka $k(x, y) = k(x_1, x_1) =$

$$\exp\left(-\frac{\|x_1 - x_1\|^2}{2\sigma^2}\right) = \exp\left(-\frac{0}{2.1^2}\right) = \exp\left(-\frac{0}{2}\right) = \exp(0) = 1$$

$$(x_1 - x_1) = \begin{pmatrix} 0.900000 \\ 0.260000 \\ 0.100000 \\ 0.100000 \\ 0.100000 \\ 0.900000 \\ 0.900000 \\ 0.900000 \\ 0.900000 \end{pmatrix} - \begin{pmatrix} 0.900000 \\ 0.260000 \\ 0.100000 \\ 0.100000 \\ 0.100000 \\ 0.900000 \\ 0.900000 \\ 0.900000 \\ 0.900000 \end{pmatrix}$$

$$= \begin{pmatrix} 0.900000 - 0.900000 \\ 0.260000 - 0.260000 \\ 0.100000 - 0.100000 \\ 0.100000 - 0.100000 \\ 0.100000 - 0.100000 \\ 0.900000 - 0.900000 \\ 0.900000 - 0.900000 \\ 0.900000 - 0.900000 \\ 0.900000 - 0.900000 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\|x_1 - x_1\| = \sqrt{0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2} = \sqrt{0} = 0$$

$$\|x_1 - x_1\|^2 = 0^2 = 0$$

Pada Tabel 3.4 adalah tabel data *training* dengan ID pasien 2.

Data 2 (x_2)

Tabel 3.4 Data training 2

No.	Keadaan Umum	Nadi	Pernapasan /RR	...	Elastisitas Kulit	Status Gizi	Suhu
2	0.900000	0.580000	0.100000	...	0.900000	0.900000	0.820000

Sumber: [Perancangan]

$$k(x_1, x_2) = \exp\left(-\frac{\|x_1 - x_2\|^2}{2\sigma^2}\right) = \exp\left(-\frac{0.1088}{2.1^2}\right) = \exp\left(-\frac{0.1088}{2}\right) = \exp(-0.0544) = 0.947053$$

$$(x_1 - x_2) = \begin{pmatrix} 0.900000 \\ 0.260000 \\ 0.100000 \\ 0.100000 \\ 0.100000 \\ 0.900000 \\ 0.900000 \\ 0.900000 \\ 0.900000 \end{pmatrix} - \begin{pmatrix} 0.900000 \\ 0.580000 \\ 0.100000 \\ 0.100000 \\ 0.100000 \\ 0.900000 \\ 0.900000 \\ 0.900000 \\ 0.820000 \end{pmatrix} = \begin{pmatrix} 0.900000 - 0.900000 \\ 0.260000 - 0.580000 \\ 0.100000 - 0.100000 \\ 0.100000 - 0.100000 \\ 0.100000 - 0.100000 \\ 0.900000 - 0.900000 \\ 0.900000 - 0.900000 \\ 0.900000 - 0.900000 \\ 0.900000 - 0.820000 \end{pmatrix} = \begin{pmatrix} 0 \\ -0.32 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0.08 \end{pmatrix}$$

$$\|x_1 - x_2\| = \sqrt{0^2 + (-0.32)^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0.08^2} = \sqrt{0.1088} = 0.3298$$

$$\|x_1 - x_2\|^2 = 0.3298^2 = 0.1088$$

Pada Tabel 3.5 adalah tabel data *training* dengan ID pasien 3.

Data 3 (x_3)

Tabel 3.5 Data training 3

No.	Keadaan Umum	Nadi	Pernapasan /RR	...	Elastisitas Kulit	Status Gizi	Suhu
3	0.900000	0.260000	0.100000	...	0.900000	0.900000	0.900000

Sumber: [Perancangan]

$$k(x_1, x_3) = \exp\left(-\frac{\|x_1 - x_3\|^2}{2\sigma^2}\right) = \exp\left(-\frac{0}{2.1^2}\right) = \exp\left(-\frac{0}{2}\right) = 1$$

$$(x_1 - x_3) = \begin{pmatrix} 0.900000 \\ 0.260000 \\ 0.100000 \\ 0.100000 \\ 0.100000 \\ 0.900000 \\ 0.900000 \\ 0.900000 \\ 0.900000 \end{pmatrix} - \begin{pmatrix} 0.900000 \\ 0.260000 \\ 0.100000 \\ 0.100000 \\ 0.100000 \\ 0.900000 \\ 0.900000 \\ 0.900000 \\ 0.900000 \end{pmatrix}$$

$$= \begin{pmatrix} 0.900000 - 0.900000 \\ 0.260000 - 0.260000 \\ 0.100000 - 0.100000 \\ 0.100000 - 0.100000 \\ 0.100000 - 0.100000 \\ 0.900000 - 0.900000 \\ 0.900000 - 0.900000 \\ 0.900000 - 0.900000 \\ 0.900000 - 0.900000 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\|x_1 - x_3\| = \sqrt{0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2} = \sqrt{0} = 0$$

$$\|x_1 - x_3\|^2 = 0^2 = 0$$

Pada Tabel 3.6 adalah hasil perhitungan *Kernel Gaussian* (RBF) menggunakan Persamaan 2.17.

Tabel 3.6 Hasil perhitungan data *training* dengan *kernel* RBF

$K(x, y)$	1	2	3	...	8	9	10
1	1.000000	0.947053	1.000000	...	0.111693	0.114865	0.111693
2	0.947053	1.000000	0.947053	...	0.106459	0.109132	0.106459
3	1.000000	0.947053	1.000000	...	0.111693	0.114865	0.111693
4	0.358868	0.437622	0.358868	...	0.200448	0.201251	0.200448
5	0.418449	0.460612	0.418449	...	0.246006	0.246992	0.246006
6	0.576258	0.634321	0.576258	...	0.338781	0.340139	0.338781
7	0.105105	0.100822	0.105105	...	0.996805	0.992826	0.996805
8	0.111693	0.106459	0.111693	...	1.000000	0.999200	1.000000
9	0.114865	0.109132	0.114865	...	0.999200	1.000000	0.999200
10	0.111693	0.106459	0.111693	...	1.000000	0.999200	1.000000

Sumber: [Perancangan]

Setelah didapat nilai perhitungan *kernel*, tahap selanjutnya adalah perhitungan metode *sequential training* untuk mendapatkan parameter SVM lainnya. Proses perhitungan menggunakan metode *sequential training*, akan menggunakan Persamaan 2.19 sampai Persamaan 2.23. Berikut adalah tahapan perhitungan *sequential training* SVM.

1. Inisialisasi dengan nilai $\alpha_i = 0$
2. Menghitung komponen matriks pada Persamaan 2.19.

$D_{ij} = y_i y_j (K(x_i, x_j) + \lambda^2)$, dimana $y_i = 1$ Class pada atribut pertama dan $y_j = 1$ pada atribut kedua. (x_i, x_j) adalah *kernel* yang telah dihitung pada tahap sebelumnya. Inisialisasi nilai lamda $\lambda = 0.5$. Contoh perhitungan *record* pertama dan kedua adalah sebagai berikut:

$$\begin{aligned} D_{11} &= y_1 y_1 (K(x_1, x_1) + \lambda^2) \\ &= 1 \cdot 1 (1.000000 + 0.5^2) \\ &= 1 (1.000000 + 0.25) \\ &= 1.250000 \end{aligned}$$

$$\begin{aligned} D_{12} &= y_1 y_2 (K(x_1, x_2) + \lambda^2) \\ &= 1 \cdot 1 (0.947053 + 0.5^2) \\ &= 1 (0.947053 + 0.25) \\ &= 1.197053 \end{aligned}$$

Tabel 3.7 adalah hasil perhitungan *matriks hessian* menggunakan Persamaan 2.19.

Tabel 3.7 Hasil perhitungan *matriks hessian*

D_{ij}	1	2	3	...	8	9	10
1	1.250000	1.197053	1.250000	...	-0.361693	-0.364865	-0.361693
2	1.197053	1.250000	1.197053	...	-0.356459	-0.359132	-0.356459
3	1.250000	1.197053	1.250000	...	-0.361693	-0.364865	-0.361693
4	-0.608868	-0.687622	0.608868	...	0.450448	0.451251	0.450448
5	-0.668449	-0.710612	-0.668449	...	0.496006	0.496992	0.496006
6	-0.826258	-0.884321	-0.826258	...	0.588781	0.590139	0.588781
7	-0.355105	-0.350822	-0.355105	...	1.246805	1.242826	1.246805
8	-0.361693	-0.356459	-0.361693	...	1.250000	1.249200	1.250000
9	-0.364865	-0.359132	-0.364865	...	1.249200	1.250000	1.249200
10	-0.361693	-0.356459	-0.361693	...	1.250000	1.249200	1.250000

Sumber: [Perancangan]

Setelah menghitung nilai *Matriks Hessian*, langkah selanjutnya adalah menghitung nilai *error* menggunakan Persamaan 2.20 sampai 2.23

3. Dilakukan iterasi untuk setiap $i, = 1, 2, \dots, n$. Pada proses perhitungan ini dilakukan iterasi ($itermax$) = 100, $C = 1$, dan γ adalah nilai konstanta, dalam penelitian ini nilai $\gamma = 0.01$. Untuk mendapatkan nilai γ digunakan rumus konstanta dibagi dengan nilai max diagonal *Matriks Hessian*. Sehingga diperoleh nilai

$$\gamma = \frac{0.01}{1.250000} = 0.008000$$

Iterasi ke 0:

- a) Menghitung Persamaan 2.20 $E_i = \sum_{j=1}^n \alpha_i D_{ij}$. Contoh perhitungan *record* pertama adalah sebagai berikut:

$$E_i = ((0.000000 \times 1.250000) + (0.000000 \times 1.250000) + (0.000000 \times 1.250000) + (0.000000 \times 1.250000) + (0.000000 \times 1.250000) + (0.000000 \times 1.250000) + (0.000000 \times 1.250000) + (0.000000 \times 1.250000)) = 0$$

Perhitungan dilakukan untuk seluruh data, dan hasilnya dapat dilihat pada Tabel 3.8.

Tabel 3.8 Hasil perhitungan *iterasi* ke 1 pada persamaan 2.20

$\alpha_i D_{ij}$	1	2	3	...	9	10	E_i
1	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000
2	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000
3	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000
4	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000
5	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000
6	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000
7	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000
8	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000
9	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000
10	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000

Sumber: [Perancangan]

- b) Menghitung nilai $\delta\alpha_i$ dengan Persamaan 2.21. Contoh perhitungan *record* pertama adalah sebagai berikut:

$$\begin{aligned}\delta\alpha_1 &= \min\{\text{maks}[\gamma(1 - E_1), -\alpha_1], C - \alpha_1\} \\ &= \min\{\text{maks}[0.008000(1 - 0.000000), -0.000000], 1 - 0.000000\} \\ &= \min\{\text{maks}[0.008000(1), -0.000000], 1\} \\ &= \min\{\text{maks}[0.008000, -0.000000], 1\} \\ &= \min\{0.008000, 1\} \\ &= 0.008000\end{aligned}$$

Perhitungan dilakukan untuk seluruh data, dan hasilnya dapat dilihat pada Tabel 3.9.

Tabel 3.9 Hasil perhitungan *iterasi* ke 1 pada persamaan 2.20

	1	2	3	...	8	9	10
$\delta\alpha_i$	0.008000	0.008000	0.008000	...	0.008000	0.008000	0.008000

Sumber: [Perancangan]

- c) Memperbarui nilai α_i dengan Persamaan 2.22. Contoh perhitungan *record* pertama adalah sebagai berikut
 $\alpha_1 = 0 + 0.008000 = 0.008000$

- d) Proses selanjutnya adalah melakukan pengecekan iterasi. Nilai max yang diperoleh adalah 0.00 dan kurang dari nilai *threshold* yang ditentukan yaitu 0.0001 maka iterasi dapat dilanjutkan. Nilai *itermax* yaitu 100 sudah terpenuhi, maka seluruh proses dihentikan.

Setelah perhitungan *training* menggunakan *sequential training*, tahap selanjutnya adalah mencari nilai *w* (*weight*) dan *b* (posisi bidang relatif terhadap pusat koordinat) dengan menggunakan proses *sequential training*. Nilai $w \cdot x^+$ adalah nilai dari salah satu *support vector* dari *class positive*. x^+ memiliki nilai bobot dan nilai awal lebih besar. Sedangkan $w \cdot x^-$ adalah nilai dari salah satu *support vector* dari *class negative*. x^- memiliki nilai bobot dan nilai awal lebih kecil. Data awal setelah dinormalisasi dikali dengan data yang mempunyai nilai *support vector positive* atau *negative*.

Contoh perhitungan untuk $K(x_i, x^+)$, dan untuk perhitungan $K(x_i, x^-)$ dilakukan dengan contoh yang sama. Data nilai x_i untuk *class* $x^+ = 0.016004$, dan x_i untuk *class* $x^- = 0.015809$, maka hasil perhitungannya akan dijelaskan sebagai berikut.

Pada Tabel 3.10 adalah tabel data *training* dengan ID pasien 2.

Data 2 (x_2)

Tabel 3.10 Data training 2

No.	Keadaan Umum	Nadi	Pernapasan /RR	...	Elastisitas Kulit	Status Gizi	Suhu
1	0.900000	0.580000	0.100000	...	0.900000	0.900000	0.820000

Sumber: [Perancangan]

Misal $\sigma=1$

$k(x, x^+)$, misal $x = x_1, x^+ = x_2$ maka $k(x, x^+) = k(x_1, x_2) =$

$$\exp\left(-\frac{\|x_1 - x_2\|^2}{2\sigma^2}\right) = \exp\left(-\frac{0.1088}{2.1^2}\right) = \exp\left(-\frac{0.1088}{2}\right)$$

$$= \exp(-0.0544) = 0.947053$$

$$(x_1 - x_2) = \begin{pmatrix} 0.900000 \\ 0.260000 \\ 0.100000 \\ 0.100000 \\ 0.100000 \\ 0.900000 \\ 0.900000 \\ 0.900000 \\ 0.900000 \end{pmatrix} - \begin{pmatrix} 0.900000 \\ 0.580000 \\ 0.100000 \\ 0.100000 \\ 0.100000 \\ 0.900000 \\ 0.900000 \\ 0.900000 \\ 0.820000 \end{pmatrix}$$

$$= \begin{pmatrix} 0.900000 - 0.900000 \\ 0.260000 - 0.580000 \\ 0.100000 - 0.100000 \\ 0.100000 - 0.100000 \\ 0.100000 - 0.100000 \\ 0.900000 - 0.900000 \\ 0.900000 - 0.900000 \\ 0.900000 - 0.900000 \\ 0.900000 - 0.820000 \end{pmatrix} = \begin{pmatrix} 0 \\ -0.32 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0.08 \end{pmatrix}$$

$$\|x_1 - x_2\| = \sqrt{0^2 + (-0.32)^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + (0.08)^2}$$

$$= \sqrt{0.1088} = 0.3298$$

$$\|x_1 - x_2\|^2 = 0.3298^2 = 0.1088$$

Data nilai x_i untuk $class x^+ = 0.016004$, dan x_i untuk $class x^- = 0.015809$, maka hasil perhitungan adalah sebagai berikut.

Pada Tabel 3.11 adalah tabel data *training* dengan ID pasien 7.

Data 6 (x_6)

Tabel 3.11 Data training 6

No.	Keadaan Umum	Nadi	Pernapasan /RR	...	Elastisitas Kulit	Status Gizi	Suhu
1	0.500000	0.500000	0.100000	...	0.500000	0.500000	0.300000

Sumber: [Perancangan]

Misal $\sigma=1$

$k(x, x^-)$, misal $x = x_1, x^- = x_6$ maka $k(x, x^-) = k(x_1, x_6) =$

$$\exp\left(-\frac{\|x_1 - x_6\|^2}{2\sigma^2}\right) = \exp\left(-\frac{1.1024}{2.1^2}\right) = \exp\left(-\frac{1.1024}{2}\right)$$

$$= \exp(-0.5512) = 0.576258$$

$$(x_1 - x_6) = \begin{pmatrix} 0.900000 \\ 0.260000 \\ 0.100000 \\ 0.100000 \\ 0.100000 \\ 0.900000 \\ 0.900000 \\ 0.900000 \\ 0.900000 \end{pmatrix} - \begin{pmatrix} 0.500000 \\ 0.580000 \\ 0.100000 \\ 0.100000 \\ 0.100000 \\ 0.500000 \\ 0.500000 \\ 0.500000 \\ 0.300000 \end{pmatrix}$$

$$= \begin{pmatrix} 0.900000 - 0.500000 \\ 0.260000 - 0.580000 \\ 0.100000 - 0.100000 \\ 0.100000 - 0.100000 \\ 0.100000 - 0.100000 \\ 0.900000 - 0.500000 \\ 0.900000 - 0.500000 \\ 0.900000 - 0.500000 \\ 0.900000 - 0.300000 \end{pmatrix} = \begin{pmatrix} 0.4 \\ -0.32 \\ 0 \\ 0 \\ 0 \\ 0.4 \\ 0.4 \\ 0.4 \\ 0.6 \end{pmatrix}$$

$$\|x_1 - x_6\| = \sqrt{\frac{(0.4)^2 + (-0.32)^2 + 0^2 + 0^2 + 0^2 +}{(0.4)^2 + (0.4)^2 + (0.4)^2 + (0.6)^2}} \\ = \sqrt{1.1024} = 1.049952$$

$$\|x_1 - x_6\|^2 = (1.049952)^2 = 1.1024$$

Untuk hasil perhitungan secara keseluruhan untuk mendapatkan nilai $K(x_i, x^+)$ dan nilai $K(x_i, x^-)$ dapat dilihat pada Tabel 3.12.

Tabel 3.12 Hasil perhitungan $K(x_i, x^+)$ dan $K(x_i, x^-)$

ID	$K(x_i, x^+)$	$K(x_i, x^-)$
1	0.947053	0.576258
2	1.000000	0.634321
3	0.947053	0.576258
4	0.437622	0.689906
5	0.460612	0.726149
6	0.634321	1.000000
7	0.100822	0.334473
8	0.106459	0.338781
9	0.109132	0.340139
10	0.106459	0.338781

Sumber: [Perancangan]

Setelah didapatkan nilai dari masing-masing *class*, tahap selanjutnya adalah menghitung nilai w . Berikut akan dijelaskan perhitungan untuk salah satu data menggunakan Persamaan 2.8.

$$w = \sum_{i=1}^n \alpha_i y_i x_i$$

Data nilai, $\alpha_i = 0.015990$, $y_i = 1$, $K(x_i, x^+) = 0.947053$ maka hasil perhitungan adalah sebagai berikut.

$$w \cdot x^+ = \sum_{i=1}^n \alpha_i y_i x_i x^+ = \sum_{i=1}^n \alpha_i y_i k(x_i, x^+)$$

$$w \cdot x^+ = \begin{bmatrix} 0.015990 \times 1 \times 0.947053 + & 0.016004 \times 1 \times 1.000000 + \\ 0.015990 \times 1 \times 0.947053 + & 0.015712 \times -1 \times 0.437622 + \\ 0.015785 \times -1 \times 0.460612 + & 0.015809 \times -1 \times 0.634321 + \\ 0.015651 \times -1 \times 0.100822 + & 0.015651 \times -1 \times 0.106459 + \\ 0.015652 \times -1 \times 0.109132 + & 0.015652 \times -1 \times 0.106459 \end{bmatrix} \\ = 0.015498$$

Data nilai, $\alpha_i = 0.015990$, $y_i = 1$, $K(x_i, x^-) = 0.009215$ maka hasil perhitungan adalah sebagai berikut.

$$\begin{aligned}
 w \cdot x^- &= \\
 &\begin{bmatrix} 0.015990 \times 1 \times 0.009215 + & 0.016004 \times 1 \times 0.010152 + \\ 0.015990 \times 1 \times 0.009215 + & 0.015712 \times -1 \times -0.010840 + \\ 0.015785 \times -1 \times -0.011462 + & 0.015809 \times -1 \times -0.015809 + \\ 0.015651 \times -1 \times -0.005235 + & 0.015651 \times -1 \times -0.005302 + \\ 0.015652 \times -1 \times -0.005324 + & 0.015652 \times -1 \times -0.005302 \end{bmatrix} \\
 &= -0.030694
 \end{aligned}$$

Untuk hasil perhitungan secara keseluruhan untuk mendapatkan nilai $w \cdot x^+$ dan nilai $w \cdot x^-$ dapat dilihat pada Tabel 3.13.

Tabel 3.13 Hasil perhitungan $w \cdot x^+$ dan $w \cdot x^-$

ID	$w \cdot x^+$	$w \cdot x^-$
1	0.015144	0.009215
2	0.016004	0.010152
3	0.015144	0.009215
4	-0.006876	-0.010840
5	-0.007271	-0.011462
6	-0.010028	-0.015809
7	-0.001578	-0.005235
8	-0.001666	-0.005302
9	-0.001708	-0.005324
10	-0.001666	-0.005302
Σ	0.015498	-0.030694

Sumber: [Perancangan]

Langkah selanjutnya adalah mencari nilai b dengan menggunakan Persamaan 2.8.

$$\begin{aligned}
 b &= -\frac{1}{2}(w \cdot x^+ + w \cdot x^-) \\
 &= -\frac{1}{2}((0.015498) + (-0.030694)) = 0.007597643
 \end{aligned}$$

3.5.1.3 Perhitungan Manual Proses *Testing* SVM

Setelah dilakukan proses *training* langkah selanjutnya adalah melakukan proses *testing* dengan menggunakan data *testing*. Data *testing* sebelumnya dihitung menggunakan *kernel* dan hasil perhitungan dapat dilihat pada Tabel 3.6. Tahap proses *testing* SVM dilakukan untuk mendapatkan nilai $f(x)$ dengan mengimplementasikan Persamaan. Hasil klasifikasi dapat dihitung dengan Persamaan 2.10

$$f(x) = \sum_{i=1}^m \alpha_i y_i K(x_i, x_j) + b$$

dan Fungsi Klasifikasi= $\text{Sign}(f(x))$

Untuk contoh proses perhitungan salah satu data dengan Persamaan 2.10 adalah sebagai berikut. Untuk proses pertama adalah mencari *kernel data testing*. Data yang digunakan untuk data *testing* adalah data pasien ID 16, 32, 39.

a. Perhitungan *testing* untuk pasien ID 16

Pada nilai *record* pertama pasien data *training* = 0.900000 dan nilai pada *record* pertama pada hasil normalisasi data *testing* = 0.900000, maka hasil perhitungan *kernel* pada *record* pasien ID 1 data *training* adalah sebagai berikut.

Pada Tabel 3.14 adalah tabel data *training* dengan ID pasien 1.

Data 1 (x_1)

Tabel 3.14 Data training 1

No.	Keadaan Umum	Nadi	Pernapasan /RR	...	Elastisitas Kulit	Status Gizi	Suhu
1	0.900000	0.260000	0.100000	...	0.900000	0.900000	0.900000

Sumber: [Perancangan]

Pada Tabel 3.15 adalah tabel data *testing* dengan ID pasien 16.

Data 16 (x_{16})

Tabel 3.15 Data testing 1

No.	Keadaan Umum	Nadi	Pernapasan /RR	...	Elastisitas Kulit	Status Gizi	Suhu
1	0.900000	0.292000	0.100000	...	0.900000	0.900000	0.580000

Sumber: [Perancangan]

Misal $\sigma=1$

$k(x_i, x)$, misal $x_i = x_1, x = x_{16}$ maka $k(x_i, x) = k(x_1, x_{16}) =$

$$\exp\left(-\frac{\|x_1 - x_{16}\|^2}{2\sigma^2}\right) = \exp\left(-\frac{0.103424}{2.1^2}\right) = \exp\left(-\frac{0.103424}{2}\right) \\ = \exp(-0.051712) \\ = 0.949602$$

$$(x_1 - x_{16}) = \begin{pmatrix} 0.900000 \\ 0.260000 \\ 0.100000 \\ 0.100000 \\ 0.100000 \\ 0.900000 \\ 0.900000 \\ 0.900000 \\ 0.900000 \\ 0.900000 \end{pmatrix} - \begin{pmatrix} 0.900000 \\ 0.292000 \\ 0.100000 \\ 0.100000 \\ 0.100000 \\ 0.900000 \\ 0.900000 \\ 0.900000 \\ 0.900000 \\ 0.580000 \end{pmatrix}$$

$$= \begin{pmatrix} 0.900000 - 0.900000 \\ 0.260000 - 0.292000 \\ 0.100000 - 0.100000 \\ 0.100000 - 0.100000 \\ 0.100000 - 0.100000 \\ 0.900000 - 0.900000 \\ 0.900000 - 0.900000 \\ 0.900000 - 0.900000 \\ 0.900000 - 0.580000 \end{pmatrix} = \begin{pmatrix} 0 \\ -0.032 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0.32 \end{pmatrix}$$

$$\|x_1 - x_{16}\| = \sqrt{0^2 + (-0.032)^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + (0.32)^2} \\ = \sqrt{0.103424} = 0.321596$$

$$\|x_1 - x_{16}\|^2 = 0.321596^2 = 0.103424$$

Pada nilai *record* kedua pasien data *training* = 0.900000 dan nilai pada *record* pertama pada hasil normalisasi data *testing* = 0.9000, maka hasil perhitungan *kernel* pada *record* pasien ID 2 data *training* adalah sebagai berikut.

Pada Tabel 3.16 adalah tabel data *training* dengan ID pasien 2.

Data 2 (x_2)

Tabel 3.16 Data *training* 2

No.	Keadaan Umum	Nadi	Pernapasan /RR	...	Elastisitas Kulit	Status Gizi	Suhu
1	0.900000	0.580000	0.100000	...	0.900000	0.900000	0.820000

Sumber: [Perancangan]

Misal $\sigma=1$

$k(x_i, x)$, misal $x_i = x_2, x = x_{16}$ maka $k(x_i, x) = k(x_2, x_{16}) =$

$$\exp\left(-\frac{\|x_2 - x_{16}\|^2}{2\sigma^2}\right) = \exp\left(-\frac{0.140544}{2 \cdot 1^2}\right) = \exp\left(-\frac{0.140544}{2}\right) = \exp(-0.070272) \\ = 0.932140$$

$$(x_2 - x_{16}) = \begin{pmatrix} 0.900000 \\ 0.580000 \\ 0.100000 \\ 0.100000 \\ 0.100000 \\ 0.900000 \\ 0.900000 \\ 0.900000 \\ 0.820000 \end{pmatrix} - \begin{pmatrix} 0.900000 \\ 0.292000 \\ 0.100000 \\ 0.100000 \\ 0.100000 \\ 0.900000 \\ 0.900000 \\ 0.900000 \\ 0.580000 \end{pmatrix}$$

$$= \begin{pmatrix} 0.900000 - 0.900000 \\ 0.580000 - 0.292000 \\ 0.100000 - 0.100000 \\ 0.100000 - 0.100000 \\ 0.100000 - 0.100000 \\ 0.900000 - 0.900000 \\ 0.900000 - 0.900000 \\ 0.900000 - 0.900000 \\ 0.820000 - 0.580000 \end{pmatrix} = \begin{pmatrix} 0 \\ 0.288 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0.24 \end{pmatrix}$$

$$\|x_2 - x_{16}\| = \sqrt{0^2 + (0.288)^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + (0.24)^2} \\ = \sqrt{0.140544} = 0.374891$$

$$\|x_2 - x_{16}\|^2 = 0.374891^2 = 0.140544$$

Tahap selanjutnya mencari nilai $\alpha_i y_i K(x_i, x)$. Diketahui data nilai $\alpha_i = 0.015990$ dan nilai $y_i = 1$, maka hasil perhitungannya adalah sebagai berikut.

Perhitungan untuk data *training* pasien ID 1 $= 0.015990 \times 1 \times 0.949602 = 0.015185$

Perhitungan untuk data *training* pasien ID 2 $= 0.015990 \times 1 \times 0.932140 = 0.014918$

Hasil perhitungan pasien ID 16, untuk mendapatkan nilai *kernel* data *testing* dan mencari nilai $\alpha_i, y_i, K(x_i, x)$ dapat dilihat pada Tabel 3.17.

Tabel 3.17 Hasil perhitungan $K(x_i, x)$ dan $\alpha_i y_i K(x_i, x)$

ID	$K(x_i, x)$	$\alpha_i y_i K(x_i, x)$
1	0.949602	0.015185
2	0.932140	0.014918
3	0.949602	0.015185
4	0.421459	-0.006622
5	0.486425	-0.007678
6	0.669870	-0.010590
7	0.128268	-0.002008
8	0.132864	-0.002079
9	0.134899	-0.002111
10	0.132864	-0.002079
Σ		0.012119

Sumber: [Perancangan]

Proses selanjutnya adalah menghitung nilai $f(x)$ dengan menggunakan Persamaan 2.14. Kemudian menghitung fungsi klasifikasi dengan menggunakan Persamaan 2.11.

$$f(x) = \sum_{i=1}^m \alpha_i y_i x_i \cdot x + b$$

$$= 0.012119 + 0.007597643 = 0.019717$$

Fungsi Klasifikasi = $\text{sign}(0.019717) = 1$

Tahap ini dilakukan untuk seluruh data *testing*, dan hasil akhir dari seluruh data adalah seperti pada Tabel 3.19.

b. Perhitungan *testing* untuk pasien ID 32

Pada nilai *record* pertama pasien data *training* = 0.900000 dan nilai pada *record* pertama pada hasil normalisasi data *testing* = 0.500000, maka hasil perhitungan *kernel* pada *record* pasien ID 1 data *training* adalah sebagai berikut.

Pada Tabel 3.18 adalah tabel data *testing* dengan ID pasien 32.

Data 32 (x_{32})

Tabel 3.18 Data testing 2

No.	Keadaan Umum	Nadi	Pernapasan /RR	...	Elastisitas Kulit	Status Gizi	Suhu
32	0.500000	0.580000	0.260000	...	0.500000	0.500000	0.380000

Sumber: [Perancangan]

Misal $\sigma=1$

$k(x_i, x)$, misal $x_i = x_1, x = x_{32}$ maka $k(x_i, x) = k(x_1, x_{32}) =$

$$\exp\left(-\frac{\|x_1 - x_{32}\|^2}{2\sigma^2}\right) = \exp\left(-\frac{1.0384}{2 \cdot 1^2}\right) = \exp\left(-\frac{1.0384}{2}\right) = \exp(-0.5192)$$

$$= 0.594996$$

$$(x_1 - x_{32}) = \begin{pmatrix} 0.900000 \\ 0.260000 \\ 0.100000 \\ 0.100000 \\ 0.100000 \\ 0.900000 \\ 0.900000 \\ 0.900000 \\ 0.900000 \end{pmatrix} - \begin{pmatrix} 0.500000 \\ 0.580000 \\ 0.260000 \\ 0.100000 \\ 0.100000 \\ 0.500000 \\ 0.500000 \\ 0.500000 \\ 0.380000 \end{pmatrix}$$

$$= \begin{pmatrix} 0.900000 - 0.500000 \\ 0.260000 - 0.580000 \\ 0.100000 - 0.260000 \\ 0.100000 - 0.100000 \\ 0.100000 - 0.100000 \\ 0.900000 - 0.500000 \\ 0.900000 - 0.500000 \\ 0.900000 - 0.500000 \\ 0.900000 - 0.380000 \end{pmatrix} = \begin{pmatrix} 0.4 \\ -0.32 \\ -0.16 \\ 0 \\ 0 \\ 0.4 \\ 0.4 \\ 0.4 \\ 0.52 \end{pmatrix}$$

$$\|x_1 - x_{32}\| = \sqrt{\frac{(0.4)^2 + (-0.32)^2 + (-0.16)^2 + 0^2 + 0^2 + (0.4)^2 + (0.4)^2 + (0.4)^2 + (0.52)^2}{2}}$$

$$= \sqrt{1.0384} = 1.019019$$

$$\|x_1 - x_{32}\|^2 = 1.019019^2 = 1.0384$$

Pada nilai *record* kedua pasien data *training* = 0.900000 dan nilai pada *record* kedua pada hasil normalisasi data *testing* = 0.500000, maka hasil perhitungan *kernel* pada *record* pasien ID 2 data *training* adalah sebagai berikut.

Misal $\sigma=1$

$k(x_i, x)$, misal $x_i = x_2, x = x_{32}$ maka $k(x_i, x) = k(x_2, x_{32}) =$

$$\exp\left(-\frac{\|x_2 - x_{32}\|^2}{2\sigma^2}\right) = \exp\left(-\frac{0.8592}{2 \cdot 1^2}\right) = \exp\left(-\frac{0.8592}{2}\right) = \exp(-0.4296) = 0.650769$$

$$(x_2 - x_{32}) = \begin{pmatrix} 0.900000 \\ 0.580000 \\ 0.100000 \\ 0.100000 \\ 0.100000 \\ 0.900000 \\ 0.900000 \\ 0.900000 \\ 0.820000 \end{pmatrix} - \begin{pmatrix} 0.500000 \\ 0.580000 \\ 0.260000 \\ 0.100000 \\ 0.100000 \\ 0.500000 \\ 0.500000 \\ 0.500000 \\ 0.380000 \end{pmatrix} = \begin{pmatrix} 0.900000 - 0.500000 \\ 0.580000 - 0.580000 \\ 0.100000 - 0.260000 \\ 0.100000 - 0.100000 \\ 0.100000 - 0.100000 \\ 0.900000 - 0.500000 \\ 0.900000 - 0.500000 \\ 0.900000 - 0.500000 \\ 0.820000 - 0.380000 \end{pmatrix} = \begin{pmatrix} 0.4 \\ 0 \\ -0.16 \\ 0 \\ 0 \\ 0.4 \\ 0.4 \\ 0.4 \\ 0.44 \end{pmatrix}$$

$$\|x_2 - x_{32}\| = \sqrt{\frac{(0.4)^2 + 0^2 + (-0.16)^2 + 0^2 + 0^2 + (0.4)^2 + (0.4)^2 + (0.4)^2 + (0.44)^2}{2}}$$

$$= \sqrt{0.8592} = 0.92693$$

$$\|x_2 - x_{32}\|^2 = 0.92693^2 = 0.8592$$

Tahap selanjutnya mencari nilai $\alpha_i y_i K(x_i, x)$. Diketahui data nilai $\alpha_i = 0.015990$ dan nilai $y_i = 1$, maka hasil perhitungannya adalah sebagai berikut.

$$\begin{aligned} \text{Perhitungan untuk data } \textit{training} \text{ pasien ID 1} &= 0.015990 \times 1 \times 0.594996 \\ &= 0.009514 \end{aligned}$$

$$\begin{aligned} \text{Perhitungan untuk data } \textit{training} \text{ pasien ID 2} &= 0.016004 \times 1 \times 0.650769 \\ &= 0.010415 \end{aligned}$$

Hasil perhitungan pasien ID 32, untuk mendapatkan nilai *kernel* data *testing* dan mencari nilai $\alpha_i y_i K(x_i, x)$ dapat dilihat pada Tabel 3.19.

Tabel 3.19 hasil perhitungan $K(x_i, x)$ dan $\alpha_i y_i K(x_i, x)$

ID	$K(x_i, x)$	$\alpha_i y_i K(x_i, x)$
1	0.594996	0.009514
2	0.650769	0.010415
3	0.594996	0.009514
4	0.771669	-0.012124
5	0.812207	-0.012821
6	0.984127	-0.015558
7	0.323939	-0.005070
8	0.330219	-0.005168
9	0.332605	-0.005206
10	0.330219	-0.005168
Σ		-0.031672

Sumber: [Perancangan]

Proses selanjutnya adalah menghitung nilai $f(x)$ dengan menggunakan Persamaan 2.14. Kemudian menghitung fungsi klasifikasi dengan menggunakan Persamaan 2.11.

$$f(x) = \sum_{i=1}^m \alpha_i y_i x_i \cdot x + b$$

$$= 0.031672 + 0.007597643 = -0.024075$$

Fungsi Klasifikasi = $\text{sign}(-0.024075) = -1$

Proses perhitungan untuk mendapatkan hasil klasifikasi dilakukan untuk keseluruhan data *testing* yang telah terpilih. Hasil klasifikasi dengan nilai 1 adalah termasuk kelompok pasien dengan derajat dehidrasi ringan. Sedangkan hasil klasifikasi dengan nilai -1 termasuk dalam kelompok pasien yang menderita dehidrasi dengan derajat antara sedang sampai berat dan harus dilakukan perhitungan untuk level 2. Pada Tabel 3.20 akan dijelaskan *actual class* dan hasil klasifikasi pasien dehidrasi dari perhitungan data *testing*.

Tabel 3.20 Data hasil perhitungan $f(x)$ dan klasifikasi

ID Pasien	$f^1(x)$	$f^2(x)$	Hasil klasifikasi	Actual Class
16	0.019717	-	1 dilevel 1 = Dehidrasi ringan	ringan
32	-0.024075	0.036811	1 dilevel 2 = Dehidrasi sedang	sedang
36	-0.062058	-0.033840	-1 dilevel 2 = Dehidrasi berat	berat

Sumber: [Perancangan]

3.5.1.4 Perhitungan Manual Nilai Akurasi

Mengukur performansi dan mengukur tingkat keberhasilan atau membandingkan hasil dari proses implementasi yang telah dilakukan dengan standar yang telah ditetapkan dilakukan proses evaluasi atau pengujian sistem. Proses pengujian dilakukan untuk menentukan nilai akurasi sistem dari nilai yang diklasifikasikan secara tepat. Salah satu model evaluasi adalah dengan membuat tabel *confusion matriks* seperti pada Tabel 3.21.

Tabel 3.21 Model evaluasi *confusion matriks*

Actual Class	Predicted Class			Jumlah
	ringan	sedang	berat	
ringan	1	0	0	1
sedang	0	1	0	1
berat	0	0	1	1
Jumlah	1	1	1	3

Sumber: [Perancangan]

Untuk menghitung nilai akurasi sistem dilakukan perhitungan dengan menggunakan persamaan 2.24.

$$\text{Akurasi} = \frac{TP + TN}{TP + TN + FP + FN}$$

Maka nilai akurasi yang diperoleh adalah

- $TP = C_{1,1} + C_{2,2} + C_{3,3} = 1 + 1 + 1 = 3$
- $TN = (C_{1,1} + C_{2,2}) + (C_{1,1} + C_{3,3}) + (C_{2,2} + C_{3,3}) = 2 + 2 + 2 = 6$
- $FP = (C_{2,1} + C_{3,1}) + (C_{1,2} + C_{3,2}) + (C_{1,3} + C_{2,3}) = 0 + 0 + 0 = 0$
- $FN = (C_{1,2} + C_{1,3}) + (C_{2,1} + C_{2,3}) + (C_{3,1} + C_{3,2}) = 0 + 0 + 0 = 0$

$$\text{Akurasi} = \frac{3+6}{3+6+0+0} \times 100\% = 100\%$$

3.5.2 Perancangan Antarmuka

Rancangan antarmuka dari aplikasi ini ada 4 (empat) *form*. *Form* pertama digunakan untuk menjalankan proses *load data training*, *load data testing*, perhitungan SVM level 1, pengujian dan ditampilkan pada *form* Level 1, dapat dilihat pada Gambar 3.12. Sedangkan untuk *form* yang kedua digunakan untuk menjalankan proses perhitungan SVM level 2 dan ditampilkan pada *form* Level 2, dapat dilihat pada Gambar 3.12. *Form* ketiga digunakan untuk menampilkan data *testing* dan ditampilkan pada *form* Data *Testing*, dapat dilihat pada Gambar 3.12. *Form* keempat digunakan untuk menampilkan tabel hasil perhitungan $f(x)$ dan klasifikasi, serta hasil akurasi yang ditampilkan pada *form* Hasil, dapat dilihat pada Gambar 3.12.

Gambar 3.12 Form perhitungan SVM level 1

Sumber: [Perancangan]

Keterangan:

1. *Tab Level 1* digunakan untuk menampilkan perhitungan SVM level 1.
2. *Button browse* digunakan untuk memilih data *training* dan data *testing* dengan format .csv.
3. *Button upload*, digunakan untuk memasukkan data *testing* ke dalam *database*.

Pada Gambar 3.13 adalah gambar perancangan antarmuka *form* perhitungan SVM Level 2.

Gambar 3.13 Form perhitungan SVM level 2

Sumber: [Perancangan]

Keterangan:

1. *Tab Level 2* digunakan untuk menampilkan perhitungan SVM level 2.

Pada Gambar 3.14 adalah gambar perancangan antarmuka *form* data *testing*.

IMPLEMENTASI METODE <i>SUPPORT VECTOR MACHINE</i> UNTUK REKOMENDASI PEMILIHAN TERAPI DEHIDRASI PADA ANAK			
Level1	Level2	Data Testing	Hasil

TABEL

Gambar 3.14 Form data testing
Sumber: [Perancangan]

Keterangan:

1. *Tab Data Testing* digunakan untuk menampilkan seluruh data *testing*.
Pada Gambar 3.15 adalah gambar perancangan antarmuka *form* hasil.

IMPLEMENTASI METODE <i>SUPPORT VECTOR MACHINE</i> UNTUK REKOMENDASI PEMILIHAN TERAPI DEHIDRASI PADA ANAK			
Level1	Level2	Data Testing	Hasil

TABEL

Gambar 3.15 Form hasil
Sumber: [Perancangan]

Keterangan:

1. *Tab Hasil* digunakan untuk menampilkan hasil pengujian yang berisi tabel hasil perhitungan $f(x)$, *confusion matriks* dan nilai akurasi.

3.6 Pengujian Sistem

Pada tahap ini adalah pengujian sistem. Pengujian ini dilakukan untuk mengetahui tingkat akurasi klasifikasi menggunakan *Support Vector Machine*. Pengujian dilakukan secara *sequential*. Adapun pengujian yang dilakukan adalah sebagai berikut:

1. Pengujian terhadap data *training* dan data *testing* berdasarkan rasio perbandingan.
2. Pengujian terhadap parameter *Kernel Gaussian* RBF.
3. Pengujian terhadap parameter *Sequential Training* SVM.

3.6.1 Pengujian terhadap Rasio Perbandingan

Pengujian berdasarkan data *training* dan data *testing* digunakan untuk melihat pada rasio perbandingan data berapa mendapatkan akurasi terbaik.. *Dataset* yang digunakan sebanyak 150 data yang terbagi berdasarkan jumlah perbandingan data *training* dan data *testing*. Rasio perbandingan yang digunakan adalah 90%:10%, 80%:20%, 70%:30%, 60%:40%, 50%:50%, 40%:60%, 30%:70%, 20%:80% dan 10%:90%. Nilai parameter *Sequential Training* SVM yang digunakan dalam pengujian ini adalah $\lambda = 0.5$, $\gamma = 0.01$, $\varepsilon = 0.0001$, $itermax = 100$, $C = 1$. Adapun perancangannya ditunjukkan pada Tabel 3.22.

Tabel 3.22 Pengaruh rasio perbandingan terhadap rata-rata akurasi

Percobaan ke- <i>i</i>	Rasio Perbandingan Data <i>Training</i> dan Data <i>Testing</i> (%)								
	90:10	80:20	70:30	60:40	50:50	40:60	30:70	20:80	10:90
1.									
2.									
3.									
4.									
5.									
Rerata									

Sumber: [Perancangan]

3.6.2 Pengujian terhadap Parameter *Kernel Gaussian* RBF

Pengujian terhadap parameter *Kernel Gaussian* RBF dilakukan untuk melihat pada skenario mana yang mendapatkan akurasi terbaik. Skenario untuk nilai σ *sigma* yaitu 0.5, 0.6, 0.7, 0.8, 0.9, 1, 2, 3, 4 dan 5. Adapun perancangannya ditunjukkan pada Tabel 3.23. Nilai *default* yang digunakan yaitu 1.

Tabel 3.23 Pengaruh nilai σ (*sigma*) terhadap rata-rata akurasi

Percobaan ke- <i>i</i>	Nilai σ (<i>sigma</i>)									
	0.5	0.6	0.7	0.8	0.9	1	2	3	4	5
1.										
2.										
3.										
4.										
5.										
Rerata										

Sumber: [Perancangan]

3.6.3 Pengujian terhadap Parameter *Sequential Training SVM*

Pengujian terhadap parameter *sequential training SVM* dilakukan untuk melihat pada skenario mana yang mendapatkan nilai terbaik untuk nilai λ (*lamda*) yaitu 0.1, 0.3, 0.5, 1, 1.5, 2, 2.5, 3, 3.5 dan 4. Adapun perancangannya ditunjukkan pada Tabel 3.24. Nilai *default* yang digunakan yaitu 0.5.

Tabel 3.24 Pengaruh nilai λ (*lamda*) terhadap rata-rata akurasi

Percobaan ke- <i>i</i>	Nilai λ (<i>lamda</i>)									
	0.1	0.3	0.5	1	1.5	2	2.5	3	3.5	4
1.										
2.										
3.										
4.										
5.										
Rerata										

Sumber: [Perancangan]

Pengujian selanjutnya digunakan untuk melihat pada skenario mana yang mendapatkan nilai terbaik untuk konstanta γ (*gamma*) yaitu $1 \cdot 10^{-5}$, $1 \cdot 10^{-4}$, $5 \cdot 10^{-4}$, $1 \cdot 10^{-3}$, $5 \cdot 10^{-3}$, $1 \cdot 10^{-2}$, $5 \cdot 10^{-2}$, $1 \cdot 10^{-1}$, $5 \cdot 10^{-1}$ dan 1. Adapun perancangannya ditunjukkan pada Tabel 3.25. Nilai *default* yang digunakan yaitu 0.01.

Tabel 3.25 Pengaruh konstanta γ (*gamma*) terhadap rata-rata akurasi

Percobaan ke- <i>i</i>	Konstanta γ (<i>gamma</i>)									
	$1 \cdot 10^{-5}$	$1 \cdot 10^{-4}$	$5 \cdot 10^{-4}$	$1 \cdot 10^{-3}$	$5 \cdot 10^{-3}$	$1 \cdot 10^{-2}$	$5 \cdot 10^{-2}$	$1 \cdot 10^{-1}$	$5 \cdot 10^{-1}$	1
1.										
2.										
3.										
4.										
5.										
Rerata										

Sumber: [Perancangan]

Pengujian selanjutnya digunakan untuk melihat pada skenario mana yang mendapatkan nilai terbaik untuk nilai ϵ (*epsilon*) yaitu $1 \cdot 10^{-9}$, $1 \cdot 10^{-8}$, $1 \cdot 10^{-7}$, $1 \cdot 10^{-6}$, $1 \cdot 10^{-5}$, $1 \cdot 10^{-4}$, $5 \cdot 10^{-4}$, $1 \cdot 10^{-3}$, $5 \cdot 10^{-3}$ dan $1 \cdot 10^{-2}$. Adapun perancangannya ditunjukkan pada Tabel 3.26. Nilai *default* yang digunakan yaitu $1 \cdot 10^{-4}$.

Tabel 3.26 Pengaruh nilai ϵ (*epsilon*) terhadap rata-rata akurasi

Percobaan ke- <i>i</i>	Nilai ϵ (<i>epsilon</i>)									
	1.10^{-9}	1.10^{-8}	1.10^{-7}	1.10^{-6}	1.10^{-5}	1.10^{-4}	5.10^{-4}	1.10^{-3}	5.10^{-3}	1.10^{-2}
1.										
2.										
3.										
4.										
5.										
Rerata										

Sumber: [Perancangan]

Pengujian selanjutnya digunakan untuk melihat pada skenario mana yang mendapatkan nilai terbaik untuk nilai Maksimum iterasi (*itermax*) yaitu 100, 200, 300, 400, 500, 550, 600, 650, 700 dan 750. Adapun perancangannya ditunjukkan pada Tabel 3.27. Nilai *default* yang digunakan yaitu 100.

Tabel 3.27 Pengaruh nilai (*itermax*) terhadap rata-rata akurasi

Percobaan ke- <i>i</i>	Maksimum iterasi (<i>itermax</i>)									
	100	200	300	400	500	550	600	650	700	750
1.										
2.										
3.										
4.										
5.										
Rerata										

Sumber: [Perancangan]

Pengujian selanjutnya digunakan untuk melihat pada skenario mana yang mendapatkan nilai terbaik untuk nilai *C* (*complexity*) yaitu 1, 10, 20, 30, 40, 50, 60, 70, 80 dan 90. Adapun perancangannya ditunjukkan pada Tabel 3.28. Nilai *default* untuk parameter *C* (*complexity*) yang digunakan yaitu 1.

Tabel 3.28 Pengaruh nilai *c* (*complexity*) terhadap rata-rata akurasi

Percobaan ke- <i>i</i>	Nilai <i>C</i> (<i>complexity</i>)									
	1	10	20	30	40	50	60	70	80	90
1.										
2.										
3.										
4.										
5.										

Tabel 3.29 Pengaruh nilai C (*complexity*) terhadap rata-rata akurasi (lanjutan)

Percobaan ke- i	Nilai C (<i>complexity</i>)									
	1	10	20	30	40	50	60	70	80	90
Rerata										

Sumber: [Perancangan]

3.7 Evaluasi Hasil

Hasil evaluasi digunakan untuk mengevaluasi dari kinerja sistem. Proses evaluasi dilakukan dengan 3 tahap. Tahap pertama adalah mencari nilai akurasi terbaik dari rasio perbandingan data *testing* dan *training*, dimana ada 9 variasi data yang berbeda. Tahap kedua adalah mencari nilai akurasi terbaik dari parameter *kernel* RBF, yaitu σ (*sigma*). Tahap terakhir adalah mencari nilai akurasi terbaik dari parameter *sequential training* SVM, yaitu nilai λ (*lamda*), γ (*gamma*), ϵ (*epsilon*), iterasi maksimum (*itermax*), C (*complexity*). Semua tahapan evaluasi hasil dilakukan secara *sequential* dari evaluasi rasio perbandingan data sampai evaluasi parameter *sequential training* SVM.



BAB 4 IMPLEMENTASI

Bab ini membahas mengenai implementasi perangkat lunak berdasarkan perancangan yang dibuat. Isi dari bab implementasi meliputi spesifikasi sistem yang dibangun, batasan-batasan implementasi, implementasi algoritma, dan implementasi antarmuka.

4.1 Spesifikasi Sistem

Proses implementasi sistem membutuhkan spesifikasi perangkat yang sesuai agar sistem yang dibangun dapat berfungsi sesuai dengan kebutuhan. Spesifikasi perangkat yang dibutuhkan oleh sistem terdiri dari spesifikasi perangkat keras dan spesifikasi perangkat lunak.

4.1.1 Spesifikasi Perangkat Keras

Pengembangan implementasi metode SVM untuk rekomendasi pemilihan terapi dehidrasi pada anak menggunakan perangkat keras dengan spesifikasi pada Tabel 4.1.

Tabel 4.1 Spesifikasi perangkat keras

Komponen	Spesifikasi
Prosesor	Intel® Core™ i3-3217U CPU @ 1.80GHz
Memori	5.89 GB
Kartu Grafis	Intel® HD Graphic Family
Harddisk	500 GB

Sumber: [Implementasi]

4.1.2 Spesifikasi Perangkat Lunak

Pengembangan implementasi metode SVM untuk rekomendasi pemilihan terapi dehidrasi pada anak menggunakan perangkat lunak dengan spesifikasi pada Tabel 4.2.

Tabel 4.2 Spesifikasi perangkat lunak

Nama	Spesifikasi
Sistem operasi	Microsoft Windows 8.1 64-bit
Bahasa pemrograman	PHP
Tools Pemrograman	Notepad++
Database	MySQL
Tools DBMS	MySql 1.7.7

Sumber: [Implementasi]

4.2 Batasan Implementasi

Beberapa batasan yang digunakan dalam mengimplementasikan metode SVM untuk rekomendasi pemilihan terapi dehidrasi pada anak sebagai berikut:

1. Sistem dibangun aplikasi berbasis *web* menggunakan Bahasa pemrograman PHP.
2. Penyimpanan data menggunakan DBMS MySQL.
3. Metode yang digunakan dalam menyelesaikan masalah adalah metode *Support Vector Machine* (SVM).
4. Data masukan yang digunakan dalam sistem adalah data pasien Ruang Anggrek RSUD Dr. Soegiri Lamongan Tahun 2015.
5. Hasil *output* dari hasil perhitungan metode SVM adalah klasifikasi terapi berdasarkan kelas yang di rankingkan.
6. Kriteria yang digunakan di penelitian ini sejumlah 9 kriteria.

4.3 Implementasi Algoritma

Implementasi metode SVM untuk rekomendasi pemilihan terapi dehidrasi pada anak, dengan 7 algoritma yang akan dikerjakan seperti yang dijelaskan pada diagram implementasi. Berikut akan dijelaskan implementasi algoritma masing-masing.

4.3.1 Implementasi Algoritma Perhitungan *Kernel*

Algoritma perhitungan *kernel* adalah proses awal perhitungan menggunakan metode SVM yang melibatkan *dot product* antar 2 data *training* pada ruang *vector* berdimensi. *Kernel* yang digunakan adalah *kernel* Radial Basis Function (RBF). Keluaran yang dihasilkan berupa matriks *kernel* dengan index $n \times n$, dan n merupakan banyaknya data *training*. Kode program algoritma *kernel* dijelaskan pada Gambar 4.1.

```

1.   for ($i=0;$i<$jmlData;$i++){
2.       for ($j=0;$j<$jmlData;$j++){
3.           $temp=0;
4.           for ($k=0;$k<9;$k++){
5.               $temp=$temp+pow($baru[$i][$k]-$baru[$j][$k],2);
6.           }
7.           $matriks [$i][$j] = pow (2.718282, (-1 * pow(sqrt($temp),2)/
8.               2*(pow($sigma,2))))
9.           $matriks [$jmlData][$j]=$baru[$j][9];
10.        }
11.        $matriks [$i][$jmlData]=$baru[$i][9];
12.    }

```

Gambar 4.1 Implementasi algoritma perhitungan *kernel* RBF

Sumber: [Implementasi]

Berikut penjelasan dari Gambar 4.1:

1. Baris 1 sampai baris 2 menjelaskan tentang perulangan untuk data latih ke- i dan ke- j .

2. Baris 4 sampai baris 5 menjelaskan tentang operasi pengurangan nilai tiap fitur.
3. Baris 7 sampai 8 menjelaskan tentang rumus *kernel* RBF.

4.3.2 Implementasi Algoritma Perhitungan *Matriks Hessian*

Proses ini dilakukan untuk mencari nilai *hessian* yang nanti akan digunakan untuk perhitungan pada proses algoritma *Sequential Training SVM*. Variabel dari perhitungan *matriks hessian* didapat dari hasil perhitungan fungsi *kernel* dan nilai dari konstanta *lambda*. Berikut kode program yang dijelaskan pada Gambar 4.2.

```

1.   for ($i=0;$i<$jmlData;$i++){
2.       for ($j=0;$j<$jmlData;$j++){
3.           $hessian[$i][$j]=
4.           ($diagnosa[$i]*$diagnosa[$j])*( $matriks[$i][$j]+pow($lambda,2));
5.       }
6.   }
```

Gambar 4.2 Implementasi algoritma perhitungan matriks *hessian*

Sumber: [Implementasi]

Berikut penjelasan dari Gambar 4.2:

1. Baris 1 sampai baris 2 menjelaskan tentang perulangan untuk data latih ke-*i* dan ke-*j*.
2. Baris 3 sampai baris 4 menjelaskan tentang rumus matriks *hessian*.

4.3.3 Implementasi Algoritma Perhitungan Iterasi

Proses iterasi menghitung sejumlah iterasi maksimum. Iterasi pertama melakukan perhitungan mencari nilai E_i untuk setiap data *training*-nya, setelah itu menghitung nilai $\delta\alpha_i$ dan yang terakhir proses menghitung α_i baru.

a. Implementasi Algoritma Perhitungan E_i

Proses ini merupakan proses pertama pada tahap iterasi. Pada proses ini perhitungan nilai E_i dilakukan dengan menjumlahkan hasil perkalian dari matriks *hessian* dengan alpha ke-*i*. Proses ini merupakan proses awal iterasi dari *Sequential Training SVM*. Berikut kode program perhitungan iterasi E_i pada Gambar 4.3.

```

1.   $ai[$i]=0;
2.   for ($i=0;$i<$jmlData;$i++){
3.       for ($j=0;$j<$jmlData;$j++){
4.           $iter1 [$i][$j] = $hessian[$i][$j]*$ai[$j];
5.           $total=$total+$iter1[$i][$j];
6.       }
7.   }
```

Gambar 4.3 Implementasi algoritma perhitungan E_i

Sumber: [Implementasi]

Berikut penjelasan dari Gambar 4.3:

1. Baris 1 menjelaskan tentang inisialisasi nilai α_i .
2. Baris 2 sampai baris 3 menjelaskan tentang perulangan untuk data latih ke-*i* dan ke-*j*.

3. Baris 4 sampai baris 5 menjelaskan tentang rumus E_i .

b. Implementasi Algoritma Perhitungan Nilai $\delta\alpha_i$

Proses perhitungan untuk mencari nilai $\delta\alpha_i$ yaitu melakukan perkalian antara gamma dengan hasil dari 1 dikurangi dengan nilai E_i ke-i dan dibandingkan dengan nilai alpha ke-i untuk dicari nilai paling maksimum. Hasil nilai maksimum dibandingkan dengan hasil dari variable C dikurangi dengan nilai alpha ke-i. Berikut kode program perhitungan nilai $\delta\alpha_i$ pada Gambar 4.4.

1.	$\$gamma2 = \$gamma / \$hessian[0][0];$
2.	for ($\$i=0; \$i < \$jmlData; \$i++$) {
3.	$\$etalpa[\$i] = \min(\max(\$gamma2 * (1 - \$iter1[\$i][\$jmlData+1]), -\$ai[\$i]), (\$C - \$ai[\$i]));$
4.	}
5.	

Gambar 4.4 Implementasi algoritma perhitungan nilai $\delta\alpha_i$
Sumber: [Implementasi]

Berikut penjelasan dari Gambar 4.4:

1. Baris 1 menjelaskan tentang inisialisasi nilai γ .
2. Baris 2 menjelaskan tentang perulangan untuk data latih ke-i.
3. Baris 3 sampai baris 5 menjelaskan tentang rumus $\delta\alpha_i$.

c. Implementasi Algoritma Perhitungan Nilai α_i Baru

Perhitungan untuk menentukan nilai α_i baru yaitu dengan menambahkan nilai dari $\delta\alpha_i$ dengan nilai α_i sebelumnya. Berikut kode program perhitungan nilai α_i baru pada Gambar 4.5.

1.	for ($\$i=0; \$i < \$jmlData; \$i++$) {
2.	$\$ai[\$i] = \$etalpa[\$i] + \$ai[\$i];$
3.	}

Gambar 4.5 Implementasi algoritma perhitungan nilai α_i baru
Sumber: [Implementasi]

Berikut penjelasan dari Gambar 4.5:

1. Baris 1 menjelaskan tentang perulangan untuk data latih ke-i.
2. Baris 2 sampai baris 3 menjelaskan tentang rumus menghitung α_i baru.

4.3.4 Implementasi Algoritma Perhitungan Nilai wx^+ dan wx^-

Data training memiliki bobot sejumlah 2, yaitu bobot *training* di-dot *product* dengan nilai data *training* yang memiliki nilai alpha tertinggi pada kelas negatif (wx^-) dan bobot data *training* di-dot *product* dengan nilai data *training* yang memiliki nilai alpha tertinggi pada kelas positif (wx^+). Proses bobot negatif yaitu melakukan perhitungan menggunakan kernel dari data *training* yang memiliki kelas negatif dengan nilai alpha tertinggi dikelas negatif. Proses bobot positif melakukan perhitungan menggunakan *kernel* RBF dari data *training* yang memiliki kelas positif dengan nilai alpha tertinggi di kelas positif. Berikut kode program perhitungan nilai alpha baru pada Gambar 4.6.

```

1. $temp=0;
2. for ($i=0;$i<$jmlData;$i++){
3.     if($diagnosa[$i]==1){
4.         if($ai[$i]>$temp){
5.             $xmax=$i;
6.             $temp= $ai[$i];
7.         }
8.     }
9. }
10. $temp=0;
11. for ($i=0;$i<$jmlData;$i++){
12.     if($diagnosa[$i]==-1){
13.         if($ai[$i]>$temp){
14.             $xmin=$i;
15.             $temp= $ai[$i];
16.         }
17.     }
18. }
19. for ($i=0;$i<$jmlData;$i++){
20.     for ($xmax=0;$xmax<$jmlData;$xmax++){
21.         $temp=0;
22.         for ($k=0;$k<9;$k++){
23.             $temp=$temp+pow($baru[$i][$k]-$xmax[$j][$k],2);
24.         }
25.         $matriks [$i][$j] = pow (2.718282, (-1 * pow(sqrt($temp),2)/
26.             2*(pow($sigma,2))))
27.         $matriks [$jmlData][$j]=$xmax[$j][9];
28.     }
29.     $matriks [$i][$jmlData]=$baru[$i][9];
30. }
31. for ($i=0;$i<$jmlData;$i++){
32.     for ($xmax=0;$xmin<$jmlData;$xmin++){
33.         $temp=0;
34.         for ($k=0;$k<9;$k++){
35.             $temp=$temp+pow($baru[$i][$k]-$xmin[$j][$k],2);
36.         }
37.         $matriks [$i][$k] = pow (2.718282, (-1 * pow(sqrt($temp),2)/
38.             2*(pow($sigma,2))))
39.         $matriks [$jmlData][$j]=$xmin[$j][9];
40.     }
41.     $matriks [$i][$jmlData]=$baru[$i][9];
42. }
43. for ($i=0;$i<$jmlData;$i++){
44.     ($Wx[$i][0]=$ai[$i]*$diagnosa[$i]*$matriks[$i][$j],6);
45.     $total=$total+$Wx[$i][0];
46.     ($Wx[$i][0]=$ai[$i]*$diagnosa[$i]*$matriks[$i][$k],6);
47.     $total=$total+$Wx[$i][0];
48. }
49. $b=($total+$total1)/-2;

```

Gambar 4.6 Implementasi algoritma perhitungan nilai wx^+ dan wx^-
Sumber: [Implementasi]

Berikut penjelasan dari Gambar 4.6:

1. Baris 1 sampai 9 menjelaskan pengambilan data yang memiliki kelas level positif (1).
2. Baris 10 sampai 18 menjelaskan pengambilan data yang memiliki kelas level negatif (-1).
3. Baris 19 sampai 30 menjelaskan perhitungan $k(x_i, x^+)$.
4. Baris 31 sampai 42 menjelaskan perhitungan $k(x_i, x^-)$.
5. Baris 43 sampai 48 menjelaskan perhitungan wx^+ dan wx^- .
6. Baris 49 menjelaskan proses perhitungan nilai bias.

4.3.5 Implementasi Algoritma Perhitungan Nilai $f(x)$

Perhitungan nilai fungsi linier $f(x)$ ditahap pengujian yaitu menggunakan data *testing* dengan parameter nilai bias. Hasil dari nilai fungsi linier ini adalah nilai positif atau negatif. Jika hasil nilai positif, maka data yang diuji masuk di kelas 1, jika hasil bernilai negatif maka akan diproses lagi di level 2 yang nantinya akan menghasilkan nilai $f(x)$ lagi untuk menentukan masuk di kelas 2 atau kelas 3. Berikut kode program perhitungan nilai α_i baru pada Gambar 4.8.

```

1.  for ($i=0;$i<$jmlData;$i++){
2.      for ($j=0;$j<$jmlData;$j++){
3.          $temp=0;
4.          for ($k=0;$k<9;$k++){
5.              $temp=$temp+pow($baru[$i][$k]-$baru[$j][$k],2);
6.          }
7.          $matriks [$i][$j] = pow (2.718282, (-1 * pow(sqrt($temp),2)/
8.              2*(pow($sigma,2))))
9.          $matriks [$jmlData][$j]=$baru[$j][9];
10.     }
11.     $matriks [$i][$jmlData]=$baru[$i][9];
12. }
13. for ($i=0;$i<$jmlData;$i++){
14.     ($matriks [$i][$j] +1)=$ai[$i]*$diagnosa[$i]* $matriks [$i][$j],6);
15.     $total2=$total2+$matriks [$i][$j];
16. }
17. $fxlvl1[$j]=$total2;
18. $fxlvl1[$j]+=$b;
19. if($total2<0){
20.     $diagnosaujilvl1[$j]=-1;
21. }
22. else{
23.     $diagnosaujilvl1[$j]=1;
24. }

```

Gambar 4.7 Implementasi algoritma perhitungan nilai fungsi linier $f(x)$

Sumber: [Implementasi]

Berikut penjelasan dari Gambar 4.8:

1. Baris 1 sampai baris 12 menjelaskan perhitungan nilai $k(x_i, x_j)$.
2. Baris 13 sampai baris 24 menjelaskan perhitungan nilai $f(x)$.

4.3.6 Implementasi Algoritma *Confusion Matrix*

Perhitungan nilai *confusion matrix* ditahap pengujian yaitu mencocokkan nilai hasil dari *predicted class* dengan *actual class*. Berikut kode program perhitungan *confusion matrix* pada Gambar 4.9.

```

1. $matconf=array(array("ringan",0,0,0,0),array("sedang",0,0,0,0),
2. array("berat",0,0,0,0),array("jumlah",0,0,0,0));
3. for ($i=0;$i<$jmlDataUji;$i++){
4.     if($conf[$i][1]==1&&$conf[$i][2]==1){
5.         $matconf[0][1]++; }
6.     else if($conf[$i][1]==1&&$conf[$i][2]==2){
7.         $matconf[0][2]++; }
8.     else if($conf[$i][1]==1&&$conf[$i][2]==3){
9.         $matconf[0][3]++; }
10.    else if($conf[$i][1]==2&&$conf[$i][2]==1){
11.        $matconf[1][1]++; }
12.    else if($conf[$i][1]==2&&$conf[$i][2]==2){
13.        $matconf[1][2]++; }
14.    else if($conf[$i][1]==2&&$conf[$i][2]==3){
15.        $matconf[1][3]++; }
16.    else if($conf[$i][1]==3&&$conf[$i][2]==1){
17.        $matconf[2][1]++; }
18.    else if($conf[$i][1]==3&&$conf[$i][2]==2){
19.        $matconf[2][2]++; }
20.    else{
21.        $matconf[2][3]++; }
22. }
23. for ($i=0;$i<3;$i++){
24.     for ($j=1;$j<4;$j++){
25.         $matconf[$i][4]=$matconf[$i][4]+$matconf[$i][$j];
26.     }
27. }
28. for ($i=1;$i<5;$i++){
29.     for ($j=0;$j<3;$j++){
30.         $matconf[3][$i]=$matconf[3][$i]+$matconf[$j][$i];
31.     }
32. }
33. $tp=$matconf[0][1]+$matconf[1][2]+$matconf[2][3];
34. $tn=$matconf[0][1]+$matconf[1][2]+$matconf[0][1]+
35.     $matconf[2][3]+$matconf[1][2]+$matconf[2][3];
36. $fp=$matconf[1][1]+$matconf[2][1]+$matconf[0][2]+
37.     $matconf[2][2]+$matconf[0][3]+$matconf[1][3];
38. $fn=$matconf[0][2]+$matconf[0][3]+$matconf[1][1]+
39.     $matconf[1][3]+$matconf[2][1]+$matconf[2][2];
40. ($tp+$tn)/($tp+$tn+$fp+$fn)*100

```

Gambar 4.8 Implementasi algoritma *confusion matrix*

Sumber: [Implementasi]

Berikut penjelasan dari Gambar 4.9:

1. Baris 1 sampai baris 2 menjelaskan inisialisasi *array*.
2. Baris 3 sampai baris 22 menjelaskan kondisi *actual class* dengan *predicted class*.

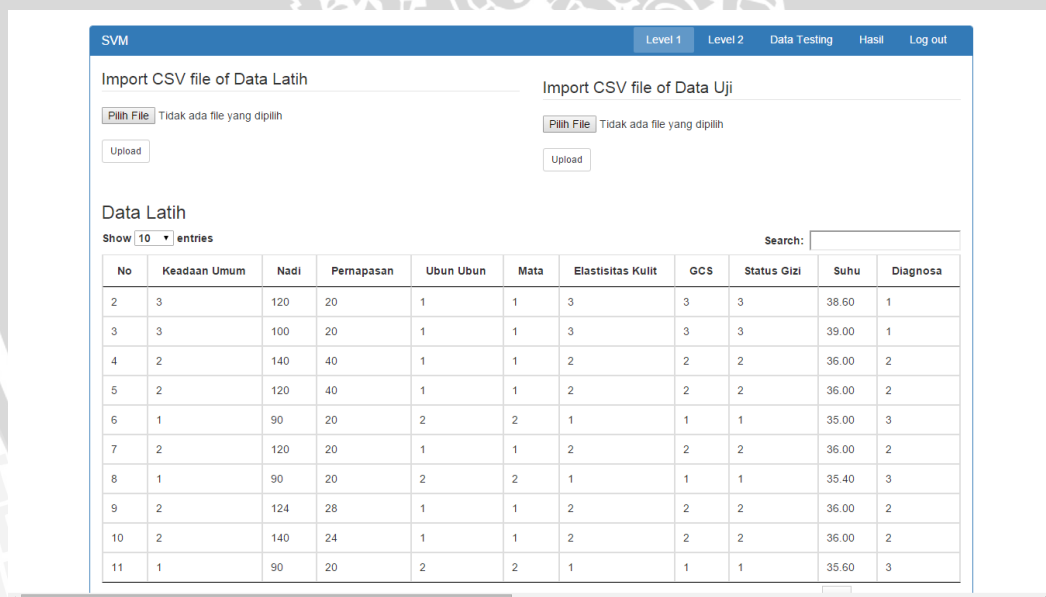
3. Baris 23 sampai baris 40 menjelaskan perhitungan nilai akurasi.

4.4 Implementasi Antarmuka

Implementasi antarmuka merupakan penghubung dengan *user* secara langsung. Antar muka dari Implementasi Metode *Support Vector Machine* Untuk Rekomendasi Pemilihan Terapi Dehidrasi Pada Anak terdiri dari empat bagian utama, yaitu halaman perhitungan SVM level 1, halaman perhitungan SVM level 2, halaman data *testing* dan halaman hasil.

4.4.1 Implementasi Antarmuka Halaman Perhitungan SVM Level 1

Halaman Perhitungan SVM Level 1 adalah halaman yang berisi semua perhitungan SVM pada level 1. Terdapat 2 *button* untuk *upload* data latih dan data uji. Data latih atau uji harus memiliki format *.csv*. Data latih yang sudah dinormalisasi seluruhnya ditampilkan pada halaman ini. Selanjutnya menampilkan perhitungan matriks *kernel* RBF, matriks *hessian*, perhitungan nilai E_i , $\delta\alpha_i$, α_i baru, wx^+ dan wx^- , nilai b (*bias*), perhitungan proses *testing* SVM $f(x)$. *Button* parameter λ (*lamda*), σ (*sigma*), γ (*konstanta gamma*), ϵ (*epsilon*), C (*complexity*), *itermax* (*iterasi maksimum*) digunakan untuk pengujian terhadap parameter *kernel* RBF dan parameter *sequential training* SVM. Implementasi halaman perhitungan SVM level 1 terdapat pada Gambar 4.10.



The screenshot shows the 'SVM' interface with tabs for 'Level 1', 'Level 2', 'Data Testing', 'Hasil', and 'Log out'. Under 'Level 1', there are sections for 'Import CSV file of Data Latih' and 'Import CSV file of Data Uji', each with a 'Pilih File' button and an 'Upload' button. Below these is a 'Data Latih' section with a 'Show 10 entries' dropdown and a search bar. The table below contains 11 rows of training data.

No	Keadaan Umum	Nadi	Pernapasan	Udun Udun	Mata	Elastisitas Kulit	GCS	Status Gizi	Suhu	Diagnosa
2	3	120	20	1	1	3	3	3	38.60	1
3	3	100	20	1	1	3	3	3	39.00	1
4	2	140	40	1	1	2	2	2	36.00	2
5	2	120	40	1	1	2	2	2	36.00	2
6	1	90	20	2	2	1	1	1	35.00	3
7	2	120	20	1	1	2	2	2	36.00	2
8	1	90	20	2	2	1	1	1	35.40	3
9	2	124	28	1	1	2	2	2	36.00	2
10	2	140	24	1	1	2	2	2	36.00	2
11	1	90	20	2	2	1	1	1	35.60	3

Gambar 4.9 Tampilan implementasi halaman perhitungan SVM level 1

Sumber: [Implementasi]

4.4.2 Implementasi Antarmuka Halaman Perhitungan SVM Level 2

Halaman Perhitungan SVM Level 2 adalah halaman yang berisi semua perhitungan SVM pada level 2. Untuk isi perhitungannya, sama dengan perhitungan SVM level 1. Implementasi halaman perhitungan SVM level 2 terdapat pada Gambar 4.11.

SVM		Level 1															Level 2	Data Testing	Hasil	Log out
Matriks Kernel RBF Level 2																				
K (x,y)	4	5	6	7	8	9	10	11	12	14	15	18	19	21	22	23				
4	1.000000	0.950089	0.197899	0.689906	0.200448	0.862459	0.814810	0.201251	0.874240	0.418449	0.689906	0.402685	0.812207	0.396611	0.366106	0.3				
5	0.950089	1.000000	0.242877	0.726149	0.246006	0.889365	0.774142	0.246992	0.920167	0.440432	0.726149	0.469541	0.854875	0.462458	0.422540	0.4				
6	0.197899	0.242877	1.000000	0.334473	0.996805	0.307534	0.269066	0.992826	0.258102	0.094875	0.334473	0.103768	0.324977	0.120899	0.104514	0.1				
7	0.689906	0.726149	0.334473	1.000000	0.338781	0.948145	0.938005	0.340139	0.668178	0.606531	1.000000	0.568929	0.971611	0.636863	0.581891	0.6				
8	0.200448	0.246006	0.996805	0.338781	1.000000	0.311496	0.272532	0.999200	0.263106	0.100822	0.338781	0.110273	0.329164	0.126438	0.111066	0.1				
9	0.862459	0.889365	0.307534	0.948145	0.311496	1.000000	0.955455	0.312745	0.818364	0.575079	0.948145	0.556255	0.994766	0.591598	0.541641	0.5				
10	0.814810	0.774142	0.269066	0.938005	0.272532	0.955455	1.000000	0.273624	0.712340	0.568929	0.938005	0.494207	0.947053	0.539237	0.497763	0.5				
11	0.201251	0.246992	0.992826	0.340139	0.999200	0.312745	0.273624	1.000000	0.265007	0.103685	0.340139	0.113404	0.330483	0.128993	0.114220	0.1				
12	0.874240	0.920167	0.258102	0.668178	0.263106	0.818364	0.712340	0.265007	1.000000	0.362330	0.668178	0.386277	0.786628	0.374411	0.347611	0.3				
14	0.418449	0.440432	0.094875	0.606531	0.100822	0.575079	0.568929	0.103685	0.362330	1.000000	0.606531	0.938005	0.589312	0.931276	0.959376	0.5				
15	0.689906	0.726149	0.334473	1.000000	0.338781	0.948145	0.938005	0.340139	0.668178	0.606531	1.000000	0.568929	0.971611	0.636863	0.581891	0.6				
18	0.402685	0.469541	0.103768	0.568929	0.110273	0.556255	0.494207	0.113404	0.386277	0.938005	0.568929	1.000000	0.574417	0.967732	0.986776	0.5				
19	0.812207	0.854875	0.324977	0.971611	0.329164	0.994766	0.947053	0.330483	0.786628	0.589312	0.971611	0.574417	1.000000	0.618783	0.565372	0.6				
21	0.396611	0.462458	0.120899	0.636863	0.126438	0.591598	0.539237	0.128993	0.374411	0.931276	0.636863	0.967732	0.618783	1.000000	0.979697	0.5				
22	0.366106	0.422540	0.104514	0.581891	0.111066	0.541641	0.497763	0.114220	0.347611	0.959376	0.581891	0.986776	0.565372	0.979697	1.000000	0.5				

Gambar 4.10 Tampilan implementasi halaman perhitungan SVM level 2
Sumber: [Implementasi]

4.4.3 Implementasi Antarmuka Halaman Data Testing

Halaman Data *testing* adalah halaman yang berisi semua data *testing* yang digunakan pada proses perhitungan. Implementasi halaman data *testing* terdapat pada Gambar 4.12.

SVM

Level 1

Level 2

Data Testing

Hasil

Log out

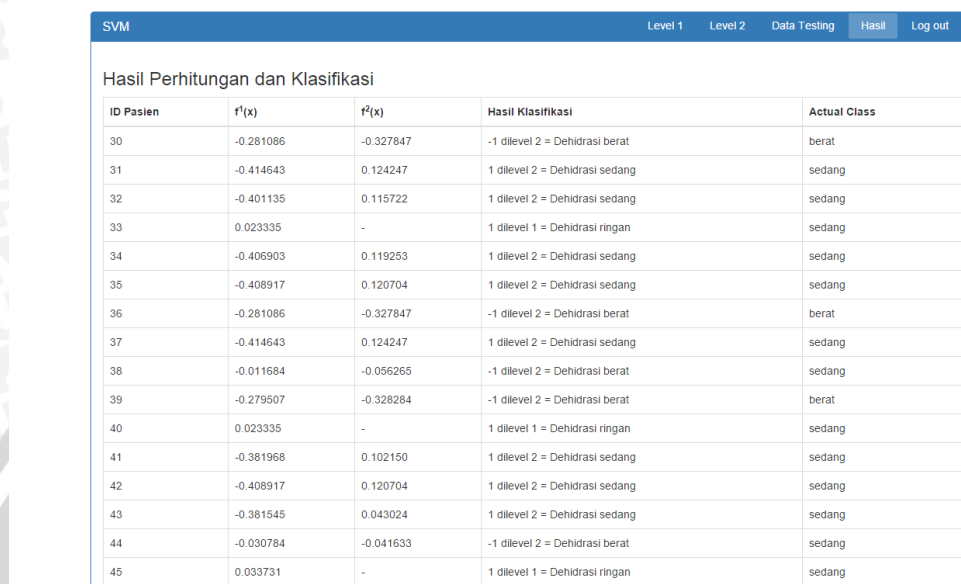
Data Uji

No	Kedaaan Umum	Nadi	Pernapasan	Ubun Ubun	Mata	Elastisitas Kulit	GCS	Status Gizi	Suhu	Diagnosa
30	1	90	20	2	2	1	1	1	35.40	3
31	2	120	26	1	1	2	2	2	36.00	2
32	2	120	24	1	1	2	2	2	36.40	2
33	3	100	24	1	1	3	3	3	39.00	2
34	2	120	26	1	1	2	2	2	36.40	2
35	2	120	24	1	1	2	2	2	36.00	2
36	1	90	20	2	2	1	1	1	35.40	3
37	2	120	26	1	1	2	2	2	36.00	2
38	3	102	20	1	1	3	3	3	37.40	2
39	1	90	20	2	2	1	1	1	35.00	3
40	3	100	24	1	1	3	3	3	39.00	2
41	2	120	20	1	1	2	2	2	36.40	2
42	2	120	24	1	1	2	2	2	36.00	2
43	2	120	40	1	1	1	2	2	36.40	2
44	3	100	24	1	1	3	3	3	37.00	2
45	3	100	20	1	1	3	3	3	39.00	2

Gambar 4.11 Tampilan implementasi halaman data *testing*
Sumber: [Implementasi]

4.4.4 Implementasi Antarmuka Halaman Hasil

Halaman hasil adalah halaman yang berisi tabel hasil perhitungan dan klasifikasi. Terdapat tabel *confusion matrix* yang digunakan untuk mengetahui nilai hasil akurasi. Implementasi halaman hasil terdapat pada Gambar 4.13.



ID Pasien	$f^1(x)$	$f^2(x)$	Hasil Klasifikasi	Actual Class
30	-0.281086	-0.327847	-1 dilevel 2 = Dehidrasi berat	berat
31	-0.414643	0.124247	1 dilevel 2 = Dehidrasi sedang	sedang
32	-0.401135	0.115722	1 dilevel 2 = Dehidrasi sedang	sedang
33	0.023335	-	1 dilevel 1 = Dehidrasi ringan	sedang
34	-0.406903	0.119253	1 dilevel 2 = Dehidrasi sedang	sedang
35	-0.408917	0.120704	1 dilevel 2 = Dehidrasi sedang	sedang
36	-0.281086	-0.327847	-1 dilevel 2 = Dehidrasi berat	berat
37	-0.414643	0.124247	1 dilevel 2 = Dehidrasi sedang	sedang
38	-0.011684	-0.056265	-1 dilevel 2 = Dehidrasi berat	sedang
39	-0.279507	-0.328264	-1 dilevel 2 = Dehidrasi berat	berat
40	0.023335	-	1 dilevel 1 = Dehidrasi ringan	sedang
41	-0.381968	0.102150	1 dilevel 2 = Dehidrasi sedang	sedang
42	-0.408917	0.120704	1 dilevel 2 = Dehidrasi sedang	sedang
43	-0.381545	0.043024	1 dilevel 2 = Dehidrasi sedang	sedang
44	-0.030784	-0.041633	-1 dilevel 2 = Dehidrasi berat	sedang
45	0.033731	-	1 dilevel 1 = Dehidrasi ringan	sedang

Gambar 4.12 Tampilan implementasi halaman hasil
Sumber: [Implementasi]

BAB 5 PENGUJIAN DAN ANALISIS

Bab ini akan membahas mengenai analisis dan pengujian terhadap sistem yang telah diimplementasikan pada tahap sebelumnya.

5.1 Pengujian Tingkat Akurasi terhadap Rasio Perbandingan

Pengujian rasio perbandingan data *training* dan data *testing* ini menjelaskan tentang hasil skenario pengujian rasio perbandingan, dan analisis skenario pengujian rasio perbandingan.

5.1.1 Skenario Pengujian Rasio Perbandingan

Pengujian berdasarkan data *training* dan data *testing* digunakan untuk melihat pada rasio perbandingan data berapa mendapatkan akurasi terbaik. *Dataset* yang digunakan sebanyak 150 data yang terbagi berdasarkan jumlah perbandingan data *training* dan data *testing*. Rasio perbandingan yang digunakan adalah 90%:10%, 80%:20%, 70%:30%, 60%:40%, 50%:50%, 40%:60%, 30%:70%, 20%:80% dan 10%:90%. Nilai parameter *Sequential Training SVM* yang digunakan dalam pengujian ini adalah $\lambda = 0.5$, $\gamma = 0.01$, $\varepsilon = 0.0001$, *itermax* = 100, $C = 1$. Hasil pengujian terhadap rasio perbandingan data *training* dan data *testing* dapat dilihat pada Tabel 5.1.

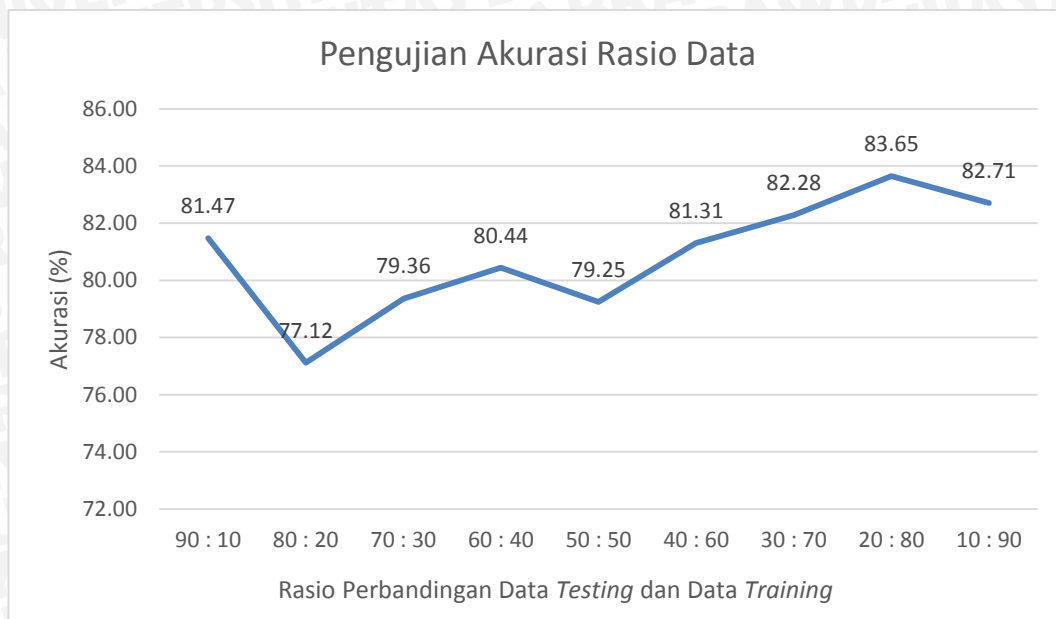
Tabel 5.1 Hasil pengujian rasio perbandingan

Percobaan ke- <i>i</i>	Rasio Perbandingan Data <i>Training</i> dan Data <i>Testing</i> (%)								
	90:10	80:20	70:30	60:40	50:50	40:60	30:70	20:80	10:90
1.	80.48	69.23	75.00	77.77	76.88	84.00	81.25	80.48	82.25
2.	80.48	77.77	82.25	84.43	78.80	81.37	79.72	86.47	82.25
3.	85.71	83.13	76.85	79.14	73.25	77.58	81.25	84.43	85.71
4.	85.71	80.48	80.48	83.13	84.11	83.13	83.50	85.07	81.67
5.	75.00	75.00	82.25	77.77	83.25	80.48	85.71	81.81	81.67
Rerata	81.47	77.12	79.36	80.44	79.25	81.31	82.28	83.65	82.71

Sumber: [Pengujian]

5.1.2 Analisis Pengujian Rasio Perbandingan

Berdasarkan grafik pada Gambar 5.1 diketahui bahwa rata-rata tingkat akurasi tertinggi sebesar 83,65% dan akurasi terbaik sebesar 86,47% pada rasio perbandingan 20:80. Sehingga tingkat akurasi terbaik dengan rasio perbandingan 20:80 akan digunakan untuk pengujian nilai *sigma*. Hasil akurasi pengujian rasio perbandingan dijelaskan pada Gambar 5.1.



Gambar 5.1 Grafik tingkat akurasi hasil pengujian rasio perbandingan
Sumber: [Pengujian]

Analisis dilakukan pada pengujian rasio perbandingan yaitu tingkat akurasi terhadap pengaruh rasio perbandingan. Pada skenario pengujian ini menunjukkan nilai akurasi paling optimal ditunjukkan pada rasio perbandingan yang memiliki nilai terbesar yaitu 83,65%. Data *training* dan data *testing* yang digunakan dipilih dari setiap *class* secara *random*, sehingga terdapat minimal satu data *training* atau data *testing* yang mewakili setiap *class*. Hal ini menunjukkan bahwa, pada rasio 20:80, sebaran data *training* dan data *testing* sangat baik. Data dikatakan baik, ketika sebuah data memiliki nilai fitur-fitur yang berada pada interval sesuai dengan *actual class*-nya. Misal data pasien dengan ID 31 yang merupakan data *testing*, *actual class*-nya adalah dehidrasi sedang. Nilai fitur-fitur yang dimiliki pasien ID 31 untuk keadaan umum = sedang, nadi = 120, pernafasan = 26, ubun-ubun = tidak cekung, mata = tidak cowong, elastisitas kulit = lambat, GCS = apatis, status gizi = kurang dan suhu = 36° , semua fitur berada pada interval dehidrasi sedang.

5.2 Pengujian terhadap Parameter *Kernel Gaussian RBF*

Pengujian nilai *sigma* ini menjelaskan tentang hasil skenario pengujian nilai *sigma*, dan analisis skenario pengujian nilai *sigma*.

5.2.1 Skenario Pengujian *Sigma*

Pengujian terhadap parameter *Kernel Gaussian RBF* dilakukan untuk melihat pada skenario mana yang mendapatkan akurasi terbaik. Skenario untuk nilai σ (*sigma*) yaitu 0.5, 0.6, 0.7, 0.8, 0.9, 1, 2, 3, 4 dan 5. Nilai parameter *Sequential Training SVM* yang digunakan dalam pengujian ini adalah $\lambda = 0.5$, $\gamma = 0.01$, $\epsilon = 0.0001$, *itermax* = 100, *C* = 1 dan rasio perbandingan 20:80. Nilai *default* untuk *sigma* adalah 1. Hasil pengujian terhadap nilai *sigma* dapat dilihat pada Tabel 5.2.

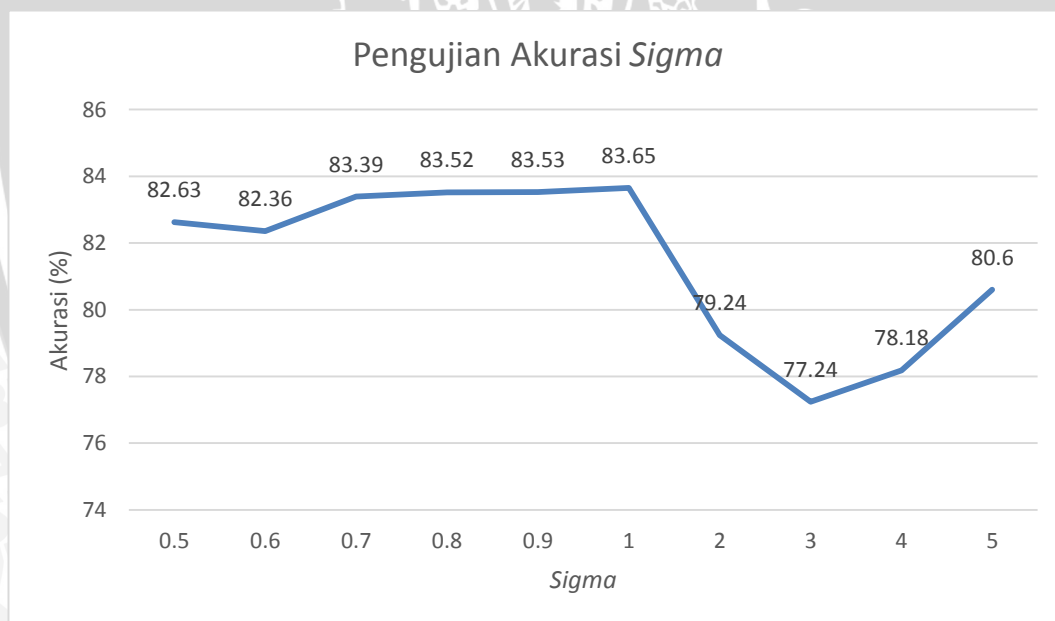
Tabel 5.2 Hasil pengujian *sigma*

Percobaan ke- <i>i</i>	Nilai σ (<i>sigma</i>)									
	0.5	0.6	0.7	0.8	0.9	1	2	3	4	5
1.	83.13	82.47	87.61	87.61	85.07	80.48	75.70	73.58	73.58	77.08
2.	80.66	80.00	80.00	80.00	80.66	86.47	74.53	75.23	77.98	77.98
3.	83.13	83.13	83.13	83.13	85.07	84.43	79.14	77.77	77.77	78.46
4.	84.43	84.43	84.43	85.07	85.07	85.07	86.35	80.48	83.13	86.35
5.	81.81	81.81	81.81	81.81	81.81	81.81	80.48	79.14	78.46	83.13
Rerata	82.63	82.36	83.39	83.52	83.53	83.65	79.24	77.24	78.18	80.60

Sumber: [Pengujian]

5.2.2 Analisis Pengujian *Sigma*

Berdasarkan grafik pada Gambar 5.2 diketahui bahwa rata-rata tingkat akurasi tertinggi sebesar 83,65% dan akurasi terbaik sebesar 86,47% pada nilai *sigma* 1. Sehingga tingkat akurasi terbaik dengan nilai *sigma*=1 akan digunakan untuk pengujian nilai *lamda*. Hasil akurasi pengujian nilai *sigma* dijelaskan pada Gambar 5.2.



Gambar 5.2 Grafik tingkat akurasi hasil pengujian nilai *sigma*

Sumber: [Pengujian]

Analisis dilakukan pada pengujian nilai σ (*sigma*) yaitu tingkat akurasi terhadap pengaruh nilai σ (*sigma*). Pada skenario pengujian ini menunjukkan nilai akurasi paling optimal ditunjukkan pada nilai σ (*sigma*) yang memiliki nilai terbesar yaitu 83,65%. Hal ini disebabkan karena σ (*sigma*) digunakan untuk menyatakan variasi/ketidakpastian sekelompok data. *Kernel* RBF digunakan untuk mengatasi permasalahan *non-linier*, khususnya untuk *mapping* data menyerupai distribusi

normal. Pada proses pemetaan data ke ruang berdimensi tinggi, ketika nilai σ (*sigma*) lebih tinggi/melewati nilai *default*, data yang dipetakan cenderung rendah.

5.3 Pengujian terhadap Parameter *Sequential Training SVM*

Pengujian nilai parameter *sequential training SVM* ini menjelaskan tentang hasil skenario pengujian nilai parameter *sequential training SVM*, dan analisis skenario pengujian nilai parameter *sequential training SVM*.

5.3.1 Skenario Pengujian *Lamda*

Pengujian terhadap parameter *Sequential Training SVM* dilakukan untuk melihat pada skenario mana yang mendapatkan nilai terbaik untuk nilai λ (*lamda*) yaitu 0.1, 0.3, 0.5, 1, 1.5, 2, 2.5, 3, 3.5 dan 4. Nilai parameter *Sequential Training SVM* yang digunakan dalam pengujian ini adalah $\gamma = 0.01$, $\epsilon = 0.0001$, *itermax* = 100, $C = 1$, rasio perbandingan 20:80 dan $\sigma = 1$. Nilai *default* untuk *lamda* adalah 0.5. Hasil pengujian terhadap nilai *lamda* dapat dilihat pada Tabel 5.3.

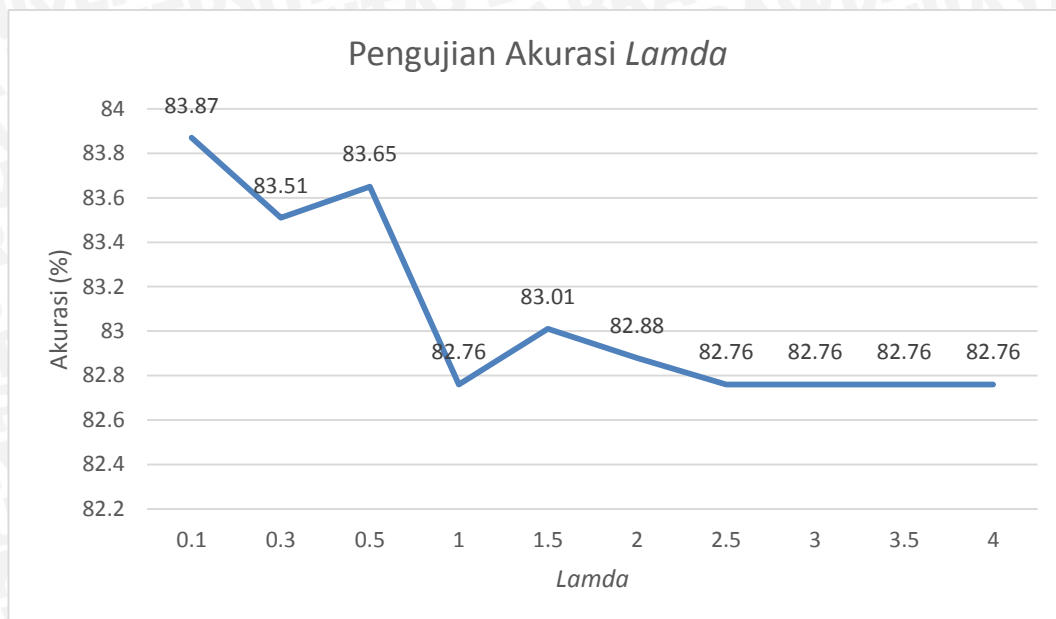
Tabel 5.3 Hasil pengujian *lamda*

Percobaan ke- <i>i</i>	Nilai λ (<i>lamda</i>)									
	0.1	0.3	0.5	1	1.5	2	2.5	3	3.5	4
1.	79.14	79.14	80.48	83.13	85.07	84.43	84.43	84.43	84.43	84.43
2.	89.56	86.47	86.47	80.66	80.00	80.00	80.00	80.00	80.00	80.00
3.	83.78	84.43	84.43	83.13	83.13	83.13	83.13	83.13	83.13	83.13
4.	85.07	85.07	85.07	85.07	85.07	85.07	84.43	84.43	84.43	84.43
5.	81.81	82.47	81.81	81.81	81.81	81.81	81.81	81.81	81.81	81.81
Rerata	83.87	83.51	83.65	82.76	83.01	82.88	82.76	82.76	82.76	82.76

Sumber: [Pengujian]

5.3.2 Analisis Pengujian *Lamda*

Berdasarkan grafik pada Gambar 5.3 diketahui bahwa rata-rata tingkat akurasi tertinggi sebesar 83,87% dan akurasi terbaik sebesar 89,56% pada nilai λ (*lamda*) 0.1. Sehingga tingkat akurasi terbaik dengan nilai λ (*lamda*) = 0.1 akan digunakan untuk pengujian nilai konstanta γ (*gamma*) dengan rasio perbandingan data *training* dan data *testing* 20:80, nilai σ (*sigma*) = 1, ϵ (*epsilon*) = 1, iterasi maksimum (*itermax*) = 100 dan C (*complexity*) = 1. Hasil akurasi pengujian λ (*lamda*) dijelaskan pada Gambar 5.3.



Gambar 5.3 Grafik tingkat akurasi hasil pengujian nilai λ (lamda)
Sumber: [Pengujian]

Analisis dilakukan pada pengujian nilai λ (*lamda*) yaitu tingkat akurasi terhadap pengaruh nilai λ (*lamda*). Pada skenario pengujian ini menunjukkan nilai akurasi paling optimal ditunjukkan pada nilai λ (*lamda*) yang memiliki nilai terbesar yaitu 83,87%. Hal ini menunjukkan bahwa, semakin besar nilai λ (*lamda*) tidak membuat akurasi menjadi baik. Terbukti dari Gambar 5.3, akurasi terbesar pada nilai λ (*lamda*) 0.1. Dan setelah melewati nilai *default* atau semakin besar nilai λ (*lamda*), akurasi akan cenderung menurun. Karena semakin besar λ (*lamda*) akan membuat proses komputasi pada tahap perhitungan *matriks hessian sequential training* SVM cenderung lebih lama. Hal ini disebabkan *augmented factor* (*lamda*) menjadikan sistem sangat lambat untuk mencapai nilai konvergen dan ketidakstabilan dalam proses pembelajaran. Solusi yang tepat dalam memilih *augmented factor* adalah sama besarnya dengan *input vector* (Vijayakumar, 1999). Namun ketika menggunakan *kernel non-linier*, nilai *augmented factor* tidak langsung ditetapkan, tergantung pada *kernel* yang kita gunakan.

5.3.3 Skenario Pengujian Konstanta Γ

Pengujian terhadap parameter *Sequential Training* SVM dilakukan untuk melihat pada skenario mana yang mendapatkan nilai terbaik untuk nilai konstanta γ (*gamma*) yaitu 1.10^{-5} , 1.10^{-4} , 5.10^{-4} , 1.10^{-3} , 5.10^{-3} , 1.10^{-2} , 5.10^{-2} , 1.10^{-1} , 5.10^{-1} dan 1. Nilai parameter *Sequential Training* SVM yang digunakan dalam pengujian ini adalah $\epsilon = 0.0001$, *itermax* = 100, $C = 1$, rasio perbandingan 20:80, $\sigma=1$ dan $\lambda=0.1$. Nilai *default* untuk konstanta γ (*gamma*) adalah 0.01. Hasil pengujian terhadap nilai konstanta γ (*gamma*) dapat dilihat pada Tabel 5.4.

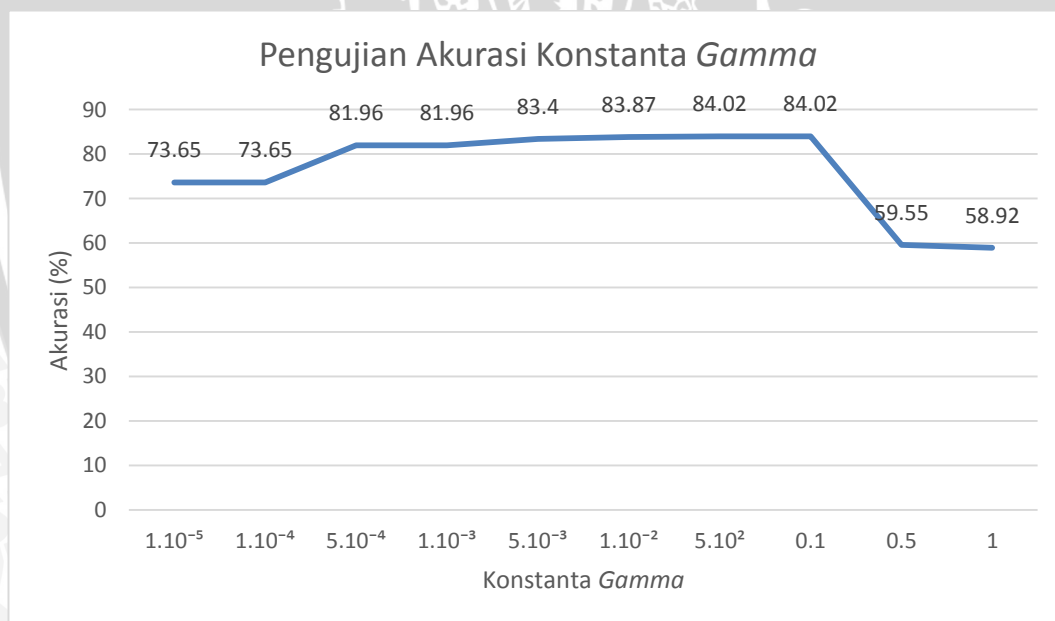
Tabel 5.4 Hasil pengujian konstanta γ (*gamma*)

Percobaan ke- <i>i</i>	Konstanta γ (<i>gamma</i>)									
	1.10^{-5}	1.10^{-4}	5.10^{-4}	1.10^{-3}	5.10^{-3}	1.10^{-2}	5.10^{-2}	1.10^{-1}	5.10^{-1}	1
1.	49.12	49.12	79.81	79.81	85.07	79.14	79.14	79.14	67.74	67.74
2.	80.66	80.66	80.66	80.66	80.00	89.56	87.09	87.09	69.52	69.52
3.	73.58	73.58	83.13	83.13	85.07	83.78	85.71	85.71	63.15	60.00
4.	84.43	84.43	84.43	84.43	85.07	85.07	85.07	85.07	59.19	59.19
5.	80.48	80.48	81.81	81.81	81.81	81.81	83.13	83.13	38.18	38.18
Rerata	73.65	73.65	81.96	81.96	83.40	83.87	84.02	84.02	59.55	58.92

Sumber: [Pengujian]

5.3.4 Analisis Pengujian Konstanta *Gamma*

Berdasarkan grafik pada Gambar 5.4 diketahui bahwa rata-rata tingkat akurasi tertinggi sebesar 84,02% dan akurasi terbaik sebesar 87,09% pada nilai λ (*lamda*) 0.1 dan 0.05. Sehingga tingkat akurasi terbaik dengan nilai γ konstanta (*gamma*) = 0.05 akan digunakan untuk pengujian nilai ϵ (*epsilon*). Hasil akurasi pengujian nilai γ konstanta (*gamma*) dijelaskan pada Gambar 5.4.



Gambar 5.4 Grafik tingkat akurasi hasil pengujian nilai konstanta *gamma*

Sumber: [Pengujian]

Analisis dilakukan pada pengujian nilai γ konstanta (*gamma*) yaitu tingkat akurasi terhadap pengaruh nilai γ konstanta (*gamma*). Pada skenario pengujian ini menunjukkan nilai akurasi paling optimal ditunjukkan pada nilai γ konstanta (*gamma*) yang memiliki nilai terbesar yaitu 84,02%. Nilai γ konstanta (*gamma*) berpengaruh terhadap proses iterasi yang terjadi. Hal ini diketahui pada saat nilai konstanta untuk mengitung γ (*gamma*) = 1.10^{-5} dan 1.10^{-4} , proses berhenti pada

iterasi ke-3. Sedangkan pada nilai konstanta 5.10^{-4} , 1.10^{-3} , 5.10^{-3} , 1.10^{-2} , 5.10^{-2} , 1.10^{-1} , 5.10^{-1} dan 1, proses berhenti pada iterasi ke-100. Sehingga hasil pengujian pada nilai 1.10^{-5} dan 1.10^{-4} , cenderung tidak stabil, karena semakin kecil nilai konstanta, maka iterasi berhenti mendekati nilai iterasi maksimum (*itermax*). Semakin besar γ konstanta (*gamma*), maka semakin besar nilai *learning rate*. *Learning rate* adalah laju pembelajaran. Dimana semakin besar nilai *learning rate*, maka proses pembelajaran akan berjalan semakin cepat. Namun apabila nilai *learning rate* relatif terlalu besar, pada umumnya proses *training* dapat melampaui keadaan optimal yaitu pada nilai *error* yang paling minimal. Dengan kata lain, *learning rate* mempengaruhi ketelitian suatu sistem. Semakin besar *learning rate*, maka ketelitian jaringan akan semakin berkurang tetapi berlaku sebaliknya, apabila *learning rate*-nya semakin kecil, maka ketelitian suatu sistem akan semakin besar dengan konsekuensi proses pembelajaran membutuhkan waktu yang semakin lama.

5.3.5 Skenario Pengujian Epsilon

Pengujian terhadap parameter *Sequential Training SVM* dilakukan untuk melihat pada skenario mana yang mendapatkan nilai terbaik untuk nilai ϵ (*epsilon*) yaitu 1.10^{-9} , 1.10^{-8} , 1.10^{-7} , 1.10^{-6} , 1.10^{-5} , 1.10^{-4} , 5.10^{-4} , 1.10^{-3} , 5.10^{-3} dan 1.10^{-2} . Nilai parameter *Sequential Training SVM* yang digunakan dalam pengujian ini adalah iterasi maksimum (*itermax*) = 100, *C (complexity)* = 1, dengan rasio perbandingan data *training* dan data *testing* 20:80, σ (*sigma*) = 1, λ (*lamda*) = 0.1 dan γ konstanta (*gamma*) = 0.05. Nilai *default* untuk ϵ (*epsilon*) adalah 0.0001. Hasil pengujian terhadap nilai ϵ (*epsilon*) dapat dilihat pada Tabel 5.5.

Tabel 5.5 Hasil pengujian *epsilon*

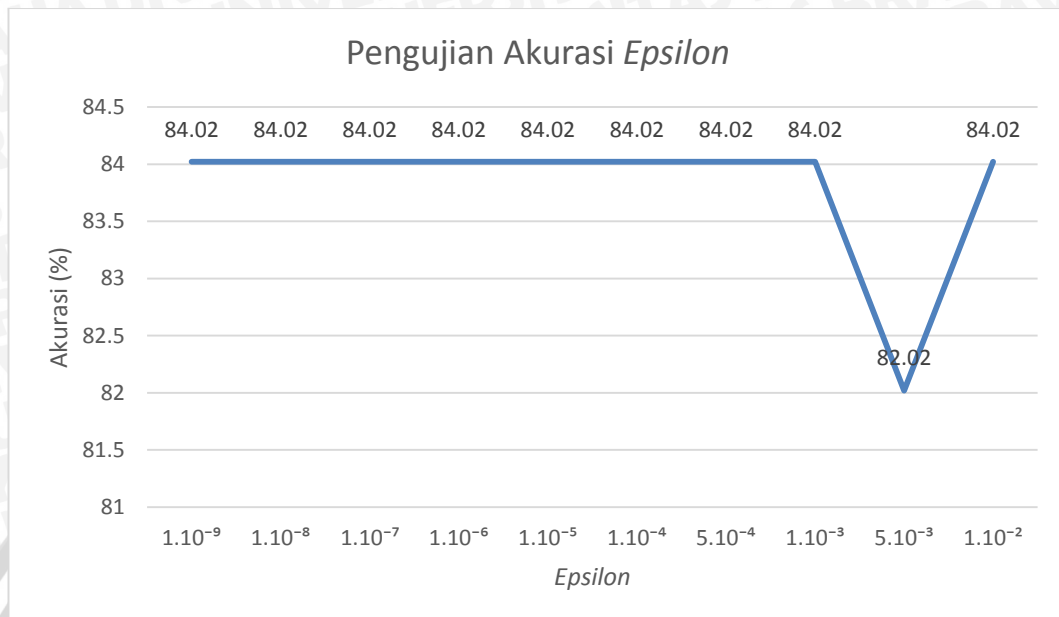
Percobaan ke-i	Nilai ϵ (<i>epsilon</i>)									
	1.10^{-9}	1.10^{-8}	1.10^{-7}	1.10^{-6}	1.10^{-5}	1.10^{-4}	5.10^{-4}	1.10^{-3}	5.10^{-3}	1.10^{-2}
1.	79.14	79.14	79.14	79.14	79.14	79.14	79.14	79.14	74.29	79.14
2.	87.09	87.09	87.09	87.09	87.09	87.09	87.09	87.09	87.09	87.09
3.	85.71	85.71	85.71	85.71	85.71	85.71	85.71	85.71	82.47	85.71
4.	85.07	85.07	85.07	85.07	85.07	85.07	85.07	85.07	83.13	85.07
5.	83.13	83.13	83.13	83.13	83.13	83.13	83.13	83.13	83.13	83.13
Rerata	84.02	84.02	84.02	84.02	84.02	84.02	84.02	84.02	82.02	84.02

Sumber: [Pengujian]

5.3.6 Analisis Pengujian Epsilon

Berdasarkan grafik pada Gambar 5.5 diketahui bahwa rata-rata tingkat akurasi tertinggi sebesar 84,02% dan akurasi terbaik sebesar 87,09% pada nilai *epsilon* 1.10^{-9} , 1.10^{-8} , 1.10^{-7} , 1.10^{-6} , 1.10^{-5} , 1.10^{-4} , 5.10^{-4} , 1.10^{-3} dan 1.10^{-2} . Sehingga tingkat akurasi terbaik dengan nilai *epsilon*=0.0001, rasio perbandingan data *training* dan data *testing* 20:80, σ (*sigma*) = 1, λ (*lamda*) = 0.1, γ konstanta (*gamma*) = 0.05 dan *C (complexity)* = 1, akan digunakan untuk pengujian nilai iterasi maksimum

(*itermax*) dengan nilai *default* untuk iterasi maksimum (*itermax*) = 100 . Hasil akurasi pengujian nilai ϵ (*epsilon*) dijelaskan pada Gambar 5.5.



Gambar 5.5 Grafik tingkat akurasi hasil pengujian nilai ϵ (*epsilon*)
Sumber: [Pengujian]

Analisis dilakukan pada pengujian nilai ϵ (*epsilon*) yaitu tingkat akurasi terhadap pengaruh nilai *epsilon*. Pada skenario pengujian ini menunjukkan nilai akurasi paling optimal ditunjukkan pada nilai ϵ (*epsilon*) yang memiliki nilai terbesar yaitu 84,02% . Hal ini diketahui bahwa nilai ϵ (*epsilon*) berpengaruh terhadap nilai akurasi yang dihasilkan. Semakin besar nilai ϵ (*epsilon*), maka banyaknya iterasi akan semakin kecil, sedangkan semakin kecil nilai ϵ (*epsilon*), maka jumlah iterasi semakin banyak. Apabila nilai ϵ (*epsilon*) semakin kecil, maka proses *learning* akan berlangsung lama. Sehingga akan memungkinkan data yang dihasilkan semakin konvergen dan nilai α_i serta nilai *support vector* akan semakin optimal. Iterasi berhenti pada nilai $\epsilon = 5.10^{-3}$ dan 1.10^{-2} karena telah mencapai nilai konvergen maks ($|\delta\alpha_i| < \epsilon$). Hasil nilai α_i serta nilai *support vector* yang tidak optimal, akan mempengaruhi nilai akurasi, sehingga nilai akurasi yang dihasilkan semakin rendah.

5.3.7 Skenario Pengujian *Itermax*

Pengujian terhadap parameter *Sequential Training SVM* dilakukan untuk melihat pada skenario mana yang mendapatkan nilai terbaik untuk nilai *itermax* yaitu 100, 200, 300, 400, 500, 550, 600, 650, 700 dan 750. Nilai parameter *Sequential Training SVM* yang digunakan dalam pengujian ini adalah $C = 1$, rasio perbandingan data *training* dan data *testing* 20:80, σ (*sigma*) = 1, λ (*lamda*) = 0.1, γ konstanta (*gamma*) = 0.05 dan ϵ (*epsilon*) = 0.0001. Nilai *default* untuk iterasi maksimum (*itermax*) adalah 100. Hasil pengujian terhadap nilai iterasi maksimum (*itermax*) dapat dilihat pada Tabel 5.6.

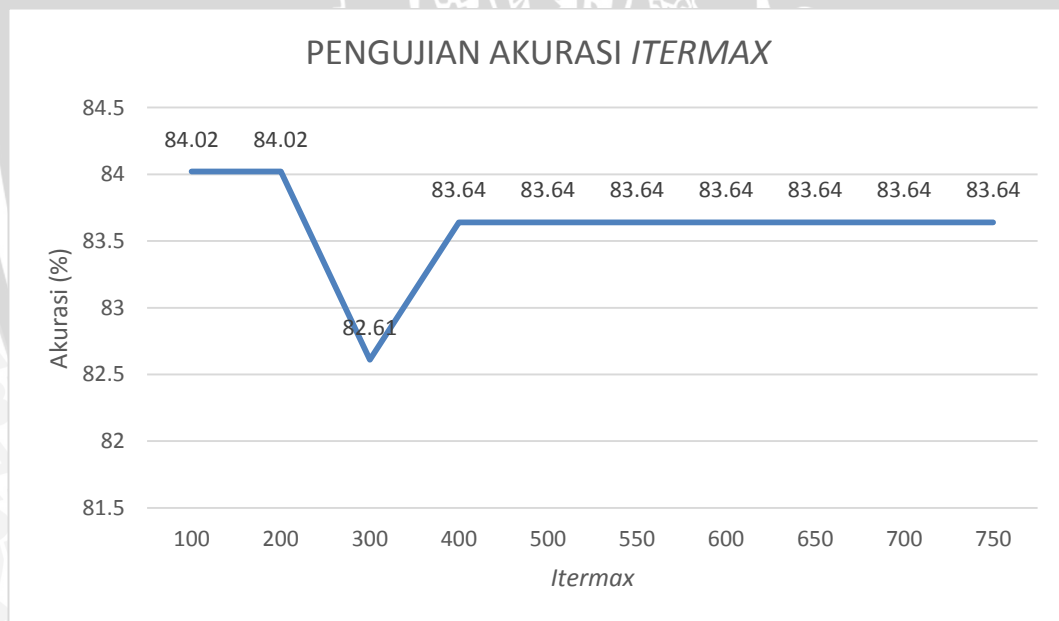
Tabel 5.6 Hasil pengujian *itermax*

Percobaan ke- <i>i</i>	Maksimum iterasi (<i>itermax</i>)									
	100	200	300	400	500	550	600	650	700	750
1.	79.14	79.14	79.14	79.14	79.14	79.14	79.14	79.14	79.14	79.14
2.	87.09	87.09	82.63	82.63	82.63	82.63	82.63	82.63	82.63	82.63
3.	85.71	85.71	85.71	85.71	85.71	85.71	85.71	85.71	85.71	85.71
4.	85.07	85.07	82.47	87.61	87.61	87.61	87.61	87.61	87.61	87.61
5.	83.13	83.13	83.13	83.13	83.13	83.13	83.13	83.13	83.13	83.13
Rerata	84.02	84.02	82.61	83.64	83.64	83.64	83.64	83.64	83.64	83.64

Sumber: [Pengujian]

5.3.8 Analisis Pengujian *Itermax*

Berdasarkan grafik pada Gambar 5.6 diketahui bahwa rata-rata tingkat akurasi tertinggi sebesar 84,02% dan akurasi terbaik sebesar 87,09% pada nilai *itermax* 100 dan 200. Sehingga tingkat akurasi terbaik dengan nilai *itermax*=100 akan digunakan untuk pengujian nilai *complexity*. Hasil akurasi pengujian nilai *itermax* dijelaskan pada Gambar 5.6.



Gambar 5.6 Grafik tingkat akurasi hasil pengujian nilai *itermax*

Sumber: [Pengujian]

Analisis dilakukan pada pengujian nilai *itermax* yaitu tingkat akurasi terhadap pengaruh nilai *itermax*. Pada skenario pengujian ini menunjukkan nilai akurasi paling stabil ditunjukkan pada nilai *itermax* yang memiliki nilai rata-rata akurasi 84,02%. Nilai *itermax* dipengaruhi oleh nilai ϵ (*epsilon*) yang digunakan. Dengan menggunakan nilai ϵ (*epsilon*) = 0.0001 dan iterasi maksimum (*itermax*) = 100, proses iterasi berhenti pada iterasi ke-100. Sedangkan pada pengujian iterasi

maksimum (*itermax*) antara 200 sampai 750, iterasi berhenti pada masing-masing batas maksimum iterasi, hal ini terjadi karena nilai α_i belum mencapai nilai konvergen. Konvergen dapat didefinisikan dari tingkat perubahan pada nilai α_i . Untuk pengujian pada iterasi ke-300 mengalami penurunan akurasi, karena terjadi *overfitting*. *Overfitting* terjadi dimana data yang dipisahkan membentuk *hyperplane* yang dekat dengan salah satu *class*-nya.

5.3.9 Skenario Pengujian Complexity

Pengujian terhadap parameter *Sequential Training* SVM dilakukan untuk melihat pada skenario mana yang mendapatkan nilai terbaik untuk nilai *C* (*complexity*) yaitu 1, 10, 20, 30, 40, 50, 60, 70, 80 dan 90. Nilai parameter *Sequential Training* SVM yang digunakan dalam pengujian ini adalah *itermax* = 100, rasio perbandingan data *training* dan data *testing* 20:80, σ (*sigma*) = 1, λ (*lamda*) = 0.1, γ konstanta (*gamma*) = 0.05 dan ϵ (*epsilon*) = 0.0001. Nilai *default* untuk *C* (*complexity*) adalah 1. Hasil pengujian terhadap nilai *C* (*complexity*) dapat dilihat pada Tabel 5.7.

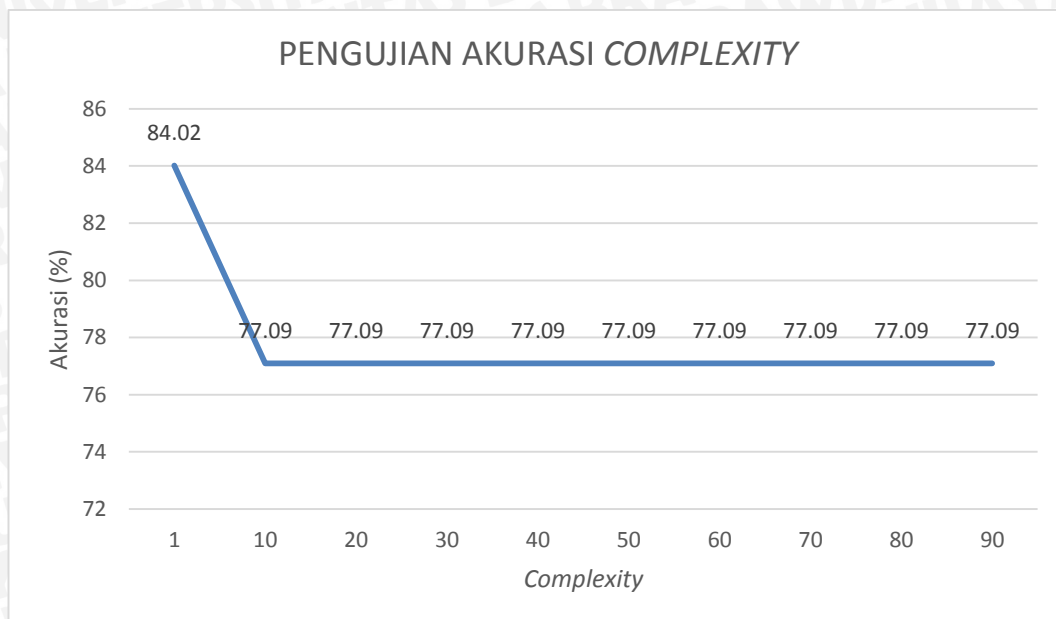
Tabel 5.7 Hasil pengujian complexity

Percobaan ke- <i>i</i>	Nilai <i>C</i> (<i>complexity</i>)									
	1	10	20	30	40	50	60	70	80	90
1.	79.14	75.70	75.70	75.70	75.70	75.70	75.70	75.70	75.70	75.70
2.	87.09	74.53	74.53	74.53	74.53	74.53	74.53	74.53	74.53	74.53
3.	85.71	74.29	74.29	74.29	74.29	74.29	74.29	74.29	74.29	74.29
4.	85.07	80.48	80.48	80.48	80.48	80.48	80.48	80.48	80.48	80.48
5.	83.13	80.48	80.48	80.48	80.48	80.48	80.48	80.48	80.48	80.48
Rerata	84.02	77.09	77.09	77.09	77.09	77.09	77.09	77.09	77.09	77.09

Sumber: [Pengujian]

5.3.10 Analisis Pengujian Complexity

Berdasarkan grafik pada Gambar 5.7 diketahui bahwa rata-rata tingkat akurasi tertinggi sebesar 84,02% dan akurasi terbaik sebesar 87,09% pada nilai *C* (*complexity*) = 1, rasio perbandingan data *training* dan data *testing* 20:80, σ (*sigma*) = 1, λ (*lamda*) = 0.1, γ konstanta (*gamma*) = 0.05 dan ϵ (*epsilon*) = 0.0001, iterasi maksimum (*itermax*) = 100. Hasil akurasi pengujian nilai *C* (*complexity*) dijelaskan pada Gambar 5.7.



Gambar 5.7 Grafik tingkat akurasi hasil pengujian nilai *complexity*
Sumber: [Pengujian]

Analisis dilakukan pada pengujian nilai *complexity* yaitu tingkat akurasi terhadap pengaruh nilai *complexity*. Pada skenario pengujian ini menunjukkan nilai akurasi paling optimal ditunjukkan pada nilai *complexity* yang memiliki nilai terbesar yaitu 84,02%. Tujuan adanya nilai *C (complexity)* adalah untuk meminimalkan *error* dan memperkecil nilai *slack variabel*. Pada saat nilai *C (complexity)* menjauhi nilai 0, akurasi cenderung menurun. Hal ini disebabkan, jika nilai *C (complexity)* mendekati nilai nol, maka lebar margin pada bidang pemisah (*hyperplane*) menjadi maksimum dan pada waktu yang sama banyak jumlah data yang dilatih yang berada dalam *margin* atau yang ada pada posisi yang salah tidak akan dipedulikan. Hal ini berarti akan mengurangi tingkat akurasi pada proses *training*, sehingga mengakibatkan data *testing* tidak dapat diklasifikasikan dengan baik.

BAB 6 PENUTUP

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

6.1 Kesimpulan

Berdasarkan hasil penelitian dan pengujian yang dilakukan, dapat diambil kesimpulan sebagai berikut.

1. Untuk mengimplementasikan metode SVM, langkah pertama adalah mengumpulkan data dari RSUD Dr. Soegiri Lamongan. Langkah selanjutnya adalah pemilihan data *training* dan data *testing* secara acak berdasarkan rasio perbandingan yang ditentukan. Kemudian melakukan normalisasi data dengan skala interval [0.1, 0.9]. Setelah diperoleh hasil normalisasi dilanjutkan dengan proses perhitungan *kernel* RBF dan dilanjutkan proses perhitungan *sequential training* SVM. Setelah proses *training* selesai diperoleh nilai α_i yang merupakan bobot data ke- i dan nilai b (*bias*) yang digunakan untuk proses *testing*. Hasil dari proses *testing* adalah berupa klasifikasi masing-masing *predicted class* data *testing*. Hasil akhir berupa akurasi sistem yang diperoleh dengan membandingkan *class actual* dengan *predicted class* hasil pengujian.
2. Untuk mengetahui hasil akurasi menggunakan metode SVM pada klasifikasi terapi dehidrasi dapat menggunakan rasio perbandingan terbaik antara data *training* dan data *testing* yaitu rasio 20%:80% dengan skenario pengujian ke-2, diperoleh nilai akurasi sebesar 86.47%. Hasil pengujian terhadap parameter *kernel* RBF dengan nilai $\sigma=1$, menunjukkan nilai rata-rata akurasi terbaik adalah 83.65%. Pemilihan nilai parameter λ (*lamda*), konstanta γ (*gamma*), ϵ (*epsilon*), *itermax* dan C (*complexity*), pada metode *sequential training* SVM berpengaruh terhadap perubahan nilai α_i (bobot data ke- i) dan nilai b (*bias*) serta nilai akurasi. Berdasarkan hasil pengujian, nilai akurasi terbaik terdapat pada pengujian parameter SVM untuk nilai λ (*lamda*) adalah 0.1, nilai konstanta γ (*gamma*) adalah 0.05, nilai ϵ (*epsilon*) adalah 0.0001, banyaknya iterasi (*itermax*) adalah 100 dan nilai C (*complexity*) adalah 1, sehingga diperoleh akurasi terbaik sebesar 87.09% dan rata-rata akurasi terbaik sebesar 84.02%.

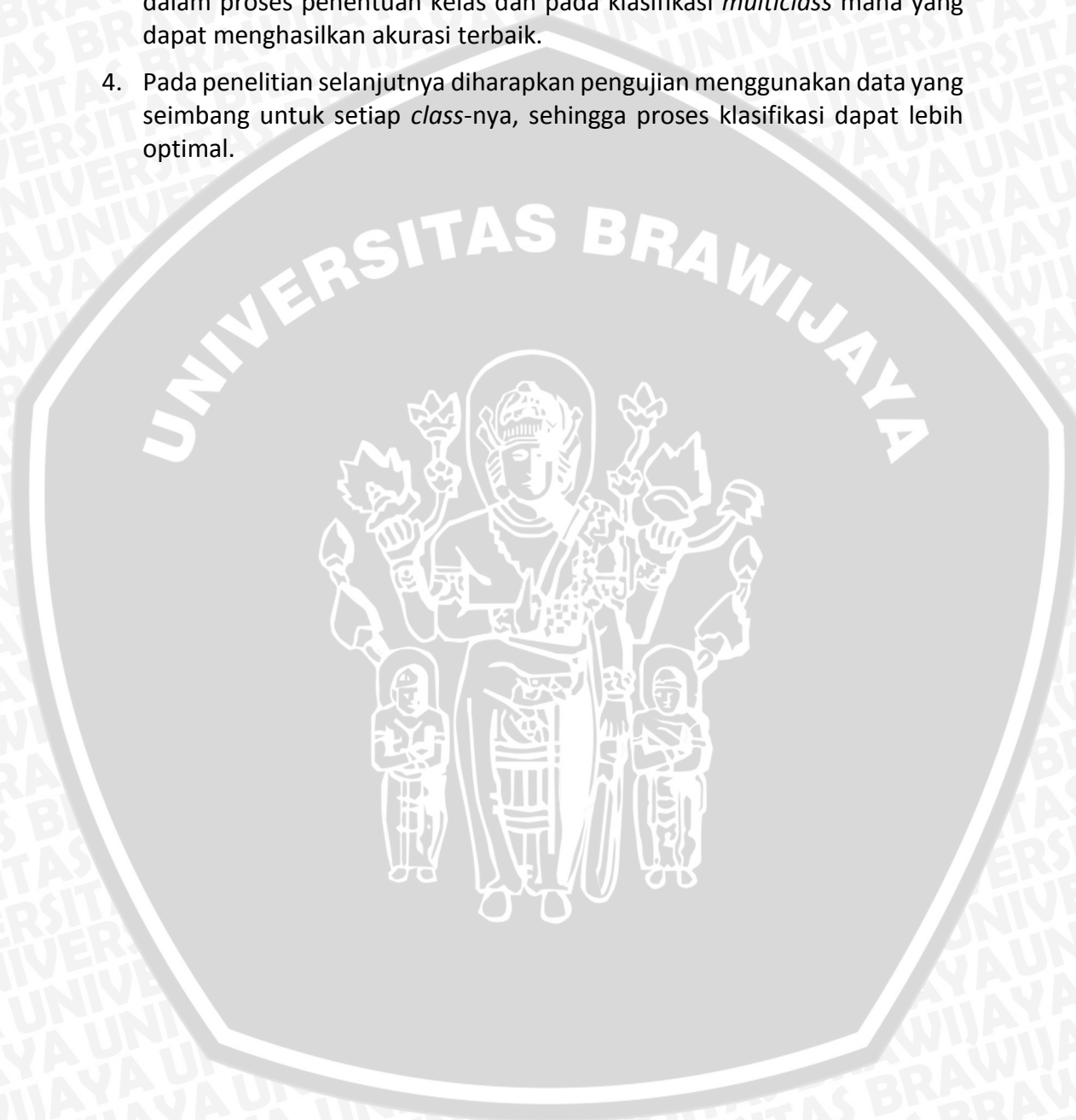
6.2 Saran

Saran yang dapat diberikan untuk pengembangan penelitian ini selanjutnya antara lain:

1. Pada penelitian selanjutnya diharapkan jumlah *dataset* ditambah, sehingga proses klasifikasi dapat lebih optimal.
2. Penelitian selanjutnya diharapkan menggunakan *kernel* yang berbeda, sehingga bisa mendapatkan perbandingan hasil akurasi. Pada *kernel* mana akurasi terbaik didapatkan, karena pada kasus tertentu *kernel* RBF tidak

menghasilkan nilai akurasi yang baik, tergantung dengan jenis data yang digunakan.

3. Pada penelitian selanjutnya diharapkan menggunakan metode klasifikasi *multiclass* yang berbeda seperti *One Against One* dan *Directed Acyclic Graph Support Vector Machine* (DAGSVM). Karena agar lebih spesifik dalam proses penentuan kelas dan pada klasifikasi *multiclass* mana yang dapat menghasilkan akurasi terbaik.
4. Pada penelitian selanjutnya diharapkan pengujian menggunakan data yang seimbang untuk setiap *class*-nya, sehingga proses klasifikasi dapat lebih optimal.



DAFTAR PUSTAKA

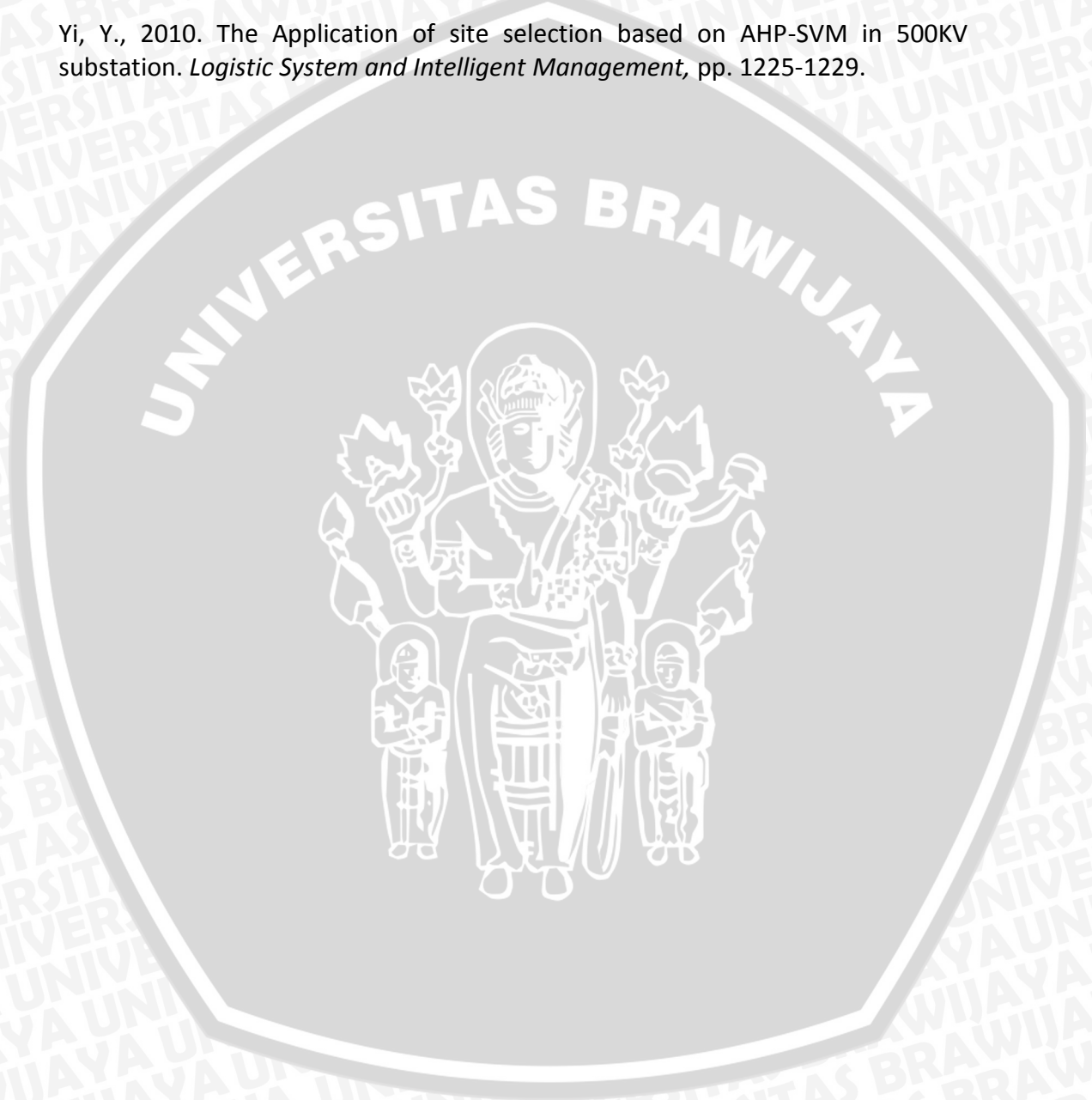
- Andayani, K., 2013. Hubungan Konsumsi Cairan Dengan Status Hidrasi Pada Pekerja Industri Laki-Laki.
- Damayanti, F., 2010. Pengenalan Citra Wajah Menggunakan Metode Two-Dimensional Linier Discriminant Analysis Dan Support Vector Machine. *Jurnal Ilmiah Kursor*, pp. 147-156.
- Dimiyati, V., 2014. Masyarakat Minim Pengetahuan Soal Air Minum. *Jurnal Nasional*, p. 20.
- Efraim, T., 2005. *Decision Support System And Intelligent System*. Yogyakarta: Andi.
- Hermaduanty, N., n.d. Sistem Pendukung Keputusan Berbasis Sms Untuk Menentukan Status Gizi Dengan Metode K-Nearest Neighbor. *Universitas Islam Indonesia*.
- Hsu, 2002. A Comparison Methods for Multi-Class Support Vector Machine. *IEEE Transactions on Neural Network*, pp. 415-425.
- Indonesia, K. K. R., 2014. Diare. 10 FEBRUARI, p. 1.
- Istara, A. W., 2013. Sistem Pendukung Keputusan Penentuan Pemberian Bantuan Raskin Menggunakan Metode SMARTER. *Jurnal Sarjana Teknik Informatika*, pp. 1-8.
- Kartal, H. B., 2013. Support Vector Machines for Multi-Attribute ABC Analysis. *International Journal of Machine Learning and Computing*.
- Kurniawati, M., 2015. Klasifikasi Dokumen E-Complaint Kampus Menggunakan Directed Acyclic Graph Multi-Class SVM.
- Kusrini, 2007. *Konsep dan Aplikasi Sistem Pendukung Keputusan*. Yogyakarta: Andi.
- Muscari, M. E., 2005. *Keperawatan Pediatrik*. 3rd ed. Jakarta: EGC.
- Novianti, F. A., 2012. Analisis Diagnosis Pasien Kanker Payudara Menggunakan Regresi Logistik dan Support Vector Machine (SVM) Berdasarkan Hasil Mamografi. *Jurnal Sains dan Seni ITS*, pp. D-147 – D-152.
- Putri, I. R., 2015. Optimasi Metode Adaptive Fuzzy K-Nearest Neighbor Dengan Particle Swarm Optimization Untuk Klasifikasi Status Sosial Ekonomi Keluarga.
- Qing Yan, H. Y. F. H. X. W. T. Z., 2009. SVM-Based Decision Support System for Clinic Aided Tracheal Intubation Prediction with Multiple Features. *Science Direct*, pp. 6588-6592.
- rachman, F., 2012. Perbandingan Klasifikasi Tingkat Keganasan Breast Cancer Dengan Menggunakan Regresi Logistik Ordinal Dan Support Vector Machine (SVM). *Jurnal Sains dan Seni ITS*, pp. D-130 – D-135.

Sembiring, 2007. Penerapan Teknik Support Vector Machine untuk Pendeteksian Intrusi pada Jaringan. *Institute Teknologi Bandung*.

Vijayakumar, S., 1999. Sequential Support Vector Classi. *Proc. International Conference on Soft Computing (SOCO'99)*, pp. 610-619.

Yessivirna, R., 2013. Klasifikasi Suara Berdasarkan Gender (Jenis Kelamin) dengan Metode K-Nearest Neighbor (KNN). *Prodi Ilmu Komputer, Fakultas PTIHK*.

Yi, Y., 2010. The Application of site selection based on AHP-SVM in 500KV substation. *Logistic System and Intelligent Management*, pp. 1225-1229.



LAMPIRAN A SURAT KETERANGAN DARI RUMAH SAKIT



PEMERINTAH KABUPATEN LAMONGAN
RUMAH SAKIT UMUM DAERAH Dr. SOEGIRI
 Jl. Kusuma Bangsa No.7 Lamongan-Kode Pos 62214
 Telp. (0322) 321718, 322582, Fax (0322)322582 E-mail : rsud@lamongankab.go.id
 Website : www.lamongsanikab.go.id



Lamongan, 9 Maret 2015

Nomor : 445/2st/413.209/2015
 Lampiran : -
 Perihal : Pemberitahuan

Kepada
 Yth. Kepala Poli Anak
 RSUD Dr. Soegiri
 di -
LAMONGAN

Memperhatikan surat Wakil Ketua Bidang Akademik Universitas Brawijaya Prodi Teknologi Informasi dan Ilmu komputer Nomor : 0727/UN10.36/AK/2015, Tertanggal 4 Maret 2015, Perihal permohonan data skripsi. Pada prinsipnya kami tidak keberatan untuk memberi pemingambilan data skripsi di RSUD dr. Soegiri Lamongan atas nama :

NO	NAMA	N I M	GAMBARAN MASALAH/ SASARAN
1.	RANI ANGGI NILAM SARI	135150109111001	Data Pasien Poli Anak yang Menderita Diare

Demikian untuk menjadikan maklum dan periksa adanya.

An. DIREKTUR RSUD Dr. SOEGIRI LAMONGAN
 Wakil Pelayanan dan Penunjang,
 Wakil Kepala Bidang Penunjang


Alfin, SKM/MM.Kes
 Pembina
 NIP. 19610421 198303 1 023

Tembusan disampaikan kepada :
 Yth. 1. Sdr. RANI ANGGI NILAM SARI
 2. Wakil Ketua Bidang Akademik Prodi Teknologi informasi dan Komputer UNIBRAW
 3. Ka. Bagian Keuangan
 4. Ka. Bag. Program
 5. Arsip

LAMPIRAN B SURAT KETERANGAN DARI KESBANGPOL



PEMERINTAH KABUPATEN LAMONGAN
BADAN KESATUAN BANGSA DAN POLITIK
 Jl. Lamongrejo No. 92 Lamongan - Kode Pos 62217
 Telp. (0322) 321708 e-mail : www.bakesbangpol@lamongankab.go.id
 website: www.lamongankab.go.id

REKOMENDASI PENELITIAN/SURVEY/KEGIATAN
 Nomor : 070/ 244 /413.204/2015

Dasar :

1. Peraturan Menteri Dalam Negeri Nomor 64 tahun 2011 tentang Pedoman Penerbitan Rekomendasi Penelitian, sebagaimana telah diubah dengan Peraturan Menteri Dalam Negeri Nomor 7 tahun 2014 tentang Perubahan atas Peraturan Menteri Dalam Negeri Nomor 64 tahun 2011;
2. Peraturan Bupati Lamongan Nomor 27 Tahun 2011 tentang Kedudukan Tugas dan Fungsi Badan Kesatuan Bangsa dan Politik Kabupaten Lamongan.

Menimbang : Surat dari Ketua Program Studi Informatika/Ilmu Komputer Universitas Brawijaya Malang tanggal 29 April 2015 Nomor : 1583/UN10.36/AK/2015 Perihal Permohonan Ijin Penelitian atas nama RANI ANGGI NILAM SARI.

Bupati Lamongan, memberikan rekomendasi kepada :

a. Nama : RANI ANGGI NILAM SARI
 b. NIK/NIM : 135150109111001
 c. Alamat : Made Karyo III/52 002/006 Ds. Made Kec. Lamongan Kab. Lamongan
 d. Pekerjaan/Jabatan : Mahasiswa
 e. Instansi/Organisasi : Universitas Brawijaya Malang
 f. Kebangsaan : Indonesia

Untuk melakukan penelitian/survey/kegiatan dengan :

a. Judul Proposal/Kegiatan : Pemodelan Sistem Pendukung Keputusan Rekomendasi Pemilihan Terapi Dehidrasi pada Anak Menggunakan Fuzzy Bayesian (Studi Kasus : RSD Dr. Soegiri Lamongan)
 b. Tujuan Penelitian/Survey/Kegiatan : Untuk Mencari Data Penyusunan Skripsi
 c. Bidang Penelitian/Survey/Kegiatan : Kesehatan
 d. Penanggungjawab : RANI ANGGI NILAM SARI
 e. Anggota/Peserta :
 f. Waktu Penelitian/Survey/Kegiatan : 04 Mei s/d 04 Agustus 2015
 g. Lokasi Penelitian/Survey/Kegiatan : RSD dr. Soegiri Lamongan

Dengan ketentuan :

1. Berkewajiban menghormati dan mentaati peraturan dan tata tertib di Lokasi Penelitian/Survey/Kegiatan.
2. Pelaksanaan Penelitian agar tidak disalahgunakan untuk tujuan tertentu yang dapat mengganggu kestabilan keamanan dan ketertiban di lokasi Penelitian/Survey/Kegiatan;
3. Wajib melaporkan hasil penelitian dan sejenisknya kepada Bupati Lamongan melalui Badan Kesatuan Bangsa dan Politik Kabupaten Lamongan dalam kesempatan pertama.

Demikian rekomendasi ini dibuat untuk dipergunakan seperfunya.

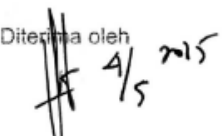
Lamongan, 04 Mei 2015
 an. KEPALA BADAN KESBANG DAN POLITIK
 KABUPATEN LAMONGAN
 Kepala Kajian Strategis dan Kewaspadaan,

MOH. AL MURTADLO, S.Pd.
 Pembina
 NIP. 19680227 199803 1 007

Tembusan :

1. Bpk. Bupati Lamongan (sebagai laporan)
2. Sdr. Kepala Kantor Litbang Daerah Kab. Lamongan
3. Sdr. Kepala Dinas Kesehatan Kab. Lamongan
4. Sdr. Ketua Program Studi Informatika/Ilmu Komputer Universitas Brawijaya Malang

LAMPIRAN C SURAT IJIN PENELITIAN

	PEMERINTAH KABUPATEN LAMONGAN KANTOR PENELITIAN DAN PENGEMBANGAN DAERAH JL. Basuki Rahmat NO. 1 LAMONGAN Kode Pos 62251 Telp (0322) 313449 e-mail : kppd@lamongankab.go.id Web site : www.lamongankab.go.id
	Formulir Tanda Terima Surat Ijin Penelitian
Telah diterima Surat Ijin Penelitian dari :	
NAMA LENGKAP	: RANI ANGGI NILAM SARI
ALAMAT	: JL. MADE KARYO III/62 LAMONGAN
No. Telp/HP	: 081232949949
Catatan :	
Pada akhir penelitian, yang bersangkutan WAJIB menyerahkan 1 (satu) berkas copy buku hasil penelitian kepada Kantor Litbangda Kab. Lamongan.	
Diterima oleh  (EDIE SUGIARTO, ST, SE.MM)	 RANI ANGGI NILAM SARI



LAMPIRAN D DATASET

No.	Keadaan Umum	Nadi	Pernapasan/RR	Udun-Udun	Mata	Elastisitas Kulit	GCS	Status Gizi	Suhu	Diagnosa
1	3	100	20	1	1	3	3	3	39	Ringan
2	3	120	20	1	1	3	3	3	38.6	Ringan
3	3	100	20	1	1	3	3	3	39	Ringan
4	2	140	40	1	1	2	2	2	36	Sedang
5	2	120	40	1	1	2	2	2	36	Sedang
6	1	90	20	2	2	1	1	1	35	Berat
7	2	120	20	1	1	2	2	2	36	Sedang
8	1	90	20	2	2	1	1	1	35.4	Berat
9	2	124	28	1	1	2	2	2	36	Sedang
10	2	140	24	1	1	2	2	2	36	Sedang
11	1	90	20	2	2	1	1	1	35.6	Berat
12	2	120	40	1	1	1	2	2	36.4	Berat
13	3	100	24	1	1	3	3	3	37	Ringan
14	3	120	20	1	1	3	3	3	39	Berat
15	2	120	20	1	1	2	2	2	36	Sedang
16	3	102	20	1	1	3	3	3	37.4	Ringan
17	1	90	20	2	2	1	1	1	35	Berat
18	3	100	24	1	1	3	3	3	39	Berat
19	2	120	26	1	1	2	2	2	36	Sedang
20	3	100	20	1	1	3	3	3	37	Ringan
21	3	100	20	1	1	3	3	3	38	Berat
22	3	102	20	1	1	3	3	3	39	Sedang

23	3	100	20	1	1	3	3	3	38.1	Sedang
24	2	120	24	1	1	2	2	2	36	Sedang
25	2	120	20	1	1	2	2	2	36.4	Berat
26	2	120	24	1	1	2	2	2	36	Sedang
27	2	120	26	1	1	2	2	2	36	Sedang
28	3	120	24	1	1	3	3	3	39	Ringan
29	2	120	26	1	1	2	2	2	36	Sedang
30	1	90	20	2	2	1	1	1	35.4	Berat
31	2	120	26	1	1	2	2	2	36	Sedang
32	2	120	24	1	1	2	2	2	36.4	Sedang
33	3	100	24	1	1	3	3	3	39	Sedang
34	2	120	26	1	1	2	2	2	36.4	Sedang
35	2	120	24	1	1	2	2	2	36	Sedang
36	1	90	20	2	2	1	1	1	35.4	Berat
37	2	120	26	1	1	2	2	2	36	Sedang
38	3	102	20	1	1	3	3	3	37.4	Sedang
39	1	90	20	2	2	1	1	1	35	Berat
40	3	100	24	1	1	3	3	3	39	Sedang
41	2	120	20	1	1	2	2	2	36.4	Sedang
42	2	120	24	1	1	2	2	2	36	Sedang
43	2	120	40	1	1	1	2	2	36.4	Sedang
44	3	100	24	1	1	3	3	3	37	Sedang
45	3	100	20	1	1	3	3	3	39	Sedang
46	3	120	20	1	1	3	3	3	38.6	Ringan
47	3	100	20	1	1	3	3	3	39	Sedang

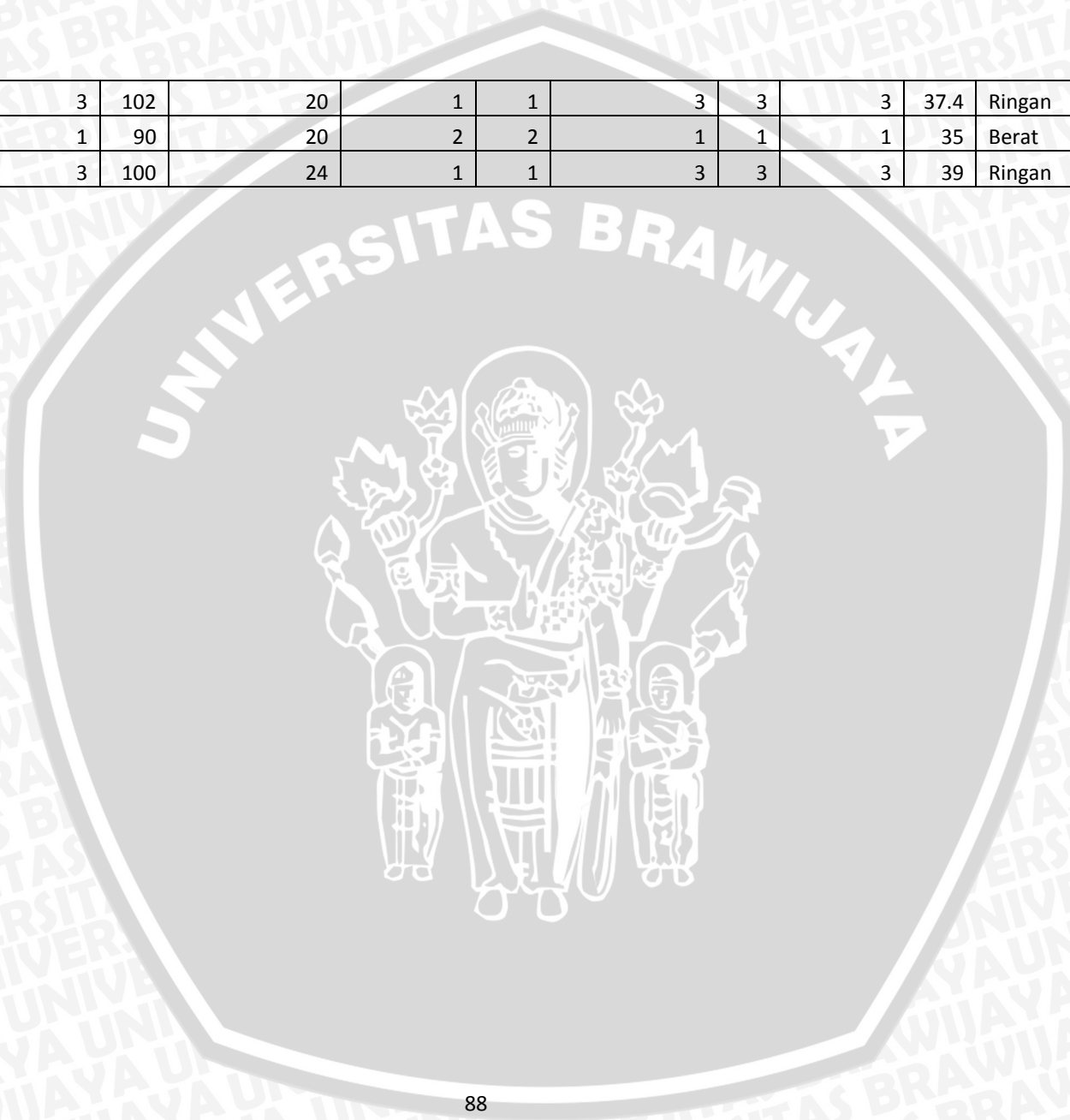
48	2	140	40	1	1	2	2	2	36	Sedang
49	2	120	40	1	1	2	2	2	36	Sedang
50	1	90	20	2	2	1	1	1	35	Berat
51	3	100	20	1	1	3	3	3	37	Sedang
52	3	100	20	1	1	3	3	3	38	Sedang
53	3	102	20	1	1	3	3	3	39	Ringan
54	3	100	20	1	1	3	3	3	38.1	Sedang
55	2	120	24	1	1	2	2	2	36	Sedang
56	3	100	20	1	1	3	3	3	39	Sedang
57	2	140	40	1	1	2	2	2	36	Sedang
58	2	120	40	1	1	2	2	2	36	Sedang
59	1	90	20	2	2	1	1	1	35.4	Berat
60	2	124	28	1	1	2	2	2	36	Sedang
61	2	140	24	1	1	2	2	2	36	Sedang
62	2	140	24	1	1	2	2	2	36	Sedang
63	1	90	20	2	2	1	1	1	35.6	Berat
64	2	120	40	1	1	1	2	2	36.4	Sedang
65	3	100	24	1	1	3	3	3	37	Sedang
66	3	120	20	1	1	3	3	3	39	Sedang
67	2	120	20	1	1	2	2	2	36	Sedang
68	3	102	20	1	1	3	3	3	37.4	Sedang
69	1	90	20	2	2	1	1	1	35	Berat
70	3	100	24	1	1	3	3	3	39	Sedang
71	2	120	26	1	1	2	2	2	36	Sedang
72	2	120	40	1	1	1	2	2	36.4	Sedang

73	3	100	24	1	1	3	3	3	37	Sedang
74	3	100	20	1	1	3	3	3	39	Sedang
75	3	120	20	1	1	3	3	3	38.6	Sedang
76	2	120	26	1	1	2	2	2	36	Sedang
77	2	120	24	1	1	2	2	2	36.4	Sedang
78	3	100	24	1	1	3	3	3	39	Sedang
79	2	120	26	1	1	2	2	2	36.4	Sedang
80	2	120	24	1	1	2	2	2	36	Sedang
81	1	90	20	2	2	1	1	1	35.4	Berat
82	2	120	26	1	1	2	2	2	36	Sedang
83	1	90	20	2	2	1	1	1	35.4	Berat
84	2	120	26	1	1	2	2	2	36	Sedang
85	2	120	24	1	1	2	2	2	36.4	Sedang
86	2	120	26	1	1	2	2	2	36	Sedang
87	2	120	40	1	1	1	2	2	36.4	Sedang
88	3	100	24	1	1	3	3	3	37	Ringan
89	3	100	20	1	1	3	3	3	39	Ringan
90	2	120	26	1	1	2	2	2	36	Sedang
91	3	100	24	1	1	3	3	3	37	Ringan
92	3	100	20	1	1	3	3	3	39	Ringan
93	3	120	20	1	1	3	3	3	38.6	Ringan
94	2	120	26	1	1	2	2	2	36	Sedang
95	2	120	24	1	1	2	2	2	36.4	Sedang
96	3	100	24	1	1	3	3	3	39	Ringan
97	3	100	24	1	1	3	3	3	37	Ringan

98	3	100	20	1	1	3	3	3	39	Ringan
99	3	120	20	1	1	3	3	3	38.6	Ringan
100	2	120	26	1	1	2	2	2	36	Sedang
101	2	120	26	1	1	2	2	2	36	Sedang
102	2	120	40	1	1	1	2	2	36.4	Sedang
103	3	100	24	1	1	3	3	3	37	Ringan
104	3	100	24	1	1	3	3	3	39	Ringan
105	2	120	26	1	1	2	2	2	36.4	Sedang
106	2	120	24	1	1	2	2	2	36	Sedang
107	1	90	20	2	2	1	1	1	35.4	Berat
108	3	100	24	1	1	3	3	3	37	Ringan
109	3	100	20	1	1	3	3	3	39	Ringan
110	3	102	20	1	1	3	3	3	39	Ringan
111	3	100	20	1	1	3	3	3	38.1	Ringan
112	2	120	24	1	1	2	2	2	36	Sedang
113	3	100	20	1	1	3	3	3	39	Ringan
114	2	140	40	1	1	2	2	2	36	Sedang
115	2	120	40	1	1	2	2	2	36	Sedang
116	1	90	20	2	2	1	1	1	35	Berat
117	3	100	20	1	1	3	3	3	37	Ringan
118	3	100	20	1	1	3	3	3	38.1	Ringan
119	2	120	26	1	1	2	2	2	36.4	Sedang
120	2	120	24	1	1	2	2	2	36	Sedang
121	2	120	26	1	1	2	2	2	36	Sedang
122	2	120	40	1	1	1	2	2	36.4	Sedang

123	3	100	24	1	1	3	3	3	37	Ringan
124	3	100	20	1	1	3	3	3	39	Ringan
125	2	120	26	1	1	2	2	2	36	Sedang
126	3	100	24	1	1	3	3	3	37	Ringan
127	3	100	20	1	1	3	3	3	39	Ringan
128	3	120	20	1	1	3	3	3	38.6	Ringan
129	2	120	26	1	1	2	2	2	36	Sedang
130	2	120	24	1	1	2	2	2	36.4	Sedang
131	3	100	24	1	1	3	3	3	39	Ringan
132	3	100	20	1	1	3	3	3	38	Ringan
133	3	102	20	1	1	3	3	3	39	Ringan
134	3	100	20	1	1	3	3	3	38.1	Ringan
135	2	120	24	1	1	2	2	2	36	Sedang
136	3	100	20	1	1	3	3	3	39	Ringan
137	2	140	40	1	1	2	2	2	36	Sedang
138	2	120	40	1	1	2	2	2	36	Sedang
139	1	90	20	2	2	1	1	1	35.4	Berat
140	2	124	28	1	1	2	2	2	36	Sedang
141	2	140	24	1	1	2	2	2	36	Sedang
142	2	140	24	1	1	2	2	2	36	Sedang
143	1	90	20	2	2	1	1	1	35.6	Berat
144	2	120	40	1	1	1	2	2	36.4	Sedang
145	3	100	24	1	1	3	3	3	37	Ringan
146	3	120	20	1	1	3	3	3	39	Ringan
147	2	120	20	1	1	2	2	2	36	Sedang

148	3	102	20	1	1	3	3	3	37.4	Ringan
149	1	90	20	2	2	1	1	1	35	Berat
150	3	100	24	1	1	3	3	3	39	Ringan



LAMPIRAN E SAMPEL *DATASET* PADA PERHITUNGAN MANUAL

Data Training

No.	Keadaan Umum	Nadi	Pernapasan/RR	Udun-udun	Mata	elastisitas kulit	GCS	Status Gizi	Suhu	Diagnosa
1	3	100	20	1	1	3	3	3	39	1
2	3	120	20	1	1	3	3	3	38.6	1
3	3	100	20	1	1	3	3	3	39	1
4	2	140	40	1	1	2	2	2	36	2
5	2	120	40	1	1	2	2	2	36	2
7	2	120	20	1	1	2	2	2	36	2
6	1	90	20	2	2	1	1	1	35	3
8	1	90	20	2	2	1	1	1	35.4	3
11	1	90	20	2	2	1	1	1	35.6	3
30	1	90	20	2	2	1	1	1	35.4	3

Data Testing

No.	Keadaan Umum	Nadi	Pernapasan/RR	Udun-udun	Mata	elastisitas kulit	GCS	Status Gizi	Suhu	Diagnosa
16	3	102	20	1	1	3	3	3	37.4	1
32	2	120	24	1	1	2	2	2	36.4	2
39	1	90	20	2	2	1	1	1	35	3

LAMPIRAN F HASIL NORMALISASI PADA PERHITUNGAN MANUAL

Hasil Normalisasi Data *Training*

No.	F1	F2	F3	F4	F5	F6	F7	F8	F9	Class
1	0.900000	0.260000	0.100000	0.100000	0.100000	0.900000	0.900000	0.900000	0.900000	1
2	0.900000	0.580000	0.100000	0.100000	0.100000	0.900000	0.900000	0.900000	0.820000	1
3	0.900000	0.260000	0.100000	0.100000	0.100000	0.900000	0.900000	0.900000	0.900000	1
4	0.500000	0.900000	0.900000	0.100000	0.100000	0.500000	0.500000	0.500000	0.300000	2
5	0.500000	0.580000	0.900000	0.100000	0.100000	0.500000	0.500000	0.500000	0.300000	2
7	0.500000	0.580000	0.100000	0.100000	0.100000	0.500000	0.500000	0.500000	0.300000	2
6	0.100000	0.100000	0.100000	0.900000	0.900000	0.100000	0.100000	0.100000	0.100000	3
8	0.100000	0.100000	0.100000	0.900000	0.900000	0.100000	0.100000	0.100000	0.180000	3
11	0.100000	0.100000	0.100000	0.900000	0.900000	0.100000	0.100000	0.100000	0.220000	3
30	0.100000	0.100000	0.100000	0.900000	0.900000	0.100000	0.100000	0.100000	0.180000	3

Hasil Normalisasi Data *Testing*

No.	F1	F2	F3	F4	F5	F6	F7	F8	F9	Class
16	0.900000	0.292000	0.100000	0.100000	0.100000	0.900000	0.900000	0.900000	0.580000	1
32	0.500000	0.580000	0.260000	0.100000	0.100000	0.500000	0.500000	0.500000	0.380000	2
36	0.100000	0.100000	0.100000	0.900000	0.900000	0.100000	0.100000	0.100000	0.180000	3

LAMPIRAN G HASIL PERHITUNGAN MANUAL LEVEL 1

Hasil Perhitungan *Kernel* RBF

$K(x, y)$	1	2	3	4	5	6	7	8	9	10
1	1.000000	0.947053	1.000000	0.358868	0.418449	0.576258	0.105105	0.111693	0.114865	0.111693
2	0.947053	1.000000	0.947053	0.437622	0.460612	0.634321	0.100822	0.106459	0.109132	0.106459
3	1.000000	0.947053	1.000000	0.358868	0.418449	0.576258	0.105105	0.111693	0.114865	0.111693
4	0.358868	0.437622	0.358868	1.000000	0.950089	0.689906	0.197899	0.200448	0.201251	0.200448
5	0.418449	0.460612	0.418449	0.950089	1.000000	0.726149	0.242877	0.246006	0.246992	0.246006
6	0.576258	0.634321	0.576258	0.689906	0.726149	1.000000	0.334473	0.338781	0.340139	0.338781
7	0.105105	0.100822	0.105105	0.197899	0.242877	0.334473	1.000000	0.996805	0.992826	0.996805
8	0.111693	0.106459	0.111693	0.200448	0.246006	0.338781	0.996805	1.000000	0.999200	1.000000
9	0.114865	0.109132	0.114865	0.201251	0.246992	0.340139	0.992826	0.999200	1.000000	0.999200
10	0.111693	0.106459	0.111693	0.200448	0.246006	0.338781	0.996805	1.000000	0.999200	1.000000

Hasil Perhitungan Matriks *Hessian*

D_{ij}	1	2	3	4	5	6	7	8	9	10
1	1.250000	1.197053	1.250000	-0.608868	-0.668449	-0.826258	-0.355105	-0.361693	-0.364865	-0.361693
2	1.197053	1.250000	1.197053	-0.687622	-0.710612	-0.884321	-0.350822	-0.356459	-0.359132	-0.356459
3	1.250000	1.197053	1.250000	-0.608868	-0.668449	-0.826258	-0.355105	-0.361693	-0.364865	-0.361693
4	-0.608868	-0.687622	-0.608868	1.250000	1.200089	0.939906	0.447899	0.450448	0.451251	0.450448
5	-0.668449	-0.710612	-0.668449	1.200089	1.250000	0.976149	0.492877	0.496006	0.496992	0.496006
6	-0.826258	-0.884321	-0.826258	0.939906	0.976149	1.250000	0.584473	0.588781	0.590139	0.588781
7	-0.355105	-0.350822	-0.355105	0.447899	0.492877	0.584473	1.250000	1.246805	1.242826	1.246805

8	-0.361693	-0.356459	-0.361693	0.450448	0.496006	0.588781	1.246805	1.250000	1.249200	1.250000
9	-0.364865	-0.359132	-0.364865	0.451251	0.496992	0.590139	1.242826	1.249200	1.250000	1.249200
10	-0.361693	-0.356459	-0.361693	0.450448	0.496006	0.588781	1.246805	1.250000	1.249200	1.250000

Hasil Perhitungan E_i Iterasi 1

$\alpha_i D_{ij}$	1	2	3	4	5	6	7	8	9	10	E_i
1	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
2	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
3	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
4	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
5	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
6	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
7	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
8	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
9	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
10	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

Hasil Perhitungan $\delta\alpha_i$ Iterasi 1

$\delta\alpha_i$	1	2	3	4	5	6	7	8	9	10
1	0.008000	0.008000	0.008000	0.008000	0.008000	0.008000	0.008000	0.008000	0.008000	0.008000

Hasil Perhitungan α_i Baru Iterasi 1

α_i	1	2	3	4	5	6	7	8	9	10
1	0.008000	0.008000	0.008000	0.008000	0.008000	0.008000	0.008000	0.008000	0.008000	0.008000

UNIVERSITAS BRAWIJAYA



LAMPIRAN H HASIL PENGUJIAN

Hasil Perhitungan $f(x)$

Id Pasien	$f^1(x)$	$f^2(x)$	Hasil Klasifikasi	Actual Class
31	-1.885668	0.618965	1 Dilevel 2 = Dehidrasi Sedang	Sedang
32	-1.836643	0.565461	1 Dilevel 2 = Dehidrasi Sedang	Sedang
33	0.108738	-	1 Dilevel 1 = Dehidrasi Ringan	Sedang
34	-1.848576	0.581394	1 Dilevel 2 = Dehidrasi Sedang	Sedang
35	-1.873721	0.602853	1 Dilevel 2 = Dehidrasi Sedang	Sedang
36	-1.867577	-1.397617	-1 Dilevel 2 = Dehidrasi Berat	Berat
37	-1.885668	0.618965	1 Dilevel 2 = Dehidrasi Sedang	Sedang
38	0.067454	-	1 Dilevel 1 = Dehidrasi Ringan	Sedang
39	-1.864539	-1.393250	-1 Dilevel 2 = Dehidrasi Berat	Berat
40	0.108738	-	1 Dilevel 1 = Dehidrasi Ringan	Sedang
41	-1.798269	0.516143	1 Dilevel 2 = Dehidrasi Sedang	Sedang
42	-1.873721	0.602853	1 Dilevel 2 = Dehidrasi Sedang	Sedang
43	-1.956246	0.308279	1 Dilevel 2 = Dehidrasi Sedang	Sedang
44	0.028595	-	1 Dilevel 1 = Dehidrasi Ringan	Sedang
45	0.121474	-	1 Dilevel 1 = Dehidrasi Ringan	Sedang
46	-0.003425	-0.200961	-1 Dilevel 2 = Dehidrasi Berat	Ringan
47	0.121474	-	1 Dilevel 1 = Dehidrasi Ringan	Sedang

48	-1.868560	0.601163	1 Dilevel 2 = Dehidrasi Sedang	Sedang
49	-1.819900	0.549761	1 Dilevel 2 = Dehidrasi Sedang	Sedang
50	-1.864539	-1.393250	-1 Dilevel 2 = Dehidrasi Berat	Berat
51	0.048545	-	1 Dilevel 1 = Dehidrasi Ringan	Sedang
52	0.107608	-	1 Dilevel 1 = Dehidrasi Ringan	Sedang
53	0.114190	-	1 Dilevel 1 = Dehidrasi Ringan	Ringan
54	0.111087	-	1 Dilevel 1 = Dehidrasi Ringan	Sedang
55	-1.873721	0.602853	1 Dilevel 2 = Dehidrasi Sedang	Sedang
56	0.121474	-	1 Dilevel 1 = Dehidrasi Ringan	Sedang
57	-1.868560	0.601163	1 Dilevel 2 = Dehidrasi Sedang	Sedang
58	-1.819900	0.549761	1 Dilevel 2 = Dehidrasi Sedang	Sedang
59	-1.867577	-1.397617	-1 Dilevel 2 = Dehidrasi Berat	Berat
60	-1.914858	0.650198	1 Dilevel 2 = Dehidrasi Sedang	Sedang
61	-1.939251	0.660860	1 Dilevel 2 = Dehidrasi Sedang	Sedang
62	-1.939251	0.660860	1 Dilevel 2 = Dehidrasi Sedang	Sedang
63	-1.866651	-1.397559	-1 Dilevel 2 = Dehidrasi Berat	Berat
64	-1.956246	0.308279	1 Dilevel 2 = Dehidrasi Sedang	Sedang
65	0.028595	-	1 Dilevel 1 = Dehidrasi Ringan	Sedang
66	0.003425	-	1 Dilevel 1 = Dehidrasi Ringan	Sedang
67	-1.834619	0.552507	1 Dilevel 2 = Dehidrasi Sedang	Sedang
68	0.067454	-	1 Dilevel 1 = Dehidrasi Ringan	Sedang

69	-1.864539	-1.39325	-1 Dilevel 2 = Dehidrasi Berat	Berat
70	0.108738	-	1 Dilevel 1 = Dehidrasi Ringan	Sedang
71	-1.885668	0.618965	1 Dilevel 2 = Dehidrasi Sedang	Sedang
72	-1.956246	0.308279	1 Dilevel 2 = Dehidrasi Sedang	Sedang
73	0.028595	-	1 Dilevel 1 = Dehidrasi Ringan	Sedang
74	0.121474	-	1 Dilevel 1 = Dehidrasi Ringan	Sedang
75	-0.003425	-0.200961	-1 Dilevel 2 = Dehidrasi Berat	Sedang
76	-1.885668	0.618965	1 Dilevel 2 = Dehidrasi Sedang	Sedang
77	-1.836643	0.565461	1 Dilevel 2 = Dehidrasi Sedang	Sedang
78	0.108738	-	1 Dilevel 1 = Dehidrasi Ringan	Sedang
79	-1.848576	0.581394	1 Dilevel 2 = Dehidrasi Sedang	Sedang
80	-1.873721	0.602853	1 Dilevel 2 = Dehidrasi Sedang	Sedang
81	-1.867577	-1.397617	-1 Dilevel 2 = Dehidrasi Berat	Berat
82	-1.885668	0.618965	1 Dilevel 2 = Dehidrasi Sedang	Sedang
83	-1.867577	-1.397617	-1 Dilevel 2 = Dehidrasi Berat	Berat
84	-1.885668	0.618965	1 Dilevel 2 = Dehidrasi Sedang	Sedang
85	-1.836643	0.565461	1 Dilevel 2 = Dehidrasi Sedang	Sedang
86	-1.885668	0.618965	1 Dilevel 2 = Dehidrasi Sedang	Sedang
87	-1.956246	0.308279	1 Dilevel 2 = Dehidrasi Sedang	Sedang
88	0.028595	-	1 Dilevel 1 = Dehidrasi Ringan	Ringan
89	0.121474	-	1 Dilevel 1 = Dehidrasi Ringan	Ringan

90	-1.885668	0.618965	1 Dilevel 2 = Dehidrasi Sedang	Sedang
91	0.028595	-	1 Dilevel 1 = Dehidrasi Ringan	Ringan
92	0.121474	-	1 Dilevel 1 = Dehidrasi Ringan	Ringan
93	-0.003425	-0.200961	-1 Dilevel 2 = Dehidrasi Berat	Ringan
94	-1.885668	0.618965	1 Dilevel 2 = Dehidrasi Sedang	Sedang
95	-1.836643	0.565461	1 Dilevel 2 = Dehidrasi Sedang	Sedang
96	0.108738	-	1 Dilevel 1 = Dehidrasi Ringan	Ringan
97	0.028595	-	1 Dilevel 1 = Dehidrasi Ringan	Ringan
98	0.121474	-	1 Dilevel 1 = Dehidrasi Ringan	Ringan
99	-0.003425	-0.003425	-1 Dilevel 2 = Dehidrasi Berat	Ringan
100	-1.885668	0.618965	1 Dilevel 2 = Dehidrasi Sedang	Sedang
101	-1.885668	0.618965	1 Dilevel 2 = Dehidrasi Sedang	Sedang
102	-1.956246	0.308279	1 Dilevel 2 = Dehidrasi Sedang	Sedang
103	0.028595	-	1 Dilevel 1 = Dehidrasi Ringan	Ringan
104	0.108738	-	1 Dilevel 1 = Dehidrasi Ringan	Ringan
105	-1.848576	0.581394	1 Dilevel 2 = Dehidrasi Sedang	Sedang
106	-1.873721	0.602853	1 Dilevel 2 = Dehidrasi Sedang	Sedang
107	-1.867577	-1.397617	-1 Dilevel 2 = Dehidrasi Berat	Berat
108	0.028595	-	1 Dilevel 1 = Dehidrasi Ringan	Ringan
109	0.121474	-	1 Dilevel 1 = Dehidrasi Ringan	Ringan
110	0.114190	-	1 Dilevel 1 = Dehidrasi Ringan	Ringan

111	0.111087	-	1 Dilevel 1 = Dehidrasi Ringan	Ringan
112	-1.873721	0.602853	1 Dilevel 2 = Dehidrasi Sedang	Sedang
113	0.121474	-	1 Dilevel 1 = Dehidrasi Ringan	Ringan
114	-1.86856	0.601163	1 Dilevel 2 = Dehidrasi Sedang	Sedang
115	-1.81990	0.549761	1 Dilevel 2 = Dehidrasi Sedang	Sedang
116	-1.864539	-1.393250	-1 Dilevel 2 = Dehidrasi Berat	Berat
117	0.048545	-	1 Dilevel 1 = Dehidrasi Ringan	Ringan
118	0.111087	-	1 Dilevel 1 = Dehidrasi Ringan	Ringan
119	-1.848576	0.581394	1 Dilevel 2 = Dehidrasi Sedang	Sedang
120	-1.873721	0.602853	1 Dilevel 2 = Dehidrasi Sedang	Sedang
121	-1.885668	0.618965	1 Dilevel 2 = Dehidrasi Sedang	Sedang
122	-1.956246	0.308279	1 Dilevel 2 = Dehidrasi Sedang	Sedang
123	0.028595	-	1 Dilevel 1 = Dehidrasi Ringan	Ringan
124	0.121474	-	1 Dilevel 1 = Dehidrasi Ringan	Ringan
125	-1.885668	0.618965	1 Dilevel 2 = Dehidrasi Sedang	Sedang
126	0.028595	-	1 Dilevel 1 = Dehidrasi Ringan	Ringan
127	0.121474	-	1 Dilevel 1 = Dehidrasi Ringan	Ringan
128	-0.003425	-0.200961	-1 Dilevel 2 = Dehidrasi Berat	Ringan
129	-1.885668	0.618965	1 Dilevel 2 = Dehidrasi Sedang	Sedang
130	-1.836643	0.565461	1 Dilevel 2 = Dehidrasi Sedang	Sedang
131	0.108738	-	1 Dilevel 1 = Dehidrasi Ringan	Ringan

132	0.107608	-	1 Dilevel 1 = Dehidrasi Ringan	Ringan
133	0.114190	-	1 Dilevel 1 = Dehidrasi Ringan	Ringan
134	0.111087	-	1 Dilevel 1 = Dehidrasi Ringan	Ringan
135	-1.873721	0.602853	1 Dilevel 2 = Dehidrasi Sedang	Sedang
136	0.121474	-	1 Dilevel 1 = Dehidrasi Ringan	Ringan
137	-1.868560	0.601163	1 Dilevel 2 = Dehidrasi Sedang	Sedang
138	-1.819900	0.549761	1 Dilevel 2 = Dehidrasi Sedang	Sedang
139	-1.867577	-1.397617	-1 Dilevel 2 = Dehidrasi Berat	Berat
140	-1.914858	0.650198	1 Dilevel 2 = Dehidrasi Sedang	Sedang
141	-1.939251	0.660860	1 Dilevel 2 = Dehidrasi Sedang	Sedang
142	-1.939251	0.660860	1 Dilevel 2 = Dehidrasi Sedang	Sedang
143	-1.866651	-1.397559	-1 Dilevel 2 = Dehidrasi Berat	Berat
144	-1.956246	0.308279	1 Dilevel 2 = Dehidrasi Sedang	Sedang
145	0.028595	-	1 Dilevel 1 = Dehidrasi Ringan	Ringan
146	0.003425	-	1 Dilevel 1 = Dehidrasi Ringan	Ringan
147	-1.834619	0.552507	1 Dilevel 2 = Dehidrasi Sedang	Sedang
148	0.067454	-	1 Dilevel 1 = Dehidrasi Ringan	Ringan
149	-1.864539	-1.39325	-1 Dilevel 2 = Dehidrasi Berat	Berat
150	0.108738	-	1 Dilevel 1 = Dehidrasi Ringan	Ringan

Hasil Perhitungan α_i Baru Level 1

α_i	2	3	4	5	6	7	8	9	10	11	12	13	14
	1.000000	0.984388	0.155683	0.368204	0.153042	0.438015	0.138014	0.000000	0.000000	0.142031	0.000000	1.000000	1.000000
α_i	15	16	18	19	20	21	22	23	24	25	26	27	28
	0.438015	1.000000	1.000000	0.147357	1.000000	1.000000	1.000000	1.000000	0.220337	0.589177	0.220337	0.147357	1.000000
α_i	29	31	33	38									
	0.147357	0.147357	1.000000	1.000000									

Hasil Perhitungan α_i Baru Level 2

α_i	4	5	6	7	8	9	10	11	12	14	15	18	19
	0.049504	0.049504	0.049504	0.049504	0.049504	0.049504	0.049504	0.049504	0.049504	0.049504	0.049504	0.049504	0.049504
α_i	21	22	23	24	25	26	27	29	31				
	0.049504	0.049504	0.049504	0.049504	0.049504	0.049504	0.049504	0.049504	0.049504				

Nilai b (bias) Level 1 = -0.86957984837491

Nilai b (bias) Level 2 = -0.40265843281893